تفهيم الخوارزميات 2

مع ترجمة بعض الخوارزميات إلى لغة جافا



د. جمال بن نوار

كتاب "تفهيم الخوارزميات : تنظيم وهيكلة المعلومات" الجزء الثاني © 2022 للدكتور جمال بن نوّار مرخص برخصة المشاع الإبداعي CC BY-NC-SA 4.0

المؤلف : البروفيسور جمال الدين بن نوّار، قسم الإعلام الآلي، جامعة البويرة، الجزائر

البريد الالكتروني : <dbennouar@gmail.com>

حساب على Djamal-Bennouar-3@

تصميم الغلاف : هيثم بن حليمة :مصمم haithem_bhm@

تفهيم الخوارزميات

الجزء 2

تنظيم وهيكلة المعلومات

جمال بن نوار
قسم الإعلام الألي
جامعة البويرة، الجزائر

2022

حق الاستعمال والاستغلال للميسر:
دفع ثمن إطعام ثلاثة مساكين من أوسط ما تطعم، إلى مسكين أو جمعية خيرية

بسم الله الرحمان الرحيم

هذا الجزء مخصص للتعاطي مع المعلومات الموجودة في الواقع المعاش، وهي في اغلبها معلومات مركبة (او مهيكلة) يمكن ان تكون معقدة جدا، وليس باليسير تمثيلها بما عرفناه من أنماط بدائية خلال الفصل الأول، وهذه الصعوبة في تناول معلومات الواقع المعاش باستعمال الأنماط البدائية للمعلومات وسلاسل الحروف نُبينها في مستهل هذا الجزء الثاني ونبين ضرورة توفير آليات تمكن من التمثيل السهل والفعال للمعلومات المهيكلة.

في اول تعاملنا مع آليات تمثيل المعلومات المهيكلة، نقتصر على تمثيل خصائص المعلومات المهيكلة والتي نسميها سمات المعلومات المهيكلة، وهنا تظهر المعلومات المهيكلة كجزء ساكن غير فعال في البرنامج، تتناوله اجراءات ووظائف الخوارزم اما للاطلاع على سيماته او لتغييرها لهدف ما، وهنا تبقي هيكلة الخوارزم (او البرنامج) على حالته التقليدية، أي انه مكون من قسمين: قسم المعطيات وهي الجانب الساكن في البرنامج والقسم الثاني هو قسم حيوي فعال، تمثله الإجراءات والوظائف، وفي هذه الهيكلة تكون معالجة المعلومات معالجة مركزية، فكل الوظائف والإجراءات موجودة في جانب واحد من البرنامج .

في مرحلة ثانية، وهي مرحلة نحضر فيها الطالب للبرمجة التصنيفية (او البرمجة بالأشياء)، نجعل من المعلومات المهيكلة معلومات ذكية غير ساكنة، قادرة على التفاعل مع غيرها من مكونات البرنامج، والمعلومات المعقدة الذكية هي التي يمكنها ان تنجز وظائف واجراءات خاصة بها، نسميها الوظائف او الإجراءات المنتسبة لنمط مهيكل ما، وهنا تصبح هيكلة البرنامج مختلفة عن الهيكلة التقليدية لكون الإجراءات والوظائف منتشرة او موزعة داخل المعلومات المهيكلة، واذا اصبح البرنامج مكون فقط من معلومات مهيكلة ذكية، تصبح معالجة المعلومات معالجة غير مركزية، فكل الوظائف والإجراءات المنتسبة تتفاعل مع سمات المعلومات المهيكلة التي فيها أنشئت.

جدول المحتويات

المعلومات في الواقع المعاش 1
المعلومات المهيكلة 26
تهيئة المتغيرات المهيكلة 48
العمليات الأساسية على المتغيرات المهيكلة 70
الخوارزميات المنتسبة للأنماط المهيكلة 108
الأوعية: الجداول 145
الأوعية: تمثيل الجداول والخانات في الذاكرة 167
الأوعية: تمثيل الجدول ذو بعدين او اكثر في ذاكرة الخوارزم 186
الأوعية: التجوال في الجداول (1) 195
الأوعية: التجوال في الجداول (2) 220
الأوعية: تمثيل واخراج جداول الواقع المعاش 234
الأوعية: سلاسل الحروف 269
الأوعية: اهم وظائف النمط سلسلة 284
الأوعية: العمليات المعتادة على الجداول 304
الأوعية: العمليات المعتادة على الجداول: عمليات البحث 318
الأوعية: العمليات المعتادة على الجداول: ترتيب المعلومات 334
الملحق الأول 353
الكلمات الشائعة الاستعمال بالعربية والإنجليزية 353
الملحق الثاني 356
ترجمة خوارزميات الجزء الثاني الى لغة "جافا" 356
المعلومات في الواقع المعاش 357
المعلومات المهيكلة 365
تهيئة المتغيرات المهيكلة 369
العمليات الأساسية على المتغيرات المهيكلة 376
الخوارزميات المنتسبة للأنماط المهيكلة 378
الأوعية: الجداول 387
الأوعية: التجوال في الجداول (1) 393
الأوعية: التجوال في الجداول (2) 400
الأوعية: تمثيل واخراج جداول الواقع المعاش 406
الأوعية: سلاسل الحروف 415
الأوعية: اهم وظائف النمط سلسلة 417
الأوعية: العمليات المعتادة على الجداول 424
الأوعية: العمليات المعتادة على الجداول: عمليات البحث 431
الأوعية: العمليات المعتادة على الجداول: ترتيب المعلومات 439

الفصل الأول

المعلومات في الواقع المعاش

بسم الله الرحمان الرحيم

1 – مقدمة

في الجزء الأول تعرفنا على كيفية تمثيل عدد كبير من المعلومات باستعمال الأنواع (او الأنماط) الأساسية (ونسميها ايضا: الأنماط البدائية) كالنمط **طبيعي** والنمط **حقيقي** والنمط **منطقي** والنمط **حرف**، وأشرنا ايضا الى ان **'الصلة'** في حد ذاتها نمط بدائي.

ملاحظة: سميت هذه الأنماط بالأساسية أو البدائية لكون كل لغات كتابة الخوارزميات وكل لغات البرمجة، حتى تلك المسماة بلغة الآلة، تعرف جيدا كيف تمثل في الذاكرة متغيرات هذه الأنماط البدائية، وكيف تجري عليها مختلف العمليات الحسابية والمنطقية، فالمعلومات البدائية من المكونات الاساسية للغات كتابة الخوارزميات ولغات البرمجة.

تنبيه: مع كون النمط **سلسلة** نمط مألوف وواسع الاستعمال، الا انه **ليس بنمط بدائي**، فسلسلة حروف مركبة من حروف، وكل حرف من النمط **حرف**، اي ان النمط **سلسلة** هو نمط مركب من النمط **حرف**

كما اشرنا اليه في بداية الجزء الأول، فانه لا يخلو ميدان من ميادين الحياة الا وقد انجزت له أنظمة معلوماتية، والمعلومات المتداولة في هذه الميادين معلومات معقدة جدا، فعلى سبيل المثال:

- في **الواقع المتعلق بالجامعات** نجد نشاط هام متعلق بتسيير الكليات والأقسام والنشاط التعليمي، ومن بين المعلومات المعقدة المتداولة في الجامعات نجد المعلومات التي تصف **الطلبة**، و**الأساتذة**، و**الدروس**، و**البرامج**، **وجداول التوقيت** الخ...
- في **الواقع المتعلق بصناعة السيارات** نجد نشاط هام متعلق بتصميم وتركيب السيارات، ومن بين المعلومات نجد مكونات السيارات: **المحركات**، و**الهياكل**، و**الأبواب**، و**الكراسي** الخ...

2– ضرورة التمثيل الفعال للمعلومات المعقدة:

لا يمكن للأنظمة المعلوماتية ان تكون ذو فعالية في مختلف ميادين الحياة الا اذا كانت قادرة على ان تمثل المعلومات المعقدة في ذاكرتها وتتعاطى معها لتنفذ عليها مختلف العمليات.

اكتساح الأنظمة المعلوماتية لكل ميادين الحياة جاء بعد ان مكنت لغات الخوارزميات ولغات البرمجة من التمثيل الفعال لمختلف المعلومات المعقدة، وتوفير مختلف التعليمات والعمليات للتعاطي مع هذه المعلومات المعقدة، ونسرد فيما يلي امثلة حقيقية لبعض هذه المعلومات المعقدة:

- المعلومات التي تمثل **الصور** في إطار نظام معلوماتي يعالج الصور (تكبير او تصغير الصورة، تسديد الألوان، اضافة تعليق في الصورة، إزالة العيوب، مقارنة الصور، تشفير الصور الخ...).

- المعلومات التي تمثل **الصوت** في إطار نظام معلوماتي يعالج الصوت (التقاط وتخزين الأصوات، مقارنة الأصوات، البحث عن صوت محدد، تشفير الأصوات قبل ارسالها الخ...)
- المعلومات التي تمثل **الطالب** في إطار نظام معلوماتي متخصص في تسيير المسار الدراسي للطالب.
- المعلومات التي تمثل **سيارة** ما في إطار نظام معلوماتي يحاكي قيادة السيارات.
- المعلومات التي تمثل **كل علامات الطلبة** في إطار نظام معلوماتي متخصص في تسيير المسار الدراسي للطالب.
- المعلومات التي تمثل **عدد كبير من الجنود** في إطار نظام معلوماتي يحاكي معركة بين جيشين الخ..

3- أشكال التعقيد في معلومات الواقع المعاش.

يتضح من هذه الأمثلة ان التعقيد يكون على شكلين اساسيين يمكن مزجهما في شكل ثالث:

- **الشكل الأول:** التعقيد يكون على مستوى المعلومة الواحدة، كالمعلومة الممثلة لطالب ما، او لسيارة ما، او لصورة ما.
- **الشكل الثاني:** يكون التعقيد في تواجد اكثر من معلومة من نفس النمط، وكأن هذه المعلومات موجودة في **وعاء**، فيه تنتظم هذه المعلومات، ومن الأمثلة لهذا الشكل الثاني قولنا **'مجموعة علامات الطالب'**، ففي مثل هذه المجموعة نجد عدد كبير من العلامات، مع ان العلامة في حد ذاتها معلومة بدائية، والتعامل مع عدد ضخم من العلامات الممثلة لمفهوم ما، مثلا علامات كل الطلبة، يكون بالتأكيد مغايرا تماما عن التعامل مع علامة واحدة او علامتين.
- **الشكل الثالث** (الممزوج): يكون التعقيد في وجود عدد كبير من معلومات هي في حد ذاتها معقدة، كقولنا **'قائمة الطلبة'**، او **'قائمة السيارات'**، او **'سلسلة بشرية'**، او "**كومة من الكتب**"، ففي هذه المعلومات المعقدة نجد تردد معلومات هي في ذاتها معقدة، او كقولنا **'مجموعة علامات الطلبة'**، وهنا نجد تردد معلومتان: الأولى هي المعلومة الممثلة للطالب والثانية هي المعلومة الممثلة لعلامات الطالب.

وهنا يطرح السؤال التالي: كيف تُمثّل يا تُرى هذه المعلومات المعقدة عند كتابة خوارزم ما؟ فهل نستعمل الأنماط البدائية كما هي أم أنه لا بد ان تتاح لكاتب الخوارزم آليات أخرى تمكنه من تمثيل هذه المعلومات المعقدة مباشرة؟

في سياق الإجابة على السؤالين الأخيرين، نحاول في بداية الأمر ان نمثل المعلومات المعقدة باستعمال ما نعرف من **انماط بدائية** لنرى جليا ضرورة ايجاد آلية تُسهل على كاتب الخوارزميات والبرامج تمثيل المعلومات المعقدة.

4 – تمثيل المعلومات المعقدة باستعمال الأنماط البدائية

حتى نبين محدودية الأنماط البدائية في تمثيل المعطيات المعقدة وضرورة توفير آلية فعالة لتمثيل المعلومات المعقدة، وفي إطار أمثلة بسيطة، نقوم فيما يلي بالاستعمال الحصري للأنماط البدائية في تمثيل المعلومات المعقدة.

4-1 المثال الأول: خوارزم يحسب نتائج طلبة قسم ما في مادة ما.

من خلال هذا المثال الأول، نريد انجاز خوارزم يقوم بالأنشطة التالية:

- حساب معدل الطلبة المنتسبين لنفس القسم في مادة ما.
- ايجاد عدد العلامات التي هي اقل من 80 بالمئة من معدل القسم، وهو عدد الراسبين في المادة.
- كتابة النتائج على الشاشة، كمعدل القسم وعدد الراسبين.

ونوكل هذه المهمات **لإجراء** نسميه ' **نتائج_القسم_في_مادة** '.

4-1-1 السلوك النظري للإجراء ' نتائج_القسم_في_مادة ':

اول ما يقوم به الإجراء ' **نتائج_القسم_في_مادة** ' هو الحصول على معدلات التلاميذ، ثم يحسب معدل القسم، ثم يحسب القيمة 80 بالمئة من معدل القسم، وبعدها يبحث في المعدلات المتحصل عليها فيحسب عدد المعدلات التي هي اقل من 80 بالمئة من معدل القسم، ثم يخبر المستعمل بثلاث نتائج: **معدل القسم، المعدل الأدنى المسموح به، عدد الراسبين في المادة.**

لإنجاز هذا الخوارزم يمكن ان ننتهج طرق عديدة، من بينها احدى الطريقتين التاليتين:

- **الطريقة الأولى** تنطلق من خوارزم حساب معدل النقاط الذي انجزناه في الجزء الأول، فنعاود استعماله.
- **الطريقة الثانية** تنطلق من كتابة الملامح الكبرى (او المراحل الكبرى) للخوارزم ثم تفصيل كل مرحلة والتنسيق بينها.

4-1-2 الطريقة الأولى (النص 1)

في اطار هذه الطريقة يقوم الخوارزم بقراءة قيم المعدلات (اي اخذها من **'لوحة الحروف'**)، وجمعها في المتغيرة **'مجموع_المعدلات'** التي صرحت بقيمة أولية (السطر 2)، فكلما قُرأ ت قيمة أُضيفت الى المتغيرة **'مجموع_المعدلات'**، وأُضيف 1 الى المتغيرة **'عدد_المعدلات'** التي تخبر عن عدد المعدلات التي ادخلها المستعمل، وعند الانتهاء من ادخال كل القيم يدخل المستعمل القيمة –1، وعند التقاط الخوارزم القيمة –1 ، يفهم ان المستعمل قد ادخل كل المعدلات، فيقوم بعد ذلك:

- اولا: بحساب معدل المادة باستعمال العبارة **'مجموع_المعدلات / عدد_المعدلات'**،

- ثانيا: بحساب معدل القيمة 80 بالمئة من معدل القسم ويضع النتيجة في المتغيرة **'ادنى_معدل_مسموح_به'**.
- ثالثا واخيرا: يخرج النتائج على الشاشة (خطأ: لم يُعثر على مصدر المرجع).

```
1  اجراء نتائج_القسم_في_مادة (){
2  حقيقي معدل، مجموع_المعدلات=0.00، معدل_القسم، ادنى_معدل_مسموح_به ؛
3  طبيعي عدد_المعدلات=0؛
4  اكتب "هذا الخوارزم يحسب معدل القسم في مادة ما، و يعطيك عدد الطلبة الراسبين في المادة"
5  اكتب "ادخل تباعا معدلات الطلبة، و عند اتمام ادخال كل المعدلات ادخل -1 "
6  اكتب (" ادخل المعدل")
7  معدل = اقرأ_حقيقي () // الوظيفة اقرأ_حقيقي() ترجع الحقيقي الذي يكتب على لوحة المفاتيح، وهي مطابة للتعليمة أقرأ(معدل)
8  مادام (معدل =!  -1) {
9      مجموع_المعدلات = مجموع_المعدلات + معدل
10     عدد_المعدلات = عدد_المعدلات + 1
11     اكتب (" ادخل المعدل")
12     معدل = اقرأ_حقيقي ()
13 }
14 معدل_القسم = مجموع_المعدلات / عدد_ المعدلات
15 ادنى_معدل_مسموح_به = معدل_القسم * 0.8
16 اكتب ("عدد المعدلات التي ادخلت هي :" + عدد_المعدلات)
17 اكتب ("معدل القسم هو:" + عدد_المعدلات}
18 اكتب ("ادنى معدل مقبول هو:" + ادنى_معدل_مسموح_به)
19 /* في ما يلي سوف نحسب عدد  الطلبة الذين لم يتحصلوا على المادة */
20
21 }
```

النص 1 : المرحلة الأولى من الإجراء: حساب المعدل

```
هذا الخوارزم  يحسب معدلا لقسم في مادة ما، ويعطيك عدد الطلبة الراسبين في المادة
ادخل تباعا معدلات الطلبة، وعند اتمام ادخالك للمعدلات ادخل -1
ادخل المعدل
12.50
ادخل المعدل
16.00
ادخل المعدل
18.50
ادخل المعدل
-1
```

الشكل 1: نتيجة تنفيذ الجزء الأول من خوارزم 'نتائج_القسم_في_مادة' (النص 1)

بعد العمليات التي تنتج ما يظهر. في خطأ: لم يُعثر على مصدر المرجع، يجب على الخوارزم **'نتائج_القسم_في_مادة'** ان يحسب عدد المعدلات الأقل من قيمة المتغيرة **'ادنى_معدل_مسموح_به'**، وليتمكن من تحقيق هذا الهدف، عليه اولا ان يتخذ **عدّادًا** قد شُحن بصفر كقيمة اولية، **ثم يراجع كل المعدلات** التي ادخلها المستعمل وكلما وجد في هذه المعدلات معدلا أقل من قيمة المتغيرة **'ادنى_معدل_مسموح_به'**، يضيف القيمة 1 **للعدّاد**.

ملاحظة: نلاحظ في السطرين 7 و 12 من النص 1، استعمال وظيفة جديدة هي الوظيفة **'اقرأ_حقيقي()'**، وترجع هذه الوظيفة العدد الحقيقي الذي يكتب على لوحة الحروف، ولكل نمط بدائي، توفر اللغة الشرمزية وظائف مشابهة: **'اقرأ_طبيعي()'، 'اقرأ_سلسلة()'، 'اقرأ _منطقي()'.**

السؤال الذي يُطرح هنا: **كيف يمكن للخوارزم ان يراجع كل ما ادخل المستعمل** من معدلات، علما انه باستثناء آخر معدل ادخله المستعمل، كل المعدلات التي ادخلت قد محيت بفعل تعليمتي السطر 7 والسطر 12، اي **'معدل = اقرأ_حقيقي ()'.**

أ – الحل البدائي: هو حل مباشر و بديهي ويكمن **في ضرورة الحفاظ على كل القيم التي ادخلها المستعمل**، وللحفاظ على هذه القيم يجب استعمال متغيرات توضع فيها كل القيم التي يدخلها المستعمل، وبالطبع تكون كل هذه المتغيرات من النمط **'حقيقي'** حتى تتمكن من الحفظ السليم لكل المعدلات التي يدخلها المستعمل.

في اطار هذا الحل الأخير يَبرُز السؤال البديهي التالي: **ما هو عدد المتغيرات** التي نستعملها للحفاظ على المعدلات؟ للإجابة على مثل هذا سؤال، يجب علينا ان نرجع الى الواقع الذي سوف يستعمل فيه هذا الخوارزم و ندرسه ونفهمه جيدا، فمثلا اذا كان مثل هذا الخوارزم يستعمل في الجامعة او الثانوية، فالعدد المعتاد من الطلبة في القسم هو 20 طالبا، و في اقصى حالة غير عادية، يمكن ان يصل العدد الى 35، وهذه حالة شاذة جدا، وفي كل الأحوال يستحيل ان يتجاوز عدد الطلبة في قسم واحد العدد 40، ومن خلال هذه المعرفة عن الواقع الذي سوف يستعمل فيه الخوارزم، ولجعل الخوارزم يتكيف حتى مع الحالات الشاذة جدا، نقوم برصد 40 متغيرة لحفظ ما يدخله المستعمل، وهكذا اذا ادخل المستعمل 15 معدلا سوف نستعمل 15 متغيرة، واذا ادخل 30 معدلا سوف نستعمل 30 متغيرة من بين المتغيرات الأربعين.

حتى يتمكن الخوارزم **نتائج_القسم_في_مادة** من استغلال الاربعين متغيرة، يجب علينا ان نصرح بها كلها، وهنا تبرز **صعوبة اولى وهي كتابة 40 تصريحا بالمتغيرات**، وحسب ما علمناه يجب ان نوفر اسما لكل متغيرة، ولنفرض ان الأسماء هي : **معدل1، معدل2، معدل3، معدل4،** الى **معدل40** كما يظهر في النص 2 من السطر 2 الى السطر 5.

بعد ان تمكنا من اجتياز الصعوبة الأولى، اي ايجاد الحل لتخزين كل ما يدخله المستعمل، تواجهنا صعوبة ثانية، تتمثل في ضرورة ايجاد حل لإشكال مهم وهو البت في تعيين المتغيرة المناسبة لتخزين كل معدل يدخله المستعمل.

```
1   اجراء  نتائج_القسم_في_مادة (){
2   حقيقي معدل1، معدل2، معدل3، معدل4، معدل5، معدل6، معدل7، معدل8، معدل9، معدل10،
3   حقيقي معدل11، معدل12، معدل13، معدل14، معدل15، معدل16، معدل17، معدل18، معدل19، معدل20
4   حقيقي معدل21، معدل22، معدل23، معدل24، معدل25، معدل26، معدل27، معدل28، معدل29، معدل30
5   حقيقي معدل31، معدل32، معدل33، معدل34، معدل35، معدل36، معدل37، معدل38، معدل39، معدل40
6   حقيقي معدل،  مجموع_المعدلات=0.00، معدل_القسم، ادنى_معدل_مسموح_به
7   طبيعي عدد_المعدلات=0
8   طبيعي عدد_الراسبين=0
9   اكتب "هذا الخوارزم يحسب معدل القسم في مادة ما، و يعطيك عدد الطلبة الراسبين في المادة"
10  اكتب "ادخل تباعا معدلات الطلبة، وعند اتمام ادخال كل المعدلات ادخل -1 "
11  اكتب (" ادخل المعدل")
12  اقرأ (معدل)
13  مادام (معدل =!  -1) }
14      مجموع_المعدلات = مجموع_المعدلات + معدل
15      عدد_المعدلات = عدد_المعدلات + 1
16      /* بعد التحصل على المعدل يجب علينا ان  نخزنه في المتغيرة المناسبة */
17      تحول الى (عدد_المعدلات)}  /*بداية جسم التعليمة الشرطية الرقمية */
18          الخيار 1:  معدل1 = معدل
19          غادر
20          الخيار 2:  معدل2 = معدل
21          غادر
...         .....
....        .....
96          الخيار 40:  معدل40 = معدل
97      { /*نهاية جسم التعليمة الشرطية الرقمية */
98      اكتب (" ادخل المعدل")
99      اقرأ (معدل)
100 {/*نهاية جسم التعليمة  مادام (معدل =!  -1) */
101 معدل_القسم = مجموع_المعدلات / عدد_ المعدلات
102 ادنى_معدل_مسموح_به = معدل_القسم * 0.8
103     /* في ما يلي سوف نحسب عدد  الطلبة الذي لم يتحصلوا على المادة */
104 اذكان (عدد_المعدلات< 0 )
105 اذكان (معدل1>ادنى_معدل_مسموح_به ) عدد_الراسبين = عدد_الراسبين + 1
106 اذكان (عدد_المعدلات< 1 )
107 اذكان (معدل2>ادنى_معدل_مسموح_به ) عدد_الراسبين = عدد_الراسبين + 1
....        .....
....        .....
182 اذكان (عدد_المعدلات< 39 )
183 اذكان (معدل40>ادنى_معدل_مسموح_به ) عدد_الراسبين = عدد_الراسبين + 1
184     اكتب ("عدد المعدلات التي ادخلت هي :" + عدد_المعدلات}
185     اكتب ("معدل القسم هو:" + معدل_القسم)
186     اكتب ("ادنى معدل مقبول هو:" + ادنى_معدل_مسموح_به)
187     اكتب ("عدد الراسبين في المادة:" + عدد_الراسبين)
188 }
```

النص 2 : انجاز الخوارزم نتائج_القسم_في_مادة باستعمال الانماط البدائية فقط

وكنتيجة للطريقة الجيدة والطبيعية والبديهية في تسمية المتغيرات، يظهر لنا جليا وبصورة فورية ان المتغيرة المسماة **'معدل1'** هي المرشحة لحفظ اول قيمة يدخلها المستعمل، والمتغيرة **'معدل 21'** سوف تستعمل لحفظ القيمة التي يدخلها المستعمل في التكرار. 21، وانطلاقا من هذه الملاحظات

يصبح خوارزم حساب المعدل على الصيغة الظاهرة في النص 3، والتعليمات التي تتكفل بوضع ما يدخله المستعمل في المتغيرة المناسبة هي تلك التي تظهر من السطر. 17 الى السطر. 97 في اطار تعليمة شرطية رقمية.

ب – تكلفة الحل البدائي: تكلفة هذا الحل الأول هو كتابة 40 تصريحا. و80 تعليمة (من السطر 17 الى السطر 97).

```
1    اجراء  نتائج_القسم_في_مادة (){
....  .....
17    تحول الى (عدد_المعدلات) }  /*بداية جسم التعليمة الشرطية الرقمية */
18    الخيار 1:  معدل1 = معدل
19    غادر
20    الخيار 2:  معدل2 = معدل
21    غادر
...   .....
....  .....
96    الخيار 40:  معدل40 = معدل
97    غادر
98    الخيار 41:  اكتب "لا يمكن لهذا الخوارزم التعاطي مع اكثر من 40 معدلا"
99    ارجع  /* نهاية مبكرة للإجراء */
100   { /*نهاية جسم التعليمة الشرطية الرقمية */
101   اكتب (" ادخل المعدل")
102   اقرأ (معدل)
....  .....
190   اكتب ("عدد الراسبين في المادة:" + عدد_الراسبين)
191   }
```

النص 3 : الخوارزم: حساب المعدل : نسخة محسنة للتنبيه اذا تجاوز المستعمل ادخال اكثر من 40 معدلا

ج – تحديات الحل البدائي: الآن وقد انجزنا شطر الخوارزم الذي يتكفل بحفظ كل ما يدخله المستعمل، علينا ان نبحث عن المعدلات التي هي اصغر من قيمة المتغيرة **' ادنى_معدل_مسموح_به '**، ولبلوغ هذا الهدف يجب علينا زيارة كل المتغيرات التي استعملت لحفظ ما ادخله المستعمل لتصفح محتواها، فمثلا اذا ادخل المستعمل 15 معدلا يجب علينا زيارة المتغيرات **'معدل1'، 'معدل2'** الى **'معدل15'**، وفي هذا الإطار تكون 15 هي قيمة المتغيرة **'عدد_المعدلات'**، ويمكن استعمال قيمة هذه المتغيرة لمعرفة هل استعملت متغيرة ما لتخزين قيمة ادخلها المستعمل، واذا تبين ان متغيرة ما قد استعملت، نتأكد من محتواها هل هو اصغر من محتوى المتغيرة **'ادنى_معدل_مسموح_به'**، وفي النص نستعمل لهذين الغرضين تعليمتين شرطيتين منطقيتين، واذا صحت العبارتين المنطقيتين التابعتين للتعليمتين الشرطيتين، يقوم الخوارزم بإضافة 1 للمتغيرة **'عدد_الراسبين'** التي صرح بها بقيمة اولية تساوي 0 .

فعلى سبيل المثال، اذا ادخل المستعمل 20 معدلا، تكون 20 هي قيمة المتغيرة **'عدد_المعدلات'**، و بذلك تكون **'صحيحة'** العبارة الشرطية للتعليمة **اذكان (عدد_المعدلات< 0)**، والتي تُبَين ان المتغيرة **'معدل1'** قد استُعملت، وَوَجَب اذًا زيارتها لمعرفة هل قيمتها اصغر من قيمة **'ادنى_معدل_مسموح_به'**، و نتحصل على نفس النتيجة، اي **'صحيح'**، في تقييم العبارات المنطقية

للتعليمات الشرطية **'اذكان (عدد_المعدلات< 1)'** و **'اذكان (عدد_المعدلات< 2)'** حتى التعليمة الشرطية **'اذكان (عدد_المعدلات < 19)'**، اما باقي التعليمات الشرطية، اي من التعليمة **'اذكان (عدد_المعدلات < 20)'** الى التعليمة **'اذكان (عدد_المعدلات < 39)'**، تكون خطأ هي نتيجة تقييم عباراتها المنطقية، ومعنى هذا ان المتغيرات من **'معدل21'** الى **'معدل 40'**، لم تُستعمل لحفظ ما يدخله المستعمل ولا يوجد معنى لزيارتها.

د- تكلفة زيارة المتغيرات الحاملة للمعدلات: يكلف اشكال زيارة المتغيرات الحاضنة لما يدخله المستعمل من علامات 120 تعليمة، فقد خصص لكل متغيرة تعليمتين شرطيتين وتعليمة شحن.

هـ- خلاصة حل المثال الأول بالطريقة الأولى: نرى جليا من خلال المثال البسيط الأول صعوبة كتابة خوارزميات تعالج مسائل تحتوي على معلومات معقدة، فالاستعمال الحصري للأنماط البدائية يؤدي بنا الى كتابة خوارزميات ضخمة، فمثلا ، لو كَيّفنا خوارزم النص 2 لمعالجة دفعات الطلبة التي في الغالب يتجاوز عدد الطلبة فيها الألف، فان حجم الخوارزم سوف يتجاوز 5000 تعليمة، وهذا حجم لا يمكن تقبله كحجم لإجراء او وظيفة.

و- تحسين الإجراء: ماذا لو أُدخل اكثر من 40 معدلا ؟

يلاحظ من خلال النص 2 ان المستعمل يمكنه ادخال اي عدد من المعدلات، اي اكثر من 40، لكون الخوارزم قد ترك امر انتهاء ادخال المعدلات تحت السيطرة الوحيدة للمستعمل، اي ان المستعمل يبين انتهاء عمله بإدخال القيمة -1، وهكذا لو استُعمل هذا الخوارزم في اطار غير الإطار الطبيعي الذي انشأ له، يمكن للمستعمل ادخال اكثر من 40 معدلا، وبعدها يدخل القيمة -1، وفي هذه الحالة لا يحتفظ الخوارزم بالقيم التي تدخل بعد القيمة رقم 40، ولا ينبه الخوارزم المستعمل عن هذه الحالة التي لا يحتملها، فيمكن ان يدخل الخوارزم 45 معدلا، لكن الخوارزم يعطيه معدل قسم فيه 40 طالبا (اي 40 معدلا).

لتفادي مثل هذه الحالات وجعل الخوارزم يُنبه المستعمل عن محدوديته، نضيف الى التعليمة الشرطية الرقمية (السطر 17) الخيار 41 ، وفي هذا الخيار يُنبه الخوارزم المستعمل عن محدوديته كما يظهر في النص 3، واذا حدث وان حاول المستعمل ادخال اكثر من 40 معدلا، ينبه الخوارزم المستعمل بعدم قدرته التعاطي مع اكثر من 40 معدلا ثم ينته بفضل التعليمة 'ارجع' (السطر 98).

4 – 1 –3 الطريقة الثانية

في هذه الطريقة الثانية، نكتب اولا الملامح الكبرى للحل الذي تظهر من خلاله المراحل الكبرى التي يمر بها الإجراء ليصل الى النتيجة المرجوة، فنكتب في اول الأمر هذه الملامح بلغة

عادية، اي غير رمزية وغير شبه رمزية، ثم بعدها نكتب الخوارزم باستعمال اللغة الشرمزية، وكلما كانت في النص تعليمات غير معلومة، قمنا بتفصيلها، حتى نتحصل على خوارزم يمكن تنفيذه.

يبرز النص 4 المراحل الكبرى لخوارزم' **نتائج_القسم_في_مادة** ' وقد كتبت الملامح الكبرى بطريقة الترقيم، والترقيم هنا غير ضروري.

```
1 : تحصل على المعدلات
2 : احسب معدل القسم
3 : احسب 80 بالمئة من معدل القسم،
4 : قم بتعداد المعدلات التي هي اقل من 80 بالمئة من معدل القسم
5 : اخراج النتائج
```

النص 4 : المراحل الكبرى لخوارزم نتائج المادة

عندما نعيد كتابة هذا النص باللغة **الشرمزية**، نخصص لكل مرحلة كبرى وظيفة او اجراء، فنتحصل على الصيغة الأولية الظاهرة في النص 5، وهي صيغة غير كاملة، ونرى من خلال هذا النص الناقص كيف اننا خصصنا لكل مرحلة إجراءا أو وظيفة.

```
1  اجراء نتائج_القسم_في_مادة (){
2      تحصل_على_المعدلات()
3      احسب_معدل_القسم()
4      احسب_ادنى_معدل()
5      عد_معدلات_الراسبين()
6      اخرج_النتائج()
7  }
```

النص 5 : نص عير كامل للخوارزم على الطريقة الثانية.

أ – البت في نوعية كل خوارزم جزئي: هل هو اجراء او وظيفة:

للبت في نوعية كل خوارزم جزئي المكلف بمرحلة ما، هل من الأفضل ان يكون **اجراء** او **وظيفة**، نستعرض اولا وبعمق السلوك العام للخوارزم الجزئي، اي:

- ما هو هدف الخوارزم الجزئي.
- ما هي المعلومات التي يحتاجها الخوارزم الجزئي حتى يُنجز مهمته.
- كيفية توصيل هذه المعلومات الى الخوارزم الجزئي.
- ما هي المعلومات التي ينتجها الخوارزم الجزئي.
- كيفية توصيل المعلومات التي ينتجها الخوارزم الجزئي الى المستفيد، اي ذلك الذي طلب تنفيذه.

ب– استعراض سلوك الخوارزم الجزئي 'تحصل_على_المعدلات'

- **هدف الخوارزم:** التفاعل مع المستعمل لجعله يدخل المعدلات.
- **المعلومات التي يحتاجها الخوارزم:** لا يحتاج لأي معلومة خاصة.
- **المعلومات التي ينتجها الخوارزم:** ينتج الخوارزم معلومتين:
  - المعلومة الأولى معقدة وهي المعدلات وقد حفظت في متغيرات يمكن الوصول اليها بعد انتهاء تنفيذ الخوارزم الجزئي وبعد ان يُمحى مجاله الخاص من الذاكرة.

- المعلومة الثانية هي عدد المعدلات التي ادخلت وهي معلومة طبيعية (بدائية).

- **كيفية توصيل المعلومات التي ينتجها الخوارزم:** هناك ثلاثة طرق لتوصيل النتائج:
  - مخارج الخوارزم الجزئي (الوظيفة او الإجراء)
  - المتغيرات العامة
  - الشاشة.

- **استعمال الشاشة:** اذا استعملنا **الشاشة**، فلن يستفيد من المعلومات المخرجة الا المستعمل البشري، والمستعمل البشري اصلا لا حاجة له بها، فهو مصدر هذه المعلومات، اي المعدلات، فالشاشة اذا مستبعدة لتوصيل نتائج الخوارزم.

- **استعمال مخارج الخوارزم الجزئي:** اذا استعملنا **المخارج**، فيمكن مثلا ان نستعمل:
  - **'منفذ الرجوع'** لإخراج عدد المعدلات، وعدد المعدلات قيمة من النمط **'طبيعي'.**
  - المخارج العادية لإخراج المعدلات.

وفي هذه الحالة يكون الخوارزم الجزئي ' **تحصل_على_المعدلات'** وظيفة، ويكون رأس هذه الوظيفة على الشكل التالي:

'طبيعي تحصل_على_المعدلات (حقيقي &معدل1، &معدل2،.....، &معدل40)'

و هنا نلاحظ الصعوبة الكبيرة في كتابة قائمة المخارج والتحكم فيها، و**ضرورة التصريح بنفس العدد من المتغيرات المحلية** في اي خوارزم يطلب خدمة الوظيفة ' **تحصل_على_المعدلات'**، واذا اردنا تكييف هذا الأخير ليتعامل مع اكثر من الف معدل، فان حجم رأس الخوارزم يصبح ضخما جدا، وهذا امر جد غريب في عالم الخوارزميات ولا يطاق، زيادة على هذا، فعلى كل خوارزم يريد استعمال الوظيفة **'تحصل_على_المعدلات'** ان يوفر له العدد الضخم من المخارج وهي متغيرات قد صرح بها قبل وضعها في مخارج طلب تنفيذ الوظيفة **'تحصل_على_المعدلات'.**

- **استعمال المتغيرات العامة:** اذا استعملنا المتغيرات العامة، سيكون رأس **'تحصل_على_المعدلات'** على الشكل التالي:

' طبيعي تحصل_على_المعدلات ()'

و هذا الشكل سهل الكتابة، لكن محتوى الخوارزم يبق كبيرا كمحتوى الجزء الأول من النص 3، اي حوالي 100 تعليمة ليتحصل على 40 معدلا وحوالي 2000 تعليمة ليتحصل على 1000 معدل، اما الخوارزميات التي تريد خدمة من الوظيفة **'تحصل_على_المعدلات'**، فلن تكون مجبرة على

التصريح بالمتغيرات لكون المعلومات المتعلقة بالمعدلات موجودة في المجال العام مخزنة في المتغيرات العامة.

```
1  /* التصريح بالمتغيرات المخصصة لتخزين المعدلات في المجال العام */
2  حقيقي معدل1، معدل2، معدل3، معدل4، معدل5، معدل6، معدل7، معدل8، معدل9، معدل10،
3  حقيقي معدل11، معدل12، معدل13، معدل14، معدل15، معدل16، معدل17، معدل18، معدل19، معدل20
4  حقيقي معدل21، معدل22، معدل23، معدل24، معدل25، معدل26، معدل27، معدل28، معدل29، معدل30
5  حقيقي معدل31، معدل32، معدل33، معدل34، معدل35، معدل36، معدل37، معدل38، معدل39، معدل40
6  اجراء نتائج_القسم_في_مادة (){
7      طبيعي عدد_المعدلات=0، عدد_الراسبين=0
8      حقيقي معدل_القسم، ادنى_معدل_مسموح_به
9      عدد_المعدلات = تحصل_على_المعدلات()
10     معدل_القسم = احسب_معدل_القسم(عدد_المعدلات)
11     ادنى_معدل_مسموح_به= احسب_ادنى_معدل(معدل_القسم)
12     عدد_الراسبين = عد_معدلات_الراسبين(ادنى_معدل_مسموح_به، عدد_المعدلات)
13     اخرج_النتائج (عدد_المعدلات، معدل_القسم، ادنى_معدل_مسموح_به، عدد_الراسبين)
14 }
```

النص 6 : نص غير كامل للخوارزم نتائج_القسم_في_مادةعلى الطريقة الثانية، مستعملا المتغيرات العامة

ج – خلاصة فيما يخص خصائص الوظيفة 'تحصل_على_المعدلات':

اذا اخذنا بعين الاعتبار سهولة كتابة رأس الخوارزم ' **نتائج_القسم_في_مادة '**، فإننا نميل الى استعمال طريقة غير مرغوبة فيها وهي طريقة المتغيرات العامة، وهكذا يصبح الخوارزم **'نتائج_القسم_في_مادة'** على صيغة النص 6 وهي طبعا صيغة غير كاملة لحد الآن.

د – سلوك الخوارزم ' نتائج_القسم_في_مادة ' في نسخته على الطريقة الثانية (النص 6):

بعد تهيئة مختلف المتغيرات العامة والخاصة، يطلب الخوارزم في اول الأمر خدمة الوظيفة **'تحصل_على_المعدلات'** التي تقوم بالتقاط ما يدخله المستعمل وتخزينه في متغيرات المجال العام وفي نهايتها وعبر منفذ الرجوع، ترجع الى الإجراء' **نتائج_القسم_في_مادة'** معلومة تفيد بعدد المعدلات التي ادخلها المستعمل، ويقوم الإجراء' **نتائج_القسم_في_مادة '** بحفظ هذه المعلومة في المتغيرة **'عدد_المعدلات'** (السطر 9 من النص 6).

بعد ان شُحنت متغيرات المجال العام بمختلف المعدلات التي ادخلها المستعمل، يطلب الإجراء **'نتائج_القسم_في_مادة'** من الوظيفة **'احسب_معدل_القسم'** ان تحسُب له معدل القسم، فتقوم هذه الوظيفة بزيارة كل المتغيرات العامة التي استعملت في تخزين ما ادخله المستعمل، ولمعرفة المتغيرات التي استعملت في تخزين المعدلات من التي لم تستعمل، تشترط الوظيفة من طالبها توفير عدد المعدلات، ولا يمكن للوظيفة **'احسب_معدل_القسم'** ان تحسب المعدل بدون هذه المعلومة.

اثر انتهاء الوظيفة **'احسب_معدل_القسم'** من انجاز عملها يتحصل **'نتائج_القسم_في_مادة'** على معدل القسم عبر منفذ الرجوع التابع للوظيفة **'احسب_معدل_القسم'**، ويخزنه في المتغيرة المحلية **'معدل_القسم'**.

بعد ان تحصل على معدل القسم، يطلب الإجراء ' **نتائج_القسم_في_مادة'** خدمة من الوظيفة **'احسب_ادنى_معدل'**، و فيما يخصنا تقوم هذه الوظيفة بضرب معدل القسم في القيمة **0.8** فقط وترجع القيمة التي تحصلت عليها، وتشترط هذه الوظيفة اعطاءها معدل القسم، وبما ان محتوى هذه الوظيفة صغير جدا، يمكن ان نستبدل طلب التنفيذ بالتعليمة:

```
ادنى_معدل_مسموح_به = معدل_القسم * 0.8 ؛
```

ومع هذا يستحسن ان نعتمد الحل الذي يستعمل وظيفة قائمة بذاتها لحساب ادنى معدل، فيمكن مثلا ان نغير طريقة حساب ادنى معدل مسموح به، وفي هذه الحالة لا نغير شيئا على مستوى الإجراء **'نتائج_القسم_في_مادة'**، بل نسدد فقط الوظيفة **'احسب_ادنى_معدل'.**

بعد هذا، ينتقل الخوارزم الى نشاط آخر وهو التعرف على عدد الراسبين في المادة، ولبلوغ هذا الهدف، يطلب الإجراء **'نتائج_القسم_في_مادة'** خدمة من الوظيفة **'عد_معدلات_الراسبين'**، التي تقوم بزيارة كل المتغيرات العامة التي استعملت في تخزين المعدلات وتحسب عدد المعدلات التي تكون قيمتها اقل من محتوى المتغيرة **'ادنى_معدل_مسموح_به'**، ولتمكينها من معرفة المتغيرات التي استعملت في تخزين المعدلات، تشترط الوظيفة **'عد_معدلات_الراسبين'** من طالب خدمتها ان يُوفر لها عدد المعدلات التي ادخلها المستعمل.

2-4 المثال الثاني: خوارزم تحديد من هو الأصغر سنا:

من خلال هذا المثال نريد بلوغ الأهداف التالية:

- اظهار مزايا هيكلة الخوارزميات الى اجراءات ووظائف.
- تبيان صعوبة استعمال الأنماط الأساسية للتمثيل الفعال للمعلومات المعقدة.
- تبيان كيفية اعادة استعمال الإجراءات والوظائف التي انجزت.

نسمي **'ايهما_الأصغر'** خوارزم هذا المثال، ويقوم هذا الخوارزم بقبض المعلومات الشخصية لشخصين، ثم يخرج على الشاشة المعلومات المتعلقة بالشخص الأصغر سنا، وفي هذا المثال نكتف بالمعلومات التالية حول الشخص: **'الاسم'**، **'اللقب'**، **'تاريخ الميلاد'**، **'العنوان الشخصي'** و **'الطول'.**

نرى جليا انه ما عدا معلومة **تاريخ الميلاد**، كل المعلومات الأخرى يمكن تمثيلها مستعملين الانماط البدائية او النمط **سلسلة**، **فالاسم**، **واللقب والعنوان الشخصي** تُمثّل بالنمط **سلسلة حروف**، و**الطول** يمثل بالنمط **حقيقي**.

الإشكال الأول المطروح في هذا المثال هو كيفية تمثيل **تاريخ الميلاد**، ولحل هذا الإشكال علينا ان نتمعن في مفهوم تاريخ الميلاد، فعند تمعننا فيه نراه مكونا (او مركبا) من ثلاث معلومات هي : **السنة**، **والشهر**، **و اليوم**، وهنا نرى انه بإمكاننا ان نمثل هذه المعلومات الثلاث مستعملين الانماط البدائية او النمط **سلسلة**، فمثلا يمكن ان نستعمل النمط **سلسلة** لتمثيل **الشهر** والنمط **طبيعي** لتمثيل **السنة**

و**اليوم**، كما يمكننا ان نستعمل النمط **طبيعي** لتمثيل المعلومات الثلاث، وفي هذا المثال نأخذ بهذا الحل الأخير، اي تمثيل **السنة** و **الشهر** و **اليوم** باستعمال النمط **طبيعي**.

1-2-4 السلوك النظري للخوارزم 'ايهما_الأصغر':

يقوم الخوارزم **'ايهما_الأصغر'**: بقبض المعلومات حول الشخص الأول، ثم يقبض معلومات الشخص الثاني، وبعدها يقوم بمقارنة تاريخ ميلاد الشخص الأول بتاريخ ميلاد الشخص الآخر ليحدد اصغرهما، وفي الأخير يكتب على الشاشة لقب واسم اصغر شخص، ويظهر. هذا السلوك في **خطأ: لم يُعثر على مصدر المرجع**.

```
هذا الخوارزم  يقبض معلومات عن شخصين ثم يبين اصغرهما
اليوم هو : الخميس 3  جمادى الأول 1440 الموافق ل10 جانفي 2016
فضلا ادخل تباعا معلومات الشخص الأول:
ادخل اسم الشخص
سليم
ادخل لقب الشخص
عبد المومن
ادخل سنة الميلاد الشخص
1995
ادخل شهر الميلاد الشخص
3
ادخل يوم الميلاد الشخص
26
ادخل عنوان الشخص
شارع عبد المومن بن علي، البويرة، رقم 28، 10000
ادخل طول  الشخص
1.76
فضلا ادخل تباعا معلومات الشخص الثاني:
ادخل اسم شخص
سمير
ادخل لقب شخص
بن عبد الله
ادخل سنة الميلاد شخص
1997
ادخل شهر الميلاد شخص
7
ادخل يوم الميلاد شخص
9
ادخل عنوان شخص
شارع يوسف بن تاشفين، البويرة، رقم 11، 10008
ادخل طول  الشخص
1.89
اصغر شخص هو: سمير  بن عبد الله
الى اللقاء
```

الشكل 2: نتيجة تنفيذ خوارزم اصغر_طالب

2-2-4 المراحل الكبرى للخوارزم:

تنبيه هام: في هذه المرحلة التي نفكر في كيفية انجاز المراحل الكبرى للخوارزم، لسنا مطالبين بسرد التفاصيل فيما يخص كل اجراء او وظيفة مكلفة بمرحلة من مراحل الخوارزم، ولذا، نستعمل ما يطلق عليه بإجراءات او وظائف فارغة، اي ان جسم الوظيفة او الإجراء لا يحتوي على اي تصريح او تعليمة، فيحتوي على الحاضنتين فقط، وفي الوقت المناسب سوف نتطرق لإنجاز محتوى كل إجراء او وظيفة.

يمر هذا الخوارزم بخمسة مراحل للوصول الى النتيجة التي يبرزها على الشاشة (خطأ: لم يُعثر على مصدر المرجع)، وفي النص 7، وعلى مستوى الإجراء **اساسي**، نرى ان كل مرحلة ممثلة بطلب تنفيذ اجراء او وظيفة.

4 -2 - 3 المراحل الخمسة هي على التوالي:

- المرحلة الأولى **مخصصة لتقديم الخوارزم نفسة، ونوكلها الى الإجراء** مقدمة_عن_الخوارزم()
- **المرحلة الثانية:** يطلب معلومات الشخص الأول، ونوكل هذه المهمة الى الإجراء **'معلومات_الشخص'**، و يكون راس هذا الإجراء على الشكل التالي:

اجراء معلومات_الشخص(سلسلة &الاسم،&اللقب،&العنوان؛ طبيعي &السنة،&الشهر،&اليوم؛ حقيقي &طول)

ونلاحظ هنا استعمال الصلات في المنافذ لكون كل منافذ الإجراء مخارج.

- **المرحلة الثالثة:** يطلب معلومات الشخص الثاني، ونستعمل نفس الإجراء: **'معلومات_الشخص'**
- **المرحلة الرابعة** : يقارن بين اعمار الشخصين لإيجاد اصغرهما، ونوكل هذا العمل الى الوظيفة **'قارن_تاريخين()'**، ويكون رأس هذه الوظيفة على الشكل التالي:

طبيعي قارن_تاريحين(طبيعي سنة1، شهر1، يوم1، سنة2، سهر2، يوم2)

```
خواروم ايهما_اصغر {
    اجراء مقدمة_عن_الخوارزم() { }
    اجراء معلومات_الشخص(سلسلة &الاسم، &اللقب & العنوان؛ طبيعي &السنة، &الشهر، &اليوم؛ حقيقي &طول) { }
    طبيعي قارن_تاريحين(طبيعي سنة1، شهر1، يوم1، سنة2، شهر2، يوم2) { }
    اجراء اخرج_النتيجة(سلسلة القب، الاسم) { }
    اجراء اساسي (){
        // التصريح بالمتغيرات الممثلة للشخصين
        .....
        مقدمة_عن_الخوارزم() ؛ // المرحلة الأولى
        // الحصول على معلومات الشخص الأول
        اكتب " فضلا ادخل تباعا معلومات الشخص الأول"
        معلومات_الشخص(الاسم1، اللقب1 ،العنوان1،السنة1، الشهر1، اليوم1، طول1) ؛ // المرحلة الثانية
        // الحصول على معلومات الشخص الثاني
        اكتب " فضلا ادخل تباعا معلومات الشخص الثاني"
        معلومات_الشخص(الاسم2، اللقب2 ،العنوان2،السنة2، الشهر2، اليوم2، طول2) ؛ // المرحلة الثالثة
        طبيعي نتيجة = قارن_تاريحين(سنة1، شهر1، يوم1، سنة2، شهر2، يوم2)؛ //المرحلة الرابعة
```

```
        اذاكان (نتيجة == 0)  اكتب "التاريخين متساويين'
        والا  اذاكان  (نتيجة_مقارنة == 1) اخرج_النتيجة(اسم1، لقب1)
              والا اخرج_النتيجة(اسم2، لقب2)؛  // المرحلة الخامسة
        { // نهاية الإجراء اساسي
}// نهاية الخوارزم ايهما_اصغر
```

النص 7 : نص غير كامل للخوارزم ايهما_اصغر

وترجع الوظيفة **'قارن_تاريخين'** إحدى القيم التالية:

- 0 اذا كان التارخين متساويين
- 1 اذا كان التاريخ الأول (**سنة1، شهر1، يوم1**) احدث من التاريخ الثاني (**سنة2، سهر2، يوم2**).
- –1 اذا كان التاريخ الأول (**سنة1، شهر1، يوم1**) أقدم على التاريخ الثاني (**سنة2، سهر2، يوم2**).

- **المرحلة الخامسة**:اخراج اسم اصغر شخص، ويتحمل الإجراء **'اخرج_النتيجة(سلسلة اسم، لقب)'** انجاز هذه المرحلة.

4 –2 – 4 كيفية تمثيل الشخصين:

لحد الآن ليس لنا الا الأنماط البدائية لتمثيل مختلف خصائص الشخصين، وكما يظهر في رأس الإجراء **'معلومات_الشخص'**، فان مختلف خصائص الشخصين تُمثل بالمتغيرات التالية:

- بالنسبة للشخص الأول، نجد المتغيرات من النمط سلسلة: **الاسم1، اللقب1، العنوان1** وبالنسبة للشخص الثاني نجد المتغيرات من النمط سلسلة **الاسم2، اللقب2، العنوان2**
- المتغيرات الممثلة لتاريخ الميلاد وهي من النمط **طبيعي: السنة1، الشهر1، اليوم1** بالنسبة للشخص الأول و **السنة2، الشهر2، اليوم2** بالنسبة للشخص الثاني.
- المتغيرة من النمط **حقيقي** الموجهة لحفظ الطول وهي:**طول1** بالنسبة للشخص الأول و**طول2** بالنسبة للشخص الثاني.

فيما يخصنا، نصرح بكل هذه المتغيرات في الإجراء اساسي كما يظهر ذلك في النص 8.

```
خواروم ايهما_اصغر {
        اجراء مقدمة_عن_الخوارزم() { }
        اجراء معلومات_الشخص(سلسلة &الاسم، &اللقب & العنوان؛ طبيعي &السنة، &الشهر، &اليوم؛ حقيقي &طول) { }
        طبيعي  قارن_تاريخين(طبيعي سنة1، سهر1، يوم1، سنة2، سهر2، يوم2) { }
        اجراء اخرج_النتيجة(سلسلة القب، الاسم) { }
        اجراء  اساسي ()}
            // التصريح بالمتغيرات الممثلة للشخصين
            سلسلة  الاسم1، اللقب1، العنوان1 ، الاسم2، اللقب2، العنوان2 ؛
            طبيعي  السنة1، الشهر1، اليوم1، السنة2، الشهر2، اليوم2 ؛
            حقيقي  طول1، طول2 ؛
                .....
        { // نهاية الإجراء اساسي
```

```
} // نهاية الخوارزم خواروم ايهما_اصغر
```

النص 8 : نص غير كامل للخوارزم ايهما_اصغرز التصريح بالمتغيرات الممثلة للشخصين في الإجراء اساسي

4 -2 – 5 انجاز إجراءات و وظائف الخوارزم 'ايهما_اصغر':

أ – الإجراء مقدمة_عن_الخوارزم(): يُظهر النص 9 محتوى الإجراء **'مقدمة_عن_الخوارزم()'**، ومن خلال هذا المحتوى نرى استعمال وظيفتين معتادتين توفرها اللغة الشرمزية: الوظيفة **'تاريخ_اليوم(سلسلة نوعية_التاريخ)'** والوظيفة **'اسم_اليوم(سلسلة تاريخ، نوعية_التاريخ)'.**

الوظيفة المعتادة **'تاريخ_اليوم(سلسلة نوعية_التاريخ)'** ترجع تاريخ اليوم حسب نوعية التاريخ، **هجري** او **ميلادي**، فان نفذت الوظيفة **تاريخ_اليوم("هجري")** في يوم الخميس 3 جمادى الأول 1440 ترجع الوظيفة السلسلة "**3 جمادى الأول 1440**" وان نفذت الوظيفة **تاريخ_اليوم("ميلادي")** في نفس اليوم، ترجع هذه الوظيفة السلسلة "**10 جانفي 2016**".

الوظيفةالمعتادة **'اسم_اليوم'** ترجع اسم اليوم انطلاقا من تاريخ محدد، هجري او ميلادي، فمثلا اذا كتبنا التعليمة **اسم_اليوم("3 جمادى الأول 1440")** او التعليمة **'اسم_اليوم("10 جانفي 2016")'** فان النتيجة تكون السلسلة "**الخميس**".

```
خواروم ايهما_اصغر {
    اجراء مقدمة_عن_الخوارزم() {
        اكتب " هذا الخوارزم  يقبض معلومات عن شخصين ثم يبين اصغرهما"
        سلسلة تاريخ_هجري = تاريخ_اليوم("هجري") ؛
        سلسلة تاريخ_ميلادي = تاريخ_اليوم("ميلادي") ؛
        سلسلة اليوم = اسم_اليوم(تاريخ_ميلادي) ؛
        اكتب "اليوم هو" + اليوم + تاريخ_هجري + "الموافق ل" + تاريخ_ميلادي ؛
    }
    اجراء معلومات_الشخص(سلسلة &الاسم، &اللقب & العنوان؛ طبيعي &السنة، &الشهر، &اليوم؛ حقيقي &طول) { }
    طبيعي  قارن_تاريخين(طبيعي سنة1، سهر1، يوم1، سنة2، سهر2، يوم2) { }
    اجراء اخرج_النتيجة(سلسلة القب، الاسم) { }
    اجراء  اساسي (){
        .....
    { // نهاية الإجراء اساسي
}   // نهاية الخوارزم ايهما_اصغر
```

النص 9 : المحتوى الكامل للإجراء مقدمة_عن_الخوارزم

ب – الإجراء 'معلومات_الشخص': هذا الإجراء الذي تظهر تفاصيله في النص 10، يقوم بالتفاعل مع المستعمل ويخزن ما يدخله المستعمل في متغيرات يصل اليها الإجراء عبر **الصلات**، فكل منافذ الإجراء مخارج ممثلة **بالصلات**.

```
خواروم ايهما_اصغر {
    اجراء مقدمة_عن_الخوارزم() {
        .....
    }
    اجراء معلومات_الشخص(سلسلة &الاسم، &اللقب & العنوان؛ طبيعي &السنة، &الشهر، &اليوم؛ حقيقي &طول) {
        اكتب ("ادخل اسم الشخص")
        اقرأ الاسم
        اكتب ("ادخل لقب الشخص")
```

```
        اقرأ اللقب
        اكتب ("ادخل سنة الميلاد الشخص ")
        اقرأ السنة
        اكتب ("ادخل شهر الميلاد الشخص ")
        اقرأ الشهر
        اكتب ("ادخل يوم الميلاد الشخص ")
        اقرأ اليوم
        اكتب ("ادخل عنوان الشخص ")
        اقرأ العنوان
        اكتب ("ادخل طول الشخص ")
        اقرأ طول
    }
طبيعي  قارن_تاريخين(طبيعي سنة1، سهر 1، يوم1، سنة2، سهر 2، يوم2) { }
اجراء اخرج_النتيجة(سلسلة القب، الاسم) { }
اجراء  اساسي (){
    .....
{ // نهاية الإجراء اساسي
}    // نهاية الخوارزم
```

النص 10 : المحتوى الكامل للإجراء معلومات_الشخص

ج – الوظيفة قارن_تاريخين:

يَظهر في النص 11 محتوى الوظيفة **'قارن_تاريخين'**، و تبدأ بمقارنة السنتين، فان وجدت انهما مختلفتين، تُحسم النتيجة ولا داعي لمقارنة الأشهر والأيام، ففي التعليمة الأولى ان صحت العبارة هل السنة الأولى اكبر من السنة الثانية، يُحسم الأمر وتَرجع الوظيفة **'قارن_تاريخين'** القيمة 1 مُخبرة أن التاريخ الأول أحدث من التاريخ الثاني، واذا لم تصح عبارة السطر الأول، ينتقل التنفيذ الى التعلييمة الثانية، وان صحت عبارة التعليمة الثانية، تُرجع الوظيفة القيمة –1، اي ان التاريخ الأول اقدم من التاريخ الثاني، واذا لم تصح هذه العبارة الثانية، فمعنى هذا ان السنتين متساويتين، ولحسم الأمر يجب مقارنة الأشهر. بنفس المنطق، اي اذا كان شهر التاريخ الأول اكبر من شهر التاريخ الثاني (السطر 3 من النص 11) يُحسم الأمر و ترجع الوظيفة ' **قارن_تاريخين'** القيمة 1، والا نحاول ان نرى هل شهر التاريخ الأول اصغر من شهر التاريخ الثاني، فاذا صحت هذه العبارة (السطر 4 من النص 11) يُحسم الأمر وتُرجع الوظيفة ' **قارن_تاريخين'** القيمة –1، وان لم تصح العبارتين (السطر 3 و 4 من النص 11)، فمعنى هذا ان الشهرين متساويين، اي اننا في حلة تتساوى فيها السنتين ويتساوى فيها الشهرين للتارخين، وفي هذه الحالة ننتقل الى مقارنة الأيام بنفس المنطق الذي قارنت به الوظيفة السنتين والشهرين.

```
خواروم ايهما_اصغر {
    اجراء مقدمة_عن_الخوارزم() {
        .....
    }
    اجراء معلومات_الشخص(سلسلة &الاسم، &اللقب & العنوان؛ طبيعي &السنة، &الشهر، &اليوم؛ حقيقي &طول) {
        .....
    }
    طبيعي  قارن_تاريخين(طبيعي سنة1، سهر 1، يوم1، سنة2، سهر 2، يوم2) {
        اذاكان(سنة1 < سنة2) ارجع 1 ؛
```

```
        اذاكان(سنة1 > سنة2) ارجع -1 ؛
        اذاكان(شهر1 < شهر2) ارجع 1 ؛
        اذاكان(شهر1 > شهر2) ارجع -1 ؛
        اذاكان(يوم1 < يوم2) ارجع 1 ؛
        اذاكان(يوم1 > يوم2) ارجع -1 ؛
        ارجع 0 ؛
    }
    اجراء اخرج_النتيجة(سلسلة القب، الاسم) { }
    اجراء اساسي (){
        .....
    } // نهاية الإجراء اساسي
} // نهاية الخوارزم
```

النص 11 : المحتوى الكامل للوظيفة قارن_تاريخين

د – الإجراء 'اخرج_النتيجة'

يظهر محتوى الإجراء **'اخرج_النتيجة'** في النص 12، وهو محتوى جد بسيط، اذ يكتفي هذا الإجراء بإخراج السلسلة " اصغر شخص هو:" متبوعا باسم و لقب يأخذهما من مَدْخليه.

```
خواروم ايهما_اصغر {
    اجراء مقدمة_عن_الخوارزم() {
        .....
    }
    اجراء معلومات_الشخص(سلسلة &الاسم، &اللقب &/edà العنوان؛ طبيعي &السنة، &الشهر، &اليوم؛ حقيقي &طول) {
        .....
    }
    طبيعي قارن_تاريخين(طبيعي سنة1، سهر1، يوم1، سنة2، سهر2، يوم2) {
        .....
    }
    اجراء اخرج_النتيجة(سلسلة اللقب، الاسم) {
        اكتب " اصغر شخص هو:" + الاسم + " " + اللقب
    }
    اجراء اساسي (){
        .....
    } // نهاية الإجراء اساسي
} // نهاية الخوارزم
```

النص 12 : المحتوى الكامل للإجراء اخرج_النتيجة

6-2-4 توسيع قدرات الخوارزم 'ايهما_اصغر': لإيجاد اصغر شخص من بين اكثر من شخصين:

الخوارزم **ايهما_اصغر.** في نسخته الحالية يُمكن فقط من مقارنة شخصين فقط، وانجازه باستعمال الأنماط الأساسية سهل نوعا ما كما يظهر من خلال النصوص السابقة، والآن نريد منه ان يصبح قادرا على ايجاد اصغر شخص من بين اكثر من شخصين، مثلا من بين 10 اشخاص، ولجعله كذلك يجب اعادة كتابته وكذلك اعادة تسميته، ونسميه في هذه الحالة **ايهم_اصغر.**

مشاكل أساسية عدة مطروحة في سياق التفكير في انجاز الخوارزم **ايهم_اصغر.**:

- اولا كيف نمثل كل الأشخاص
- ثانيا كيف نقبض معلومات الأشخاص
- ثالثا كيف نقارن كل الأشخاص لإيجاد اصغرهم

- هل يصلح الخوارزم **ايهم_اصغر** لمقارنة عدد غير محدد مسبقا من الأشخاص

أ – كيفية تمثيل كل الأشخاص:

اذا انتهجنا الطريقة السالفة في تمثيل الأشخاص (النص 8)، لا بد لنا من تحديد عدد الأشخاص، فإذا افترضنا ان عدد الأشخاص هو 10(النص 13)، فعلينا ان نصرح بسبعين متغيرة (كل شخص ممثل بسعة متغيرات)، وبعد هذا لن يصبح الخوارزم صالحا لمعالجة معلومات 11 شخصا او اكثر، واذا اردنا ان نكيفه مع 100 شخص علينا بالتصريح بسبعمئة متغيرة، وهذا امر فيه صعوبة كبيرة ويكون مصدرا هائلا لارتكاب الأخطاء في كتابة الخوارزم، حتى لو تم الأمر لن يصلح هذا الخوارزم الذي بذلت فيه جهود كبيرة للتعاطي مع 101 شخص او اكثر.

```
خواروم ايهم_اصغر {
    اجراء مقدمة_عن_الخوارزم() { }
    اجراء معلومات_الشخص(سلسلة &الاسم، &اللقب & العنوان؛ طبيعي &السنة، &الشهر، &اليوم؛ حقيقي &طول) { }
    طبيعي  قارن_تاريخين(طبيعي سنة1، سهر1، يوم1، سنة2، سهر2، يوم2) { }
    اجراء اخرج_النتيجة(سلسلة القب، الاسم) { }
    اجراء  اساسي (){
        // التصريح بالمتغيرات الممثلة لعشرة اشخاص
        سلسلة الاسم1، اللقب1، العنوان1 ، الاسم2، اللقب2، العنوان2 ..... الاسم10، اللقب10، العنوان10 ؛
        طبيعي السنة1، الشهر1، اليوم1، السنة2، الشهر2، اليوم2 ..... السنة10، الشهر10، اليوم10
        حقيقي طول1، طول2 ..... طول10 ؛
        .....
    } // نهاية الإجراء اساسي
} // نهاية الخوارزم
```

النص 13 : نص غير كامل للخوارزم ايهم_اصغر، صالح لعشرة اشخاص. التصريح بالمتغيرات الممثلة لعشرة اشخاص في الإجراء اساسي

```
خواروم ايهما_اصغر {
    اجراء مقدمة_عن_الخوارزم() { }
    اجراء معلومات_الشخص(سلسلة &الاسم، &اللقب & العنوان؛ طبيعي &السنة، &الشهر، &اليوم؛ حقيقي &طول) { }
    طبيعي  قارن_تاريخين(طبيعي سنة1، شهر1، يوم1، سنة2، شهر2، يوم2) { }
    اجراء اخرج_النتيجة(سلسلة القب، الاسم) { }
    اجراء  اساسي (){
        // التصريح بالمتغيرات الممثلة للشخصين
        .....
        مقدمة_عن_الخوارزم() ؛ // المرحلة الأولى
        // التحصل على معلومات الشخص الأول
        اكتب " فضلا ادخل تباعا معلومات الشخص الأول"
        معلومات_الشخص(الاسم1، اللقب1 ،العنوان1،السنة1، الشهر1، اليوم1، طول1) ؛
        // التحصل على معلومات الشخص الثاني
        اكتب " فضلا ادخل تباعا معلومات الشخص الثاني"
        معلومات_الشخص(الاسم2، اللقب2 ،العنوان2،السنة2، الشهر2، اليوم2، طول2) ؛
        .....
        اكتب " فضلا ادخل تباعا معلومات الشخص الخامس"
        معلومات_الشخص(الاسم5، اللقب5 ،العنوان5،السنة5، الشهر5، اليوم5، طول5) ؛
        .....
    } // نهاية الإجراء اساسي
} // نهاية الخوارزم
```

النص 14 : نص غير كامل للخوارزم ايهم_اصغر مهيأ لخمسة اشخاص : قبض معلومات 5 اشخاص

ب – قبض معلومات اكثر من شخصين:

كما في الخوارزم **'ايهما_اصغر'**، نكتب تعليمتين لقبض معلومات كل شخص: الأولى تعليمة تدع الى ادخال معلومات الشخص متبوعة برقم الشخص، والثانية طلب تنفيذ التعليمة **معلومات_الشخص،** وفي المثال التالي نطلب ادخال معلومات الشخص السابع.

```
اكتب " فضلا ادخل تباعا معلومات الشخص السابع" ؛
معلومات_الشخص(الاسم7، اللقب7 ،العنوان7،السنة7، الشهر7، اليوم7، طول7) ؛
```

وهكذا لنتمكن من ادخال 10 اشخاص علينا كتابة 20 تعليمة، اما اذا كان عدد الأشخاص 100 فعلينا كتابة 200 تعليمة، والنص 14 كتب لقبض 5 اشخاص، وهذا النص لا يمكنه التعامل مع 6 اشخاص او اكثر.

ج – مقارنة اكثر من شخصين لإيجاد اصغرهم:

من خلال هذا المثال نبرز اولا المشقة عند استعمال الأنماط الأساسية لتمثيل معطيات معقدة وثانيا نبين كيفية اعادة استعمال الإجراءات والوظائف التي انجزت سابقا.

لمقارنة اكثر من شخصين من اجل إيجاد اصغرهم نستعمل الوظيفة **قارن_تاريخين** ونتبع المراحل التالية:

- نحتاج لمتغيرات مهمتها حفظ معلومات اصغر شخص من بين الأشخاص التي يستهدفهم الخوارزم، ونسمي هذه المتغيرات كالتالي: **لقب_الأصغر ، اسم_الأصغر. ، سنة_الأصغر، شهر_الأصغر، يوم_الأصغر.**
- نفترض في البداية ان الشخص الأول هو الأصغر، فنضغ معلوماته في المتغيرات التي نحفظ فيها الشخص الأصغر، وتتم هذه المرحلة عند التصريح بالمتغيرات التي تبين اصغر شخص، اي المتغيرات **لقب_الأصغر ، اسم_الأصغر. ، سنة_الأصغر، شهر_الأصغر، يوم_الأصغر**
- نبدأ المقارنة بين الشخص الأصغر حاليا (اي الذي معلوماته في المتغيرات **لقب_الأصغر. ، اسم_الأصغر ، سنة_الأصغر شهر_الأصغر، يوم_الأصغر،** وهو الشخص الأول**)** والشخص الثاني، فاذا وجدنا ان الشخص الثاني اصغر من الأصغر حاليا، اي الشخص الأول في هذه المرحلة، نضع معلومات الشخص الثاني في المتغيرات المخصصة للشخص الأصغر، واذا لم يكن الشخص الثاني اصغر من الأول لا يحدث اي تغيير وتبق متغيرات الأصغر على حالها.
- ننتقل الى الشخص الثالث و نقارنه مع الأصغر حاليا، اي مع المعلومات المخزنة في المتغيرات **سنة_الأصغر، شهر_الأصغر، يوم_الأصغر.** فاذا وجدنا ان الشخص الثالث اصغر من الأصغر حاليا، اي الشخص الأول او الثاني حسب ما افرزته المرحلة السابقة، نضع

معلومات الشخص الثالث في المتغيرات المخصصة للشخص الأصغر، واذا لم يكن الشخص الثالث اصغر من الأصغر حاليا، نبق متغيرات الأصغر حاليا على حالها.

ونستمر في هذا المنطق حتى الشخص الأخير، وفي آخر المطاف نتحصل على اصغر شخص في المتغيرات المخصصة لأصغر شخص.

خوارزم النص 15 يتعاطى مع 5 اشخاص، ونرى ان عملية البحث عن الأصغر تكلف كتابة 4 تعليمات، التعليمة الشرطية التي تقارن بين شخصين والتعليمات التي تحدث تغييرا في متغيرات الشخص الأصغر اذا صحت عبارة التعليمة الشرطية، اي ان مجمل التعليمات التي تمكن من ايجاد اصغر شخص من بين 5 اشخاص هي 20 تعليمة، واذا افترضنا ان الخوارزم يجب ان يقارن بين 100 شخص، فعلينا كتابة 400 تعليمة متشابهة تماما، وهنا تبرز لنا الحاجة الى ضرورة البحث عن نمط يمكننا من اعادة استعمال نفس التعليمات في اطار تعليمة تكرار، لا اعادة كتابتها كما نفعل الآن.

```
خواروم ايهما_اصغر {
    اجراء مقدمة_عن_الخوارزم() { }
    اجراء معلومات_الشخص(سلسلة &الاسم، &اللقب & العنوان؛ طبيعي &السنة، &الشهر، &اليوم؛ حقيقي &طول) { }
    طبيعي  قارن_تاريخين(طبيعي سنة1، شهر1، يوم1، سنة2، شهر2، يوم2) { }
    اجراء اخرج_النتيجة(سلسلة القب، الاسم) { }
    اجراء اساسي (){
    // التصريح بالمتغيرات الممثلة لخمسة اشخاص
    .....
    // التصريح بمتغيرات تستعمل لتخزين معلومات اصغر شخص، بالأخص اللقب، الإس و تاريخ الميلاد
    سلسلة لقب_الأصغر = اللقب1، اسم_الأصغر =الإسم1؛
    طبيعي سنة_الأصغر =سنة1، شهر_الأصغر = شهر1، يوم_الأصغر = يوم1 ؛
    مقدمة_عن_الخوارزم() ؛ // تقديم ما يفعله الخوارزم
    // التحصل على معلومات الشخص الأول
    اكتب " فضلا ادخل تباعا معلومات الشخص الأول"
    معلومات_الشخص(الاسم1، اللقب1 ،العنوان1،السنة1، الشهر1، اليوم1، طول1) ؛
    .....
    // التحصل على معلومات الشخص الخامس
    اكتب " فضلا ادخل تباعا معلومات الشخص الخامس"
    معلومات_الشخص(الاسم5، اللقب5 ،العنوان5،السنة5، الشهر5، اليوم5، طول5) ؛
    // بداية عمليات المقارنة للحصول على اصغر شخص
    اذاكان (قارن_تاريخين( سنة2، سهر2، يوم2، سنة_الأصغر ، شهر_الأصغر ، يوم_الأصغر)) == 1 {
        // اذا صحت العبارة فمعنى هذا ان الشخص الثاني اصغر من الشخص الأول
        //وبهذا يصبح الشخص الثاني هو الشخص الأصغر
        سنة_الأصغر  = سنة 2؛
        شهر_الأصغر = شهر2 ؛
        يوم_الأصغر = يوم2 ؛
    }
    ....
    ....
    اذاكان (قارن_تاريخين( سنة5، سهر5، يوم5، سنة_الأصغر ، شهر_الأصغر ، يوم_الأصغر)) == 1 {
        // اذا صحت العبارة فمعنى هذا ان الشخص الخامس هو الأصغر
        // وبهذا يصبح الشخص الخامس هو الشخص الأصغر
        سنة_الأصغر  = سنة 2؛
        شهر_الأصغر = شهر2 ؛
        يوم_الأصغر = يوم2 ؛
    }
    // على هذا المستوى يكون الخوارزم قد تحصل على اصغر شخص و معلوماته في متغيرات اصغر شخص
```

```
        // لم يبق للخوارزم الا اخراج النتيجة
        اخرج_النتيجة(اسم_الأصغر، لقب_الأصغر)
        { // نهاية إجراء اساسي
    }   // نهاية الخوارزم
```

النص 15 : نص غير كامل للخوارزم ايهم_اصغر مهيأ لخمسة اشخاص : مقارنة 5 اشخاص

هـ – ضرورة تسديد الإجراء مقدمة_عن_الخوارزم()

نلاحظ ان أغلب الإجراءات والوظائف قد تم اعادة استعمالها دون اي تغيير، وهذا لكونها قد انجزت لتكون غير مرتبطة بأي خوارزم او محيط، اما الإجراء **مقدمة_عن_الخوارزم()** فقد انجز وفيه تعليمة مرتبطة ارتباطا وثيقا بالخوارزم **ايهما_اصغر.** وهذا الارتباط ظاهر في التعليمة الأولى التي تحدد عدد الأشخاص، وهما شخصان، فعلينا:

- اما تسديد هذا الإجراء لجعله غير مرتبط بالخوارزم **ايهما_اصغر،** فيمكن بذلك استعماله في الخوارزم **ايهم_اصغر.**
- او اعادة كتابة الإجراء ليكون مرتبطا بالخوارزم **ايهم_اصغر.** الذي يتعاطى مع 5 اشخاص

```
خواروم ايهم_اصغر {
    اجراء مقدمة_عن_الخوارزم(طبيعي عدد_الأشخاص) {
    اذاكان (عدد_الأشخاص > 2 ) اكتب_سطر"خطأ فادع... هذا الخوارزم لا يمكنه التعاطي مع اقل من شخصين "
    والا اذاكان (عدد_الأشخاص ==2)
        اكتب_سطر" هذا الخوارزم  يقبض معلومات شخصين ثم يبين اصغرهما"
    والا اذاكان (عدد_الأشخاص > 11)
        اكتب_سطر" هذا الخوارزم  يقبض معلومات " + عدد_الأشخاص + "اشخاص ثم يبين اصغرهم"
    والا اكتب_سطر" هذا الخوارزم  يقبض معلومات " + عدد_الأشخاص + "شخصا ثم يبين اصغرهم"
    سلسلة تاريخ_هجري = تاريخ_اليوم("هجري") ؛
    سلسلة تاريخ_ميلادي = تاريخ_اليوم("ميلادي"} ؛
    سلسلة اليوم = اسم_اليوم(تاريخ_ميلادي) ؛
    اكتب_سطر" "اليوم هو" + اليوم + تاريخ_هجري + "الموافق ل" + تاريخ_ميلادي ؛
                                                                    }
                                                                    ....
}   // نهاية الخوارزم  ايهم_اصغر
```

النص 16 : تسديد نص الإجراء مقدمة_عن_الخوارزم

الحل الجيد هو جعل الإجراء غير مرتبط بعدد الأشخاص، اي غير مرتبط بالخوارزم **ايهما_اصغر** والخوارزم **ايهم_اصغر** الذي يتعاطي مع 5 اشخاص او مع الخوارزم **ايهم_اصغر.** الذي يتعاطي مع 1000 شخص ، وهذا ما نراه في النص 16، اين جعل عدد الأشخاص معلومة توفر للإجراء عبر مدخل.

5– ضعف استعمال الأنماط الأساسية لتمثيل المعطيات المعقدة:

من خلال المثالين السابقين رأينا ان استعمال الأنماط الأساسية لتمثيل المعطيات المعقدة تؤدي الى المشقات التالية:

- صعوبة تمثيل المعلومات المعقدة خاصة اذا كثرت، وهنا نلجأ الى عدد ضخم من المتغيرات، واذا كان العدد ضخما وجب علينا ايجاد طريقة جيدة للتحكم في العدد الضخم للمتغيرات.
- كتابة عدد ضخم من التعليمات للتعاطي مع عدد محدد من المعلومات المعقدة.
- الخوارزميات المنتجة مرتبطة عضويا مع عدد محدد من المعطيات، ولا يمكن للخوارزم التعاطي مع عدد اكبر
- التعليمات التي تكتب هي في حقيقتها نفس التعليمات مع اختلاف طفيف موجود في اسم المتغيرات، فلو كانت هناك طريقة غير طريقة الأنماط البدائية لتمثيل المعطيات المعقدة لكتبت التعليمات مرة واحدة في اطار تعليمة تكرار، وفي كل تكرار تتعاطى التعليمات مع معطيات اخرى، لكن استعمال المتغيرات البدائية المرتبطة عضويا باسم قد حدده كاتب الخوارزم لا يُمكننا من إنجاز تعليمة تكرار، وهنا علينا البحث عن طريقة يمكن بها الوصول لمتغيرة ما بطريقة اخرى غير طريقة اسمائها، وهذا ممكن كما رأينا في آخر الجزء الأول باستعمال الصِلات، فالصلات تحتوي على عنوان متغيرة ما، فاذا تمكننا من الوصول الى متغيرة ما، فيمكننا ايضا الوصول الى متغيرة جارة بإحداث تغيير في معلومات الوصول الى المتغيرات والموجودة في الصلات.

6 – ضرورة توفير طرق قوية لتمثيل المعلومات المعقدة

من المثالين السابقين، تظهر جليا ضرورة ايجاد آليات تمكننا من التمثيل السهل والسريع للمعطيات المعقدة، ففي المثالين السابقين، لمسنا صعوبة كتابة خوارزميات مع معطيات يمكن ان نصفها كمعلومات بسيطة اذا قورنت بالمعلومات التي توجد في انظمة محاكاة امر ما من امور الواقع المعاش، كالأنظمة المعينة على قيادة الطائرات او الانظمة المستعملة في كشف الأمراض واعانه الأطباء الى غير ذلك.

وحتى نتمكن من توفير الاليات التي تمكن لغة ما من لغات الخوارزميات او لغة ما من لغات البرمجة من التعامل الفعال مع المعطيات المعقدة، علينا اولا من فهم التعقيد، وابراز نوعياته ان وجدت، فان وجدت يمكن ان نتعامل مع كل نوعية لنوفر لها آلية خاصة بتمثيلها.

7 – انواع المعلومات المعقدة:

مهما يكن تعقيد المعلومة، فهي في حقيقة امرها منظمة تنظيما محكما، وتكون على شكلين اساسين:

- معلومة مهيكلة ، مبنية على اساس معلومات اقل تعقيدا.

- معلومة مجمعة، وفيها تجتمع معلومات يمكن ان تكون بدائية، او مهيكلة، لكن من نفس الصنف، فالمعلومة المجمعة تلعب دور الوعاء لمعلومات اخرى.

الفصل الثاني

المعلومات المهيكلة

الجزء الثاني: الفصل الثاني: المعلومات المهيكلة

1 – مقدمة:

كما اوردناه في الجزء الأول، فان الانظمة المعلوماتية قد اجتاحت اليوم اغلب ميادين الحياة، فإذا كانت هذه الانظمة قد اقتصرت فيما سبق على ميادين محدودة لا تحتوي الا على اعداد قليلة من المفاهيم البسيطة والمتجانسة، والتي يمكن تمثيلها، في بعض الأحيان ببعض الصعوبات، باستعمال الانماط الأساسية (او القاعدية او البدائية) كالأعداد الطبيعية والاعداد الحقيقية والحروف، فإنها اليوم، اي الانظمة المعلوماتية، حاضرة في ميادين لا يمكن حصرها بسهولة، واكثرية هذه الميادين متسمة بمفاهيم جد معقدة، لا يمكن تمثيلها بالأنماط القاعدية الا بشق الأنفس وطول الزمن وبتكاليف جد باهظة لا جدوى من تحملها.

حتى يتمكن صانعوا الخوارزميات والبرامج من التعامل السهل والفعال مع المفاهيم المعقدة الموجودة في الواقع المعاش، طورت لغات الخوارزميات ولغات البرمجة، فأدخلت فيها آليات وطرق عدة تمكن من التمثيل السهل والمباشر لأي مفهوم ولخصائصه، ويمثل استعمال هذه الآليات والطرق نشاطا هاما في سياق انجاز الخوارزميات، وغالبا ما يسمى هذا النشاط **بتنظيم المعلومات**، او **هيكلة المعلومات**، وهدف هذا النشاط هو:

- تحديد المفهوم الذي نسعى الى تمثيله.
- تحديد الخصائص (او السمات) المهمة للمفهوم.
- تعريف المعلومة التي تمثل المفهوم ويكون هذا بتعريف هيكل المفهوم وموقع كل سمة في الهيكل الممثل للمفهوم.

1– 1 التعريف:

المعلومة المهيكلة معلومة مكونة من معلومات اخرى اقل تعقيدا، نسميها **المعلومات الجزئية** او **السمات**، وتمثل كل **سمة** خاصية من خصائص المفهوم الذي تمثله المعلومة المهيكلة، وكما ان لكل خاصية **إسما** خاص بها يدل عليها وحدها في اطار المفهوم، فان لكل **معلومة جزئية** (او **سمة**) اسما خاصا بها يدل عليها في اطار المعلومة المهيكلة، فالمعلومة المهيكلة في حقيقتها اطار تتواجد فيه **السمات** (او **المعلومات الجزئية**).

1 – 2 المثال الأول: المعلومة الممثلة لمفهوم الشخص:

مفهوم الشخص مفهوم معقد جدا من مفاهيم الواقع المعاش، ونمثله فيما يلي بمعلومة مهيكلة تحتوي على المعلومات الجزئية (او السمات) التالية:

- **السمة** الممثلة **لإسم الشخص**، واسمها في الغالب هو **الاسم**، وعلى مستوى الخوارزميات يمثل الإسم **بسلسلة من الحروف**، لذا يمكن بسهولة استعمال النمط **المعتاد** الذي سميناه **سلسلة (سلسلة حروف)** لتمثيل **السمة** الممثلة لاسم الشخص.
- **السمة** الممثلة **للقب الشخص**، واسمها في الغالب هو **اللقب**، وهذه المعلومة من النمط الشائع **سلسلة**
- **السمة** الممثلة لطول الشخص، واسمها في الغالب هو **الطول**، وهذه المعلومة من النمط القاعدي **حقيقي.**
- **السمة** الممثلة للحالة المدنية، وهي تبين هل الشخص متزوج ام لا، ويمكن ان يكون اسمها **الحالة المدنية**، والقيم التي يمكن ان تمثلها هذه المعلومة محدودة جدا، فهي على سبيل المثال: **اعزب، متزوج، مطلق**، ويمكن ان يكون النمط المستعمل في تمثيلها النمط **سلسلة**، كما يمكن ان يكون اي نمط آخر قادر على تمثيل الحالات الثلاث، فمثلا، القيمة 1 تمثل الحالة اعزب، القيمة 2 تمثل الحالة متزوج، و القيمة 3 تمثل الحالة مطلق، وهذه القيم من النمط **طبيعي**.
- **السمة** الممثلة لتاريخ الميلاد، و يمكن ان يكون اسمها **تاريخ الولادة**، وهذه المعلومة في حد ذاتها معلومة تمثل مفهوما مكونا من ثلاث مفاهيم على الأقل او خمسة مفاهيم اذا اردنا ان يكون التاريخ دقيقا، وهذه المفاهيم هي **نوعية التاريخ** (هجري، ميلادي، فلاحي الخ) **السنة، الشهر، اليوم، والساعة.**

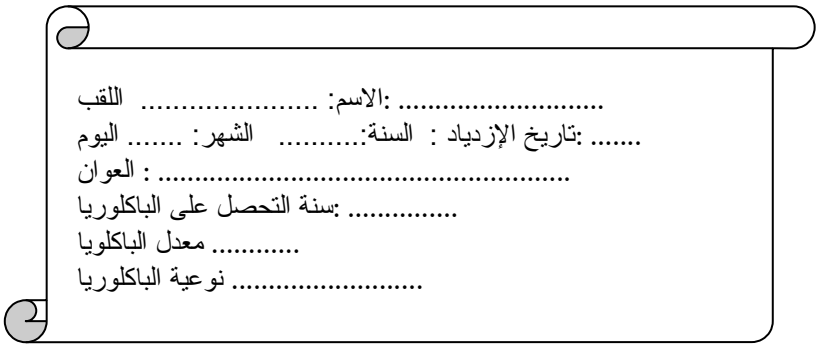
وهكذا نرى جليا ان مكونات المعلومات المهيكلة يمكن ان تكون معلومات قاعدية او معلومات مهيكلة اخرى كسمة **تاريخ الولادة.**

1 – 3 المثال الثاني: المعلومة الممثلة لمفهوم الطالب

في الواقع المعاش، عندما يريد استاذ ما معرفة جيدة لطلبته، يقوم بتحضير ورقة نموذجية تمثل طالبا ما، كتلك التي تظهر في خطأ: لم يُعثر على مصدر المرجع، ويضح في الورقة التي تصف طالبا ما **اسم لكل معلومة** مرغوب فيها من المعلومات المتعلقة بالطالب، **كالإسم، واللقب، والعنوان، ومعدل الباكالوريا** الخ... وفي الورقة الممثلة لطالب ما، يوفر الاستاذ لكل معلومة مرغوب فيها مكانا خاصا بها.

ملاحظة: هناك تشابه كبير بين **اسم المعلومة الجزئية** (او **السمة**) والمكان المخصص للمعلومة **(النقاط التي تتبع اسم المعلومة)** وبين المتغيرات في كتابة الخوارزميات، **فالمتغيرات ايضا لها اسم ومكان في الذاكرة توضع فيه المعلومة**، وهذا ما يجعلنا نقول ايضا ان **المعلومة المهيكلة** سوف يحدد هيكلتها **نمط مُهيكل**، وسوف تحتضنها. **متغيرة مهيكلة** مكونة من **متغيرات جزئية** (او **سمات**) يمكن ان تكون بدائية

او غير بدائية (اي مهيكلة) ولها مكان محدد في الذاكرة يحتوي على الأماكن المخصصة للمعلومات الجزئية (او **السمات**) وهي **متغيرات جزئية،** تكون غالبا متجاورة فيما بينها في الذاكرة.



الشكل 1 : نموذج اعده استاذ ما للحصول على بعض المعلومات حول طلبته

بعد ان يرضى الاستاذ عن النموذج الذي يستعمله للحصول على معلومات طلبته، يقوم بتصوير النموذج ليتحصل على عدد من النسخ تكف الطلبة، ونراه ايضا يحتفظ بالصورة الأصلية للنموذج، ثم يوزع النسخ على الطلبة، ليضع كل طالب المعلومات الخاصة به في النسخة التي اعطيت له.

ملاحظة: نرى هنا التشابه الكبير بين **النموذج الورقي** و**نمط المعلومات،** وبين **النسخ** التي انتجت انطلاقا من **النموذج الورقي** و**المتغيرات** التي تُنشأ انطلاقا من **نمط المعلومات،** وهذا يؤدي بنا للتفكير في انشاء **انماط للمعلومات المهيكلة** كما انشأ الاستاذ **نموذجا ورقيا** من اجل الحصول على المعلومات التي تهمه حول طلبته.

عندما يملأ الطالب النسخة التي اعطيت له، نراه يستعمل **اسم السمة** ليعرف المكان الذي تكتب في المعلومة، فبفضل الاسم **شهر** يضع الطالب المعلومة الدالة على شهر الازدياد في المكان المناسب المحاذي لكلمة **شهر**، ولا يضعها في اي مكان آخر.

2 – آليات تمثيل و استغلال المعلومات المهيكلة

2 – 1 : التعريف بالأنماط المهيكلة

تمتاز اللغات الشبه الرمزية و لغات البرمجة بخاصية التعريف بما يسمى **الأنماط المهيكلة للمعلومات** التي تستعمل لإنشاء **المتغيرات المهيكلة،** ويتم تعريف نمط مهيكل باستعمال الكلمتين **عرف نمط** متبوعتين باسم **النمط،** ثم تكتب **سمات النمط** (اي مكوناته او معلوماته الجزئية) بين حاضنتين، كما يظهر في النص 17، وتُعَرفُ سمات النمط المهيكل، بالطريقة نفسها التي تصرح بها المتغيرات، فنكتب اولا اسم نمط ما **يكون معلوما،** متبوعا باسم هو اسم **السِمة.**

النمط المعلوم، في سياق تعريف نمط مهيكل، هو احد الأنماط التي توفرها لغة الخوارزميات اولغة البرمجة (**طبيعي، حقيقي، سلسلة، منطقي** الخ) او اي **نمط مهيكل قد عُرِف من قبل،** وهذا ما

يظهر في تعريف النمط **شخص** (النص 17)، فقد عرفنا اولا بنمط اسمه **تاريخ** ثم بنمط ثان اسمه **تاريخ_دقيق** وفي سياق التعريف بالنمط **شخص** استعملنا النمط **تاريخ**.

وفي سياق تعريف نمط مهيكل ما، يمكن ان **نصف قيمة اولية** لأي سمة شئنا، كما هو الحال مع السمة المسماة **الثانية** في تعريف النمط **تاريخ_دقيق** (النص 17).

```
عرف نمط تاريخ {
    طبيعي  السنة، الشهر، اليوم
}
عرف نمط تاريخ_دقيق {
    طبيعي  السنة، الشهر، اليوم
    طبيعي  الساعة، الدقيقة، الثانية=0
}
عرف نمط شخص {
    سلسلة  الاسم، اللقب، العنوان
    حقيقي  الطول، الوزن
    منطقي  متزوج
}
```

النص 17 : كيفية انشاء نمط جديد في اللغة الشرمزية

هام جدا: فيما يلي سوف نلتزم بالقاعدتين التاليتين الخاصتين بهذا الكتاب:

1 - تُعَرفُ **الأنماط المهيكلة** خارج جسد الخوارزميات.

2 – في سياق تعريف **نمط مهيكل**، لا يمكن تعريف نمط مهيكل آخر داخله.

2 – 2 : التصريح بالمتغيرات االمهيكلة

للتصريح بالمتغيرات المهيكلة نستعمل الانماط المهيكلة ، وطريقة التصريح بالمتغيرات المهيكلة هي الطريقة ذاتها التي تستعمل في التصريح بالمتغيرات البدائية **(طبيعي، حقيقي، منطقي، حرف)**، اي نكتب اولا **اسم النمط المهيكل** متبوعا بأسماء المتغيرات المهيكلة ، كما يظهر في النص 18.

و تتكون المتغيرة المهيكلة من متغيرات جزئية يمكن ان تكون بدائية او في حد ذاتها متغيرات مهيكلة اخرى، وكل متغيرة جزئية، او **سمة**، تابعة للهيكل، لها اسم يدل عليها في اطار الهيكل، وعملية انشاء متغيرة مهيكلة (او بنائها) تتم بإنشاء **السمات**.

```
عرف نمط تاريخ {
    طبيعي  السنة، الشهر، اليوم
}

عرف نمط شخص {
    سلسلة  الاسم، اللقب، العنوان
    طبيعي  السنة، الشهر، اليوم
    حقيقي  الطول، الوزن
    منطقي  متزوج
}

خوارزم مثال_اول {
/* التصريح بمتغيرات مهيكلة عامة، و تنشأ على صورة النمط المهيكل */
    شخص ش1، ش2 ؛
    اجراء اساسي {
    شخص  ش3، ش4 ؛
```

```
تاريخ ت1، ت2 ؛
}
{
```

النص 18 : كيفية انشاء متغيرات مهيكلة في اللغة الشرمزية

الفرق بين **متغيرات الانماط الاساسية، والسمات التابعة لمتغيرة مهيكلة**، يكمن اساسا في كيفية الوصول اليهما من قبل التعليمات، فالأولى لها اسم يكف ذكره للوصول اليها واستغلال اوتغيير مافيها، اما الثانية فلا يمكن الوصول اليها الا عبر **اسم المتغيرة المهيكلة**، التي تمثل اطارا يحتوي على السمات، فيجب الولوج في الإطار للوصول الى السمة، ويتم الولوج في **المتغيرة المهيكلة** و الوصول الى **السمة** من خلال عملية خاصة.

تنبيه هام: عند التصريح **بسمة** من سمات النمط المهيكل في سياق تعريف هذا الأخير، يجب ان يكون نمط السمة معلوما، فالأنماط البدائية او النمط **سلسلة**، معلومة كونها من مفردات اللغة الشبه الرمزية اولغة البرمجة، اما اذا كان نمط السمة غير بدائي، وجب تعريفه قبل استعماله.

2 –3 استغلال سمات المتغيرات االمهيكلة

للوصول الى مكون من مكونات متغيرة مهيكلة، اي السمات، نستعمل عملية **الولوج في اطار المعلومة المهيكلة**، وتمثل **النقطة** رمز هذه العملية كما يظهر ذلك في النص 19، وكأن هدف هذه العملية هو اختراق الجدار الخارجي الممثل لأطار المتغيرة المهيكلة بغية الوصول الى محتواها، او كأن المتغيرة المهيكلة تملك مفتاحا يستعمل للولوج فيها واستغلال او تغيير السمات التي تتكون منها، وسوف نرى فيما بعد وجود آلية خاصة لمنع اي اختراق والتحكم في الولوج في المتغيرة المهيكلة.

```
عرف نمط  تاريخ_دقيق {
    طبيعي  السنة، الشهر، اليوم
    طبيعي  الساعة، الدقيقة
}
خوارزم  مثال_ثان {
/* التصريح بمتغيرات مهيكلة عامة، و تنشأ على صورة النمط المهيكل */
تاريخ_دقيق  تاريخ_الولادة، الموعد
/*بداية الإجراء الأساسي */
اجراء اساسي {
/* جعل المتغيرة تاريخ_الولادة تمثل التاريخ 12 اوت 2011 */
تاريخ_الولادة.السنة = 2011
تاريخ_الولادة.الشهر = 8
تاريخ_الولادة.اليوم = 12
/* جعل المتغيرة الموعد تمثل التاريخ 2 اكتوبر 2015 ، الساعة 1 الا الربع */
الموعد.السنة = 2015
الموعد.الشهر = 10
الموعد.اليوم = 2
الموعد.الساعة = 12
الموعد.الدقيقة = 45
}
}// نهاية الخوارزم مثال_ثاني
```

النص 19 : كيفية استغلال السمات في اللغة الشرمزية

2 –4 نوعية محتوى المتغيرات المهيكلة

كما لمحنا اليه فيما سبق، يمكن لنمط مهيكل ان يحتوي على سمات تكون انماطها بدائية او مهيكلة، فعلى سبيل المثال، يحتوي النص 20 على التعريف بنمطين، الأول خاص بالتواريخ، والثاني خاص بالأشخاص، ونرى في النص 20 كيف تستعمل النقطة مرتين للوصول الى الأجزاء التي تحتضن معلومات **يوم** و**شهر** و**سنة** الولادة، فمكنت النقطة الأولى (من اليمين الى اليسار) من اختراق اطار المتغيرة **ش1** والولوج فيا ، ومكنت النقطة الثانية من اختراق اطار السمة **الولادة** والولوج فيها.

```
عرف نمط  تاريخ_دقيق {
    طبيعي   السنة، الشهر، اليوم
    طبيعي   الساعة، الدقيقة
}
عرف نمط  شخص {
    سلسلة   الاسم، اللقب، العنوان
    تاريخ_دقيق  الولادة
    حقيقي  الطول، الوزن
    منطقي  متزوج
}
خوارزم  مثال_ثالث {
    اجراء اساسي {
// التصريح بمتغيرة مهيكلة محلية ، و تنشأ على صورة النمط المركب */
    شخص  ش1 ؛
ش1.الاسم = "عمر"
ش1.اللقب = "عمر"
ش1.الولادة.السنة = 2001
ش1.الولادة.الشهر = 8
ش1.الولادة.اليوم = 12
    }// نهاية الإجراء اساسي
} // نهاية الخوارزم مثال_ثالث
```

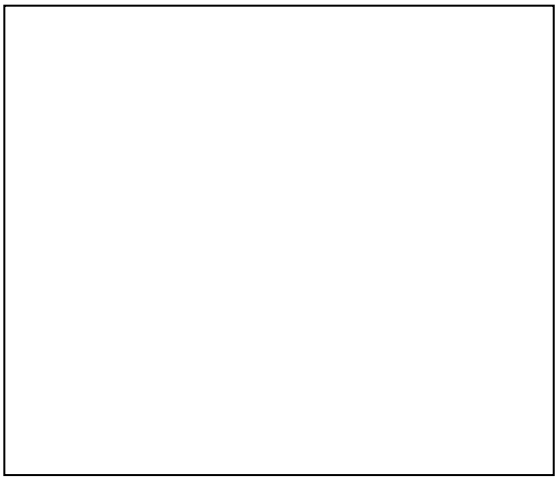
النص 20 : كيفية استغلال السمات لسمات مهيكلة في اللغة الشرمزية

3 – تمثيل نموذجي للمتغيرات المهيكلة في الذاكرة

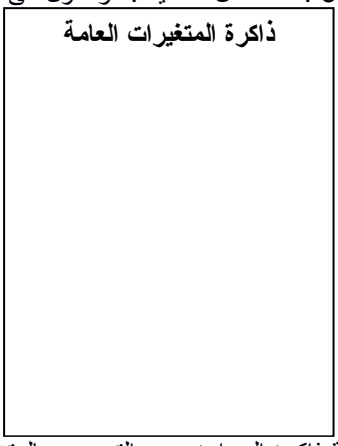
لإنشاء متغيرة مهيكلة في الذاكرة، يستمل منفذ الخوارزم نمط المتغيرة، فيستخرج منه المساحة التي تكفي لإيواء كل السمات، وفي هذه المساحة تتتابع السمات على الشكل الذي صُرح بها عند التعريف بالنمط المهيكل، فعلى سبيل المثال، يُظهر خطأ: لم يُعثر على مصدر المرجع المساحة المخصصة للمتغيرتين المهيكلتين العامتين **ت1** و **ت2** المنتميتين الى النمط **تاريخ** (خطأ: لم يُعثر على مصدر المرجع)، والمساحة الإجمالية المخصصة للمتغيرة المهيكلة هي مجموع المساحات التي وفرت للمتغيرات الممثلة للسمات، فإذا كان النمط **طبيعي** يستحوذ مثلا على **اربع وحدات** من الذاكرة، وبما ان المتغيرة **ت1** محتوية على ست سمات من النمط طبيعي، فان المساحة الإجمالية التي تخصص للمتغيرة **ت1** هي 24 وحدة من وحدات الذاكرة، ونفس هذه المساحة تُخصص للمتغيرة **ت2.**

هام: نلاحظ في خطأ: لم يُعثر على مصدر المرجع ان التصريح بمتغيرة مهيكلة يُنتج متغيرة مهيكلة تحتوي على سمات غير معلومة المحتوى الا في حالة ما اذا عُرفت السِمة بقيمة اولية، فنجد المتغيرة الممثلة للسمة قد شحنت بالقيمة الأولية التي صرح بها في تعريف سمة المتغيرة المهيكلة.

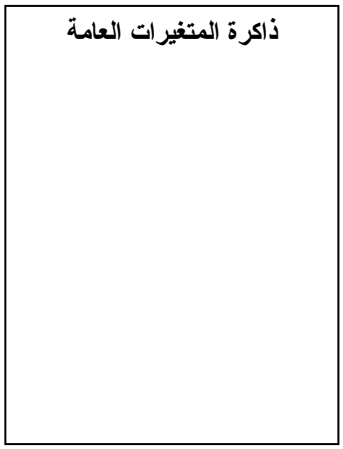
ويبين خطأ: لم يُعثر على مصدر المرجع حالة سمات المتغيرتين ت1 و ت2 بعد تنفيد الإجراء اساسي (خطأ: لم يُعثر على مصدر المرجع)، وبالتحديد التعليمات من 8 الى 13.



النص 5 : استعمال العملية . للوصول الى السمات



الشكل 2 : حالة ذاكرة الخوارزم بعد التصريح بالمتغيرتين ت1 و ت2 (السطر6 النص 5)



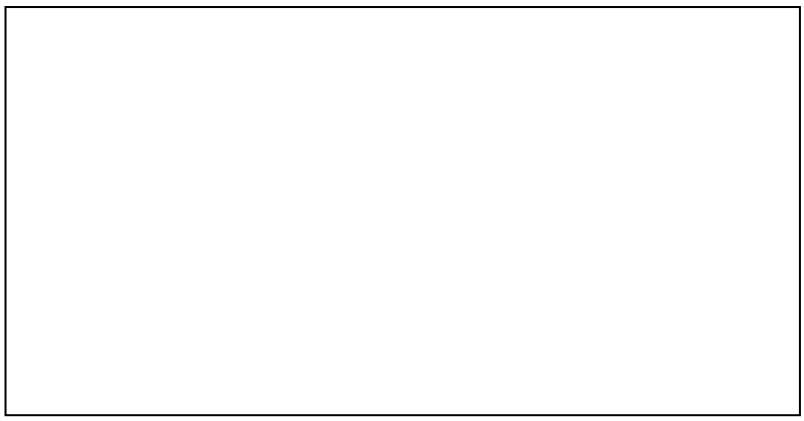
الشكل 3 : حالة ذاكرة بعد تنفيذ تعليمات الإجراء اساسي (النص 5 : من السطر8 الى السطر 13)

4 – اولوية عملية الولوج في المتغيرة المهيكلة عند تقييم العبارات

عندما نريد استعمال سمة من سمات متغيرة مهيكلة في عبارة ما، يجب علينا اولا استعمال عملية الولوج **(النقطة)** للوصول الى السمة، وهكذا تظهر. عملية الولوج مع باقي العمليات التي تحتويها.

العبارة كما يظهر ذلك في السطر 10 من خطأ: لم يُعثر على مصدر المرجع، والسؤال المطروح في سياق تقييم العبارة: بما ان تقييم العبارات يرتكز على اولوية العمليات (انظر الجزء 1، الفصل 12) ، فما هي أولوية عملية الولوج؟

عملية الولوج هي اولى في التقييم من اي عملية اخرى الا عملية التقويس، فهاتين العمليتين لهما نفس الأولوية، فاذا حضرتا في عبارة واحدة، تكون الأولية للعملية الموجودة على اليمين، فعلى سبيل المثال تظهر في خطأ: لم يُعثر على مصدر المرجع عبارتين استعملت فيهما عملية الولوج (السطر 10 و السطر 11)، ففي السطر 10، و في اول عملية تقييم (جدول 1) نجد من اليمين الى اليسار عملية الشحن (=)، ثم عملية التقويس، ثم عملية الضرب، ومن هذه العمليات الثلاث تنتخب عملية التقويس لكونها اولى من عمليتي الضرب و الشحن، وعند تقييم ما في القوسين (جدول 2) تتنافس 3 عمليات: عملية الولوج في اليمين، وعملية الجمع وعملية الولوج في اليسار، وهنا تنتخب عمليتي الولوج، كون عملية الولوج اولى من عملية الجمع، فتنتخب اولا عملية الولوج الموجودة على اليمين، تتبعها عملية الولوج في اليسار ثم عملية الجمع.



النص 6 : استعمال عملية ولوج المتغيرات المهيكلة في العبارات

المرحلة	العبارة قيد التقييم	العملية المنتخبة	العبارة		القيمة	الوعاء
1	قطر = (مس1.طول + مس 1.عرض)*2	()	(مس1.طول + مس 1.عرض)	**جدول 2**	150	مظ3
2	قطر = مظ3*2	*	مظ3*2		300	مظ4
3	قطر = مظ4	=	قطر = مظ4		300	قطر
النتيجة	300					قطر

جدول 1 : تقييم عبارة السطر 10 من النص 6

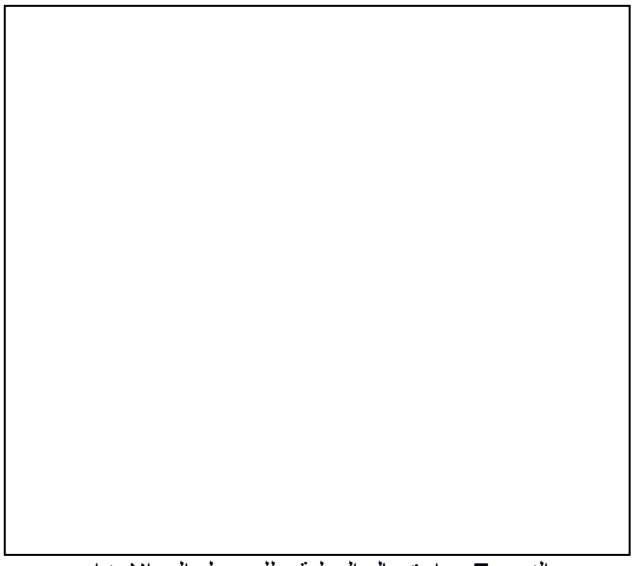
المرحلة	العبارة قيد التقييم	العملية المنتخبة	العبارة	القيمة	الوعاء

1	مس1.طول + مس1.عرض	.	مس1.طول	100	مظ1
2	مظ1 + مس1.عرض	.	مس1.عرض	50	مظ2
3	مظ1 + مظ2	+	مظ1 + مظ2	150	مظ3
النتيجة	150				مظ3

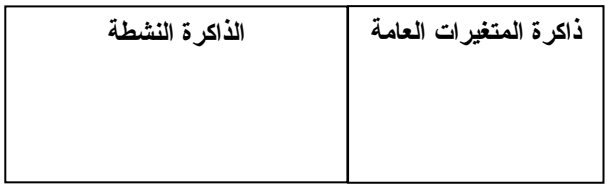
جدول 2 : تقييم عبارة ما بين قوسين في السطر 10 من النص 6

4 – المتغيرات المهيكلة الطارئة

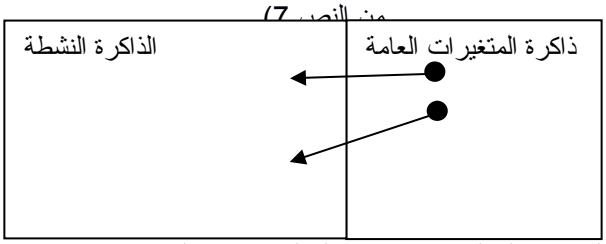
كالمتغيرات الطارئة البدائية (اي التي تنشأ انطلاقا من الأنماط الأساسية)، تُنشأ المتغيرات الطارئة المهيكلة في **الذاكرة النشطة**، وتُستغل عن طريق **الصلات**، وتُستعمل العملية **شيد** لإنشائها، وتقوم عملية **شيد** بحجز المساحة الكافية للمتغيرة الطارئة المهيكلة في الذاكرة النشطة، وتُرجع لمن طلب تنفيذها معلومات عن تواجد المتغيرة الطارئة المهيكلة في الذاكرة النشطة، وهذه المعلومات يجب الاحتفاظ بها في صلة حتى يُمْكِن الوصول الى المتغيرة الطارئة المهيكلة وما تحتويه من سمات(خطأ: لم يُعثر على مصدر المرجع، خطأ: لم يُعثر على مصدر المرجع الى خطأ: لم يُعثر على مصدر المرجع).



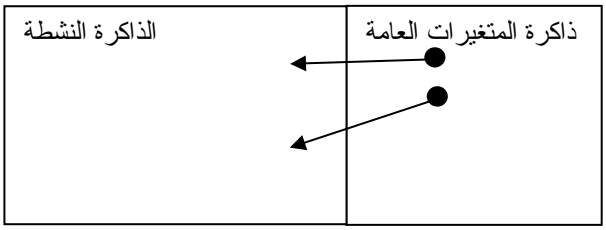
النص 7 : استعمال العملية . للوصول الى الاجزاء



الشكل 4 : حالة الذاكرة بعد التصريح بالصلتين ت1 و ت2 من النط تاريخ (السطر 7 من النص 7)



الشكل 5 : حالة الذاكرة بعد تنفيذ العملية شيد في السطرين 10 و 11 من النص 7



الشكل 6 : حالة الذاكرة بعد تنفيذ التعليمات من السطر 12 الى 17 من النص 7

العلاقة بين اسم المتغيرة المهيكلة والصلة في العبارات:

في اغلب الحالات تستعمل الصلات مع المتغيرات المهيكلة الطارئة، ونادرا ما تستعمل مع المتغيرات العادية او المتغيرات المهيكلة العادية التي لا تنشأ في الذاكرة النشطة، وفي العبارات يمكن ان تظهر اسماء المتغيرات المهيكلة او صلتها، وفي كل العبارات تلعب الصلة الدور نفسه الذي يلعبه اسم المتغيرة المهيكلة، الا في الحالة التي يكون في يمين عملية النسخ (=) صلة وفي يسارها صلة، ففي هذه الحالة لا تلعب الصلة دور المتغيرة المرتبطة بها، بل تلغب دور متغيرة بدائية، فتقوم عملية النسخ بنقل القيمة الموجودة في المتغيرة الممثلة للصلة على يسارها الى المتغيرة الممثلة للصلة على يمينها، وفيما يلي نستعرض بعض الأمثلة الواردة في النص 21.

الرقم	النص
	خوارزم المثال_السابع {
1	تاريخ ت1، ت2 /* التصريح بمتغيرات عامة من النمط تاريخ المعرف في خطأ: لم يُعثر على مصدر المرجع)
2	اجراء اساسي(){
3	تاريخ &صت1 --< ت1، &صت2 --< ت2 ، صت3 ، صت4 = صلة_حرة /* محتوى الصلة صت3 مجهول */
4	صت1 = ت2 /* و كأننا كتبنا ت1 = ت2، و هذه كتابة صحيحة شكلا و مضمونا */
5	صت4 = ت1 /* كتابة خطيرة، صحيحة شكلا و غير صحيحة في المعنى، اذ لا توجد اي متغيرة مرتبطة بالصلة الحرة صت4، */
6	صت3 = ت1 /* كتابة خطيرة، صحيحة شكلا و غير صحيحة في المعنى، فلا علم لنا بأي متغيرة الصلة صت3 مرتبطة*/
7	صت1 = صت4
8	ت1 = صت2
9	صت4 = شيد(تاريخ)
10	صت4 = ت1
11	} /* نهاية الإجراء اساسي */
	} // نهاية الخوارزم المثال_السادس

النص 21 : امثلة لشرح معنى عملية = عندما نستعمل معها الصلات

في (السطر 4) نجد الكتابة **صت1 = ت2**، وهذه الكتابة شبيهة بالكتابة **ت1 = ت2**، وسوف نرى في الفصل التالي سلوك العمليات الأساسية عندما تتناول المتغيرات المهيكلة، ويجب هنا ان تكون الصلة **صت1** مرتبطة بمتغيرة معلومة، فالكتابات الظاهرة في السطرين 5 و 6، مع كونها صحيحة في الشكل الا انها خطيرة على الخوارزم ونتائجه، وهذه الكتابة لا تغير في محتوى المتغيرة الممثلة للصلة **صت1**، فتبق دائما مرتبطة بالمتغيرة **ت1** (السطر 3).

في السطر 3 نرى ان محتوى الصلتين **صت3** و **صت4** غير مرتبط بأي متغيرة معلومة، ولهذا فان الكتابتين الظاهرتين في السطرين 5 و 6 خطيرتين على سلوك الخوارزم مع صحة شكلهما، فالصلة على اليمين تفهم في هاتين الحالتين على انها المتغيرة المرتبطة بالصلة، وهنا ايضا لا يمس محتوى الصلتين، فتبق **صت4** حرة و **صت3** حرة او مرتبطة بمجهول لكوننا لا نعرف محتواها.

في السطر السابع، نرى في يمين عملية الشحن (=) **صلة** وفي يسارها **صلة (صت1=صت 4)**، ففي هذه الحالة الفريدة لا تُفهم الصلة على انها المتغيرة المرتبطة بها بل تفهم على انها **صلة،** وعملية الشحن (=) تقوم هنا بتغيير الصلة في يمينها، وشحنها بما في المتغيرة الممثلة للصلة على اليسار، وبهذه العملية يُفك الارتباط بين الصلة **صت1** والمتغيرة **ت1** وتصبح الصلة **صت1** على حال الصلة **صت4** أي صلة حرة.

في السطر 8، و بما ان احد طرفي عملية الشحن (=) هو اسم لمتغيرة مهيكلة (**ت1** في يمين عملية الشحن=)، فإن الصلة **صت2** ينظر اليها على انها المتغيرة المرتبطة بها، اي **ت2**، ولذا تكون هذه الكتابة متساوية تماما مع الكتابة **ت1 = ت2**.

في السطر 9، نجد صلة في يمين عملية الشحن = وكذلك صلة في يسارها لأن العملية **شيد** تنتج عند انتهاء تنفيذها صلة ظرفية تحتوي على معلومات تواجد المتغيرة الطارئة في الذاكرة النشطة، وتنفيذها اولى من عملية الشحن(=)، وهكذا عند انتهاء تعليمة السطر 9، تكون الصلة مرتبطة بمتغيرة مهيكلة طارئة انتجتها الوظيفة **شيد**، وفي السطر 10، ونتيجة لما حدث في السطر 9، تفسر الصلة على انها المتغيرة الطارئة المرتبطة بها.

5 – نوعية العلاقة بين المعلومة المهيكلة وعناصرها

راينا فيما سبق ان المعلومة المهيكلة يمكن ان تكون مكونة من سمات تنتمي الى انماط بدائية وكذلك من سمات تنتمي الى انماط مهيكلة اخرى، وكان هذا شأن المعلومة الممثلة لشخص ما والتي بنيت على اساس النمط **شخص** المحتوي على سمة من النمط **تاريخ**، وهذا امر عادي، وكمثال آخر (خطأ: لم يُعثر على مصدر المرجع) يمكن ان نعرف نمطا يمثل عجلات السيارات ثم نعرف نمطا يمثل السيارات ومن محتويات هذا الأخير سمات عدة (**ع1، ع2، ع3** و **ع4**) من النمط الممثل للعجلات، ولا يكتف النمط سيارة ان يكون له مكونات من النوع عجلة، بل نلاحظ وجود مكونا من النمط **شخص** يمثل الشخص المالك للسيارة.



النص 9 : تعريف النمط سيارة والانماط الضرورية له

هنا يتبادر الى الذهن السؤال التالي: في الواقع المعاش، **هل العلاقة بين مالك السيارة والسيارة علاقة تكوينية ام انها غير ذلك؟** والجواب البديهي هو ان الشخص المالك للسيارة ليس جزءا مكونا للسيارة، وانما هو جزء من العالم الذي توجد فيه السيارة، فالعلاقة ليست تكوينية بل تكميلية، فيمكن للسيارة ان توجد دون مالك لها، ومثل هذه الملاحظة تؤدي بنا الى ضرورة التفريق بين نوعية من السمات التي تستعمل في تعريف الانماط المهيكلة، وهكذا يمكن لنا ان نقول ان السمة المذكورة في تعريف نمط مهيكل تكون على احد الأمرين التاليين:

- سمة حيوية بالنسبة للمعلومة المهيكلة، يجب ان تكون حاضرة لتكون المعلومة قائمة بذاتها.
- سمة مكملة، يمكن ان تغيب من المعلومة، واذا غابت لا تؤثر على وجود المعلومة.

فمثلا هل كل العجلات ضرورية لتكون السيارة، حقا سيارة يمكن استعمالها ام لا؟ و الجواب هنا نعم. فلا يمكن استغلال سيارة بدون عجلاتها، وان كانت كذلك فلا نسميها سيارة، بل هي في غالب الأحيان **خردة** او **سيارة معطلة**، ونفس الملاحظة بالنسبة للمحرك وخزان الوقود، فلا يمكن ان نتصور سيارة بدون محرك أو بدون خزان وقود، فهذه الاجزاء حيوية بالنسبة للسيارة ووجودها ضروري في المتغيرة المهيكلة الممثلة للسيارة.

وعلى النقيض مما سبق، هل الشخص المالك للسيارة ضروري لوجود السيارة، والجواب هنا لا، اذ يمكن ان نتصور حالة تكون فيها السيارة غير مملوكة لشخص ما، وهنا يظهر جليا ان المعلومة المتعلقة بمالك السيارة معلومة مكملة، يمكن ان نحتاج اليها كما يمكن تجاهلها عند استغلال السيارة، فهي اذا ليست حيوية، وهي ليست من مكونات السيارة.

6 تمثيل الأجزاء الحيوية و المكملة في ذاكرة الخوارزم:

اذا التزمنا بالتعريف الذي اعطيناه فيما سبق عن المساحة التي تستحوذ عليها المتغيرة المهيكلة من الذاكرة، فان متغيرة من النمط سيارة (خطأ: لم يُعثر على مصدر المرجع) سوف تستهلك ما يلي:

- مساحة لإيواء معلومات المالك، وتستنتج من النمط **شخص**، وهي المساحة الكافية لإيواء الاسم، واللقب، والعنوان، و تاريخ الولادة، و الطول، والوزن، والحالة المدنية.

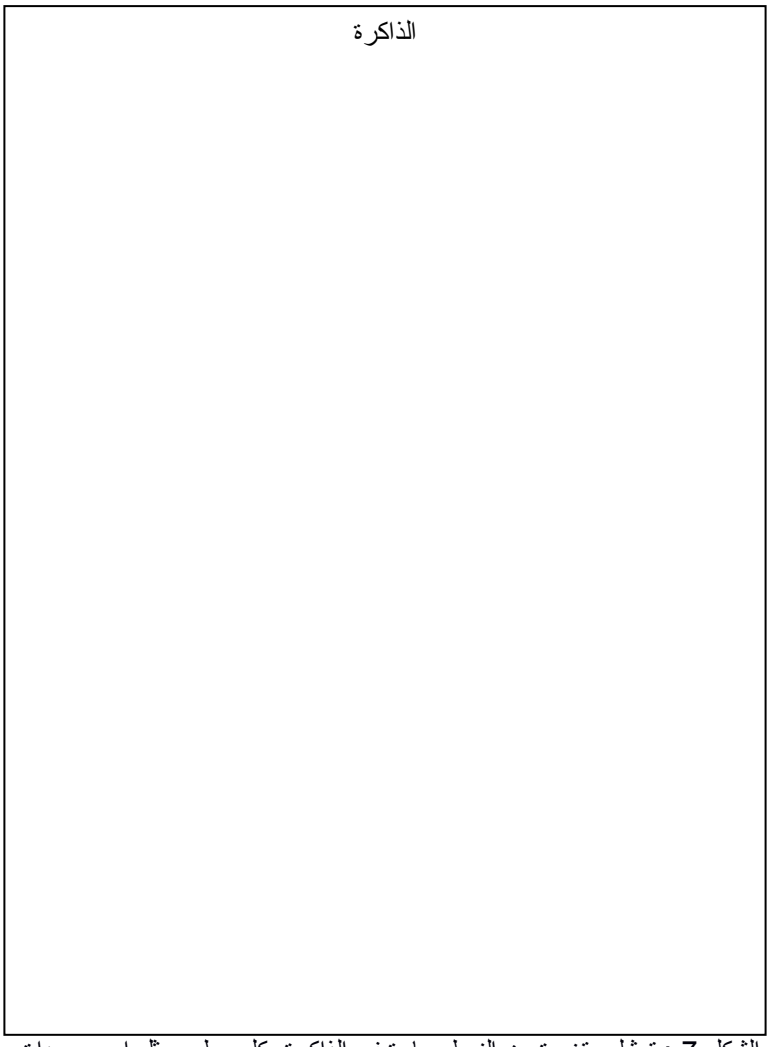
- مساحة لإيواء رقم_التسجيل، و رقم_التسجيل_السابق، و الطراز، والرقم_التسلسلي، والهيكل.
- مساحة لإيواء 4 عجلات، وكل عجلة من النمط **عجلة**، ومعلومات كل عجلة هي: المنتج، العرض، السماكة، القطر.

و اذا اعتبرنا ان وحدة قياس سعة الذاكرة هي "**البايت**" (هذه الوحدة تكف لإيواء معلومة من نمط **حرف**، اي حرف واحد)، وان الأعداد الطبيعية، والصحيحة والحقيقية والمنطقية تستهلك 4 وحدات، وان متغيرة من النمط **سلسلة** تستهلك من الذاكرة ما تحتويه السلسلة من حروف، فإننا نرى انه في احسن حال، اذا كان معدل طول سلاسل الحروف هو 4 حروف وحدات، سوف تستهلك متغيرة من النمط **سيارة** 128 وحدة من الذاكرة كما يظهر ذلك في خطأ: لم يُعثر على مصدر المرجع والجدول 3، ونلاحظ ان الجزء الغير الحيوي الممثل للمالك يستحوذ على ربع المساحة.

رقم	اسم الجزء	عدد الوحدات المستهلكة من كل جزء بدائي وكل سلسلة حروف	ملاحظة
1	المالك.الاسم	4	الجزء المخصص للمالك استهلك 44 وحدة تخزين، و فيه استهلك الجزء المخصص للولادة 12 وحدة
2	المالك.اللقب	4	
3	المالك.العنوان	4	
4	المالك.الولادة.السنة	4	
5	المالك.الولادة.الشهر	4	
6	المالك.الولادة.اليوم	4	
7	المالك.الولادة.الساعة	4	
8	المالك.الولادة.الدقيقة	4	
9	المالك.الطول	4	
10	المالك.الوزن	4	
11	المالك.متزوج	4	
12	رقم_التسجيل	4	
13	ارقم_التسجيل_السابق	4	
14	الطراز	4	
15	الرقم_التسلسلي	4	
16	الهيكل	4	
17	ع1.المنتج	4	الجزء المخصص لعجلة واحدة استهلك 16 وحدة تخزين
18	ع1.العرض	4	
19	ع1.لسماكة	4	
20	ع1.القطر	4	
21	ع2.المنتج	4	
22	ع2.العرض	4	
23	ع2.لسماكة	4	
24	ع2.القطر	4	
25	ع3.المنتج	4	
26	ع3.العرض	4	
27	ع3.لسماكة	4	
28	ع3.القطر	4	
29	ع4.المنتج	4	
30	ع4.العرض	4	
31	ع4.لسماكة	4	
32	ع4.القطر	4	

جدول 3: المساحة المستهلكة من كل جزء، مع فرضية ان الانماط البدائية و النمط سلسلة يستهلكون 4 وحدات تخزين

واذا اردنا ان ننجز خوارزما ما يحاكي سباق السيارات (خطأ: لم يُعثر على مصدر المرجع)، يكون عدد المتغيرات الممثلة للسباق يساوي على الأقل عدد السيارات المقحمة في السباق، وفي هذا الخوارزم لا نستعمل البتة المعلومات عن المالك، بل معلومات اخرى عن السيارة، وهكذا اذا افترضنا ان عدد السيارات في واقع ما هو 100 سيارة (خطأ: لم يُعثر على مصدر المرجع)، فان الحجم المستهلك لتمثيل 100 متغيرة من النمط سيارة هو 12000 وحدة تخزين، والحجم المستهلك من قبل الجزء الخاص بالمالك هو 3600 وحدة تخزين، والإشكال هنا ليس في الحجم المستهلك، بل في حجز مساحة من الذاكرة، يمكن ان تكون ضخمة، دون استغلالها.



الشكل 7 : تمثيل متغيرة من النمط سيارة في الذاكرة، كل سطر يمثل اربع وحدات تخزين



النص 10 : عرض جزئي لخوارزم محاكاة سباق 100 سيارة

السؤال الذي يتبادر الى الذهن هو كيف يمكن معالجة مثل هذا الإشكال مع ابقاء الهيكل كما هو، وبصفة ادق، كيف يمكننا ان نتحصل على هيكل يمكن في بعض الأحيان ان نجد فيه عنصرا ما عندما يصبح وجوده ضروري وفي احيان اخرى يكون فيه العنصر مغيبا عندما لا يكون وجوده ضروري، بل اكثر من ذلك كيف يكون لهيكل ما **عنصرا طارئا** يمكن ان يظهر او يغيب حسب الاحتياج.

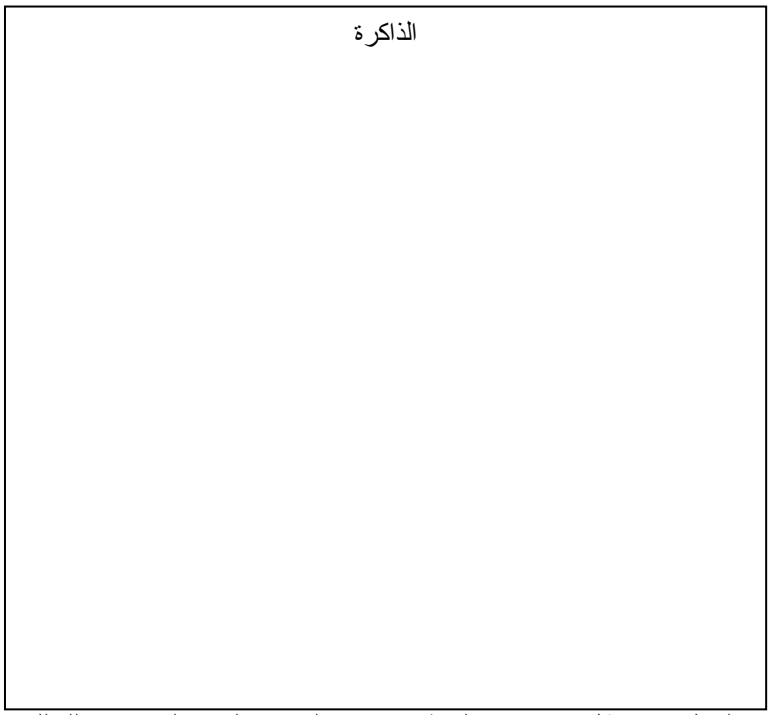
7 – 1 هيكلة المتغيرات باستعمال العناصر الطارئة

كما علمنا، فان سمات المتغيرات المهيكلة هي نفسها متغيرات، وبهذا نرى ان حل الإشكال السابق يكمن في استعمال المتغيرات الطارئة لتمثيل السمات التي يمكن ان تحضر او تغيب حسب الحاجة، ومعنى هذا اننا في تعريف النمط المهيكل، بدل استعمال تصريح بمتغيرة في تعريف سمة ما، نستعمل التصريح بصلة ، فمثلا في خطأ: لم يُعثر على مصدر المرجع، السطر 6، بدل التصريح بمتغيرة من النمط **شخص،** صُرِح بصلة من النمط **شخص** ، وهكذا، عندما تُنشأ متغيرة من النمط **شخص،** تُنشأ في موضع السمة **مالك،** صلة بدل من متغيرة ، وفي العموم لا تحتاج الصلة الا لعدد قليل من وحدات الذاكرة لتمثيلها، وفي الغالب مساحة الصلة تساوي مساحة متغيرة من النمط **طبيعي،** اي 4 وحدات فيما يخصنا، وهذا ما يُظهره خطأ: لم يُعثر على مصدر المرجع الذي نجد فيه صلة بدل من اسم متغيرة من النمط شخص.

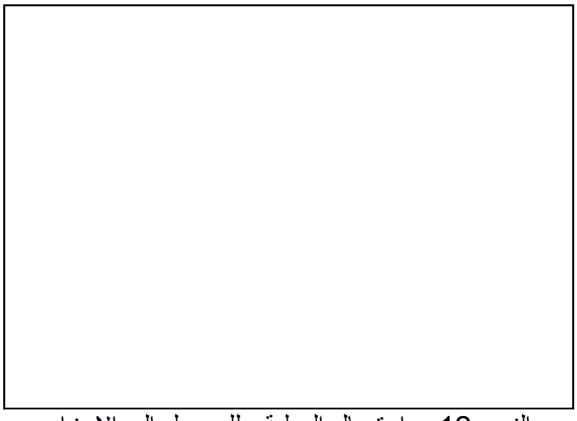
تنبيه: الصلة في حد ذاتها متغيرة بدائية النمط، تحتوي على معلومات تدل على متغيرة اخرى، بدائية كانت او مهيكلة، ونفترض انها تستهلك من الذاكرة اربعة وحدات تخزين.

النص 11 : تعريف النمطين عجلة و سيارة

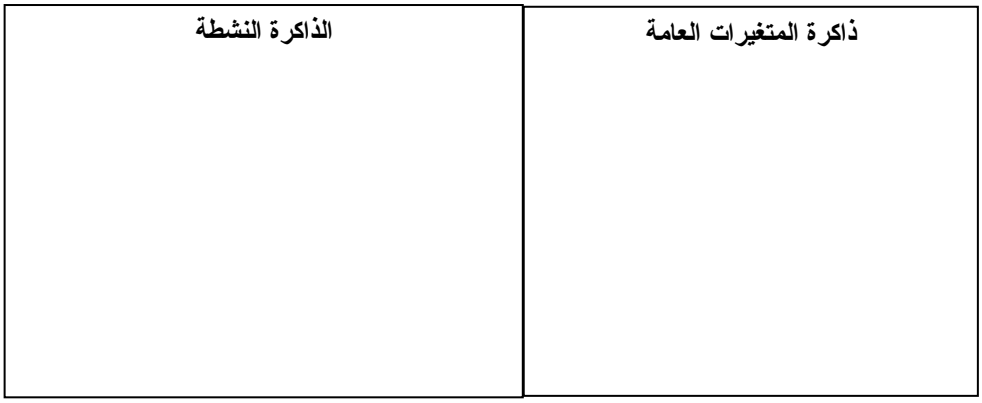
مع هذه الصورة الجديدة للمتغيرة المهيكلة من النمط **سيارة،** عندما نريد من العنصر الممثل للمالك ان يكون حاضرا، نستعمل التعليمة **شيد** ونربط المتغيرة الطارئة التي انتجتها التعليمة **شيد** بالصلة الممثلة للشخص في المتغيرة الممثلة للسيارة ، كما يبين ذلك خطأ: لم يُعثر على مصدر المرجع الذي فيه تنشأ المتغيرة **س1** الممثلة لسيارة ما (السطر 1)، ثم بعد ذلك تنشأ متغيرة من النمط **شخص** وتربط بالصلة المسماة **المالك** (السطر. 4) الموجودة في المتغيرة **س1**، وتبين التعليمات المتبقية كيف توضع المعلومات المتعلقة بالاسم، و اللقب ومعلومات الولادة في اماكنها، كما يبين خطأ: لم يُعثر على مصدر. المرجع و خطأ: لم يُعثر على مصدر المرجع حالة الذاكرة بعد انجاز بعض التصريحات وتنفيذ بعض التعليمات من خوارزم خطأ: لم يُعثر على مصدر المرجع، وهكذا بدل حجز ربع اجمالي المساحة المخصصة لتمثيل متغيرة من النمط سيارة، لن يحتجز الجزء المخصص للمالك الا جزءا ضئيلا (1 من 22 جزء) اذا لم تكن معلومات المالك ضرورية في الخوارزم.



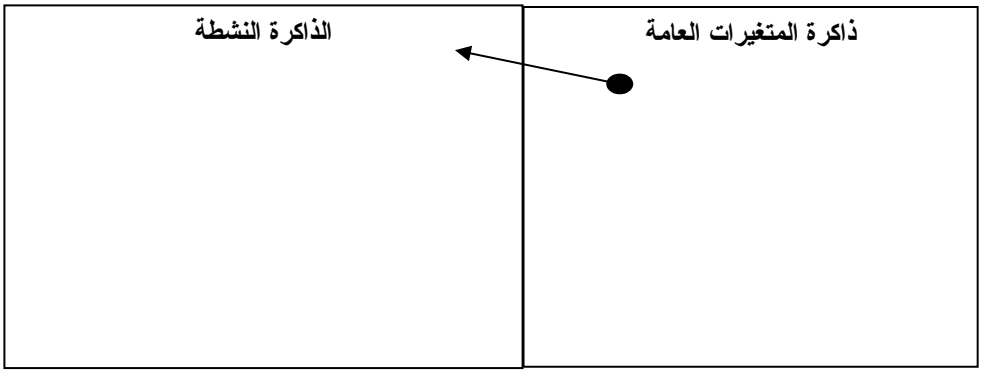
الشكل 8 : تمثيل متغيرة من النمط سيارة في الذاكرة، الجزء المخصص للمالك ممثل بصلة



النص 12 : استعمال العملية . للوصول الى الاجزاء



الشكل 9 : حالة الذاكرة بعد التصريح بالمتغيرة س1 من النط سيارة (السطر 1 من النص 15)

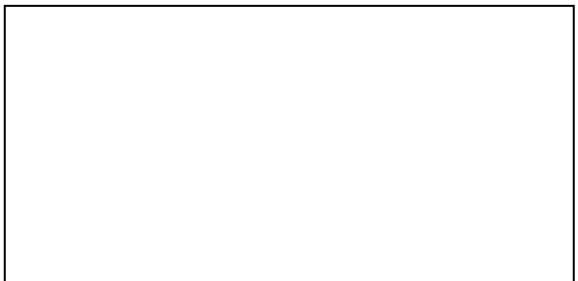


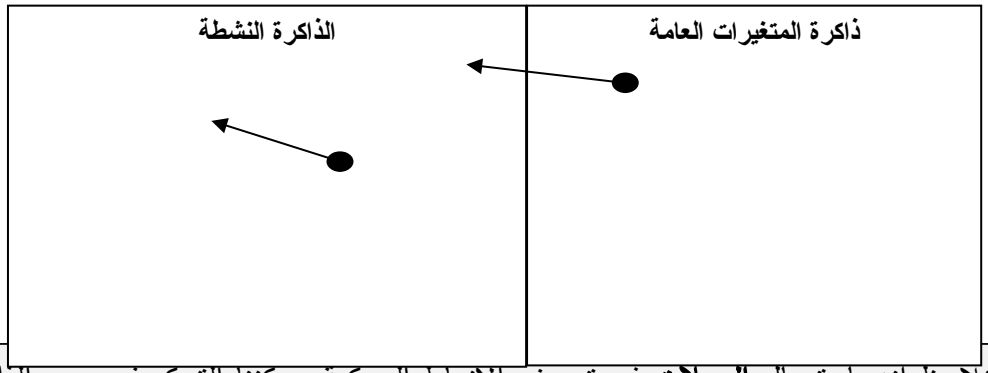
الشكل 10 : حالة الذاكرة بعد تنفيذ عملية شيد لإنشاء متغيرة طارئة من النمط شخص (السطر 4 من النص 15)

ويمكن ان نستعمل العناصر الطارئة في تعريف النمط **شخص**، فمثلا، اذا جعلنا الولادة عنصرا طارئا، يصبح التعريف بالنمط شخص على صيغة النص 22، الذي فيه عرف عنصر التاريخ على انه صلة لمتغيرة طارئة من النمط تاريخ، وفي هذه الحالة، اذا اردنا ان نعرف تاريخ ميلاد شخص ما، يجب علينا اولا ان نوفر العنصر الخاص بمعلومات الولادة المتمثل في متغيرة طارئة من النمط تاريخ كما هو ظاهر في السطر 8 من خطأ: لم يُعثر على مصدر المرجع ويبين خطأ: لم يُعثر على مصدر. المرجع نتيجة تنفيذ تعليمات خطأ: لم يُعثر على مصدر المرجع على الذاكرة .

```
عرف نمط شخص {
سلسلة الاسم، اللقب، العنوان
تاريخ &الولادة
حقيقي الطول، الوزن
منطقي متزوج
}
```

النص 22 : النمط شخص بجزء طارئ





ملاحظة: نلاحظ انه يمكن استعمال **الصلات** في تعريف الانماط المركبة، يمكننا التحكم في حجم الذاكرة التي يجب ان تستغل لتمثيل المتغيرات المهيكلة، فاذا كان جزءا ما ضروريا، انشأناه ووضعنا فيه المعلومات المناسبة، وان كان الجزء غير مهم لا ننشئه، واكثر من ذلك، ان كان جزءا ما ضروري في مرحلة ما من مراحل تنفيذ الخوارزم انشأناه، وعندما يصبح غير ضروري في مراحل قادمة حذفناه باستعمال التعليمة **حرر** التي تحرر المساحة المستعملة من الذاكرة و توفرها للتعليمة **شيد** عند تنفيذها.

7 – 1 تعميم استعمال الصلات في تعريف الانماط المهيكلة

يُمكن تعميم استعمال الصلات لجعل السمات عناصر طارئة، وفي هذه الحالة يجب ان نحتاط جيدا، فنتذكر دائما انه لا يمكن استغلال عنصرا طارئا الا بعد انشائه، فمثلا، في سياق انشاء متغيرة مهيكلة، إذا كانت سيماتها الحيوية قد صرحت باستعمال الصلات، يجب اولا انشاء السمات الحيوية قبل اي استعمال للمتغيرة المهيكلة، اما السمات المكملة فتنشأ عند الحاجة لها، وهذا ما يظهره خطأ: لم يُعثر على مصدر المرجع الذي يستعمل صيغة جديدة للنمط سيارة، التي ترتكز على الصلات لوصف العجلات، وبما ان العجلات سمات حيوية، فانه يتم انشاؤها قبل اي استعمال للمتغيرة **س1** (السطر 9) الممثلة للسيارة.

النمط 15 : استعمال الصلات للتصريح بالسمات من النمط عجلة و شخص

8 – استعمال النمط نفسه كعنصر طارئ: مدخل الى الهياكل المتسلسلة

لا يمكن ان نستعمل النمط نفسه ليكون سمة بالطريقة التي لا نستعمل فيها الصلات، فمثلا لا يمكن ان ننشأ سيارة تكون فيها سيارة اخرى.

اما اذا استعملنا الصلات، فيمكن لسمة ان تكون من نفس النمط الذي هو قيد التعريف، كما يظهر في السطر 7 من خطأ: لم يُعثر على مصدر المرجع، و هذه الهيكلة للمعلومات لا يمكن ان تترجم على ان سيارة ما يمكن ان تكون مكونة من سيارة اخرى، بل ان سيارة ما، مرتبطة بسيارة اخرى، بعلاقة غير العلاقة التكوينية، فمثلا (خطأ: لم يُعثر على مصدر المرجع) يمكن ان تكون السيارة **س1** تمثل السيارة الأولى في سباق ما وتليها السيارة **س2** المربوطة بها.

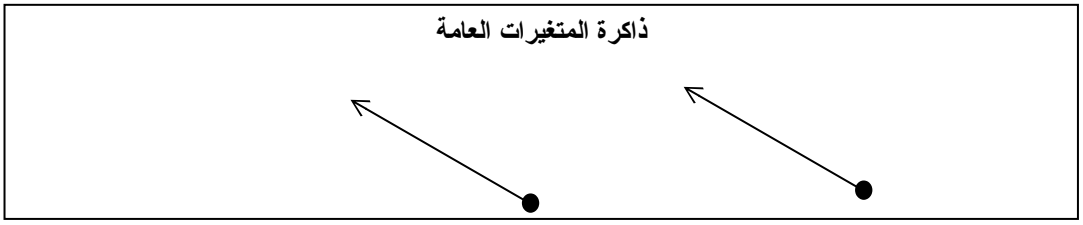
النص 16 : النمط سيارة يحتوي على صلة من النمط سيارة

ذاكرة المتغيرات العامة

الشكل 12 : حالة الذاكرة العامة بعد التصريح بثلاث بمتغيرات عامة من النمط سيارة (السطر 10 من النص 16)

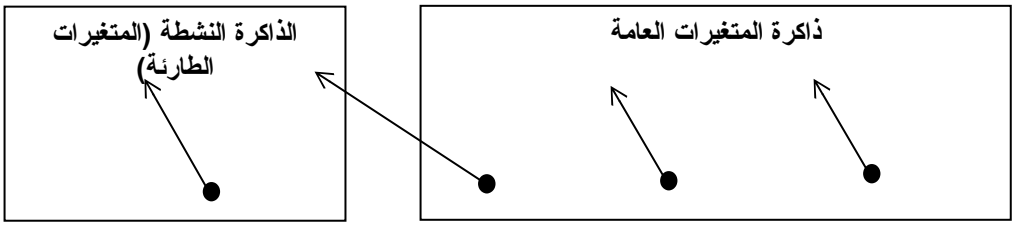
ذاكرة المتغيرات العامة

الشكل 13 : حالة الذاكرة العامة بعد ربط الصلة س من المتغيرة س1 بالمتغيرة س2 (السطر 12 من النص 16)



الشكل 14 : حالة الذاكرة العامة بعد ربط الصلة س من المتغيرة س1 بالمتغيرة س2 (السطر 13 من النص 16)

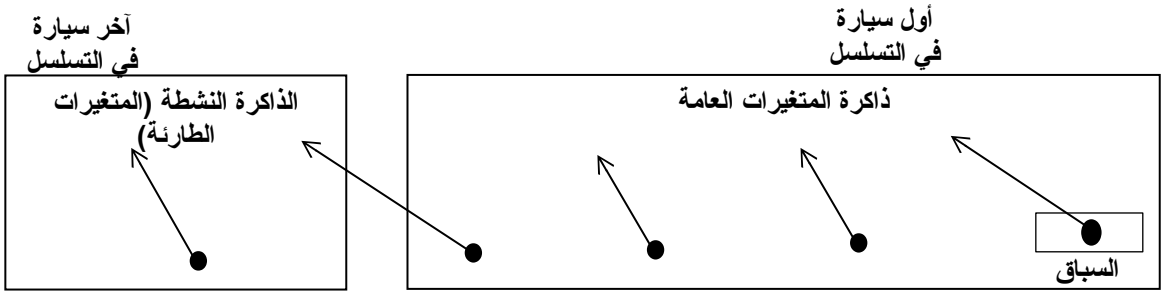
واللافت للنظر ان هذه الهيكلة التي تترابط فيها متغيرات مهيكلة من نفس النمط، قادرة على استيعاب اكثر من سيارة، فيمكن ان نضيف سيارة ثالثة ورابعة (خطأ: لم يُعثر على مصدر المرجع و خطأ: لم يُعثر على مصدر المرجع) ونتحصل بذلك على سلسة من السيارات يمكن ان تمثل في وقت ما حالة السباق، وكذلك يمكن ان ننشأ متغيرات طارئة من النوع سيارة ثم نربطها بالسلسلة كما يظهر في السطرين 14 و 15 من خطأ: لم يُعثر على مصدر المرجع و في خطأ: لم يُعثر على مصدر المرجع، ونلاحظ من خلال الشكل وجود اسماء خاصة بالمتغيرات العامة **(س1، س2 س3)** مرتبطة بها ارتباطا وجوديا وعدم وجود مثل هذه الاسماء في حال المتغيرات الطارئة.



الشكل 15 : حالة الذاكرة العامة بعد تنفيذ تعليمات السطرين 14 و 15 من النص 16

ملاحظة هامة: من خلال خطأ: لم يُعثر على مصدر المرجع نلاحظ تجمع اكثر من سيارة فيما قد نسمية وعاء يمثل سباق السيارا ت، وهذا الوعاء متماسك بفضل الصلات التي تربط مكونات الوعاء فيما بينها، اي السيارات، فهذا الوعاء يمكن مثلا ان نطلق عليه اسم "قائمة السيارا ت" او "طابور السيارات" او "مجموعة من السيارات"، وهذا الوعاء له هيكلة تسلسلية، فيمكن ان نصفه "بقائمة متسلسلة من السارات"، او "بمجموعة متسلسلة من السيارات"، وللتدليل على هذا الوعاء نحتاج الى اسم، وهذا الإسم يمكن ان يكون لصلة مرتبطة بأول مكون في هذه السلسلة من السيارات، مثلا الصلة **سباق** بالسيارة **س1** كما يظهر في خطأ: لم يُعثر على مصدر المرجع، وحتى نعلم حدود هذا الوعاء،

لا بد ان نعرف نهايته، وهذه النهاية يمكن ان تمثلها القيمة "**حرة**" لصلة الربط بين السيارات في آخر سيارة من "القائمة المتسلسلة" من السيارات (خطأ: لم يُعثر على مصدر المرجع).



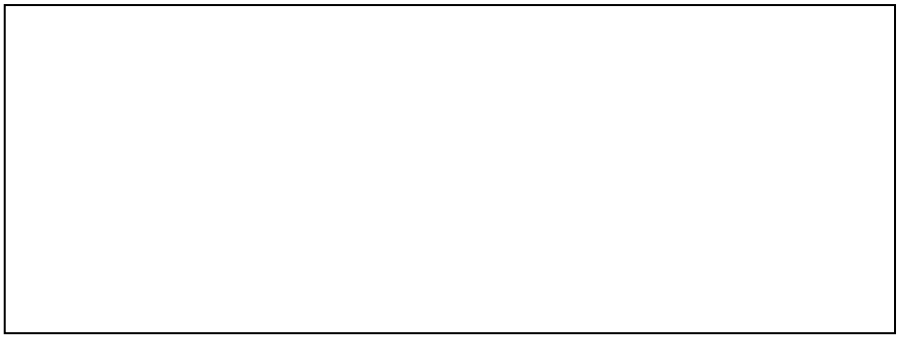
الشكل 16 : وعاء يحتوي على السارات المشاركة في سباق ما

الفصل الثالث

تهيئة المتغيرات المهيكلة

1- مقدمة: أهمية تهيئة المتغيرات قبل استعمالها في العبارات

علمنا مما سبق (انظر الجزء الأول) ان إنشاء المتغيرات بقيمة اولية، او ما نسميه بتهيئة المتغيرات، امر ذو اهمية قصوى، حتى يصبح استعمال المتغيرات استعمالا سليما في مختلف العبارات، فلا يمكن ان تكون العبارة سليمة اذا ارتكزت على متغيرات غير معلومة المحتوى، فلا بد لكل متغيرة ذكرت في عبارة ما ان تكون قد شحنت بقيمة معلومة قبل استعمالها، ولإبراز اهمية تهيئة المتغيرات قبل استعمالها في العبارات، نحلل فيما يلي، محتوى النص 1.



النص 1 : اهمية تهيئة المتغيرات بقيمة قبل استعمالها في العبارات

- في السطر 5 تكون قيمة تقييم العبارة قيمة معلومة، وهي القيمة 10 التي وضعت في المرحلة الأخيرة من عملية التقييم في المتغيرة **الف**، وتصبح المتغيرة **الف** بعد هذا التقييم ذات قيمة معلومة هي 10.
- في السطر 6 تكون نتيجة تقييم العبارة قيمة معلومة، وهي القيمة 130 (**الف** يحتوي على القيمة 10 و **عين** تحتوي على القيمة 5) التي توضع في المرحلة الأخيرة من عملية التقييم في المتغيرة **باء**، وبعد انتهاء التقييم تصبح المتغيرة **باء** محتوية على قيمة معلومة، اي 130.
- في السطر 9، تقييم العبارة يكون كالتالي (انظر جدول اولويات العمليات في الجزء الأول): العملية التي تنخب في اول الأمر هي **الف*باء**، و يفضي تقييم **الف*باء** الى القيمة 1300 (المتغيرة الف تحتوي على القيمة 10 و المتغيرة باء تحتوي على القيمة 130)، وكما سبق وان تعلمناه في الجزء الأول، توضع النتيجة في متغيرة ظرفية نسميها مثلا **مظ 1**، و تصبح العبارة التي يجب تقييمها الآن هي: **مظ1 – تاء* 6**، وهنا تنتخب عملية الضرب *، فتقيم العملية **تاء*6**، وهنا نصطدم بأمر مزعج وهو خلل في التقييم، فالمتغيرة **تاء** تحتوي على قيمة طبيعية غير معلومة، وتنفيذ العملية **تاء*6** يفضي الى قيمة طبيعية غير معلومة، توضع في متغيرة ظرفية نسميها مثلا **مظ2**، وبعد انتهاء تقييم

تاء*6، تصبح العبارة الأصلية على الشكل: **مظ1 – مظ2**، وهذه العملية الأخيرة مبنية على متغيرة محتوية على قيمة مجهولة، وبذلك تكون نتيجة تقييم عبارة السطر 9 مجهولة.

1 – 2 تذكير: متى تهيئ المتغيرات

علمنا مما سبق (انظر الجزء الأول) ان تهيئة متغيرة بقيمة معلومة يكون اما عند التصريح بها بقيمة اولية (السطر 2 من النص 1) او بتنفيذ تعليمة تحتوي على عملية شحن (الممثلة بالرمز =) وتكون المتغيرة التي يراد تهيئتها بقيمة ما في الجانب الأيمن من عملية الشحن (الأسطر 5 و 6 من النص 1)، ففي السطر 2 تأخذ المتغيرة **عين** اول قيمة لها عند التصريح بها، وفي السطر 5 تشحن المتغيرة **الف** بأول قيمة عبر تقييم عبارة هذا السطر ويتم هذا الشحن قبل اي استعمال للمتغيرة **الف**، ونفس الأمر يتم بالنسبة للمتغيرة **باء** التي تأخذ اول قيمة لها من خلال تقييم عبارة السطر 6، اما المتغيرة **تاء** فيتم استعمالها في الشطر الأيسر من عملية الشحن (=) قبل ان تشحن باي قيمة معلومة.

1 – 3 خطورة استعمال متغيرات غير مهيأة

وعلمنا ايضا ان انشاء متغيرة مهيكلة يفضي الى متغيرة مجهولة السمات اذا لم تكن السمات قد عرفت بقيم اولية، وهذا يعني ان المتغيرة المتوفرة على سمات مجهولة القيمة لا تمثل بصفة سليمة المفهوم الذي انشأت من اجله، وهذا ما يظهر في خطأ: لم يُعثر على مصدر المرجع و خطأ: لم يُعثر على مصدر المرجع، فالمتغيرة **ت1** لا تمثل تاريخا معلوما، بل ان القيم الغير المعلومة يمكن ان تكون قيم لسمات تاريخ غير صحيح بل غير معقول، إذ من الممكن ان تحتوي السمة **شهر** على القيمة 2000 والسمة **يوم** على القيمة 56789، وبهذا نرى جليا ان عملية تهيئة المتغيرات المهيكلة لتكون سيماتها معلومة امر ضروري وفي غاية الأهمية.

ذاكرة المتغيرات العامة

الشكل 1 : حالة ذاكرة بعد التصريح بالمتغيرتين
تم1 و الآن (السطر6 من النص 2)

النص 2: خوارزم حساب عدد الأيام بين تاريخين

2 – اهمية تهيئة المتغيرات عبر تحليل نص خوارزم:

لإبراز اهمية تهيئة المتغيرات المهيكلة وخطورة استعمال المتغيرات قبل ان تشحن بقيم معلومة متحكم فيها، نقوم بإنجاز خوارزما هدفه حساب عدد الأيام التي مضاها انسان ما منذ تاريخ ولادته، ونسميه **عدد_الأيام_منذ_الولادة** ويتلخص السلوك العام لهذا الخوارزم فيما يلي:

بعد تقديم نفسه، يطلب الخوارزم من المستعمل تاريخ ميلاد شخص ما، وبعد التقاط سمات تاريخ الميلاد، يقوم الخوارزم بحساب عدد الأيام التي مرت منذ هذا التاريخ الى اليوم، ويفترض الخوارزم في نسخته الأولى هذه أن عدد ايام السنة هو 365 وان كل شهر فيه 30 يوما.

الطريقة التي يتبعها خوارزم خطأ: لم يُعثر على مصدر المرجع، تتكون من ثلاث مراحل: في المرحلة الأولى يحسب الخوارزم عدد الايام التي مرت من السنة 0، الشهر 1 و اليوم 1 الى تاريخ ميلاد ما، وفي المرحلة الثانية، يحسب الخوارزم عدد الايام التي مرت من السنة 0، الشهر 1 و اليوم 1 الى اليوم الذي ينفذ فيه الخوارزم، وفي الأخير نحسب الفارق بين العددين.

2 – 1 النسخة الأولى لخوارزم عدد_الأيام_منذ_الولادة (خطأ: لم يُعثر على مصدر المرجع)

تنبيه: النسخة الأولى لخوارزم خطأ: لم يُعثر على مصدر المرجع تحتوي على أخطاء ونقائص، ومن النقائص: التعليمات التي من خلالها يتحصل الخوارزم على سمات تاريخ ميلاد المستعمل.

بعد تفحص دقيق لنص خوارزم خطأ: لم يُعثر على مصدر المرجع، نلاحظ ان المتغيرة **تم1** التي تمثل تاريخ ميلاد ما، والتي صُرِح بها في السطر 6، لم تذكر ابدا في يمين عملية الشحن(=)، ومعنى هذا ان هذه المتغيرة التي لم نقم بتهيئتها تبق محتوية على سمات ذات قيم مجهولة (خطأ: لم يُعثر على مصدر المرجع) طيلة تنفيذ الإجراء اساسي، وهذا امر يمكن ان يؤدي الى عواقب غير متحكم فيها اذا استعملت سمات هذه المتغيرة في تقييم العبارات، وعلى عكس هذا نرى ان سمات المتغيرة **الآن** قد هيأت في الأسطر 13، 14 و15، بعد ان كانت مجهولة عند التصريح بها في السطر 6 (خطأ: لم يُعثر على مصدر المرجع)، ونرى ايضا ان المتغيرتين **عدد_ايام_1** و **ايام_خلت** قد ذكرتا في يمين عملية الشحن (=) قبل ظهورها في الجانب الأيسر من عملية الشحن (=)، اي انها هيئت قبل استعمالها.

فكما هو الشأن بالنسبة للمتغيرات من الأنماط البدائية، فإن الارتكاز في العبارات على سمات غير معلومة المحتوى يؤدي الى نتائج غير معلومة وغير متحكم فيها، وهذا ما يحدث عند تقييم العبارتين الموجودتين في السطر 19 و السطر 23، ففي عبارة السطر 19، نرى استعمال السمات

سنة، شهر و **يوم** من المتغيرة المهيكلة **تم1**، وهذه المتغيرة قد صرح بها في السطر 6، و قبل استعمالها في عبارة السطر 19 لم تَعرف سيماتها اي تهيئة، وبما ان سمات المتغيرة **تم1** غير معلومة، فالنتيجة التي يفضي اليها تقييم عبارة السطر. 19 غير معلومة وغير متحكم فيها، وهكذا تصبع المتغيرة **عدد_ايام_1** تحمل قيمة غير معلومة، ولكون عبارة السطر 23 ترتكز. على المتغيرة **عدد_ايام_1**، فان عملية تقييمها تنتج قيمة غير معلومة ايضا، اي ان المتغيرة **ايام_خلت** تصبح محتوية على قيمة غير معلومة وغير متحكم فيها.

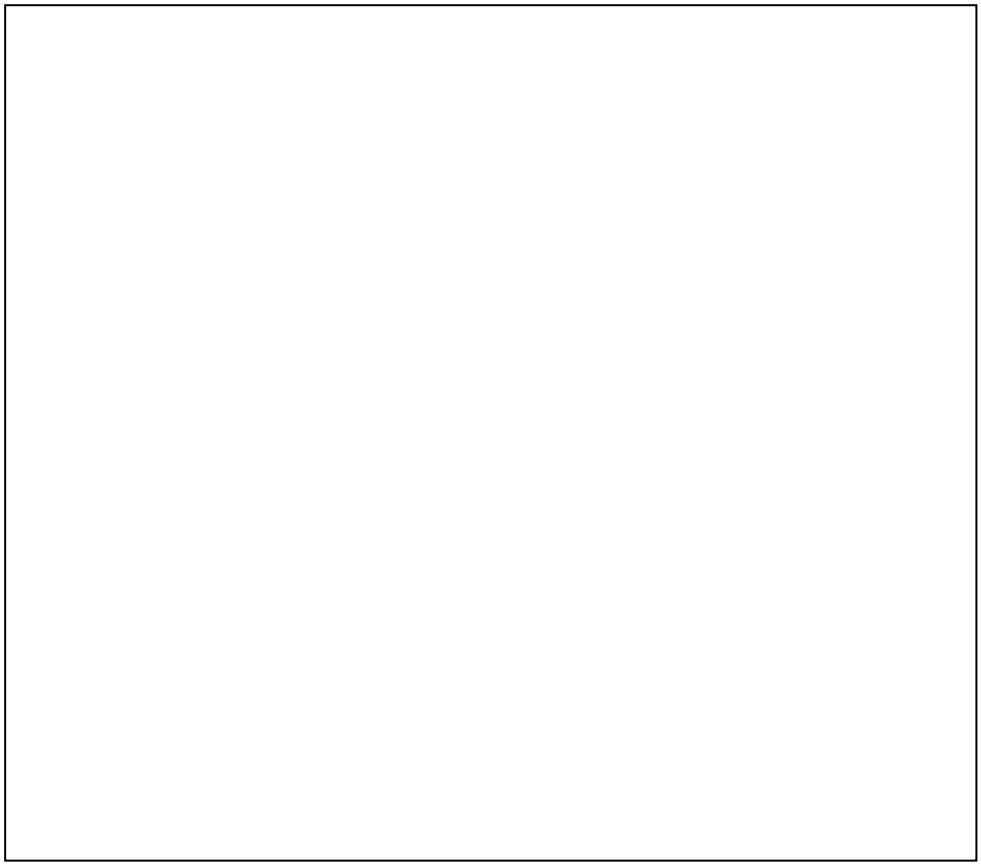
وكنتيجة لمثل هذه الحالة يصبح للخوارزم سلوكا غير متحكم فيه، لكوننا لا نعلم حقيقة كل المعطيات التي ينطلق منها، ولتجاهلنا التحكم في قيمة بعض منها، وهكذا، بعدم تحكمنا في بعض المعطيات، يظهر لنا انه بنفس المعطيات ينتج الخوارزم في كل تنفيذ نتائج مختلفة، وفي بعض الأحيان نتائج غير معقولة، وهذا يعني ان ما في خطأ: لم يُعثر على مصدر المرجع ليس بخوارزم صحيح، فمثلا في تنفيذ اولي، يمكن ان تكون القيم المجهولة لسمات المتغيرة **تم1** هي القيم 0، 0 و 0، و بما ان المتغيرة **تم1** لم تهيأ بقيم متحكم فيها، فهذه القيم هي التي ستستعمل في تقييم العبارات، وهكذا تكون نتيجة تقييم عبارة السطر 19 هي القيمة 0، و تستعمل هذه القيمة في تقييم عبارة السطر 23، و في تنفيذ آخر يمكن ان تكون القيم المجهولة لسمات المتغيرة **تم1** هي القيم 10000، 20000 و 30000، وهذه قيم غير معقولة، وهذه هي القيم التي ستستعمل في تقييم عبارة السطر. 19، ثم في السطر 23.

2 – 2 النسخة المصححة لخوارزم حساب عدد ايام العمر:

في هذه النسخة (خطأ: لم يُعثر على مصدر المرجع) نتدارك النقص الخطير الذي وجد في خطأ: لم يُعثر على مصدر المرجع، فهذه المرة نقوم بتهيئة المتغيرة **تم1** التي تمثل تاريخ ميلاد ما، وهذه المرة تتم التهيئة عبر تعليمة **اقرأ** التي تقوم بأخذ ما يكتب على لوحة الحروف ووضعه في المتغيرة التي تذكر لها، ففي السطر 15 تذكر تباعا ثلاث متغيرات التي تمثل سمات المتغيرة **تم1**، فبمثل هذه الكتابة تقوم التعليمة بقراءة ثلاث قيم، تقرأ اولا قيمة أولى لتوضع في المتغيرة التي ذكرت في الأول، اي السمة المسماة **اليوم**، ثم تقرأ قيمة ثانية فتوضع في المتغيرة التي ذكرت في المرتبة الثانية، اي السمة المسماة **الشهر** وأخيرا تقرأ قيمة ثالثة فتوضع في المتغيرة التي ذكرت في المرتبة الثالثة، اي السمة المسماة **السنة**.

في هذه النسخة(خطأ: لم يُعثر على مصدر المرجع) استبدلنا ايضا طريقة تهيئة المتغيرة المهيكلة المسماة **الآن**، فعوض تثبيتها لتاريخ ما، جعلنا تثبيت سيماتها عبر التعليمة **اقرأ** كما فعلنا مع المتغيرة **تم1**، وبهذا التغيير يصبح الخوارزم قادرا على التكيف مع اي وقت ينفذ فيه، فاذا نفذ في تاريخ 18 /06 /2016 فانه يعطي اجابة صحيحة، لأن المتغيرة المسماة **الآن** ستشحن عبر التعليمة

اقرأ بالقيم 18 و 6 و 2016، واذا اعيد تنفيذ الخوارزم في يوم 16 /08 /2016 فانه يعطي اجابة صحيحة، لأن المتغيرة **الآن** ستشحن عبر التعليمة اقرأ بالقيم 16 و 8 و 2016، وهذا الأمر لم يكن ممكنا مع النسخة الأولى اين ثبتت المتغيرة **الآن** بالقيم 12 و 12 و 2016.



النص 3: النسخة المصححة لخوارزم حساب عدد الأيام بين تاريخين

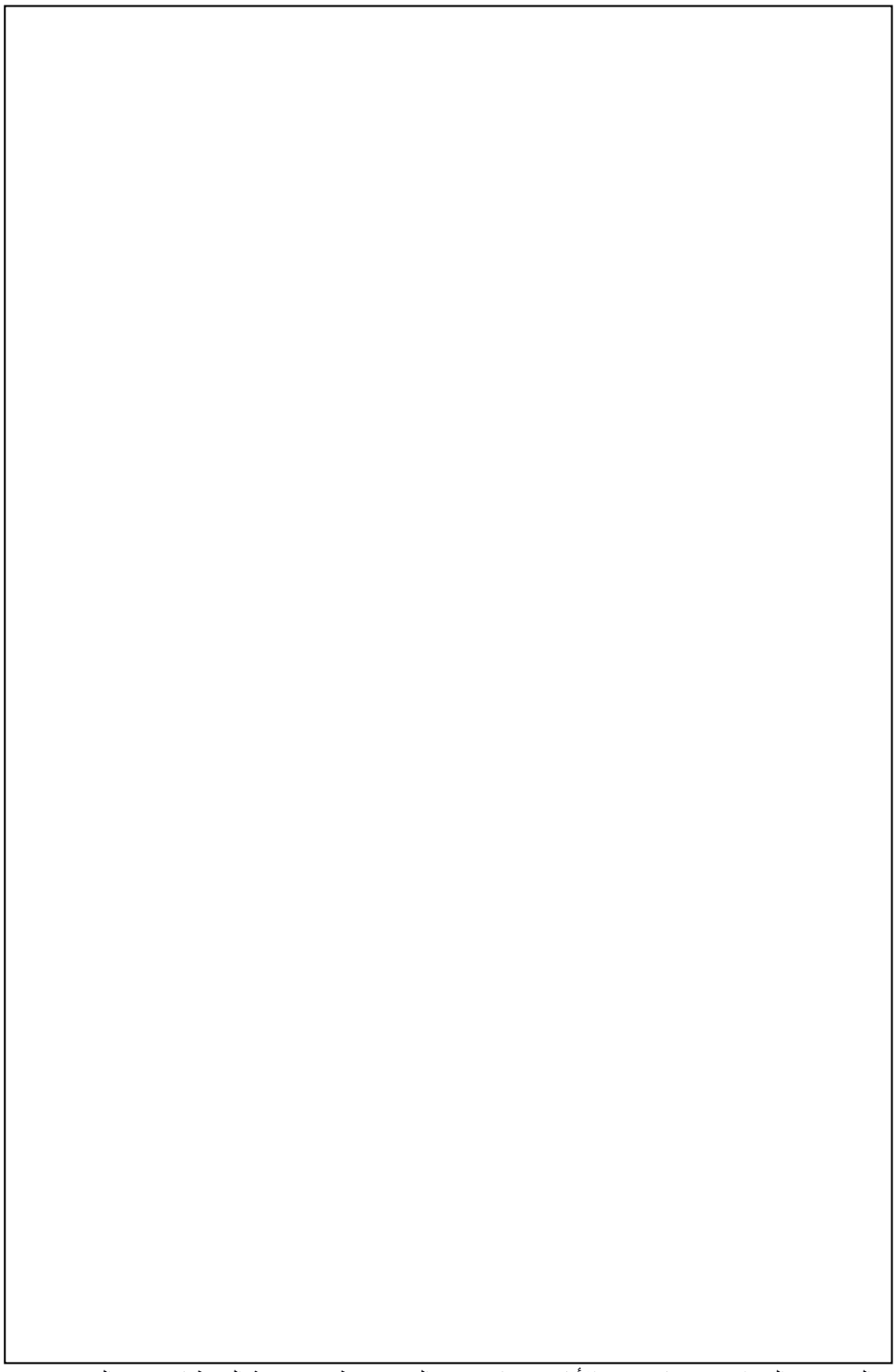
2 – 3 ضعف النسخة المصححة لخوارزم حساب عدد الأيام بين تاريخين:

عملية ادخال القيم عبر لوحة المفاتيح لا تخلو من الخطأ، فيمكن للمستعمل ان يدخل قيم غير صحيحة وغير معقولة عن قصد او غير قصد، ففي مثل هذه الحالة لن يعطي الخوارزم نتيجة صحيحة، ولنفرض مثلا ان المستعمل ادخل القيم التالية 24، 92، 2016 في المرحلة الأولى، كرد على التعليمة اقرأ (السطر 15 من خطأ: لم يُعثر على مصدر المرجع)، ففي هذه الحالة يُتم الخوارزم عمله وطبعا لن يعطي نتيجة معقولة لكون قيمة الشهر التي ادخلها المستعمل غير معقولة.

لجعل الخوارزم قادرا على العمل في ظروف معقولة، وجب علينا ان نحسنه بتزويده ببعض الذكاء ليصبح قادرا على التعامل مع الحالات الغير معقولة، ولن يقوم بعمله الا في اطار معطيات صحيحة ومعقولة، فعليه مثلا ان يتأكد ان قيمة **اليوم** هي من 1 الى 30 لكوننا اعتبرنا ان عدد ايام

الشهر هي 30 يوما، وعلى الخوارزم ان يتأكد ان قيمة السمة **شهر** هي ما بين 1 و12، وهذا ما نراه في خطأ: لم يُعثر على مصدر المرجع.

أ أخيرا على الخوارزم ان يتأكد ان التاريخ الذي وضع في المتغيرة **تم1** اقدم من التاريخ الذي وضع في المتغيرة **الآن**، وفي صورة ما اذا وجد خللا في القيم ينبه الخوارزم المستعمل ليصحح القيم التي يكتبها على لوحة الحروف.



النص 4: الخوارزم حساب عدد الأيام بين تاريخين: التحقق فقط من صحة المعطيات دون التحقق من ترتيب التاريخين

لإضافة التعليمات التي من شأنها التحقق من صحة المعطيات التي يتلقاها الخوارزم، يمكن اتباع احدى السياسيتين التاليتين:

- التحقق المبكر: في هذه السياسة، يتحقق الخوارزم من صحة كل المعلومة فور التقاطها، اي مباشرة بعد التعليمة **اقرأ**، فان كانت غير صحيحة ينبه الخوارزم المستعمل ويدعوه الى تصحيح الخطأ.
- التحقق المتأخر: في هذه السياسة يلتقط الخوارزم كل المعطيات، وقبل الشروع في استعمالها يتحقق من صحتها، فان كانت احدى المعطيات غير صحيحة ينبه الخوارزم المستعمل بالخطأ ثم يدعوه الى اعادة ادخال القيم الخاطئة او كل القيم، وهذه السياسة يمكن ان تجعل مرحلة التحقق والتصحيح في غاية التعقيد.

فيما يخصنا، فإننا نتبع السياسة الأولى التي تظهر من خلال خطأ: لم يُعثر على مصدر المرجع، ففي هذه السياسة نضع عملية قراءة قيمة ما في دائرة غير متناهية، والخروج من الدائرة يكون باستعمال التعليمة غادر عندما تكون القيمة صحيحة، فمثلا في الأسطر ما بين 15 و20 من خطأ: لم يُعثر على مصدر المرجع، نجد الدائرة المتخصصة في محاولة التقاط قيمة صحيحة للسمة **'اليوم'**، فكانت تكلفة هذه العملية، اي التحقق من صحة القيمة التي قرأت، 5 تعليمات، وهذه الطريقة تستعمل مع كل السمات التي يلتقطها الخوارزم، وبما انه يجب التحقق من 6 معطيات (3 معطيات لسمات المتغيرة **تم1** و 3 معطيات لسمات المتغيرة **الآن)**، فان تكلفة التحقق تكون 36 تعليمة، و هذا دون حساب التعليمات التي سوف تتحقق من ان التاريخ الذي تحمله المتغيرة **تم1** اقدم من التاريخ الذي تحمله المتغيرة **الآن.**

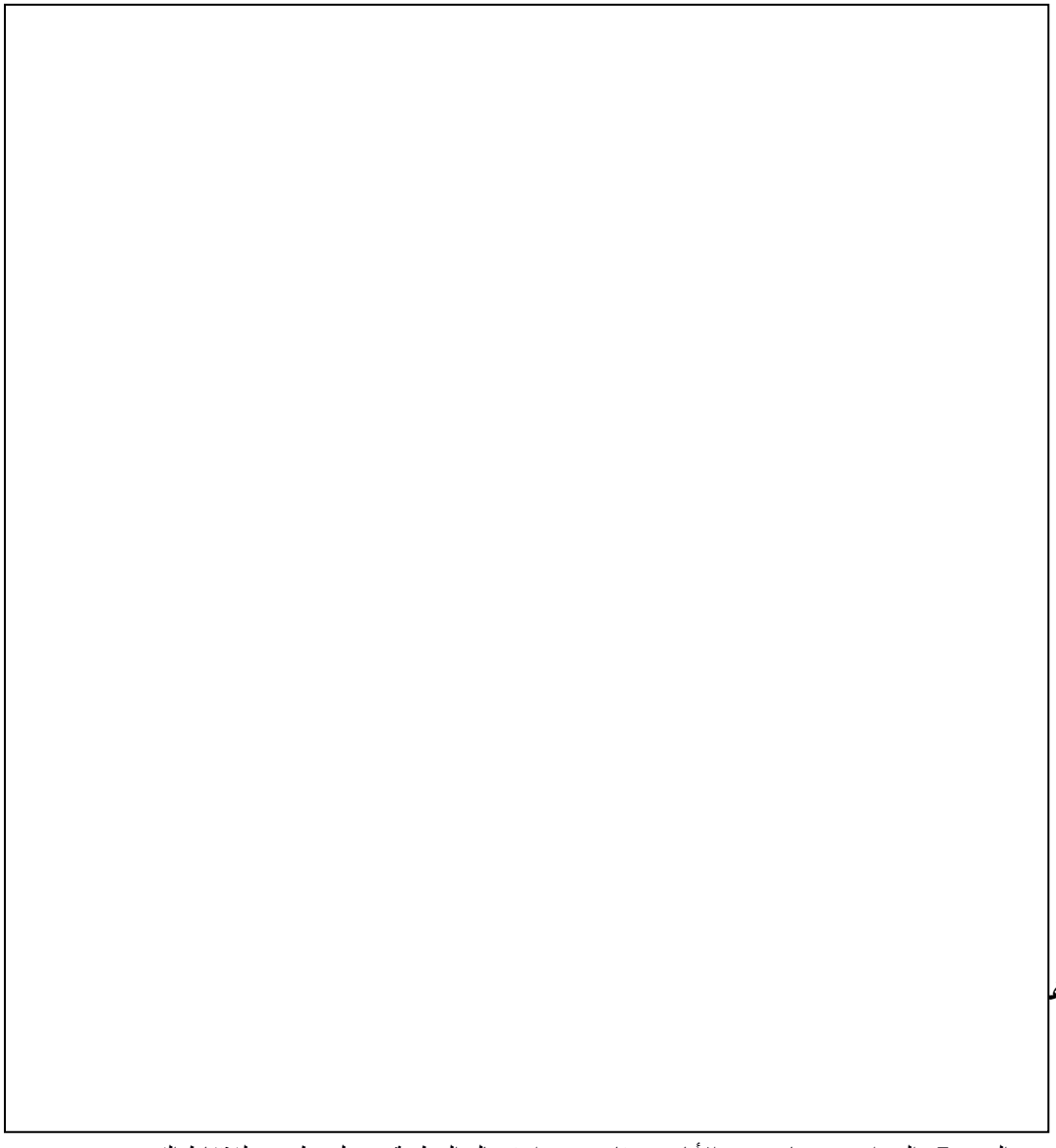
2 – 4 تحسين حجم الخوارزم:

الطريقة السابقة في كتابة التعليمات التي بموجبها يتحقق الخوارزم من صحة المعلومات طريقة مباشرة، سهلة و ساذجة، تكون مقبولة وفعالة اذا كان عدد المعطيات قليل جدا، اما اذا كان عدد المعطيات كبير، فحجم الخوارزم يصبح ضخما، ويصعب التحكم فيه، وفي مثل هذه الحالة يجب ان نفكر مليا لإيجاد طريقة تجعل طول الخوارزم غير مرتبط بعدد المعطيات التي يجب التحقق منها.

الطريقة التي توصلنا اليها بعد تفكير تكمن في انجاز وظيفة همها قراءة قيمة طبيعية واحدة وفي نفس الوقت تقوم بالتحقق من صحتها، فعندما يُطلب من هذه الوظيفة قراءة قيمة ما، تدخل هذه الوظيفة في دوامة، فيها تقرأ القيمة التي تكتب على لوحة المفاتيح وفيها تتحقق الوظيفة من صحة القيمة(أمعقولة هي ام لا)، ولبلوغ هدفها تحتاج الوظيفة الى المعطيات التالية على الأقل:

- اصغر قيمة طبيعية مسموح بها.
- اكبر قيمة طبيعية مسموح بها.

في خطأ: لم يُعثر على مصدر المرجع سمينا **حصل_طبيعي** هذه الوظيفة، واضافة الى المداخل الخاصة بأصغر واكبر قيمة، زدنا مدخلا من نوع سلسلة يوضع فيه اسم السمة او المتغيرة المستهدفة من عملية القراءة، وتستعمل الوظيفة هذا الاسم في سياق الحوار مع المستعمل بشكل فعال، فعندما يطلب من المستعمل ادخال قيمة تُمثل الشهر تكتب له الوظيفة على الشاشة **شهر:** وعندما يطلب من المستعمل ادخال قيمة تُمثل اليوم تكتب له الوظيفة على الشاشة **اليوم:**، وبفضل هذه الطريقة، كلما اردنا قراءة معطية اضافية، زدنا فقط تعليمة واحدة، ففي النص احتجنا الى 6 معطيات، فكتبنا ست تعليمات ليتحصل الخوارزم على 6 معطيات صحيحة (الأسطر من 24 الى 30)، ولنفرض ان الخوارزم يحتاج الى اربع متغيرات من النمط تاريخ تلتقط قيم سيماته عبر لوحة الحروف، ففي هذه الحالة اذا استعملنا الطريقة المباشرة الساذجة علينا كتابة 60 تعليمة لقراءة القيم ووضعها في سمات المتغيرات المهيكلة من النمط تاريخ، اما اذا استعملنا الطريقة الثانية، فعدد التعليمات المستعملة هي 17: عدد تعليمات الوظيفة **حصل_طبيعي**، اي 5 تعليمات، و12 تعليمة طلب تنفيذ الوظيفة.



الصف

الجزء الثاني: الفصل الثالث: تهيئة المتغيرات المهيكلة

ملاحظة: في السطرين 26 و 30 ذكرنا القيمة 30000 كأكبر قيمة للسمة سنة، و في حقيقة الأمر يمكن ان نضع اي قيمة، تكون نوعا ما اكبر من السنة الحالية، فمثلا اذا رأينا ان مثل هذا الخوارزم يبق مستعملا لمئة سنة بعد السنة الحالية، يمكن ان نكتب 2121 كحد اكبر للسمة سنة اذا كانت السنة الحالية هي 2021.

2 – 5 ادراج خاصية التحقق من اقدمية التاريخ الأول:

لحد الآن لم نتطرق في كتابة الخوارزم الى التحقق من ان التاريخ الأول الموجود في المتغيرة **م1** اقدم من التاريخ الموجود في المتغيرة **الآن**، ولبلوغ هذا الهدف دون احداث تغيير كبير على مستوى الإجراء اساسي، نستعمل وظيفة تقارن بين تاريخين، ومثل هذه الوظيفة قد انجزناها في الفصل الأول وهي الوظيفة **طبيعي قارن_تاريخين(طبيعي سنة1، شهر1، يوم1، سنة2، شهر2، يوم2)**، ولا مانع من اعادة استعمالها.

2 – 5 – 1 تذكير بسلوك الوظيفة قارن_تاريخين

في هذه الوظيفة تمثل المداخل الثلاث الأولى **سنة1، شهر1** و **يوم1** تاريخا اولا والمداخل الثلاث التالية **سنة2، شهر2** و **يوم2** تاريخا ثانيا، فاذا كان التاريخان متساويان ترجع الوظيفة القيمة 0، واذا كان التاريخ الأول اكبر من التاريخ الثاني (اي ان التاريخ الأول احدث من الثاني) ترجع الوظيفة القيمة 1، اما اذا كان التاريخ الأول اصغر من التاريخ الثاني (اي ان التاريخ الأول اقدم من الثاني) ترجع الوظيفة قيمة اصغر من –1.

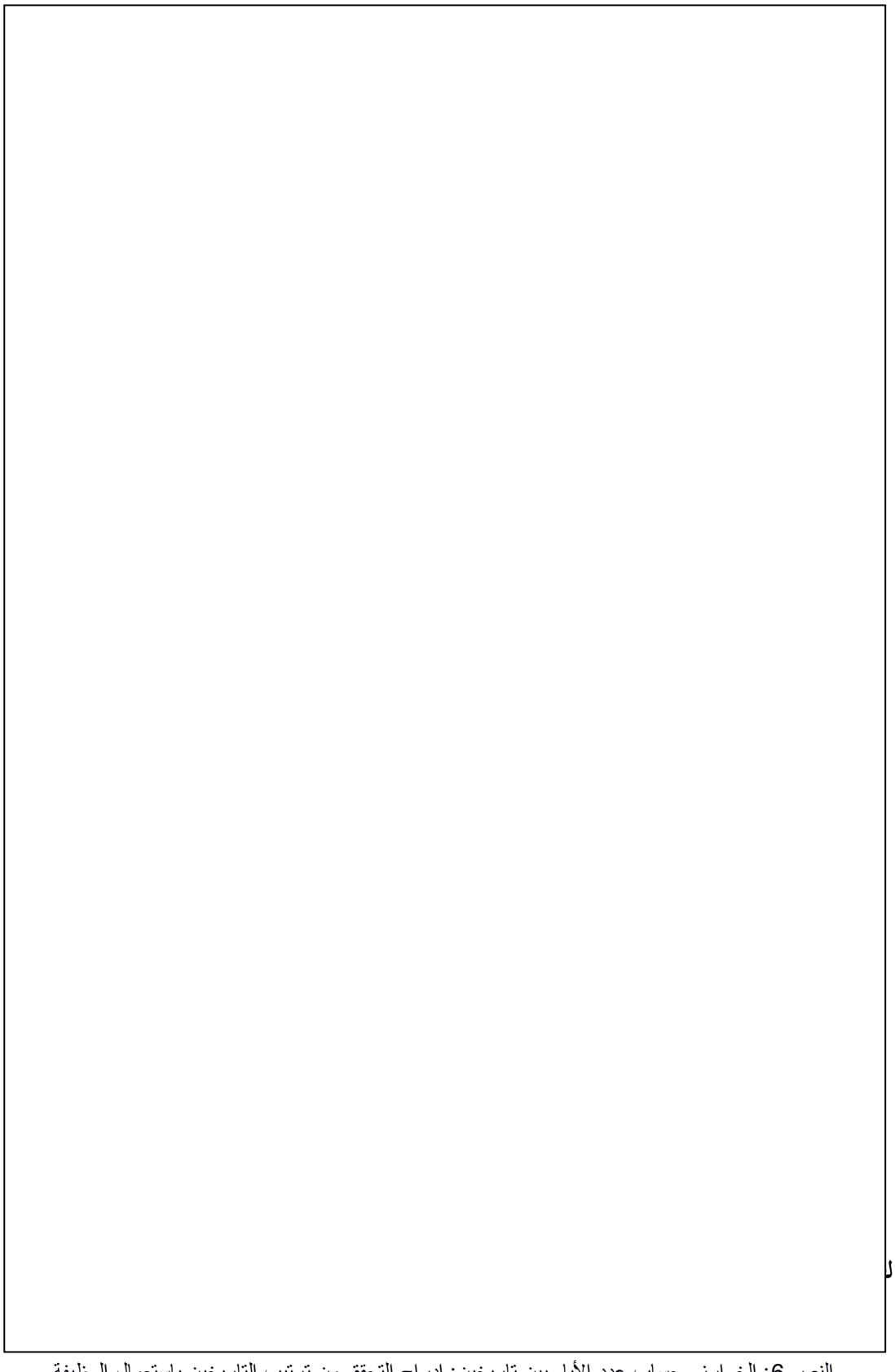
2 – 5 – 2 تحسين مداخل الوظيفة قارن_تاريخين:

انجزت الوظيفة **قارن_تاريخين** بست مداخل عندما كنا لا نعلم ان اللغة الشرمزية قادرة على تحمل وصف المعلومات المعقدة باستعمال الأنماط المهيكلة، وبوجود مثل هذه الأنماط نتسائل: هل من الجيد ترك ستة مداخل او اعادة التفكير في تنظيم المداخل باستعمال النمط تاريخ.

إذا استعملنا النمط تاريخ لا نحتاج الى 6 مداخل، بل مدخلين فقط، ولا نحتاج لذكر كل السمات في المداخل الستة، فبمدخلين من النمط **تاريخ** تسهل كتابة رأس الوظيفة وطلب تنفيذها ، وهكذا يصبح رأس الوظيفة **قارن_تاريخين** على الشكل التالي: **طبيعي قارن_تاريخين(تاريخ** تاريخ1، تاريخ2**)**.

2 – 5 – 3 تذكير بسلوك الوظيفة قارن_تاريخين:

تقوم الوظيفة **قارن_تاريخين** (خطأ: لم يُعثر على مصدر المرجع من السطر. 17 الى السطر 28) اولا بمقارنة السمة **سنة** للمدخلين **تاريخ1** و **تاريخ2** في السطرين 18 و 19، ففي السطر. 18 تحاول الوظيفة ان ترى هل سنة **تاريخ1** اكبر من سنة **تاريخ2**، فان كان الأمر كذلك تحسم الوظيفة امرها وترجع النتيجة 1 وتنته فورا منبئة ان **تاريخ1** اكبر من **تاريخ2**، وفعلا، مهما كانت قيمة الشهر واليوم لكل من **تاريخ1** و**تاريخ2** فان **تاريخ1** احدث (او اكبر) من **تاريخ2**.



النص 6: الخوارزم حساب عدد الأيام بين تاريخين: ادراج التحقق من ترتيب التاريخين باستعمال الوظيفة قارن_تاريخين

اذا لم يتحقق ما في السطر 18، فمعناه ان **تاريخ1** اما ان يكون اصغر من **تاريخ2** او يساويه، وللحسم في هذا الأمر تحاول الوظيفة ان ترى هل سنة **تاريخ1** اصغر من سنة **تاريخ2**، فان كان الأمر كذلك تحسم الوظيفة امرها في السطر 19 و ترجع النتيجة −1 وتنته فورا منبئة ان **تاريخ1** اصغر من **تاريخ 2**، و فعلا، مهما كانت قيمة الشهر واليوم لكل من **تاريخ1** و **تاريخ2** فان **تاريخ1** اقدم (او اصغر) من **تاريخ2**، اما ان لم يتحقق ما في السطر. 19، فمعناه ان السمة **سنة** في **تاريخ1** تساوي السمة سنة في **تاريخ2**، وفي هذه الحالة يجب على الخوارزم ان يقارن سمة **الشهر** لكل من **تاريخ1** و **تاريخ2**.

في السطرين 21 و 22 يعاود الخوارزم نفس الطريقة التي اتبعها في السطرين 18 و 19، لكن هذه المرة مع السمة المسماة **الشهر** كون السمة المسماة **السنة** للمدخلين **تاريخ1** و **تاريخ2** متساويتين، وهكذا ان كان شهر **تاريخ1** اكبر من شهر **تاريخ 2**، يحسم الخوارزم امره و يرجع القيمة 1 وينته فورا، اما ان لم يتحقق شرط السطر 21، ينتقل الخوارزم للسطر 22 و يحاول ان يرى هل السمة **الشهر** في **تاريخ1** اصغر من السمة **الشهر** في **تاريخ2**، فان تحقق شرط السطر 22، اي ان شهر **تاريخ1** اصغر من شهر **تاريخ2**، يحسم الخوارزم امره و يرجع القيمة −1 وينته فورا، اما ان لم يتحقق شرط السطر 22 بعد عدم تحقق شرط السطر 21، فهنا يتبين للخوارزم ان قيمة الشهر في كلا المدخلين متساويتين وكذلك الأمر بالسمة **السنة** للمدخلين، وهنا ينتقل الخوارزم لمقارنة قيم السمة المسماة **اليوم** في كلا المدخلين.

أخيرا، في السطرين 24 و 24 يعاود الخوارزم نفس الطريقة التي اتبعها اولا في السطرين 18 و19 و ثانيا في السطرين 21 و22، وهذه المرة يهتم الخوارزم بيوم كلا التاريخين، فإن كانت قيمة اليوم في المدخل **تاريخ1** اكبر من قيمة اليوم في المدخل **تاريخ2**، يقرر الخوارزم ان **تاريخ1** احدث من **تاريخ2**، فيرجع 1 و ينته، وان لم يتحقق شرط السطر 24، ينتقل الخوارزم الى السطر. 25 ليحاول معرفة هل يوم **تاريخ1** اصغر من **تاريخ2**، فان كان كذلك، يقرر الخوارزم ان **تاريخ1** اقدم من **تاريخ2**، فيرجع القيمة −1 و ينته فورا، وهنا ان لم يتحقق شرط السطر 25، بعد ان لم تتحقق اي من الشروط السابقة، يفهم الخوارزم ان كل سمات المدخلين **تاريخ1** و **تاريخ2** متساوية، فيقرر ان التاريخين متساويين فيرجع في السطر 27 القيمة 0 مخبرا ان التاريخين الواردين في مدخليه متساويين.

2 – 6 تكلفة ادراج التحقق من ترتيب التاريخين في الإجراء اساسي

بفضل الطريقة التي اتبعناها، اي استعمال الوظيفة **قارن_تاريخين**، تكون تكلفة ادراج التحقق من ترتيب التاريخين في الإجراء اساسي بسيطة جدا، وتتمثل فقط في تعليمتين: التعليمة الشرطية في السطر 45 وتعليمة السطر 46 التي تنفذ في حالة ما اذا وجد خلل في ترتيب التاريخين.

يتم تشغيل الوظيفة **قارن_تاريخين** في السطر. 45 كأحد عوامل العبارة المنطقية للتعليمة الشرطية المنطقية، والعبارة المنطقية هي **قارن_تاريخين(الآن، ت1) > 0**، وعند تقييم هذه العبارة المنطقية، وحسب جدول اولويات العمليات، فان اول عملية تنتخب للتنفيذ هي عملية تشغيل الوظيفة المتمثلة بالقوسين المحتويين على المداخل، وبعد تشغيل الوظيفة واتمام تنفيذها، وكما سبق وان شرحناه، توضع النتيجة التي ترجعها الوظيفة في متغيرة ظرفية، نسميها مثلا **مظ1**، فتصبح العبارة الشرطية على الحالة الجديدة و هي **'مظ1 > 0'**، وهذه العبارة تحتوي على عملية واحدة فقط وهي العملية التي تبين هل ما في **مظ1** اقل من **0**، اي بصورة اوضح هل **تاريخ1** اقدم من **تاريخ2**، فان صحت نتيجة التقييم (النتيجة هي القيمة المنطقية صحيح)، يفهم الخوارزم ان هناك خللا في ترتيب التاريخين، فيخبر. المستعمل عن الخلل بواسطة تعليمة السطر. 46 ثم ينته فورا من التنفيذ مرسلا الى الشاشة العبارة "**شكرا لاستعمالك الخوارزم... الى اللقاء**"، اما في صورة ما اذا لم تصح العبارة، اي ان **تاريخ1** اكبر او يساوي **تاريخ2**، فان الخوارزم ينفذ التعليمات من السطر 48 الى السطر. 56، وهي نفس التعليمات التي كتبت في النسخة السابقة، ثم ينته الخوارزم بعد إرسال البيان "**شكرا لاستعمالك الخوارزم... الى اللقاء**" الى الشاشة.

3 – كيفية تهيئة المتغيرات المهيكلة

انطلاقا. مما تحصلنا عليه من معرفة، يمكننا ان نميز بين ثلاث طرق لوصف القيم الأولية التي تهيئ بها المتغيرات المهيكلة فور. انشائها:

- وصف القيم الأولية في سياق التعريف بالنمط المهيكل.
- وصف القيم الأولية في سياق التصريح بالمتغيرة المهيكلة.
- تهيئة السمات بالقيم الأولية عبر التعليمات في جسم الوظائف او الإجراءات.

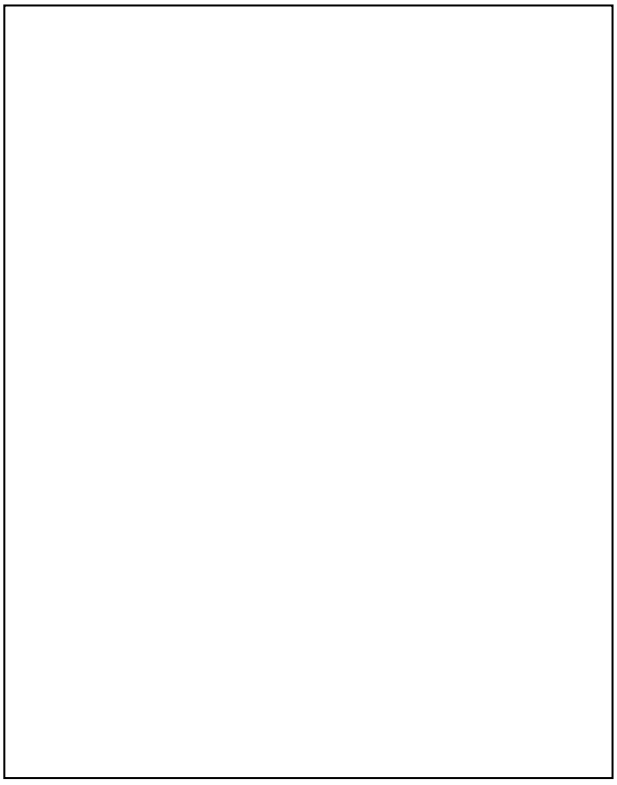
3 – 1 الطريقة الأولى: وصف القيم الأولية في سياق التعريف بالنمط المهيكل

(خطأ: لم يُعثر على مصدر. المرجع):

في هذه الحالة، عند التعريف بهيكل النمط، تُذكر لكل سمة قيمة أولية، وعند انشاء المتغيرة المهيكلة، توضع القيمة التي ذكرت لكل سمة في موضعها، اي في المتغيرة الممثلة للسمة، وبهذه الطريقة، تنشأ كل المتغيرات المهيكلة على صورة واحدة، تمثل حالة واحدة من الواقع المعاش، وهذه

الطريقة قليلة الاستعمال لضرورة اعادة تهيئة المتغيرة لتمثل حالة غير الحالة التي وصفت في تعريف النمط المهيكل، ولمزيد من الشرح، نحلل فيما يلي ما ورد في خطأ: لم يُعثر على مصدر المرجع.

يمثل خطأ: لم يُعثر على مصدر المرجع الكتابة الأولى لخوارزم ضخم هدفه محاكات سباق السيارات، ونلاحظ في الأسطر. من 1 الى 4 انه في سياق التعريف بالنمط **تاريخ** حددت قيم اولية تمثل تاريخا. محددا واحدا، و هو 1 يناير 2016، في السطر 24 صرح بسبع متغيرات من النمط سيارة الذي عرف في الأسطر. من 15 الى 22، و المتغيرات المسمات **س1**، **س2** و **س3** الى **س7** تمثل في حقيقة الأمر سبع سيارات تكون في الغالب كلها مختلفة، ونلاحظ في خطأ: لم يُعثر على مصدر المرجع التمثيل الجزئي للذاكرة المحلية للإجراء اساسي التي صرحت فيه المتغيرات من **س1** الى **س7**.



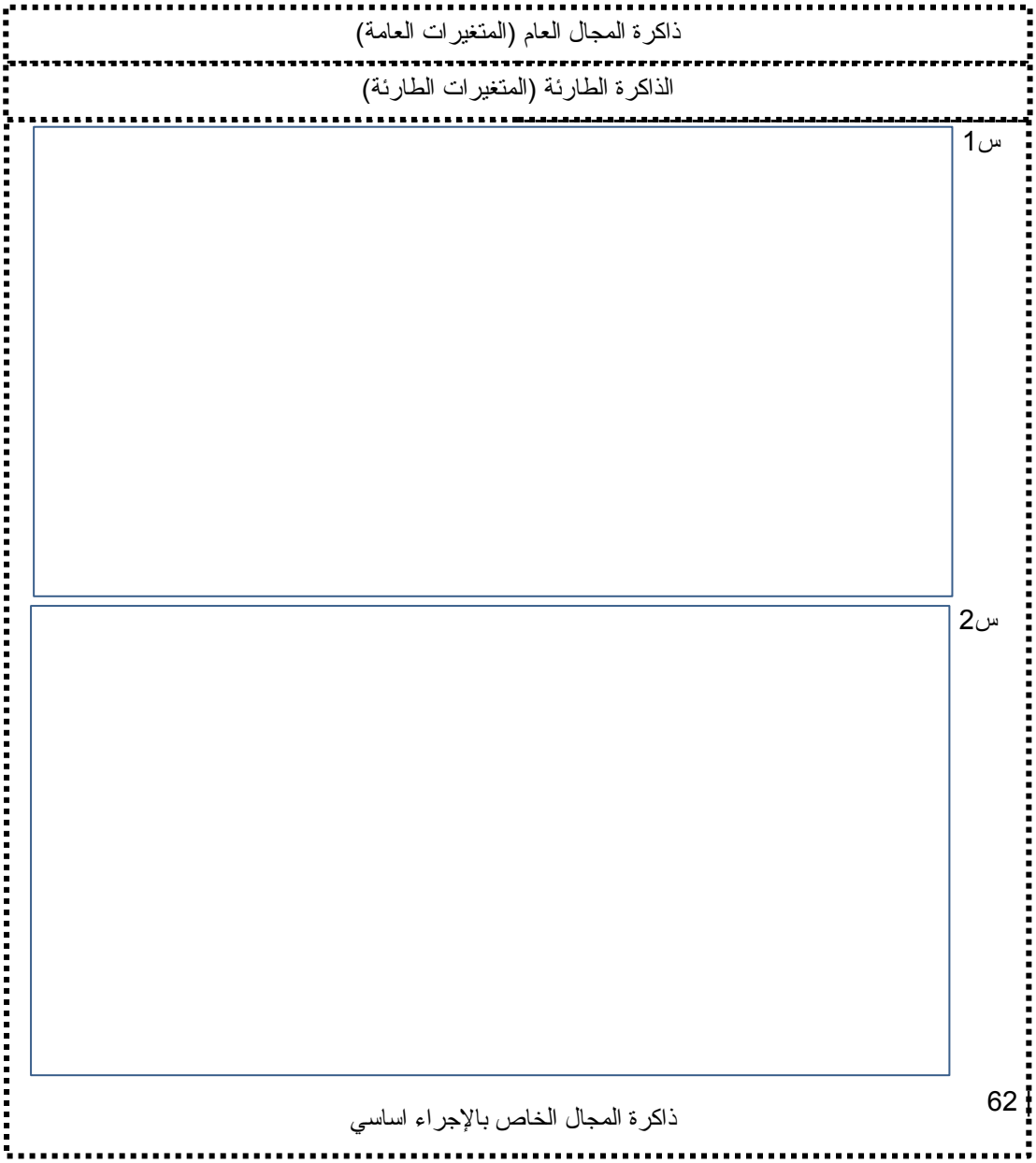
النص 7 : التعريف بالقيم الأولية لتهيئة المتغيرات من النمط تاريخ

نلاحظ ايضا من خلال تعريف النمط **سيارة**، انه لكل سيارة ثلاث تواريخ (تاريخ ولادة مالك السيارة، تاريخ التصنيع و تاريخ اول استعمال)، ونلاحظ هنا امرا غريبا ناتج عن استعمال الطريقة الأولى في تهيئة المتغيرات المهيكلة بقيم اولية معرفة في النمط: كل التواريخ (جزء منهم ظاهر في خطأ: لم يُعثر على مصدر المرجع)، وعددها 21 (3 تواريخ لكل متغيرة من المتغيرات السبع) متساوية، اي ان تاريخ تصنيع **س1** هو نفس تاريخ اول استعمال للسيارة **س1** و لكل السيارات الاخرى، وهو ايضا تاريخ ميلاد مالك السيارة **س1**، وتاريخ ميلاد مالك السيارة **س2** وتاريخ ميلاد

مالك السيارة **س3** الخ... وهذه حالة بعيدة عن الواقع المعاش، ولجعل المتغيرات تمثل واقعا حقيقيا، يتوجب اعادة تسديد قيم سمات كل المتغيرات، اي كتابة ما يجب من تعليمات لشحن القيم الصحيحة في كل سمة من سمات 21 تاريخ.

3 – 2 الطريقة الثانية: وصف القيم الأولية في سياق التصريح بالمتغيرة المهيكلة:

في مثل هذه الطريقة، علينا ان نبرز في الكتابة هيكل المتغيرة وكل مكوناتها، ونضع في كل مكون من مكونات الهيكل القيمة المناسبة للسمة الموجودة في المكون، ولإبراز مكونات هيكل المتغيرة المهيكلة نستعمل الحاضنتين كما يظهر في خطأ: لم يُعثر على مصدر المرجع من السطر 12 الى السطر 16، ففي هذه الأسطر نرى التصريح بمتغيرة من النمط **شخص**، وهذا النمط يحتوي في هيكله على مستويين : في المستوى الأول نجد **الاسم، اللقب، العنوان، الولادة، الطول، الوزن**، وحالة الزواج (السمة **متزوج**)، وفي المستوى الثاني نجد سمات **الولادة** وهي **السنة، الشهر، اليوم، الساعة** و**الدقيقة**.



الشكل 2 : تمثيل جزئي للذاكرة المحلية للإجراء اساسي من النص 7 (المتغيرتين س1 و س2 فقط)

في السطر 12 نرى بداية استعمال هذه الطريقة الثانية مع المتغيرة **ش1**، ولبدء وصف قيم المستوى الأول من المتغيرة **ش1**، نفتح حاضنة اولى (السطر 12) مباشرة بعد الرمز =، ثم نسرد تباعا قيم سمات المستوى الأول، حسب الترتيب الذي ذكر في تعريف النمط شخص، اي: **الاسم، اللقب، العنوان، الولادة، الطول، الوزن،** واخيرا السمة **متزوج.**

النص 8 : التصريح بمتغيرة مهيكلة مع التهيئة الفورية لكل سماتها

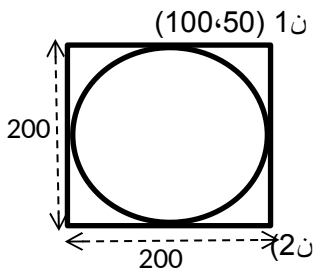
عند وصولنا الى سمة المسماة **الولادة**، نرى انها تنتمي الى نمط مهيكل آخر وهو النمط **تاريخ**، في هذه الحالة ننتقل الى مستوى اعمق من المستوى الحالي، وهو المستوى الثاني فيما يخصنا، ولإبراز المستوى الثاني نفتح حاضنة ثانية لندخل بها المستوى الثاني الممثل هنا للنمط **تاريخ**، وفي هذا المستوى الثاني الخاص بالنمط **تاريخ**، نسرد تباعا كل قيم سمات التاريخ، اي: **السنة** و **الشهر** و **اليوم** و **الساعة** و **الدقيقة** (السطر 14)، وعند الانتهاء من سرد قيم سمات التاريخ، ينته المستوى الثاني، فنستعمل حاضنة الغلق، لنعود الى المستوى الأول، فنكمل سرد باق السمات التابعة للمستوى الأول، وعند الانتهاء من سرد كافة سمات المستوى الأول نغلقه بحاضنة وتنته بذلك عملية التصريح مع التهيئة الفورية للسمات بقيم اولية.

ملاحظة: عندما لا نريد ان نصف لسمة ما قيمة اولية، لا نذكر اي قيمة في موقعها، نستعمل الفاصلة ونترك مكان السمة فارغا، فمثلا اذا كانت الساعة والدقيقة غير مهمتين في متغيرة ما من النمط تاريخ نكتب {1998، 6، 19، ، }.

2-3 -1 مثال ثاني حول الطريقة الثانية:

يحتوي خطأ: لم يُعثر على مصدر المرجع على التعريف بأربع انماط مهيكلة: **النقطة، المستطيل، الدائرة** و **الرسم**. النمط **المستطيل** يُعرف مستطيلا ما انطلاقا من نقطتين متقابلتين **ن1** و **ن 2** (خطأ: لم يُعثر على مصدر المرجع)، و معنى هذا ان هيكل المستطيل ذو مستويين، ونفس الأمر بالنسبة للنمط **الدائرة**، اما النمط **رسم** فهو ذو ثلاث مستويات، مع وجود مستويين اثنين من المستوى الثاني في النمط **رسم**: مستوى ثان خاص بالمستطيل الذي تمثله السمة **م1**، و مستوى ثان خاص بالسمة **د1**.

لتهيئة المتغيرة **ر1** بالقيم الأولية (السطر 18) نفتح اولا الحاضنة الأولى التي تدخلنا في المستوى الأول، ثم نبدأ بسرد قيم المستوى الأول وهي على التوالي: **عدد_المستطيلات، عدد_الدوائر، م1** و **د1**، فنكتب عدد المستطيلات و هو 1، ثم عدد الدوائر و هو 1، ثم نصل الى سمة **م1** وهي من النمط **'المستطيل'**، فهنا ندخل في المستوى الثاني فنفتح حاضنة لقيم **م1** وفي هذا المستوى الثاني نقوم بسرد قيم سمات المستطيل، فنبدأ بالسمة **ن1** وهي ايضا سمه مهيكلة، وعلينا هنا الدخول في مستوى اقل من المستوى الثاني، وهو المستوى الثالث، فنفتح له حاضنة وندخل فيه، ثم نقوم بسرد قيم المستوى الثالث المتعلق بالسمة **ن1**، وهذه القيم الخاصة بالسمات **س** و **ج**، هي من النمط طبيعي، فنكتب اذا على التوالي قيمة **س** و **ج**، وهي حسب خطأ: لم يُعثر على مصدر المرجع، القيمتين 100 و 50، وعند الانتهاء من سرد قيم سمات **ن1**، نغلق الحاضنة ونرجع الى المستوى السابق، اي المستوى الثاني لنكمل فيه سرد قيم السمات، وبعد السمة **ن1**، تأتي السمة **ن2** وهي ايضا من نمط مهيكل، فنعاود معها ما فعلناه مع **ن1**، لكن هذه المرة علينا استنتاج قيمة السمة **س** وقيمة السمة **ج** من الرسم الذي لا يحتوي مباشرة على قيم **ن2**، وبما ان **ن2** هي النقطة المقابلة للنقطة **ن1**، فان قيمة السمة **س** للنقطة ن 2 تستنتج بجمع قيمة سمة **س** للنقطة **ن1** وهي 100 وقيمة عرض المستطيل الظاهرة في الرسم، اي 200، وبهذا تكون قيمة السمة **س** للنقطة **ن2** هي 300، ونعاود طريقة الاستنتاج هذه مع السمة **ج** للنقطة **ن2**، و نجد ان 250 هي قيمتها.



الشكل 3 : دائرة داخل مربع

النص 9 : التهيئة عند التصريح بمتغيرة مهيكلة، المثال الثاني

عند الانتهاء من سرد قيم **ن2**، نرجع الى المستوى الثاني، ونرى اننا قد ذكرنا كل سمات المستوى الثاني الممثل للمستطيل **م1**، وهنا نغلق هذا المستوى ونرجع الى المستوى الأول من المتغيرة **ر1**، ثم نكمل سرد قيم المستوى الأول، و بعد **م1** تأتي السمة **د1**، وهي سمة مهيكلة توجب فتح مستوى ادنى وهو المستوى الثاني الخاص بالسمة **د1**، واول سمة في **د1** هي السمة **المركز**، وهذه السمة من النمط **النقطة**، فهنا يجب فتح المستوى الثالث لسرد سمات **المركز**، وهذه السمات تستنبط من قيم النقطة (100، 50) ومن والعرض(او الطول) 200، وهي (200، 150)، ثم نرجع الى المستوى الثاني لنسرد باق قيم سمات **د1**، المتمثلة فقط في قيمة السمة المسماة **الشعاع**، وقيمة هذه السمة تستنبط من عرض (او الطول)، فالشعاع في حالنا هو نصف العرض، اي 100، وتكون سمة الشعاع آخر سمة في المستوى الثاني المتمثل في **د1**، و بانتهاء سرد قيم سمات **د1**، نرجع الى المستوى الأول و نجد ان كل السمات قد زودت بقيمها، فنغلق المستوى الأول وتنته عمليه تهيئة المتغيرة **ر1** كما يظهر ذلك في خطأ: لم يُعثر على مصدر المرجع.

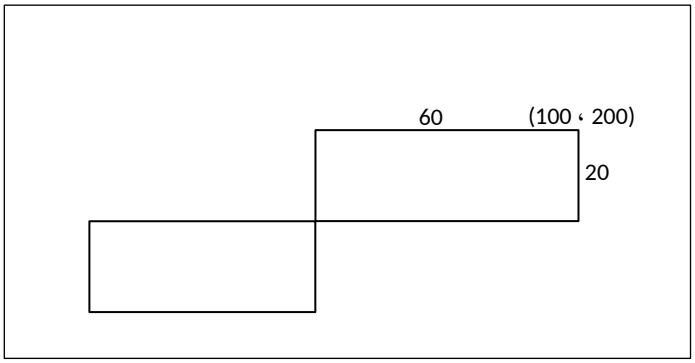
3 -3 الطريقة الثالثة: تهيئة السمات بالقيم الأولية عبر التعليمات في جسم الوظائف او الإجراءات:

يتم هذا الشكل من التهيئة اثناء تنفيذ الإجراءات او الوظائف، ونذكر ان التعليمات تكتب فقط في جسم الإجراءات او الوظائف، وفي الغالب تكون الكتابة التي بها تُهيئ سمة ما هي عبارة تحتوي على الأقل على عملية الشحن ويجب ان تكون السمة مذكورة في يمين عملية الشحن.

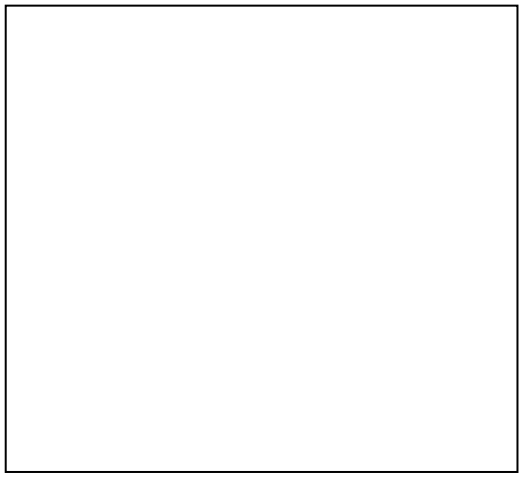
يمكن لعملية تهيئة متغيرة مهيكلة ان تكون معقدة جدا، والتعقيد في عملية التهيئة مرتبط اساسا بأمرين:

- اولا بتعقيد هيكل المتغيرة.
- ثانيا بالطريقة التي تُنظم عمليات شحن السمات بقيمها.

فاذا كان الهيكل سهلا، تكون عملية التهيئة سهلة لا تحتاج الا لعدد قليل من التعليمات، كل واحد منها تختص في شحن سمة من السمات القليلة للمتغيرة المهيكلة، وفي الغالب لا يحتاج العدد القليل من التعليمات طريقة معينة في التنظيم ، اما اذا كان الهيكل شديد التعقيد، فإن التهيئة لا يمكن اتمامها الا بعدد كبير من التعليمات، كل واحدة منها تختص بشحن سمة من السمات الكثيرة للمتغيرة المهيكلة، وكلما كبر عدد التعليمات تصبح مختلف طرق تنظيمها ذات فعالية مختلفة، وهنا علينا اختيار الطريقة الاسلم والأكثر فعالية وهذا طبعا حسب الظرف الذي من اجله يصمم الخوارزم، وباستعمال التعليمات لتهيئة المتغيرات المهيكلة، يمكن ان نستعمل ثلاث صيغ لتنظيم التعليمات، نبينها فيما يلي:



الشكل 3 : مستطيلين تمثلهما المتغيرتين م1 و م2 في النص 14

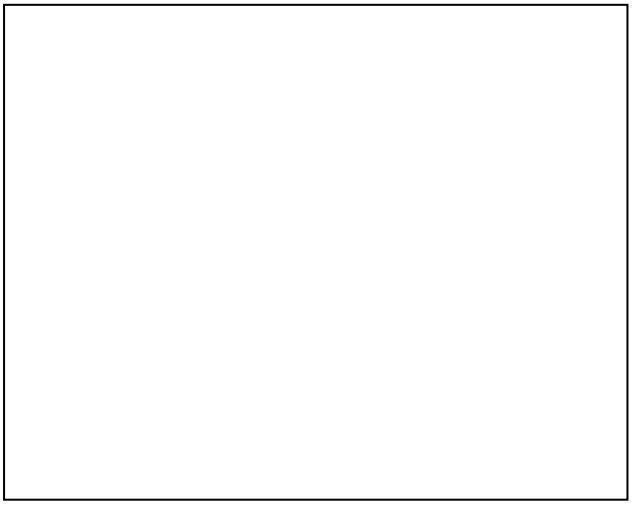


النص 10 : شحن متغيرتين من نمط مستطيل بقيم اولية لتمثيل الشكل 3

3 – 3 – 1 الصيغة الأولى: في هذه الصيغة، يتم شحن سمات المتغيرات المهيكلة بالقيم الأولية عبر تعليمة الشحن (الرمز =) عند الحاجة اليها، وعموما يتم هذا الأمر في الخوارزم (الإجراء او الوظيفة) الذي يحتوي التصريح بالمتغيرة، فتشحن تباعا كل السمات بالقيم المناسبة، وهذا ما يظهر من خلال خطأ: لم يُعثر على مصدر المرجع، ففي السطر 5 نرى التصريح بمتغيرتين **م1** و **م2** من النمط **'مستطيل'** الذي عرف في بداية خطأ: لم يُعثر على مصدر المرجع، ثم من السطر 7 الى السطر 14، تَظهر التعليمات التي تضع القيم المناسبة في سمات المتغيرتين **م1** و **م2** لتمثل هاتين المتغيرتين الرسم الظاهر في خطأ: لم يُعثر على مصدر المرجع .

ضعف الصيغة الأولى :

من خلال المثال السابق يظهر لنا ان عملية شحن عناصر متغيرة مهيكلة بقيم اولية يمكن ان يتطلب كتابة عدد غير مستهان به من التعليمات، فمثلا لشحن السمات الأربع التي تصف اي مستطيل(س، ج، طول، عرض)، يجب كتابة 4 تعليمات، و اذا اردنا ان نمثل رسما فيه اربع مستطيلات، يجب علينا كتابة 16 تعليمة، و يكون العدد اكبر كلما كان كبيرا عدد سمات المتغيرة المهيكلة، اي ان عدد التعليمات التي يجب كتابتها مرتبط ارتباطا شديدا بعدد عناصر المتغيرة المهيكلة.



النص 11 : استعمال اجراء متخصص في شحن المتغيرات من النمط مستطيل

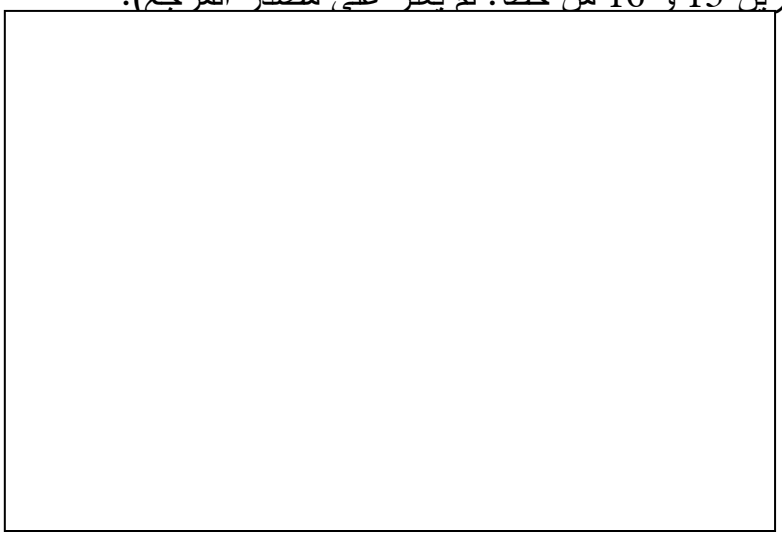
3 – 3 – 2 الصيغة الثانية: تخصيص اجراء لشحن عناصر متغيرة مهيكلة.

في هذه الصيغة الثانية، نكتب مرة واحدة العمليات التي تقوم بشحن سمات متغيرة مهيكلة، ونجعلها كإجراء (خطأ: لم يُعثر على مصدر المرجع)، وهكذا كلما اردنا ان نشحن متغيرة ما، نقوم بتنفيذ الإجراء كما يظهر في السطرين 14 و 15 من خطأ: لم يُعثر على مصدر المرجع، ونوفر للإجراء المعلومات التالية:

- صلة المتغيرة المهيكلة التي نريد تهيئتها (السطر 6 من خطأ: لم يُعثر على مصدر المرجع)، وهنا نستعمل في المنافذ صلة المستطيل لكون المستطيل الذي نريد تغيير ملامحه ليس بمتغيرة محلية لإجراء التهيئة، بل هو متغيرة انشأت عموما في خوارزم آخر، ولا يمكن الوصول اليه الا باستعمال الصلة كما شرحنا ذلك في الجزء الأول من هذا الكتاب.
- القيم التي تستعمل في شحن سمات المتغيرة المهيكلة

3 – 3 – 3 الصيغة الثالثة: تخصيص وظيفة لإنشاء وتهيئة متغيرة مهيكلة:

ترتكز هذه الصيغة على المتغيرات الطارئة، وتتمثل في وظيفة تقوم اولا بإنشاء متغيرة طارئة باستعمال التعليمة **شيد** (في خطأ: لم يُعثر على مصدر المرجع، الوظيفة **انشأ_مستطيل**، السطر 7)، ثم تُشحن كل السمات باستعمال القيم التي وضعت في المداخل (الأسطر من 8 الى 11)، وعند الانتهاء من تهيئة السمات، ترجع الوظيفة صلة المستطيل الذي أُنشأ في الذاكرة الطارئة (السطر 12)، وفي هذه الصيغة تحتوي مداخل الوظيفة على القيم التي نريد وضعها في سمات المتغيرة المهيكلة الطارئة، ونستعمل مخرج الرجوع لتوصيل صلة المتغيرة المهيكلة الطارئة التي انشأت، وكما هو الشأن بالنسبة للصيغة الثانية، نكتب فقط تعليمة واحدة اذا اردنا انتاج متغيرة مهيكلة مهيأة بالقيم التي نذكرها في التعليمة (السطرين 15 و 16 من خطأ: لم يُعثر على مصدر المرجع).



النص 12 : استعمال وظيفة متخصصة في صناعة و تهيئة متغيرات طارئة من النمط مستطيل

مزايا الصيغة الثانية و الثالثة:

نرى في النصين (خطأ: لم يُعثر على مصدر المرجع و خطأ: لم يُعثر على مصدر المرجع) ان عدد التعليمات التي تكتب لشحن عناصر متغيرة مهيكلة اصبح غير مرتبط بعدد المتغيرات المهيكلة التي نريد تهيئها، فنرى مثلا ان الإجراء اساسي في خطأ: لم يُعثر على مصدر المرجع والإجراء اساسي في خطأ: لم يُعثر على مصدر المرجع يحتويان فقط على تعليمتين لتهيئة متغيرتين مهيكلتين.

3 – 4 مزج الطريقة الأولى و الثالثة:

يمكن، حسب الظرف الذي من اجله يصمم الخوارزم، ان نمزج بين الطريقة الأولى والطريقة الثالثة بإحدى صيغها، فمثلا يمكن وصف قيمة اولية على مستوى تعريف النمط، وهكذا اذا انشأت متغيرة مهيكلة عادية او طارئة، تكون القيمة التي وصفت في النمط قد وضعت في السمة المناسبة، وعلى الطريقة الثالثة اتمام التهيئة بإحدى صيغها، وهذا ما نلاحظه في خطأ: لم يُعثر على مصدر المرجع، فنرى في تعريف النمط **تاريخ** ان أي متغيرة من هذا النمط اذا انشأت تنشأ بقيم معروفة لثلاث سمات: **الساعة، الدقيقة و الثانية.**

النص 13 : استعمال القيم المذكورة في الأنماط مع الوظائف و الإجراءات لتهيئة المتغيرات المهيكلة

على مستوى النمط **شخص** نرى ان السمة **ولادة** قد أُريد منها ان تكون ممثلة بمتغيرة طارئة، وكأن المصمم يرى ان المعلومات حول الولادة ليست لها اي اهمية في بعض الأحيان، ولهذا جعل انشاء متغيرة من النمط شخص غير محتوية على السمات التي تخبر عن تاريخ الولادة، وبشكل ادق تكون سمة الولادة، وهي ممثلة بصلة، تحتوي على القيمة **صلة_حرة** التي تبين ان الصلة غير مرتبطة بأي متغيرة.

اضافة الى تهيئة السمات بقيم اولية انطلاقا من تعريف النمط، يحتوي خطأ: لم يُعثر على مصدر. المرجع على الصيغتين من الطريقة الثالثة، فمن السطر 10 الى السطر. 16 عرفت الوظيفة **اصنع_تاريخ** وهدفها إنشاء وتهيئة متغيرة طارئة من النمط تاريخ، وقد استعملت هذه الوظيفة في الإجراء **اساسي**(السطر 26) لإنشاء متغيرة طارئة من النمط **تاريخ** مربوطة بالصلة **ت1،** ومن السطر 17 الى 24، عرف الإجراء **هيئ_شخص،** وهدف هذا الإجراء هو تهيئة متغيرة من النمط **شخص** يوصف في المدخل الأول، ونلاحظ أن كلا الخوارزميين الجزئيين(الوظيفة **اصنع_تاريخ** و الإجراء **هيئ_شخص)** لا يهتمان بتهيئة السمات التي هيأت باستعمال القيم التي وصفت في تعريف أنماطها.

الفصل الرابع

العمليات الأساسية على المتغيرات المهيكلة

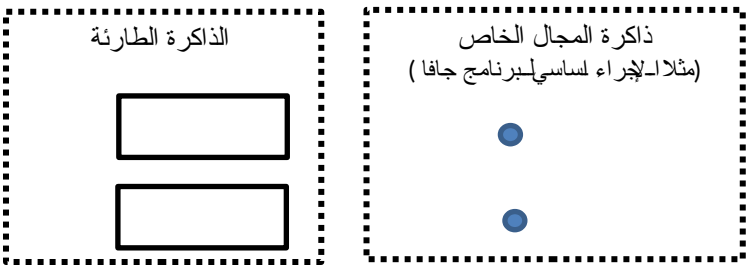
1 – مقدمة: العمليات الأساسية على المتغيرات المهيكلة:

و نعني بالعمليات الأساسية تلك التي لها رمز خاص بها، كعملية الشحن (=)، او عملية المقارنة بالمساواة (==)، او عملية الجمع (+) الخ، وفي هذه الفقرة ندرس جدوى استعمال هذه الرموز مع المتغيرات المهيكلة، وقبل ان نراها في عالم لغات الخوارزميات، نقدم نبذه سريعة عن السياسة المتبعة في بعض لغات البرمجة فيما يخص استعمال العمليات الأساسية مع المتغيرات المهيكلة.

2 – استعمال المتغيرات المهيكلة كعوامل للعمليات الأساسية في لغات البرمجة:

في عدد كبير من لغات البرمجة الحديثة، كلغة "**جافا**"، لا يمكن استعمال العمليات الأساسية مع المتغيرات المهيكلة، ففي "**جافا**" يجب ان تكون المتغيرات الأساسية (او ما سميناه ايضا بالمتغيرات البدائية) هي عوامل العمليات الأساسية ، وفي لغة "**جافا**" كل المتغيرات المهيكلة متغيرات طارئة، تنشأ في الذاكرة الطارئة، وللوصول الى متغيرة مهيكلة في **جافا** نستعمل صلات **جافا**، فعلى سبيل المثال، اذا كانت **عم1** صلة لمتغيرة من النمط **عدد_مركب**، واذا كانت **عم2** صلة لمتغيرة من النمط **عدد_مركب**، واذا افترضنا ان المتغيرتين قد انشأتا، ونفترض ايضا، كما يظهر في خطأ: لم يُعثر على مصدر المرجع، ان المتغيرتين تحتويان على نفس القيم، فان الكتابة **عم1 == عم2** مسموح بها، لكن "**جافا**" يعطيها معنى غير المعنى الظاهر للمبتدئين في عالم "**جافا**"، وتقييمها يكون غير التقييم الظاهر للمبتدئين في البرمجة بلغة "**جافا**"، ففي "**جافا**"، يفضي تقييم العبارة المنطقية **عم1 == عم2** الى نتيجة قيمتها **خطأ**، لأن الكتابة **عم1 == عم2** هي في حقيقة الأمر مقارنة بين محتوى الصلتين وليس مقارنة بين محتوى المتغيرتين الطارئتين، وبما ان المتغيرتين الطارئتين المشار اليهما بالصلتين **عم1** و **عم2** موجودتان في مواقع مختلفة من الذاكرة، فان محتوى الصلتين مختلف، ونُذكر ان محتوى صلة ما هي مجموعة من المعلومات تدل على موقع في الذاكرة للمتغيرة الطارئة المربوطة بالصلة.

في لغة "**جافا**"، الكتابة **عم1 = عم2** لا تعني شحن المتغيرة المشار اليها بالصلة **عم1** بما في المتغيرة الطارئة المشار اليه بالصلة **عم2**، بل شحن الصلة **عم1** بما في الصلة **عم2**، وكنتيجة لمثل هذه العبارة، تصبح المتغيرة التي كانت مربوطة بالصلة **عم2** مربوطة ايضا بالصلة **عم1**، وتصبح دون صلة، المتغيرة الطارئة التي كانت مربوطة بالصلة **عم1**، وتعتبر كل متغيرة طارئة دون صلة متغيرة ضائعة لا يمكن الوصول اليها.



الشكل 1 : تمثيل لمتغيرتين مهيكلتين من النمط عدد_مركب في ذاكرة برنامج بلغة جافا

اذا كانت العمليتين = و == مسموح بهما مع المتغيرات المهيكلة في **"جافا"**، لكون حقيقة امرها عمليات تستهدف الصلات، والصلات نمطها بدائي في **"جافا"**، فان باق العمليات قد حسم في امرها فهي غير مسموح بها في لغة **"جافا"**، فمثلا لا يمكن كتابة **عم1 + عم2** او **عم1*عم2** او **عم1 < عم2** مع ان ظاهر هذه الكتابات صحيح، اي انها عمليات في ظاهرها تستهدف عددين مركبين.

في لغة "**سي**"، اذا كانت المتغيرتين المهيكلتين **عم1** و **عم2** عاديتين، اي غير طارئتين، يمكن كتابة **عم1 = عم2**، والنتيجة تكون شحن كل سمات **عم1** بقيم سمات **عم2**، ومع هذا يُطلب توخي الحذر الشديد في استعمال مثل هذه الكتابة، خاصة عندما تكون السمات ممثلة بمتغيرات طارئة، اما الكتابة **عم1 == عم2** فهي غير مسموح بها، لعدم قدرة لغة **"سي"** على فهم معنى مقارنة محتوى متغيرتين مهيكلتين، وسوف نرى فيما بعد الأسباب التي جعلت هذه اللغات لا تسمع بورود المتغيرات المهيكلة مع معظم العمليات الأساسية.

في لغة "**سي++**" يمكن استعمال اي عمليات اساسية مع اي متغيرة مهيكلة بشرط ان تكتب للعملية الأساسية وظيفة تبين كيفية تعامل العملية الأساسية مع المتغيرة المهيكلة، وتحمل هذه الوظيفة اسما مكونا من الكلمة **عملية**[1] متبوعة برمز العملية، فمثلا اذا اراد كاتب البرنامج ان يجعل من الكتابة **عم1 * عم2** كتابة صحيحة، فعلية ان يكتب وظيفة اسمها **عملية*** تحتوي على التعليمات التي تبين معنى ضرب المتغيرة المهيكلة **عم1** بالمتغيرة المهيكلة **عم2**.

3 – العمليات الأساسية على المتغيرات المهيكلة في لغات الخوارزميات:

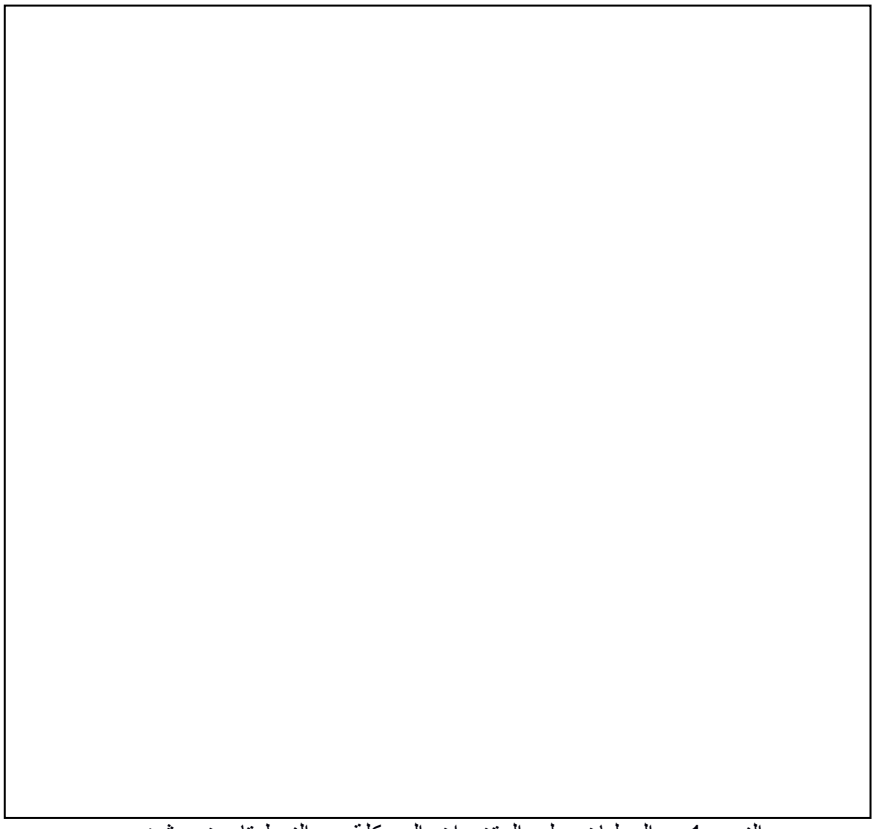
عند انجاز خوارزم ما يجد الكاتب بعض الحرية في كتابة التعليمات والعبارات، وهذه الحرية لا يجدها عند كتابة البرامج اين يجب عليه ان يحترم بدقة متناهية ما تفرضه عليه لغة البرمجة، فمثلا يمكن له ان يقارن بين متغيرتين مهيكلتين باستعمال رموز عمليات المقارنة (== ، < ، >)، كما يمكن له ان يكتب تعليمة شحن متغيرة مهيكلة بما في متغيرة مهيكلة اخرى باستعمال الرمز =، وباختصار، وبما ان نص الخوارزم موجه للإنسان، وهذا الأخير هو من سيترجم الخوارزم الى لغة برمجة، فيمكن لكاتب الخوارزم ان يستعمل اي رمز شاء في كتابة العبارات، ومع وجود مثل هذه الحرية، يجب في آخر المطاف ان تكون اي كتابة تَرِدُ في الخوارزم كتابة دقيقة وذات دلالة ومعنى واحد، فلا يسمح بوجود كتابات عليها لبس في الفهم، فإن وجد ان كتابة ما يمكن فهمها على اكثر من شكل، فمعنى هذا ان الخوارزم قد فقد خاصية هامة وحيوية من خصائصه، وهي القراءة والفهم الواحد لكل ما فيه من تعليمات.

كما اوردناه في بداية الجزء الأول، فان دورة حياة الخوارزم تقضي بان يحول الخوارزم الى لغة برمجة لينفذ فيما بعد من قبل الحاسوب، ولذا وجب ان تكون الترجمة دقيقة لا تحتمل وجود لبس

1 **operator** keyword (C++ Operator Overloading)

في نص الخوارزم، فإن وجد لبس في نص الخوارزم تصبح عملية التحويل من اللغة الخوارزمية الى لغة البرمجة غير ممكنة.

الحرية التي يجدها واضع الخوارزم عند الكتابة باللغة الخوارزمية يمكن ان تؤدي به ان لم يكن حذرا ودقيقا الى سرد تعليمات تبدو في الظاهر كاملة، وفي حقيقتها ناقصة لكونها ترتكز على مفاهيم اصبحت بديهية عند واضع الخوارزم، فتبق هذه المفاهيم في ذهن الكاتب ولا تظهر في الكتابة، وهكذا اذا كان واضع الخوارزم هو من يحول الخوارزم الى برنامج، تكون عملية التحويل دقيقة وصحيحة، اما اذا أُعطي الخوارزم الى انسان آخر وطُلب منه التحويل، فان هذا الأخير لن يجد في النص المفاهيم التي كانت في ذهن كاتب الخوارزم، وهنا اما ان يجد هذا الانسان نفسه في حيرة، فيحاول ان يتصل بكاتب الخوارزم ليطلب منه شرحا دقيقا لما كتب، او يجد نفسه دون شعور قد انجز التحويل مرتكزا على ما يفهمه هو من الكتابة، فيمكن ان يصبح التحويل على شكل آخر غير الشكل الذي يريده كاتب الخوارزم، ولشرح عملي اكثر ندرس المثالين التاليين.



النص 1 : العمليات على المتغيرات المهيكلة من النمط تاريخ و شخص

3 – 1 مثال توضيحي اول:

في السطر 13 من خطأ: لم يُعثر على مصدر المرجع، كتبت العبارة **ش.الولادة = ت** وفيها استعملت عملية الشحن بين متغيرتين من النوع **تاريخ** و هما: السمة **الولادة** والمدخل **ت**، وهذه الكتابة لا لبس فيها، فالكاتب يريد شحن قيم سمات المتغيرة **ت** في المتغيرة **الولادة**، وحسب ما توفره لغة البرمجة من قدرات يقوم من يريد تحويل الخوارزم الى برنامج بإيجاد التعليمة او التعليمات التي تحقق بالغرض.

في السطر 24، استعملت عملية المقارنة < بين متغيرتين من النمط شخص، واذا قرأنا العبارة فمفادها ما يلي: هل ما في المتغيرة **ش1** اكبر مما في المغيرة **ش2**، وبصيغة ادق: هل الشخص الممثل بالمتغيرة **ش1** اكبر من الشخص الممثل بالمتغيرة **ش2**، وهذه الكتابة في حقيقتها مرتبطة بمفهوم غير معلن بقي في ذهن كاتب الخوارزم، ولنفرض ان كاتب الخوارزم قد وضع في باله ان مثل هذه المقارنة هي في حقيقة امرها مقارنة بين اعمار الشخصين، وهكذا، إن كان كاتب الخوارزم هو من يحول الخوارزم الى برنامج، فسوف يُحول الكتابة قبل ترجمتها الى الشكل **ش1.الولادة < ش2.الولادة**، وبما ان مفهوم هذه المقارنة عنده هي مقارنة بين الأعمار فانه يترجم هذه الكتابة الأخيرة باستعمال تعليمات لغة البرمجة التي تحفظ هذا الفهم في البرنامج.

اما ان كان الذي يترجم الخوارزم الى برنامج شخص آخر غير الشخص الذي كتب الخوارزم، فانه يصدم ويحتار فعند رؤيته الكتابة **اذاكان(ش1 < ش2)**، يتسائل: هل الكتابة معناها:

- هل عمر الشخص الممثل بالمتغيرة ش1 اكبر من عمر الشخص الممثل بالمتغيرة ش2
- هل طول الشخص الممثل بالمتغيرة ش1 اكبر من طول الشخص الممثل بالمتغيرة ش2

ولنفرض ان المترجم قد استطاع التواصل مع كاتب الخوارزم، فطلب منه توضيحا كاملا ودقيقا للكتابة **ش1 < ش2**، ولنفرض ان اجابة كاتب الخوارزم كانت: معنى ش1 < ش2 هو **ش1.الولادة < ش2.الولادة**، فهل هذه الكتابة تكف لرفع اللبس والحيرة؟ كلا، فهنا ايضا يمكن للمترجم ان يحتار. في المعنى الدقيق للكتابة **ش1.الولادة < ش2.الولادة** التي تقارن بين تاريخين:

- هل عمر **ش1** اكبر من عمر **ش2**
- هل تاريخ **ش1** اكبر من تاريخ **ش2** اي تاريخ ش1 احدث من تاريخ ش2

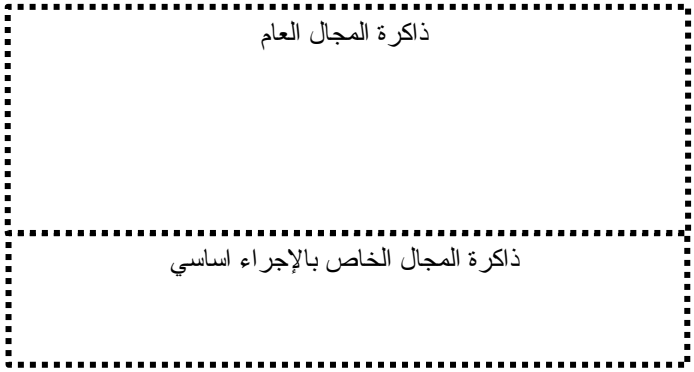
3 – 2 مثال توضيحي ثاني:

يحتوي خطأ: لم يُعثر على مصدر المرجع على التعريف بنمطين: النمط **عدد_مركب** والنمط **عدد_كسري**، ونفترض في النمط **عدد_مركب** ان قيم اجزاء الأعداد المركبة التي تهمنا هي قيم طبيعية، ويمثل خطأ: لم يُعثر على مصدر المرجع حالة ذاكرة خوارزم خطأ: لم يُعثر على مصدر

المرجع اثناء تنفيذ الإجراء **اساسي**. اذا تأملنا في خطأ: لم يُعثر على مصدر المرجع، نرى ان متغيرات النمطين **عدد_مركب** و **عدد_كسري** لهما نفس التمثيل في الذاكرة (المنطقة الملونة)، ولنفرض ان نُمتحن فتُعطى لنا حالة الذاكرة في خطأ: لم يُعثر على مصدر المرجع (المنطقة الملونة) ويطلب منا ان نجد المتغيرات التي هي من النمط **عدد_مركب**، فاننا لن نقدر على تبيان ما هي المتغيرة التي تنتمي الى النمط **عدد_مركب** او النمط **عدد_كسري**. المعلومة التي تفيد هذا المعنى، اي الانتماء الى النمط **عدد_مركب** او النمط **عدد_كسري** موجودة في النص، وهي اسم النمط.

تذكير: اسماء الأنماط المهيكلة والسمات والمتغيرات والإجراءات والوظائف هي اسماء من وحي مصمم الخوارزم، وكان بإمكان مصمم الخوارزم اعطاء اسماء أخرى للأنماط والسمات المستعملة في هذا المثال، فمثلا كان من الممكن للمصمم ان يضع الاسم **عد_مر** بدلا من **عدد_مركب** و **س** و **ج** بدلا من **ح** و **ت**، و **ع_كس** بدلا من **عدد_كسري**، الخ.. **واختيار واضع الخوارزم للأسماء له هدف هام و هو تسهيل قراءة و فهم نص الخوارزم.**

النص 2 : العمليات الأساسية على المتغيرات المهيكلة من النمطين عدد_مركب و عدد_كسري



الشكل 2 : حالة ذاكرة المجال العام لخوارزم النص 2 والمجال الخاص للإجراء اساسي

يحتوي خوارزم خطأ: لم يُعثر على مصدر المرجع في السطرين 15 و 16 على عبارتين استعملت فيهما متغيرات مهيكلة من النمطين **عدد_مركب** او النمط **عدد_كسري** مع عملية الجمع (+) و عملية الشحن (=)، فإذا كانت عملية الشحن تبدو واضحة لا لبس فيها، فهل يا ترى مثل هذا الوضوح وعدم وجود اللبس نجده مع عملية الجمع؟ للإجابة على هذا التساؤل نحاول ان نرى فيما يلي ما يمكن ان يحدث في سياق تقييم هاتين العبارتين في الحالتين التاليتين:

- الحالة الأولى عند الشروع في ترجمة الخوارزم الى لغة برمجة لا تحتمل وجود المتغيرات المهيكلة مع عملية الجمع.
- الحالة الثانية نرى ما يحدث لو افترضنا ان عملية الجمع لمتغيرتين مهيكلتين مسموح بها في لغة من لغات البرمجة.

الحالة الأولى: تحويل الخوارزم الى لغة برمجة لا تحتمل وجود المتغيرات المهيكلة مع عملية الجمع:

في عالم الخوارزميات، يمكن ان يقوم واضع الخوارزم او غيره، بعملية تحويل الخوارزم الى لغة برمجة. فإذا قام واضع الخوارزم بتحويل خطأ: لم يُعثر على مصدر المرجع، يكون التحويل مبني على اسس صحيحة حتى ولو كان اسم النمط غير حامل للهدف الذي من اجله وضع، اي ان واضع الخوارزم يمكن ان يستعمل اسماء تكون بدون معنى او بعيدة عن المعنى المحمول من قبل الأسماء **عدد_مركب** و**عدد_كسري**، فالعلاقة بين النمط مهما كان اسمه والهدف الذي وضع من اجله موجودة في ذهن واضع الخوارزم، وهذه العلاقة يستعملها واضع الخوارزم في عملية التحويل.

اما ان كان من يحول الخوارزم شخصا آخر، فهنا يمكن ان يكون هذا الشخص من احد الصنفين التاليين:

- صنف اول يفهم العمليات الحسابية على الأعداد المركبة والأعداد الكسرية
- صنف ثان لم يتعلم العمليات الحسابية الموجهة للأعداد المركبة و الأعداد الكسرية

التحويل من قبل شخص من الصنف الأول

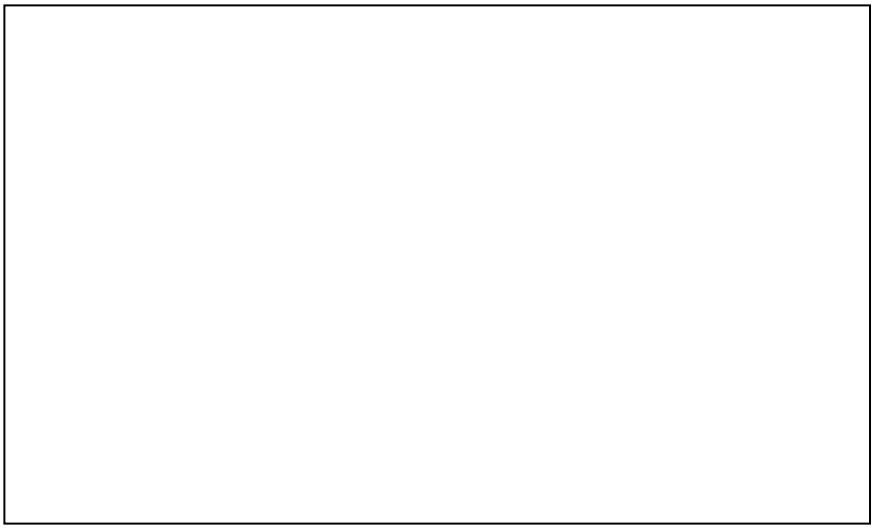
عندما يقوم شخص من الصنف الأول بعملية التحويل، يمكن ان يواجه احدى الحالتين التاليتين:

- في الحالة الأولى يكون واضع الخوارزم قد اختار للأنماط أسماء ناطقة، اي ان قارئ الخوارزم، بمجرد رؤية اسم النمط، يفهم تماما الهدف الذي من اجله وضع النمط، وطبعا يفهم كل ما يتعلق بالنمط، وهذا ما حدث في خطأ: لم يُعثر على مصدر المرجع، حيث استعملت الأسماء **عدد_مركب** و **عدد_كسري** لتمثيل الأعداد المركبة والأعداد الكسرية، ففي هذه

الحالة يستعمل الشخص ما بذهنه من علم وطرق لإنجاز التحويل، ويكون التحويل في هذه الحالة مبني على الأسس الصحيحة المخزنة في ذهن الشخص.

- في الحالة الثانية، يكون واضع الخوارزم قد اختار اسماء جامدة، لا توحي رؤيتها بأي معنى، فمثلا (خطأ: لم يُعثر على مصدر المرجع) يجد القارئ الاسم **ع_مر** بدلا من **عدد_مركب** و **ع_كس** بدلا من **عدد_كسري،** ونلاحظ هنا ان هذه الطريقة في اختيار الأسماء غير مرغوب فيها البتة في عالم الخوارزميات والبرامج، لكنها موجودة وشائعة، خاصة عند المبتدئين، ففي مثل هذه الحالة يجد الشخص نفسه امام لغز محير عندما يريد تحويل العبارات الواردة في السطرين 15 و 16 من خطأ: لم يُعثر على مصدر المرجع الى لغة برمجة، وهذا لعدم وجود اي ربط بين ما في ذهن الشخص من علم وبين اسم النمط وايضا لعدم ورود العلم الضروري لحل اللغز في نص الخوارزم، وهذا العلم الناقص في الخوارزم يتلخص في الأمرين التالين:
  - عدم التعريف في النص بكيفية انجاز عملية جمع بين متغيرتين من النمط **ع_مر** (وهنا لا نقول **عملية الجمع بين عددين مركبين** لكون الشخص لا يعلم هل النمط **ع_مر** مخصص لتمثيل الأعداد المركبة)،
  - عدم التعريف في النص بكيفية انجاز عملية جمع بين متغيرتين من النمط **ع_كس،**

وهذه الحالة الثانية هي في حقيقة الأمر، حالة الحاسوب عندما ينفذ برنامجا ما، فما يفهمه الحاسوب هو فقط العمليات الأساسية على المتغيرات الأساسية، فاذا اردناه ان ينجز عملية ما على متغيرات مهيكلة، فعلينا ان نخبره عن الكيفية باستعمال ما يفهم من عمليات أساسية على المتغيرات الأساسية، والإخبار عن الكيفية، يتم في العموم بإنجاز وظيفة، ويتم وصف الوظيفة في العبارات اما بذكر اسمها وذكر عواملها في المداخل او بذكر رمز العملية وربط الرمز بالوظيفة عند تعريف الوظيفة كما هو الشأن في لغة "**سي++**".



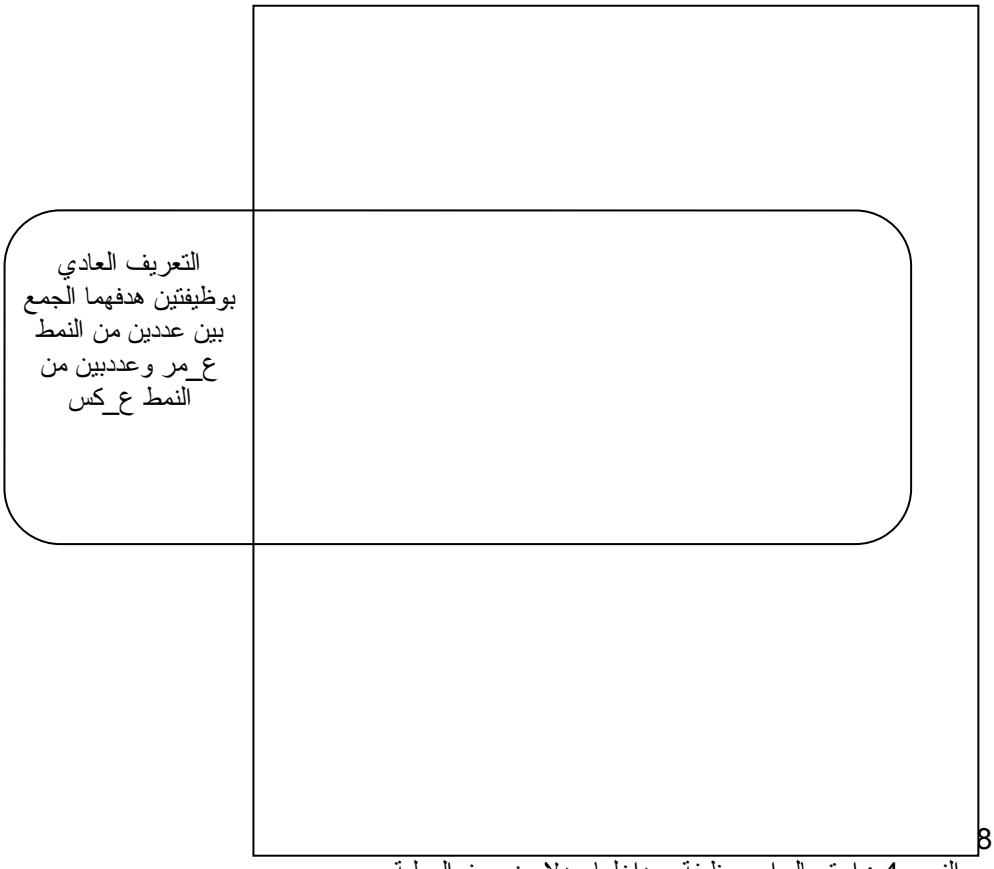
النص 3 : استعمال اسماء جامدة في تعريف الأنماط

التحويل من قبل شخص من الصنف الثاني:

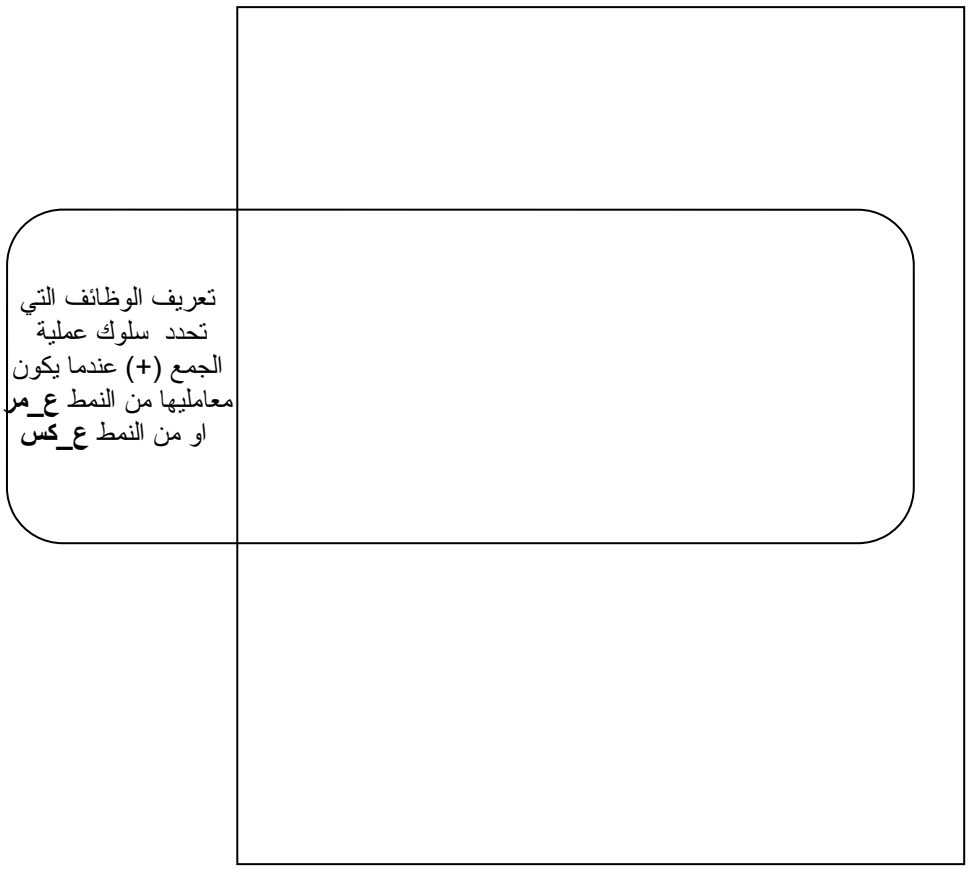
هذا الصنف من الأشخاص، طبعا يعرف معرفة جيدة لغة من لغات، لكنه مع الأسف ضعيف في الحساب والرياضيات، فليس له علم بالأعداد المركبة ولا الأعداد الكسرية، كما هو حال الحاسوب، ففي هذه الحالة، وجود اسما ناطقا او جامدا لن يكون له اي مفعول، والترجمة سوف لن تتم لعدم وجود تفسير في النص لمعنى عملية الجمع بين الأعداد المركبة ولمعنى عملية الجمع بين الأعداد الكسرية، فهنا تتوقف عملية التحويل فورا. ويطلب الشخص مزيدا من المعلومات عن كيفية انجاز عملية الجمع في كلتا الحالتين، فإذا كان هذا هو حال مع الإنسان، فالحال مع الآلة يكون اصعب، وكما اشرنا اليه سابقا، هذه الحالة تشبه ايضا موقف الحاسوب اذا طلب منه تنفيذ عملية اساسية على متغيرات مهيكلة.

3- 3 خلاصة القول عن المثالين:

عند كتابة اي خوارزم يراد منه الوصول الى الغير، اما للقراءة او للترجمة الى لغة من لغات البرمجة، يجب ان يحتوي الخوارزم على كل المعلومات الضرورية التي تجعل من فهمه فهما دقيقا لا يحتمل اي لبس، وما رأيناه من خلال المثالين السابقين، فإن كتابة عبارات تحتوي على متغيرات مهيكلة ورموز العمليات الأساسية، دون ذكر معنى رمز العملية في اطار المتغيرات المهيكلة، يعتبر نقصا في الخوارزم، يؤدي الى وجود لبس في تعليمات الخوارزم، ما يجعل هذا الأخير يفقد صفة حيوية من صفاته الا وهي وحدة القراءة والفهم، فلا يمكن ان يطلق عليه خوارزم.



النص 4 : استعمال اسم وظيفة ومداخلها بدلا من رمز العملية



النص 5 : استعمال رمز العملية (طريقة "سئ++") بدلا من ا سم الوظيفة المرتبطة بالرمز

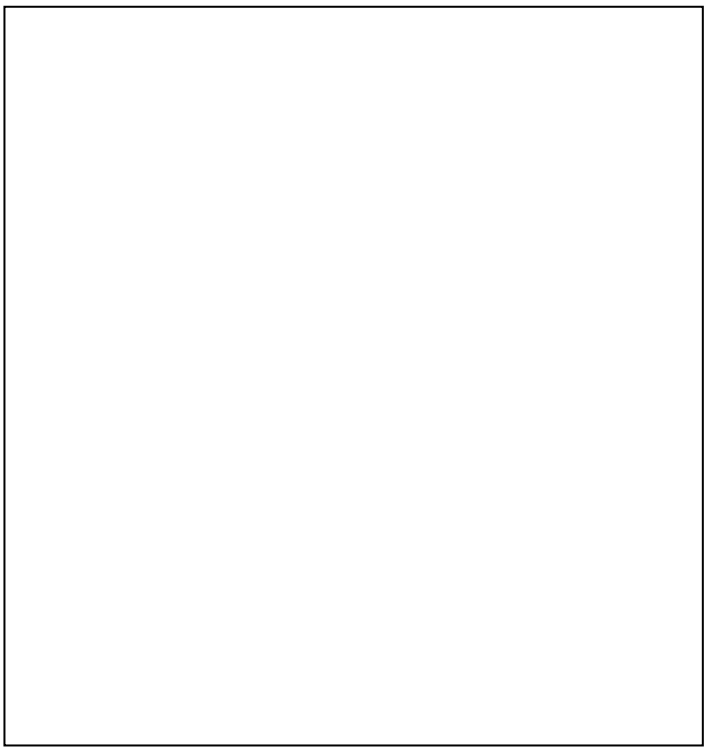
و من المثالين السابقين، توصلنا ايضا الى نتيجة تفيد بأن النقص يُصحح بإضافة كيفية انجاز عملية ما عندما تستهدف هذه العمليات المتغيرات المهيكلة، ورأينا ان الكيفية تكون على شكل وظيفة، والوظيفة يمكن ان تكون على احدى الطريقتين:

- تعريف الوظيفة بشكل عادي (السطرين 8 و 14من خطأ: لم يُعثر على مصدر المرجع) واستعمالها باسمها في العبارات، كما هو الشأن في السطرين 28 و 29 من خطأ: لم يُعثر على مصدر المرجع .
- تعريف الوظيفة والرمز الذي يمثلها كما هو الشأن في لغة "**سي++**" (السطرين 8 و 14من خطأ: لم يُعثر على مصدر المرجع)، واستعمال الرمز بدل اسم الوظيفة في العبارات (السطرين 28 و 29 من خطأ: لم يُعثر على مصدر المرجع).

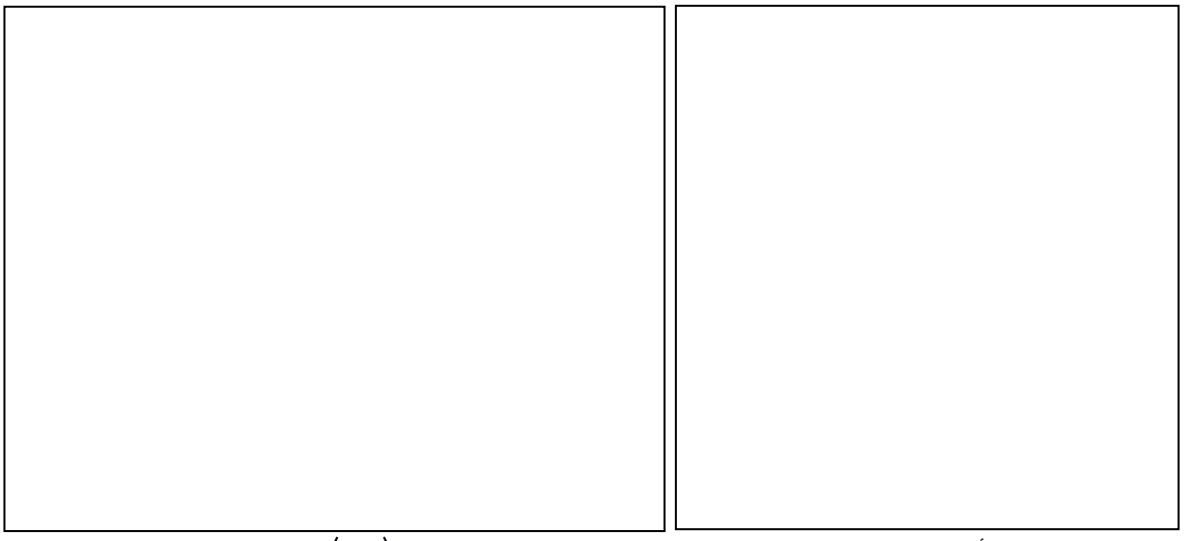
3 – 4 انعكاسات استعمال اجراء بدل وظيفة لتمثيل عملية اساسية

كان بالإمكان في الأمثلة السابقة استعمال **اجراء** بدل من **وظيفة** لتبيان كيفية انجاز عملية على متغيرتين مهيكلتين، كما يظهر في السطرين 24 و 25 من خطأ: لم يُعثر على مصدر المرجع، لكن

استعمال **اجراء** بدل **وظيفة** يحد من استعمال العملية في العبارات، فبينما يمكن ذكر الوظيفة اكثر من مرة داخل عبارة واحدة، فلا يمكن ذكر الإجراء في عبارة، فالإجراء يذكر وحيدا في تعليمة طلب تنفيذه، فمثلا، اذا استعملنا وظيفة يمكن كتابة عبارة فيها اكثر من عملية شحن او جمع كما يبينه السطر 16 من خطأ: لم يُعثر على مصدر المرجع (أ) و (ب)، اما اذا استعملنا اجراء فسوف نضطر الى كتابة اكثر من سطر كما يبينه خطأ: لم يُعثر على مصدر المرجع في الأسطر من 15 الى 18.

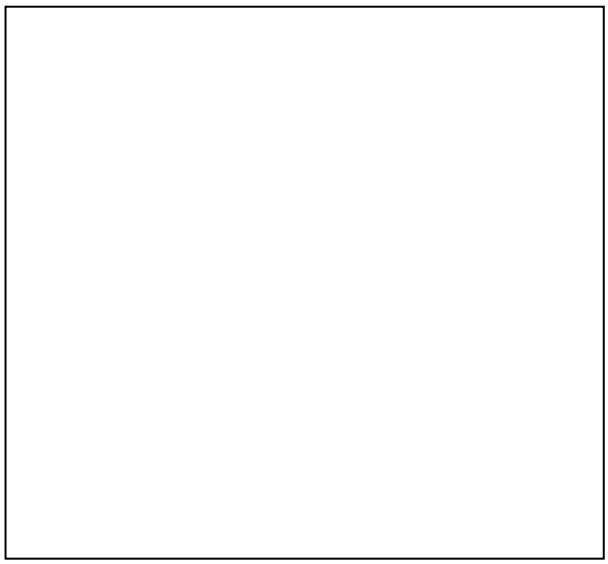


النص 6 : استعمال اجراء بدل وظيفة لإنجاز عملية الجمع



(أ) (ب)

النص 7 : عبارة تحتوي على اكثر من عملية جمع لمتغيرات مهيكلة: (أ):استعمال رمز العملية (ب):استعمال اسم الوظيفة



النص 8 : استعمال اجراء بدل وظيفة لإنجاز عملية الجمع

4 – عملية شحن متغيرة مهيكلة

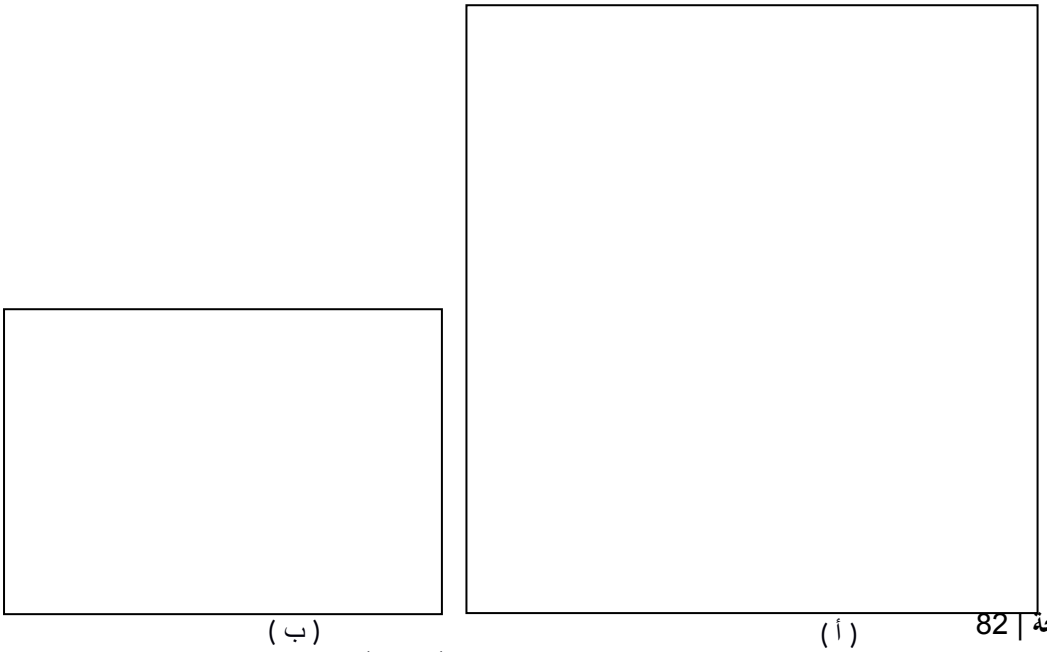
رأينا فيما سبق ان عملية شحن متغيرة مهيكلة بما في متغيرة مهيكلة اخرى كتابة سليمة لا تحتمل في اي حال الا على مفهوم واحد وهو شحن سمات المتغيرة المذكورة في يمين الرمز = بسمات المتغيرة المذكورة في يسار الرمز=، وهذه الكتابة مسموح بها في بعض لغات البرمجة كلغة "سي" و "سي++" ولغات الخوارزميات.

عندما يقوم مترجم لغة البرمجة بتحويل نص البرنامج الى نص بلغة الحاسوب (لغة الآلة)، يعتمد في الغالب على احد السلوكين التاليتين المؤديتين الى نفس النتيجة:

- في السلوك الأولى يقوم المترجم بما يلي:
  - انطلاقا من معرفة النمط المهيكل، يحسب المترجم حجم الذاكرة التي يجب ان توفر لكل متغيرة وردت في عملية الشحن، فمثلا، اذا كان النمط **طبيعي** يستهلك من الذاكرة 4 وحدات، واذا كانت المتغيرتين الواردتين في عملية الشحن من النمط **عدد_مركب**، يتوصل المترجم الى ان حجم كل متغيرة من النمط **عدد_مركب** هو 8 وحدات من الذاكرة، لكون النمط **عدد_مركب** مكون من سيمتين من النمط **طبيعي.**
  - انطلاقا من حجم الذاكرة التي تحتوي على المتغيرتين الواردتين في عملية الشحن، يحول المترجم عملية الشحن الواردة في نص البرنامج الى عدد من التعليمات هدفها

واحد هو نقل محتوى وحدات من الذاكرة من مكان اول الى مكان ثان، ولا علم لمثل هذه التعليمات لمعنى القيم المخزنة في وحدات الذاكرة التي يتم نقلها، و فيما يخصنا، المكان الأول هو المكان المخصص للمتغيرة الأولى الموجودة في يسار عملية الشحن والمكان الثاني هو المكان المخصص للمتغيرة الثانية الموجودة في يمين عملية الشحن، وعدد التعليمات التي تستعمل لنقل كل الذاكرة المخصصة للمتغيرة الأولى الى ذاكرة المتغيرة الثانية، مرتبط اساسا بقدرة تعليمات نقل محتوى الذاكرة، فمثلا، اذا كانت تعليمة نقل المحتوى لا يمكنها الا نقل محتوى وحدة واحدة فقط، واذا كان حجم المتغيرة المستهدفة 8 وحدات، فان المترجم يحول تعليمة شحن المتغيرات المهيكلة، الواردة في نص البرنامج المكتوب بلغة برمجة، الى 8 تعليمات نقل المحتوى، الأولى تنقل الوحدة الأولى، والثانية تنقل الوحدة الثانية والأخيرة تنقل الوحدة الثامنة، اما اذا كانت تعليمة نقل المحتوى قادرة على نقل 4 وحدات مرة واحدة، فان المترجم يحول تعليمة شحن المتغيرات المهيكلة الى تعليمتين لنقل محتوى الذاكرة، الأولى تنقل الوحدات الأربع الأولى و الثانية تنقل ما تبقى.

- السلوك الثاني: يقوم المترجم بنقل السمات واحدة تلو الأخرى من المتغيرة الموجودة في يسار تعليمة الشحن الى سمات المتغيرة الموجودة في يمين تعليمة الشحن، فاذا كانت تعليمة الشحن تستهدف متغيرتين من النمط **عدد_مركب**، فان الكتابة التي ينتجها المترجم تكون مكونة من تعليمتين، الأولى تستهدف السمة **ح**، و هي من نمط بدائي، الثانية تستهدف السمة **ت**، وهي ايضا من نمط بدئي، واذا كانت تعليمة الشحن تستهدف متغيرتين من النمط شخص (خطأ: لم يُعثر على مصدر المرجع،أ ، السطر. 19) فان عدد التعليمات التي يفضي اليها المترجم هي 10 (خطأ: لم يُعثر على مصدر المرجع ،ب).



(أ) (ب)

النص 9 :الطريقة الثانية في ترجمة تعليمة شحن متغيرة مهيكلة بأخري (أ) نص خوارزم شبيه بلغة برمجة (ب) نتيجة ترجمة تعليمة ش2 = ش1

ملاحظة هامة: الطريقتين السابقتين تقوم بمهامها مهما كان نمط المتغيرات المهيكلة التي تستهدفها عملية الشحن، فعمل تعليمة نقل محتوى وحدات الذاكرة تقوم بنقل أعمى لما في الذاكرة و تعليمة شحن سمة بسمة أخرى هي في حقيقتها تعليمة نقل متغيرة من نمط بدائي الى متغيرة اخرى من نفس النمط، ولا علم لها اثناء عملها هل المتغيرة هي سمة لمتغيرة مهيكلة أو لا، فالطريقتين منفصلتين تماما عما تمثله متغيرة مهيكلة وعما تمثله كل سمة.

4 – 1 عملية شحن المتغيرات المهيكلة المحتوية على سمات طارئة.

في الأمثلة السابقة استعملنا متغيرات مهيكلة خالية من السمات الطارئة الممثلة بالصلات، ففي مثل هذه الحالة، تكون تعليمة شحن متغيرة مهيكلة بأخرى كتابة صحيحة وفعالة، اما ان كانت المتغيرة تحتوي على سمات طارئة، وجب الحذر الشديد عند استعمال تعليمة شحن متغيرة مهيكلة بأخرى، مخافة الحصول على نتائج غير مرغوب فيها او نتائج يمكن ان تكون سبب في احداث خلل اثر تنفيذ الخوارزم، و يرجع هذا الأمر الى ان السمة الطارئة ممثلة بصلة، فإن كانت قيمة الصلة هي القيمة **صلة_حرة**، تكون السمة غائبة عن المتغيرة المهيكلة، اما ان كانت قيمتها مختلفة عن القيمة **صلة_حرة**، فمعنى هذا ان السمة موجودة و تمثلها متغيرة طارئة.

عند نقل محتوى المتغيرة المهيكلة اليسرى (الموجودة في يسار رمز عملية الشحن =) الى المتغيرة المهيكلة اليمنى، تنقل فقط قيمة صلة السمة الطارئة، ولا ينقل ما في المتغيرة الطارئة المرتبطة بالصلة، لكون المتغيرة الطارئة موجودة في ذاكرة أخرى غير ذاكرة المتغيرة المهيكلة، واذا افترضنا ان سلوك عملية الشحن سلوكا قويا، وليتمكن من نقل ما في المتغيرة الطارئة، يجب عليه اولا ان يتحقق على مستوى المتغيرة اليمنى من وجود متغيرة طارئة مربوطة بالصلة الممثلة للسمة، فان لم تكن موجودة فعليه انشاءها وربطها بالصلة، ثم يقوم بعملية النقل، لكن من اجل ان تتسم عملية الشحن بهذا السلوك عليها ان تتحقق من قيمة الصلة، وكما اشرنا اليه سابقا فان عملية شحن متغيرة بأخرى ليس من صلاحيتها التحقق من القيم التي تأخذها من جانبها الأيسر لتنسخ في جانبها الأيمن.

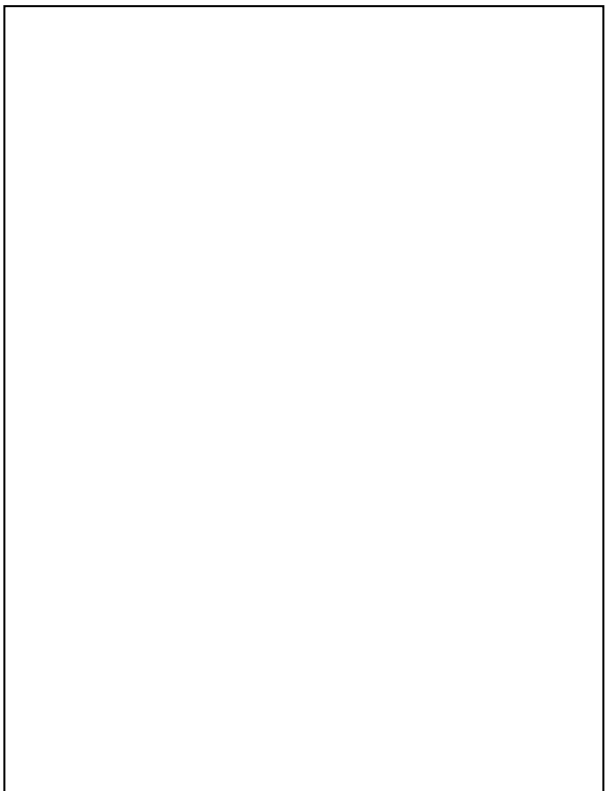
4–2 امثلة توضيحية:

للتبيان الجيد لما يحدث عند شحن متغيرة مهيكلة محتوية على سمات طارئة الى اخرى، ندرس الأمثلة التالية، التي ترتكز على ما في خطأ: لم يُعثر على مصدر المرجع وفي هذه الأمثلة **نفترض ما يلي:**

- **قيمة الصلة هي عدد طبيعي يمثل عنوان متغيرة مهيكلة في الذاكرة الطارئة**، وبالتدقيق عنوان الوحدة الأولى للمنطقة المخصصة للمتغيرة المهيكلة.
- قيمة عنوان اول وحدة في الذاكرة الطارئة هي 1000، فالقيمة 1000 هي عنوان اول وحدة في الذاكرة الطارئة، والقيمة 1009 هي عنوان الوحدة العاشرة في الذاكرة الطارئة، والقيمة 1099 هي عنوان الوحدة مئة من الذاكرة الطارئة، والقيمة 999 هي عنوان الوحدة الف من الذاكرة الطارئة، اما القيمة 0 فهي تمثل صلة حرة، وفي اللغة الشرمزية لا يمكن اجراء عمليات حسابية على الصلات حتى ولو كانت في حقيقتها قيم طبيعية ، فمثلا لا يمكننا في اللغة الشرمزية كتابة عملية جمع صلتين بالرغم من ان ما تحتويه كل صلة هي قيم طبيعية، وهذا الأمر موجود ايضا في لغة **'جافا'** التي لا تمكن من اجراء عمليات حسابية على الصلات، وهذا عكس لغة **"سي"** اين محتوى الصلة هو قيمة طبيعية يمكن ان تنفذ عليها مختلف العمليات الحسابية كالجمع، والطرح الخ... ونذكر ان نوعية قيمة الصِلات أُدخلت **فقط لتسهيل فهم الأمثلة.**

4 – 2 – 1 المثال الأول:

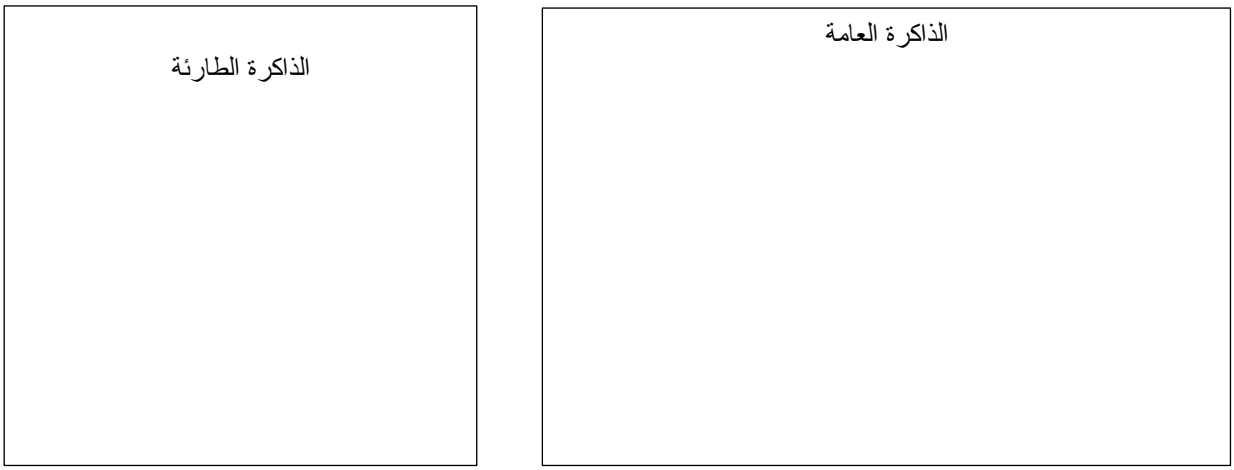
يرتكز هذا المثال على ما في خطأ: لم يُعثر على مصدر المرجع، فمن خلال خطأ: لم يُعثر على مصدر المرجع، نلاحظ ان النمط **رسم** (السطر 7) يحتوي على سمتين طارئتين (أي صلتين)، تمثل الأولى مستطيل في الرسم والثانية دائرة في الرسم.



الجزء الثاني: الفصل الرابع: العمليات الأساسية على المتغيرات المهيكلة

في السطرين 22 و 23 من خطأ: لم يُعثر على مصدر المرجع نلاحظ التصريح بمتغيرتين عامتين، **ر1** و **ر2** من النمط **رسم**، وتم التصريح بهما بقيم اولية، والقيم الأولية للسمتين **م** و **د** هي القيمة **صلة_حرة**، ومعنى هذا ان السمتين غائبتين لغياب المتغيرتين الطارئتين، وهذا ما يبينه خطأ: لم يُعثر على مصدر المرجع، اذ نرى من خلاله حالة ذاكرة الخوارزم قبل البدئ في تنفيذ اول تعليمة من الإجراء اساسي (السطر. 24 من خطأ: لم يُعثر على مصدر المرجع).

التعليمة الأولى التي تنفذ (السطر 25) تزود الرسم بمستطيل وهذا عبر انشاء السمة الطارئة **'م'**، اي انشاء متغيرة طارئة من النمط مستطيل وربطها بالصلة **'م'**، وهكذا يصبح للرسم مستطيلا واحدا، وهذه المعلومة الأخيرة تسجلها تعليمة السطر 26 في السمة **'عدد_مس'**، وبعد تنفيذ تعليمات السطرين 25 و26 تصبح الذاكرة على خطأ: لم يُعثر على مصدر المرجع.



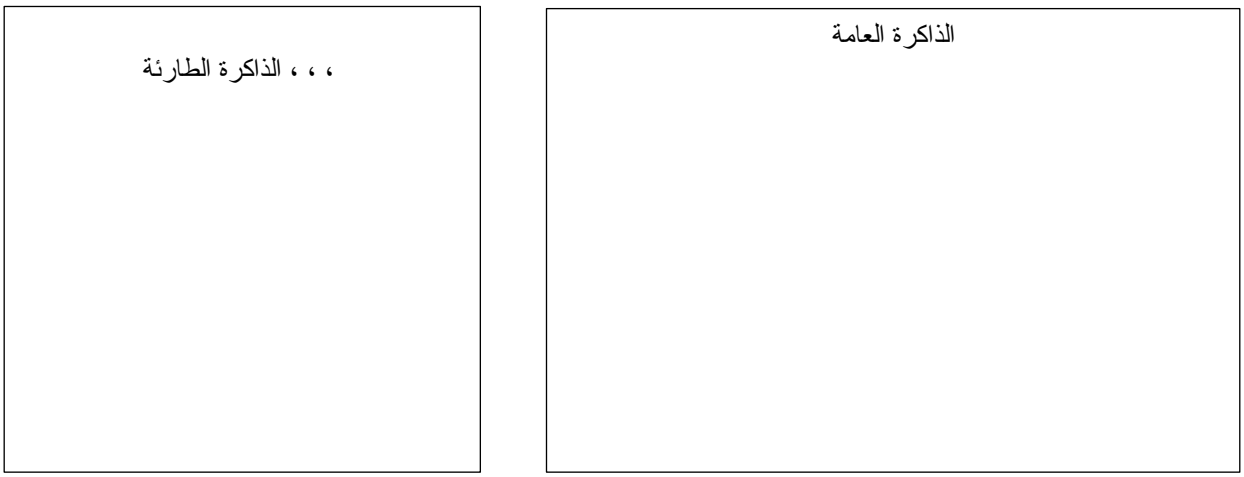
الشكل 3 : حالة الذاكرة العامة و الطارئة عند انطلاق تنفيذ الخوارزم (النص 10 و النص 11) و قبيل تنفيذ اول تعليمة من الإجراء اساسي



الشكل 4 : حالة الذاكرة بعد تنفيذ تعليمات السطرين 24 و 52 من النص 10

تنفيذ التعليمة **' ر2 = ر 1'** (السطر 27) يؤدي الى نقل كل ما في المتغيرة المهيكلة **' ر1 '** الى ما في المتغيرة المهيكلة **' ر2'**، بناءا على احدى السياستين اللتين عرضتا من قبل والتي تفسر سلوك عملية الشحن، اي نسخ ذاكرة ما في يمين العملية = بما في ذاكرة ما في يسار العملية = بالنسبة للسياسة الأولى، ونسخ سمات المتغيرة في يمين العملية = بسمات المتغيرة في يسار العملية = بالنسبة للسياسة الثانية، ويبين خطأ: لم يُعثر على مصدر المرجع نتيجة تنفيذ عملية النسخ **' ر2 = ر1'**، وفي هذه النتيجة نرى جليا ان **كلا الرسمين يتشاركان نفس المستطيل**، وهذا في حقيقة الأمر مخالف للواقع، فكل رسم مستقل، فمحتوى رسم ما لا يمكن ان يتواجد في رسوم أخرى حتى ولو كانت الرسوم تشبه بعضها البعض.

تذكير: في الجزء الأول تعلمنا انه اذا كانت صلتين، مثلا ص1 و ص2، مرتبطتين بمتغيرتين طارئتين، فان الكتابة ص1=ص2 لا تعني البتة شحن المتغيرة الطارئة المرتبطة بالصلة ص1 بما في المتغيرة الطارئة المرتبطة بالصلة ص2، بل تغيير الصلة ص1 وشحنها بما في الصلة ص2، وبعد انتهاء تنفيذ العبارة ص1 = ص2، يصبح للمتغيرة الطارئة المرتبطة اصلا بالصلة ص2 صلتين هما ص1 و ص2، و تصبح المتغيرة المرتبطة سلفا بالصلة ص1 بدون صلة، وعلمنا ايضا ان متغيرة طارئة بدون صلة هي متغيرة ضائعة.



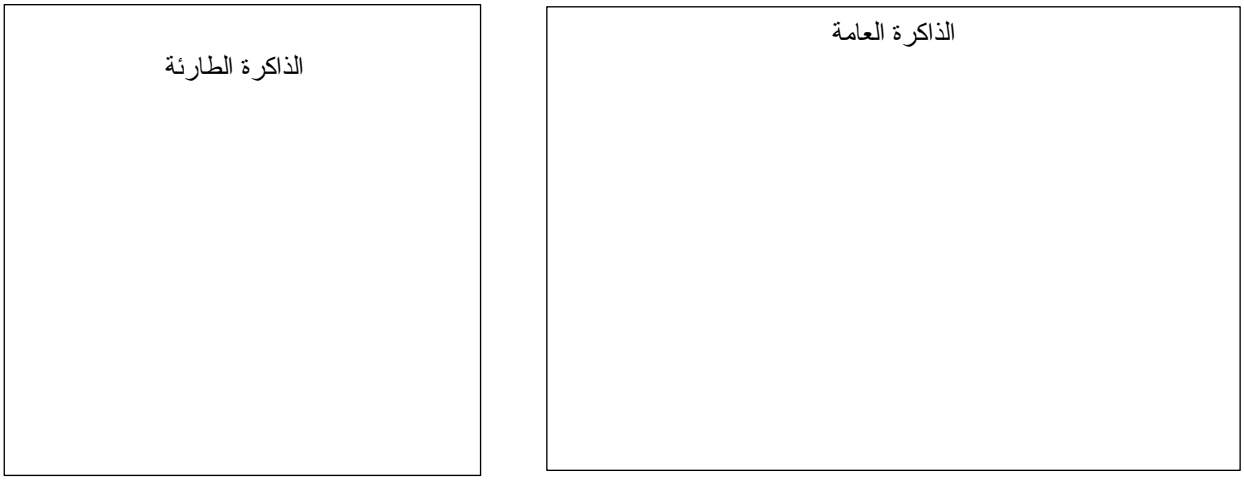
الشكل 5 : حالة الذاكرة بعد تنفيذ تعليمة بشحن المتغيرة م2 بما في المتغيرة م1 (السطر 26 من النص 10)

4 – 2 – 2 المثال الثاني:

في هذا المثال نصحح الإشكال السابق، والذي اصبح فيه رسمين مشتركين في نفس المستطيل، فمثلا، في اطار خوارزم لرسم ومعالجة الرسوم، اذا اردنا ان نغير ملامح مستطيل الرسم **ر1** (تحريكه او تكبيره)، فان ملامح مستطيل الرسم **ر2** سوف تتغير ايضا، لأن مستطيل الرسم **ر1** هو نفسه مستطيل الرسم **ر2**، وما نريده هذه المرة هو ان يكون لكل رسم مستطيلا خاصا به، لن يشاركه فيه اي رسم آخر.

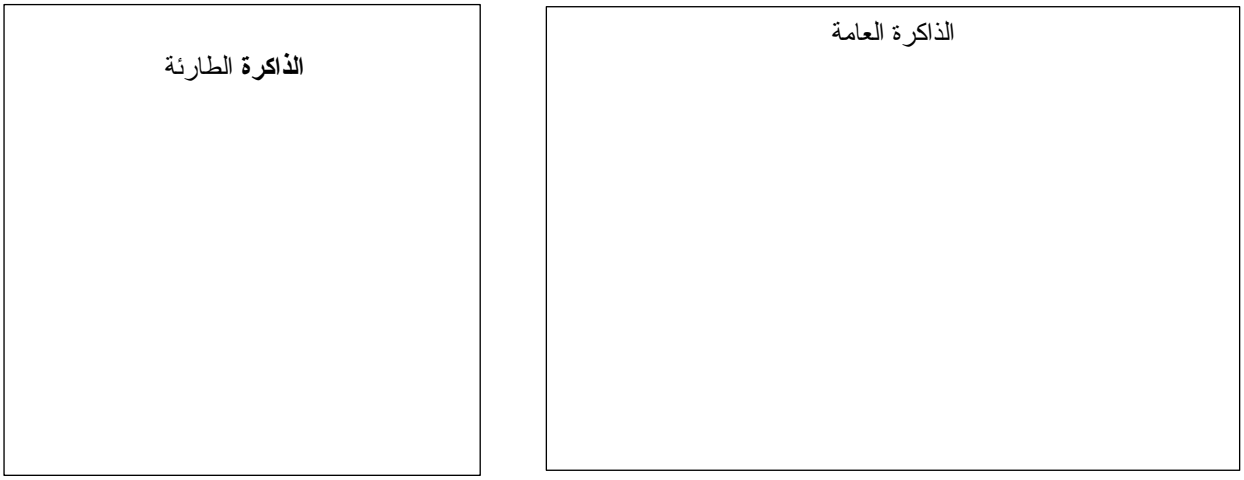
نذكر انه في المثال الأول، لم ينشأ للمتغيرة **ر2** اي سمة تمثل المستطيل الخاص بها، وفي هذا المثال نقوم اولا بتزويد الرسم **ر2** بمستطيل خاص به(السطر 27 من خطأ: لم يُعثر على مصدر المرجع)، ولا نهتم بما في هذا المستطيل لكوننا نريد فيما بعد ان نغير ملامحه لتصبح كملامح مستطيل الرسم **ر1**، ونلاحظ ان تعليمة السطر 27 لا تستعمل الوظيفة **كون_مستطيل** واستعملت مباشرة التعليمة **شيد**، وهذا راجع فقط الى كوننا لا نهتم الآن بما في هذا المستطيل الجديد و لم نرد انشاءه بقيم اولية، وهكذا بعد تنفيذ التعليمات من السطر 25 الى السطر 28 (خطأ: لم يُعثر على مصدر المرجع)، تكون الذاكرة على حالة خطأ: لم يُعثر على مصدر المرجع.

النص 11: النسخة الثانية لإجراء اساسي (الأولى في النص 10)



الشكل 6 : حالة الذاكرة بعد تنفيذ التعليمات من السطر 24 الى السطر 27 من النص 11

بعد اتمام تنفيذ التعليمة ر2 = ر1 (السطر. 28 من خطأ: لم يُعثر على مصدر المرجع)، اي نسخ ر2 بما في ر1 على احدى السياستين للنسخ واللتين عرضتا فيما سبق، تصبح ذاكرة الخوارزم على ما يظهر. في خطأ: لم يُعثر على مصدر المرجع، ونرى جليا ان الهدف المنشود لم يتحقق، بل الحال اصبح اسوأ من الحال الأول، فزيادة على ما تحصلنا عليه في المثال الأول، في هذا المثال تصبح الذاكرة الطارئة تحمل متغيرة ضائعة لا يمكن الوصول اليها، وهي المتغيرة التي أنشأت لتمثل المستطيل الخاص بالرسم ر2.



الشكل 7 : حالة الذاكرة بعد تنفيذ تعليمة نسخ ر2 بما في ر1 (السطر 28 من النص 11)

4 – 2 – 3 المثال الثالث:

في هذا المثال نستعرض الكيفية الصحيحة والفعالة لنسخ متغيرة مهيكلة تحتوي على سمات طارئة، ونُذكر. **ان أصل الإشكال الذي ادى الى النتائج الغير مرغوب فيها في المثال الأول و الثاني هو سلوك عملية النسخ**، فهي لا تستطيع ان تخرج من المنطقة التي تحتلها المتغيرة المهيكلة في الذاكرة العامة او الذاكرة الخاصة بإجراء او وظيفة، ونعني هنا بالخروج هو اتباع الصلات للوصول الى الذاكرة التي تحتلها السمة الطارئة (اي المتغيرة المرتبطة بالصلة التي تمثل السمة).

لكي تتمكن عملية النسخ من الخروج من ذاكرة المتغيرة التي في يسارها او يمينها، فعليها ان تتحلى بسلوك معقد جدا قادر على الأخذ بعين الاعتبار بعدد كبير من الحالات التي يمكن ان تكون عليها المتغيرتين التي تستهدفهما، ويتفادى واضعوا لغات البرمجة ان يعتمدوا مثل هذا السلوك المعقد لعملية النسخ، ولتبيان التعقيد، نَصِف فيما يلي بعض من خصائصه,

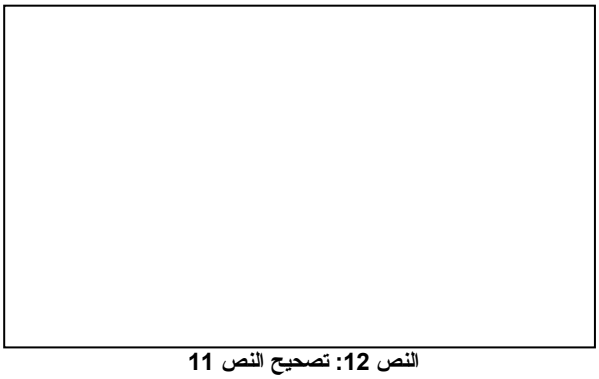
اولا: لإتباع صلة واردة في المتغيرة مصدر. عملية النسخ (المتغيرة في يسار الرمز =) والوصول الى المتغيرة المرتبطة بها، يجب على عملية النسخ فهم محتوى الصلة الممثلة لسمة ما، وهذا امر في حد ذاته غريب عن مفهوم النسخ، ومع فرضية تقبل هذا الأمر، فالفهم يجعل عملية النسخ قادرة ان تقرر

ان السمة الطارئة موجودة في المصدر او غائبة، فعلى سبيل المثال، في الأمثلة الحالية، اذا كانت قيمة الصلة هي 0 ، فالصلة غائبة، والا فهي حاضرة.

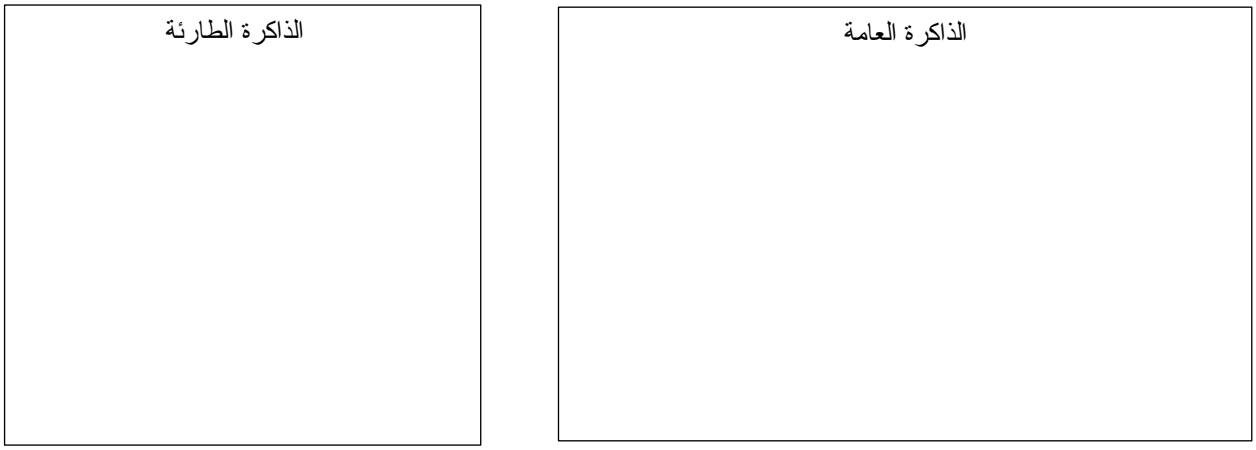
ثانيا: في حالة وجود السمة الطارئة في المصدر، تحاول عملية النسخ معرفة حالة السمة الطارئة في الهدف (اي المتغيرة التي تظهر في يمين الرمز =)، فان غابت، فعلى عملية النسخ ان تنشأ المتغيرة الطارئة الممثلة للسمة الطارئة في الهدف، وهذا امر بعيد كل البعد عن مفهوم النسخ.

ولا يقتصر التعقيد فقط عن قدرة الفهم وانشاء متغيرات طارئة، فيمكن مثلا ان تكون سمة طارئة مبنية على سمات طارئة أخرى، وهكذا يضيف تعقيد هيكل المتغيرة تعقيدا في سلوك عملية النسخ، فيمكن وجود عدد كبير من الصلات في المستوى الأول والثاني و الثالث الخ، وعلى العملية ان تتبع كل صلة وتتعمق في الإتباع.

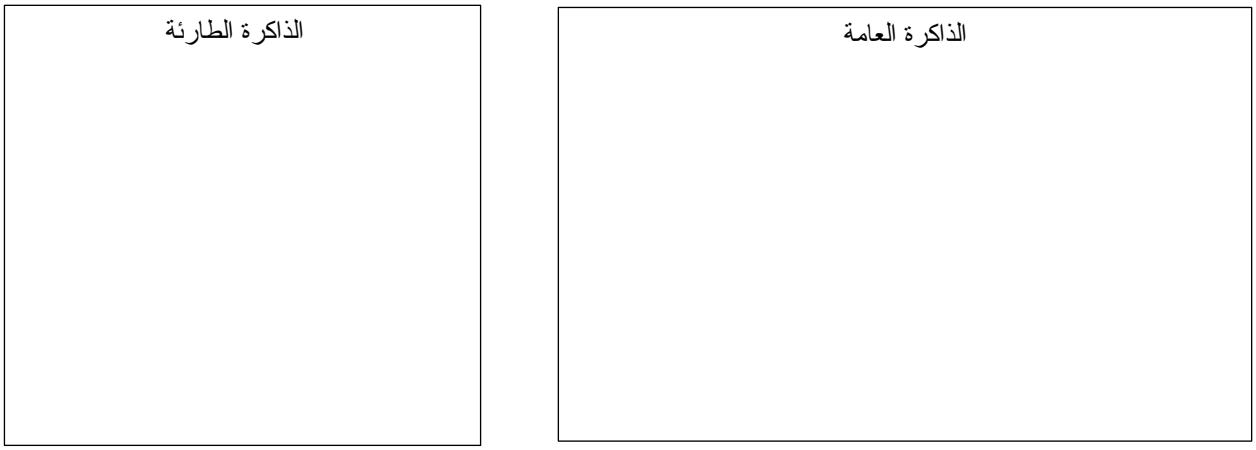
الحل: يكمن الحل في القبول بالسلوك العادي لعملية النسح، وعلى كاتب الخوارزم ان يتحمل تعقيدات الأنماط التي يُعرفها، وتحمل التعقيدات يكون بكتابة تعليمات او بإنجاز وظائف او اجراءات تقوم بعملية التتبع والإنشاء، فهاتين العمليتين في الحقيقة تخضع لما يريده مصمم الخوارزم او البرنامج، فمثلا في خطأ: لم يُعثر على مصدر المرجع، يمكن ان نكتب **ر2=ر1** (السطر. 27)، لكن علينا تسديد نتيجته، فننشأ المتغيرة الطارئة الخاصة بالمتغيرة **ر2** ، لكونها لم تكن حاضرة من قبل، وهكذا نفك ارتباط المتغيرة **ر2** بالمتغيرة الطارئة الخاصة بالمتغيرة **ر1**، فنتحصل على ذاكرة يمثل حالتها خطأ: لم يُعثر على مصدر المرجع وخطأ: لم يُعثر على مصدر المرجع.



النص 12: تصحيح النص 11



الص الشكل 8: حالة الذاكرة بعد تنفيذ التعليمات من السطر 27 الى السطر 28 من النص 13



الشكل 9: حالة الذاكرة بعد تنفيذ التعليمات من السطر 29 الى السطر 32 من النص 12

اما فيما يخص خطأ: لم يُعثر على مصدر المرجع الذي يصحح خطأ: لم يُعثر على مصدر المرجع، فقد حذفت فيه التعليمة **ر2=ر1** (الموجودة في السطر 29 من خطأ: لم يُعثر على مصدر المرجع)، لكون هذه التعليمة تقوم بنسخ قيم ر1 في ر2 وكذلك تقوم بتضييع المتغيرة التي تمثل السمة الطارئة للمتغيرة ر2، ففي مثل هذه الحالة يجب ان نحذر فيها من مثل التعليمة 29، فلا نستعملها، وعوضا عنها نستعمل تعليمات نقل السمات الغير الطارئة واحدة تلو الأخرى، وفيما يخصنا (خطأ: لم يُعثر على مصدر المرجع) ننقل سمة عدد الدائرات (السطر 29)، ولا حاجة لكتابة تعليمة نقل سمة عدد المستطيلات لأن هذه التعليمة قد كتبت في السطر 28، ثم ننقل قيم السمات الطارئة، وهذا ما نراه في الأسطر من 29 الى 32.



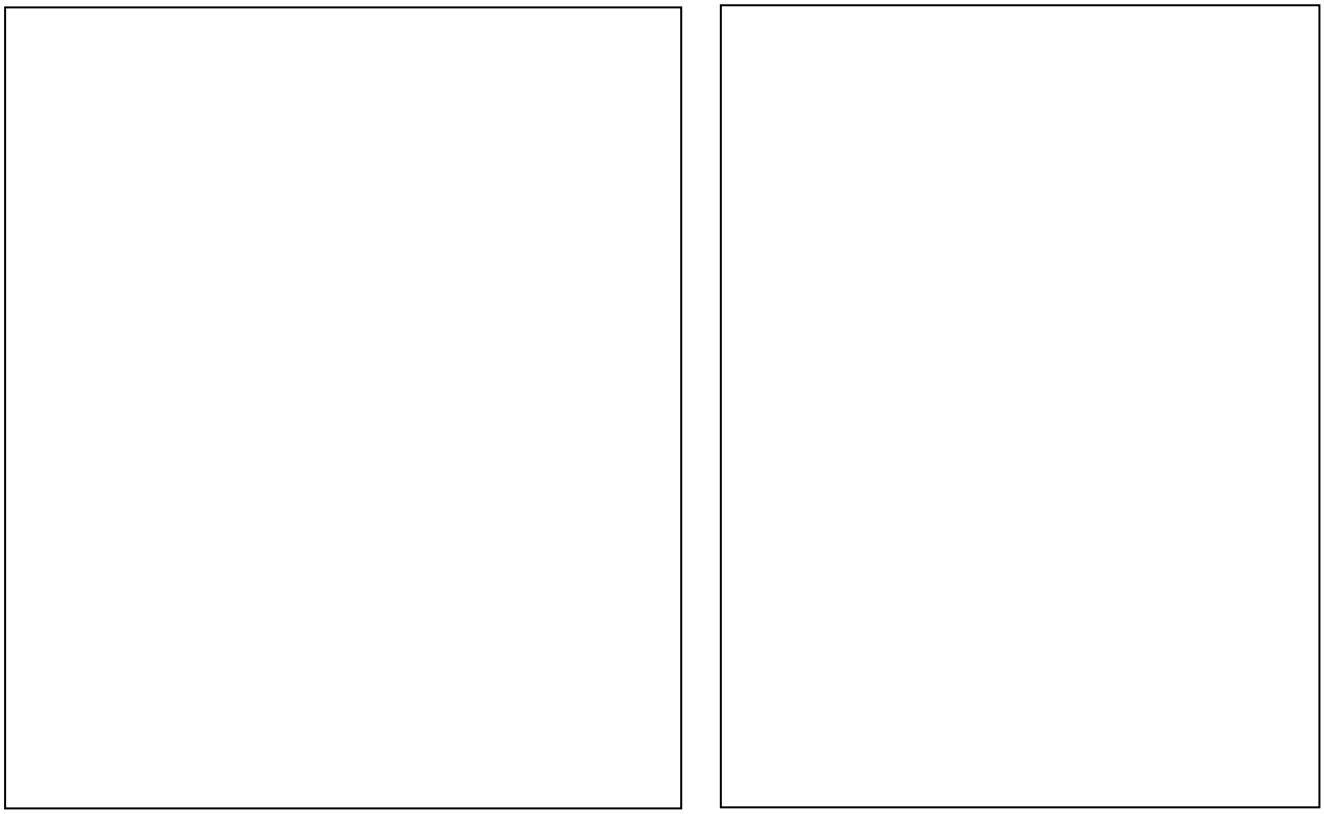
النص 13: تصحيح النص 12

4 – 2 – 4 تحسين طريقة المثال الثالث:

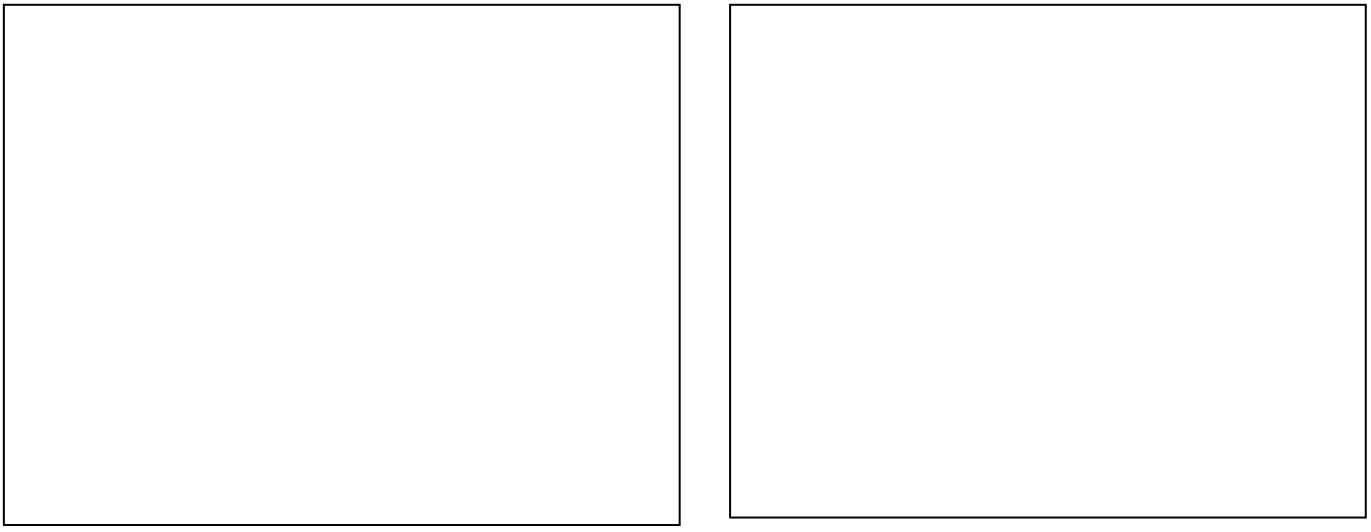
في خطأ: لم يُعثر على مصدر المرجع، ادخلنا تغييرا على النمط رسم، فاصبح يحتمل الآن مثلثا، وخطا مستقيما اضافة الى مستطيلا، ودائرة، ومن اجل تحمل مثلثا وخطا مستقيما، اضيف الى النص التعريف بالنمطين: **مثلث** و **مستقيم**، وزيادة على الوظيفة التي تمكن من صنع متغيرة طارئة تمثل مستطيلا، اضيفت وظائف تتعلق بإنشاء متغيرات مهيكلة طارئة ومهيأة بقيم اولية و هي:

الجزء الثاني: الفصل الرابع: العمليات الأساسية على المتغيرات المهيكلة

اصنع_دائرة في السطر 29، **اصنع_مثلث** في السطر 36 و **اصنع_مستقيم** في السطر 46، ولكون النمط **رسم** يحتوي الآن على 8 سمات، فقد تم ذكر 8 قيم عند التصريح بالمتغيرتين **ر1** و **ر2** بقيم اولية في السطرين 55 و 56، والقيم الأولية التي ذكرت تفيد ان كلا الرسمين **ر1** و **ر2** فارغين لا يحتويان على اي مكون.

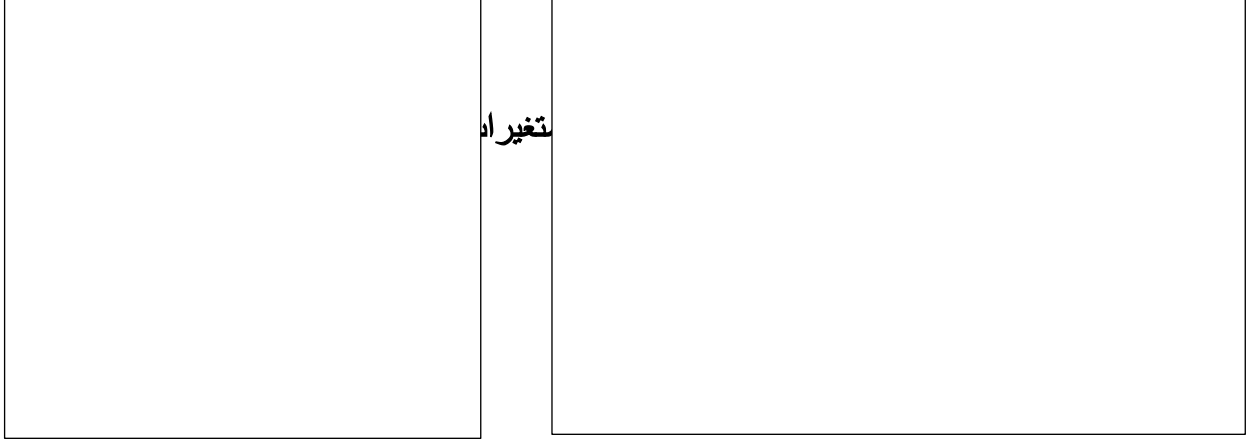


النص 14: النص الأساسي الخاص بتحسين طريقة المثال الثالث

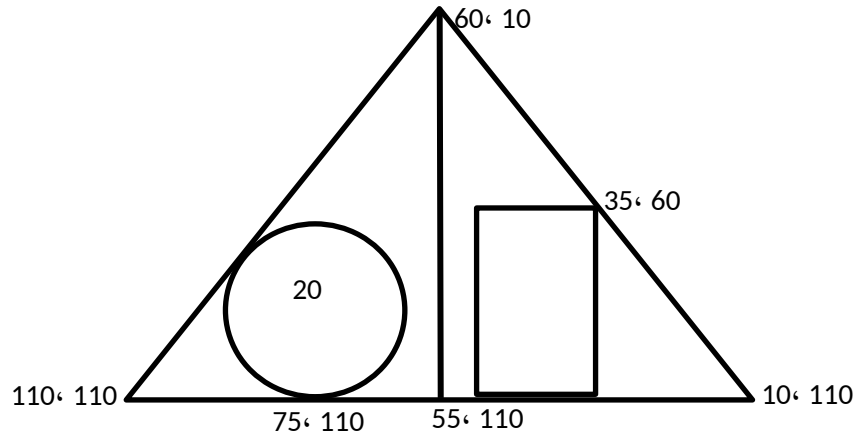


النص 15: تكملة النص 14





الشكل 10: حالة الذاكرة بعد تنفيذ التعليمات من السطر 58 الى السطر 65 من النص 15



الشكل 11: الرسم الذي تمثله المتغيرة ر1 من النص 15

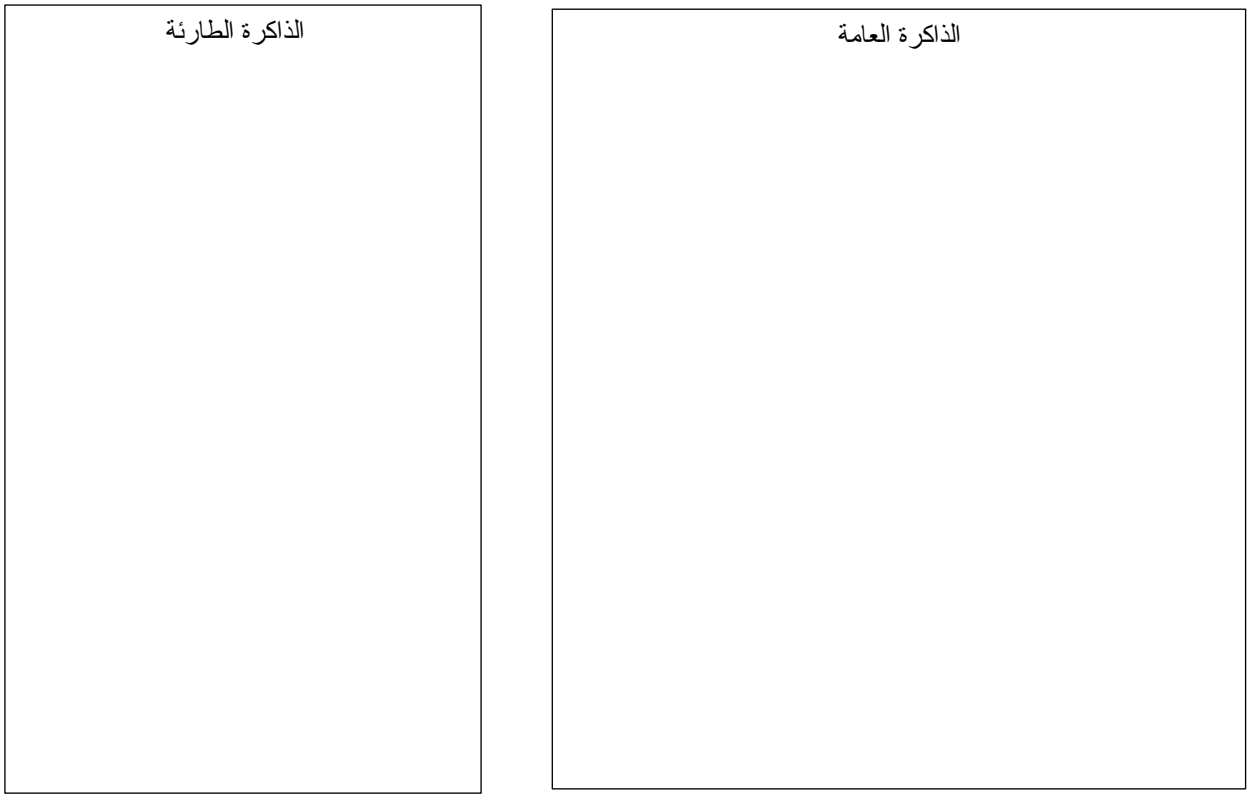
بعد التصريحين في السطرين 55 و 56 بالمتغيرتين العامتين **ر1** و **ر2**، و بعد اضافة كل السمات الطارئة الى المتغيرة **ر1** عبر تعليمات الأسطر من 58 الى 61 من خطأ: لم يُعثر على مصدر المرجع وبعد تثبيت قيم السمات **عدد_مس**، **عدد_دوائر**،**عدد_مث** و **عدد_مق** (الأسطر 62 من الى 65 من خطأ: لم يُعثر على مصدر المرجع) تصبع ذاكرة الخوارزم على الحالة الظاهرة في خطأ: لم يُعثر على مصدر المرجع و فيها تمثل المتغيرة **ر1** الرسم الموجود في خطأ: لم يُعثر على مصدر المرجع.

نريد الآن ان ننجز عملية نسخ كل ما في المتغيرة **ر1** في المتغيرة **ر2**، فنقوم اولا، بنسخ السمات الغير الطارئة، ولأن المتغيرة **ر2** لا تحتوي على أي سمة طارئة، يمكننا ان نكتب **ر2 = ر1**، ولا نخش ضياع أي متغيرة طارئة تمثل سمة طارئة، كون مثل هذه المتغيرات غائبة، وهذا ما نراه

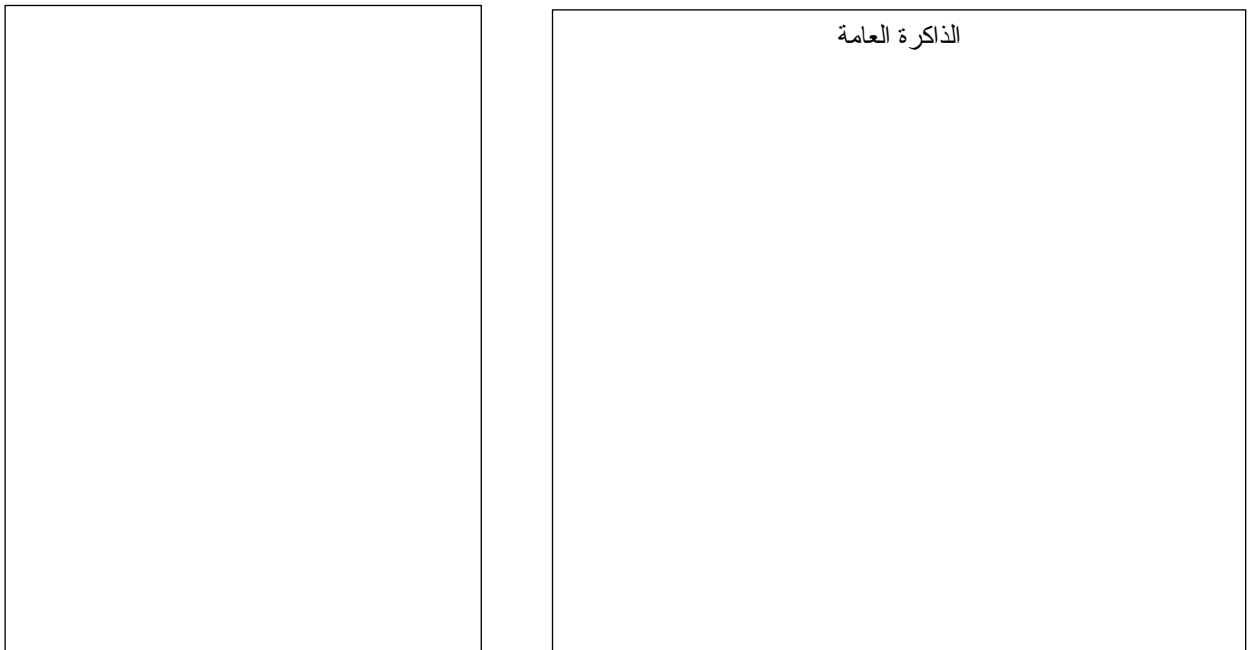
فعلا في السطر 67، وبعد تنفيذ السطر 67، تصبح الذاكرة على حال خطأ: لم يُعثر على مصدر المرجع، أي ان الرسم **ر1** و **ر2** مشتركان في نفس السمات الطارئة.

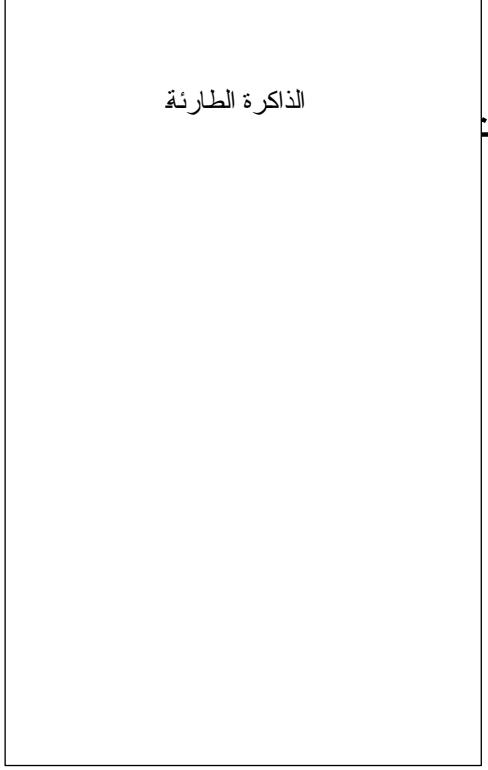
لجعل الرسم الممثل بالمتغيرة **ر2** مستقل تماما عن الرسم الممثل بالمتغيرة **ر1**، نقوم بكتاية التعليمات التي تنجز ما يلي:

- اولا إنشاء السمات الطارئة للمتغيرة **ر2**، وهذا ما تقوم به تعليمات الأسطر من 69 الى 72.
- ثانيا نسخ كل سمات **ر1** في **ر2** الواحدة تلو الأخرى، وهذا ما تقوم به التعليمات من السطر 73 الى89، و بعد تنفيذ هذه الأسطر. تصبح الذاكرة على حالة خطأ: لم يُعثر على مصدر المرجع



الشكل 12: حالة الذاكرة بعد تنفيذ تعليمة السطر 67 من النص 16





الشكل 13: حالة الذاكرة بعد تنفيذ تعليمات الأسطر من 73 الى89 من النص 16

وتبرز ملاحظتان في خطأ: لم يُعثر على مصدر المرجع:

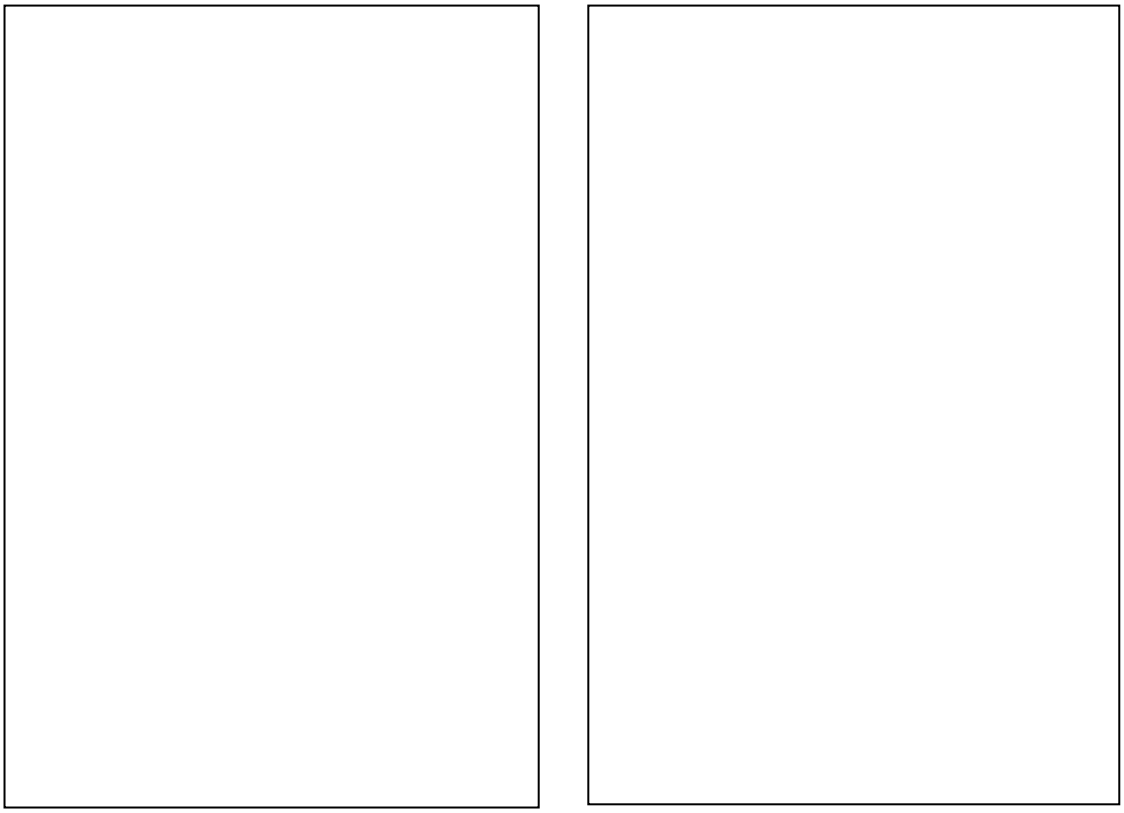
الملاحظة الأولى: انشاء السمات الطارئة للمتغيرة **ر2** بقيم اولية مجهولة: في الأسطر من 69 الى 72 نلاحظ الاستعمال المباشر للتعليمة **شيد** عوض استعمال الوظائف **اصنع_مستطيل، اصنع_دائرة، اصنع_مثلث، اصنع_مستقيم**، واخترنا الاستعمال المباشر للتعليمة **شيد** في انشاء السمات الطارئة لأن قيم هذه السمات عند الإنشاء لا تهم، فمهما كانت هذه القيم فانها سوف تمحى وتعوض بالقيم الموجودة في المتغيرة **ر1**.

الملاحظة الثانية: العدد الكبير لعمليات نسخ سمات **ر1** في **ر2**، ويكون هذا العدد اكبر كلما كبر تعقيد وارتباط هيكل المتغيرة بالسمات الطارئة، وهذا اشكال يجب علينا ان نبحث له عن حل ما امكن.

اشكال الملاحظة الثانية يشبه كثيرا الإشكال الذي صدفناه في الفصل الخاص بتهيئة المتغيرات المهيكلة بقيم اولية، والحل الذي توصلنا اليه هو انجاز وظيفة او اجراء تقوم بالعملية، وعند الحاجة لا نكون مضطرين لإعادة كتابة كل التعليمات التي تقوم بنسخ سمة بسمة اخرى، بل نطلب من الوظيفة او

الجزء الثاني: الفصل الرابع: العمليات الأساسية على المتغيرات المهيكلة

الإجراء انجاز العملية على المتغيرات التي نصفها في مداخل الوظيفة او الإجراء، وقد اوضحنا فيما سبق الانعكاسات التي ستحدث عند كتابة العبارات اذا اختار كاتب الخوارزم ان يستعمل إجراءا بدل وظيفة.

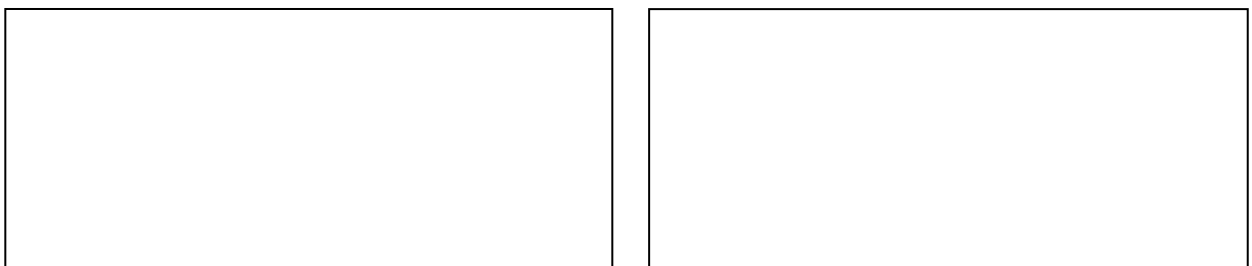


(أ) (ب)

النص 16: اضافة وظيفة نسخ متغير مهيكلة من النمط رسم بأخرى: (ا) طريقة "سي++": وظيفة تعرف سلوك الرمز (ب) طريقة الوظيفة العادية

بما ان عملية النسخ الممثلة بالرمز = هي عملية كباقي العمليات فإن انجازها يمكن ان يكون على احدى الطريقتين التاليتين:

- انجاز وظيفة على طريقة لغة البرمجة "**سي++**" (خطأ: لم يُعثر على مصدر المرجع-أ)، تحتوي على سلوك الرمز = عندما يكون معامله الأيمن ومعامله الايسر من نمط مهيكل ما، وفي هذه الحالة المشابهة لما نجده في اللغة "**سي++**" يكون اسم الوظيفة مكونا من الكلمة **عملية** متبوعة برمز العملية (مثلا: **عملية=** ، **عملية+**، **عملية*** الخ)، و بفضل هذه الطريقة يمكن ان نستعمل في العبارات رمز العملية، وعند انجاز العملية لا ينفذ السلوك العادي للعملية، بل ينفذ السلوك الذي عُرف في الوظيفة.
- انجاز وظيفة عادية بأي اسم يراه مناسبا كاتب الخوارزم(خطأ: لم يُعثر على مصدر المرجع-ب)، وفي هذه الطرقية لا نستعمل الرمز في العبارات بل نستعمل اسم الوظيفة.



يمثل خطأ: لم يُعثر على مصدر المرجع تتمة خطأ: لم يُعثر على مصدر المرجع، ويمثل الجانب (أ) من خطأ: لم يُعثر على مصدر المرجع الطريقة التي تمكن من استعمال الرمز = في العبارات، وهي طريقة "**سي++**"، اما الجانب (ب) فيمثل الطرية المباشرة او العادية و فيها لا يمكن استعمال الرمز، بل يستعمل اسم الوظيفة في العبارات، وما عملناه هو اخراج الاسطر من 73 الى 89 من خطأ: لم يُعثر على مصدر المرجع وتحويل الأسطر الى وظيفة في خطأ: لم يُعثر على مصدر المرجع.

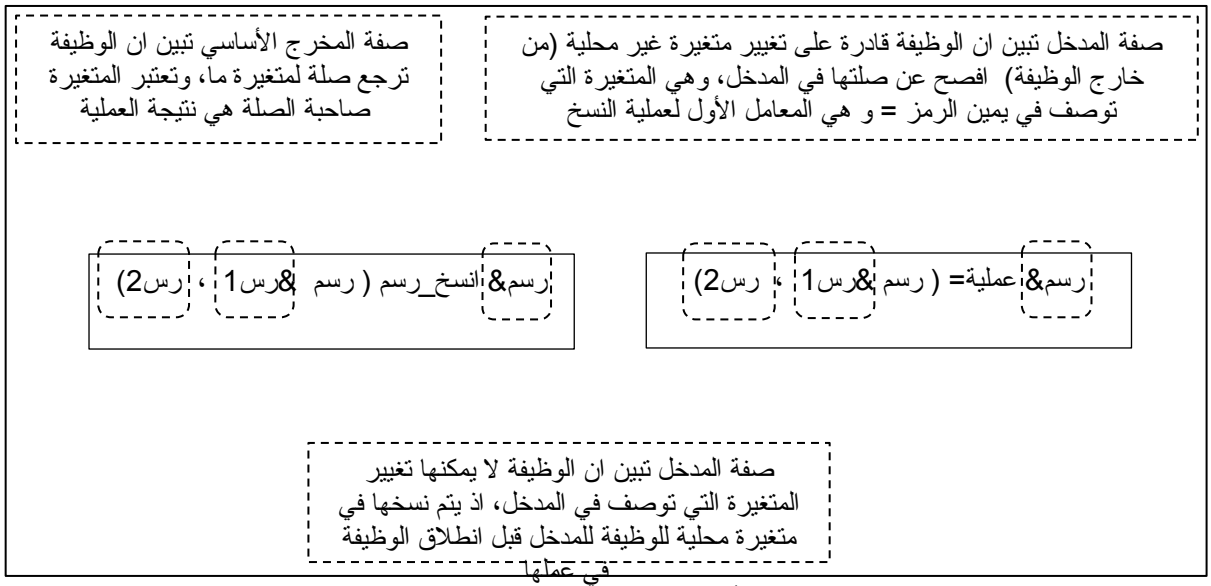
نلاحظ في آخر خطأ: لم يُعثر على مصدر المرجع التصريح بأربعة متغيرات من النط رسم: **ر 1، ر2، ر3** و **ر4**، وصرحت المتغيرة **ر1** بقيم اولية، بينما لم تُصرح باقي المتغيرات بقيم اولية، وهذا راجع الى ان قيم هذه المتغيرات لا تهمنا عند انشاءها، وان هذه المتغيرات سوف تنسخ بكتابة واحدة فقط (خطأ: لم يُعثر على مصدر المرجع، السطر 92) انطلاقا بما في المتغيرة ر1، و نلاحظ هنا انه لو لم نستعمل النسخ بوظيفة لكان علينا لزاما ان نكتب 17 تعليمة لنسخ متغيرة واحدة فقط (الأسطر من 73 الى 89 من خطأ: لم يُعثر على مصدر المرجع)، اي 51 تعليمة لنسخ ر2 و ر3 و ر4 انطلاقا من ر1.

4 – 2 – 5 دراسة معمقة لحقيقة مداخل الوظيفة الممثلة لعملية النسخ:

في الأمثلة السابقة اسردنا نص وظيفة النسخ على شكليها في خطأ: لم يُعثر على مصدر المرجع دون تفسير كاف حول الخيارات التي حددت في وصف المداخل، وفي هذه الفقرة نعيد النظر في ما ورد في رأس الوظيفة لنرى هل الأوصاف التي حددت للمنافذ تفي بالغرض ام ان هناك نقص يجب تداركه ام خطأ يجب اصلاحه.

نلاحظ في منافذ وظيفة النسخ (السطر 54 من خطأ: لم يُعثر على مصدر المرجع) وجود صلتين: الأولى في المدخل **رس1** والثانية في **مخرج الرجوع** للوظيفة كما يظهر في خطأ: لم يُعثر على مصدر المرجع، وبواسطة الصلة **رس1** يمكن للوظيفة الوصول الى متغيرة خارجية (غير محلية) وتغييرها، ونلاحظ ان المدخل **رس2** لم يوصف على انه صلة بل وصف على انه متغيرة محلية للوظيفة، ولا يمكن للوظيفة الوصول الى المتغيرة التي توصف في مدخل الوظيفة في تعليمة

طلب تشغيلها، فتعليمة طلب التشغيل تقوم بنقل قيم المتغيرة التي وصفت في المدخل الى المتغيرة المحلية، وتعليمات الوظيفة تستهدف المتغيرة المحلية.



الشكل 13: رأس وظيفة على شكليها الاثنين (النص 17)

تمثل الصلة **رس1** المعامل الموجود على يمين الرمز = ويمثل المدخل **رس2** المعامل الموجود في يسار الرمز =، وفي عملية النسخ يكون المعامل على يمين الرمز = هو المعامل الذي يجب ان يتغير، اما محتوى المعامل الموجود في يسار الرمز = فإنه لا يتغير، وهذا هو الذي يفسر استعمال الصلة في المدخل **رس1**.

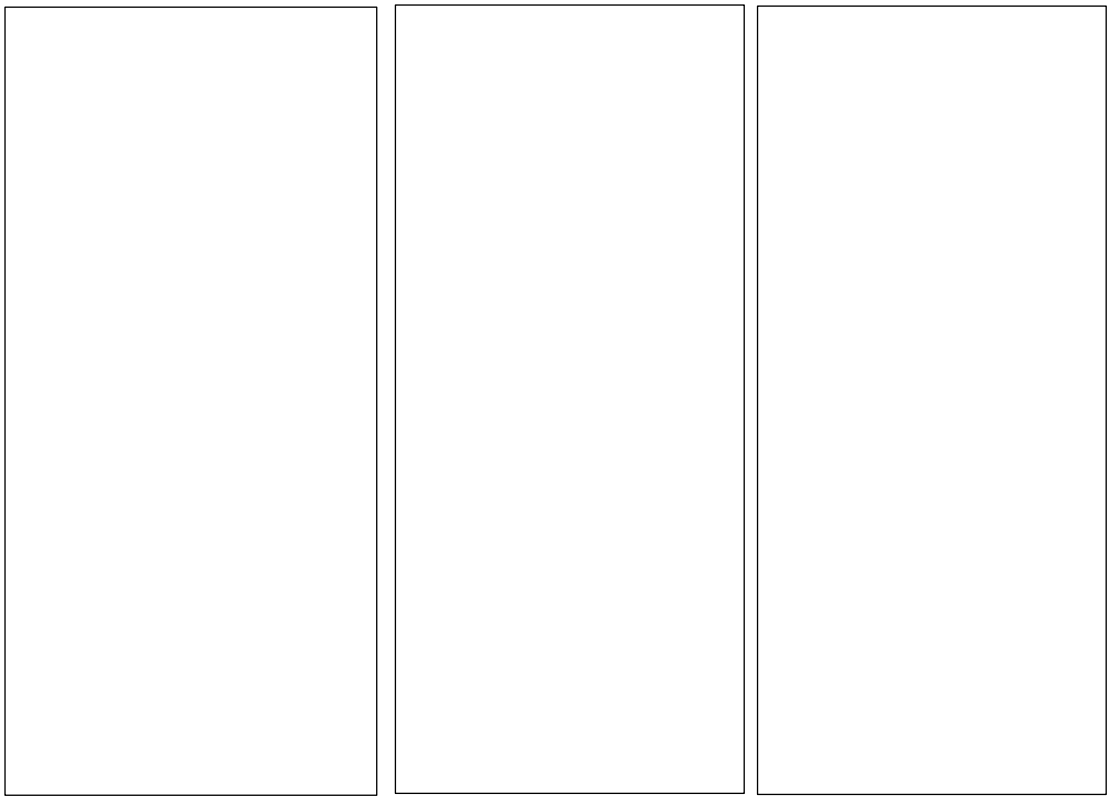
و نلاحظ ايضا في السطر 54 من خطأ: لم يُعثر على مصدر المرجع ان الوظيفة ترجع الصلة **رس1** بعد اتمام عملها، وهذا يتوافق مع المفهوم المتعارف عليه في تقييم عبارة مبنية على العملية =، اذ ان تقييم عبارة النسخ يفضي الى نتيجة ما تكون مخزنة دائما في المتغيرة المذكورة على يمين الرمز =، وليس في متغيرة طارئة كما هو الشأن لأكثر العمليات، فعلى سبيل المثال اذا اردنا ان نقيم التعليمة **ر4 = ر3= ر2= ر1**، فان مسار التقييم، حسب جدول أوليات العمليات الأساسية، يبدأ بتقييم العبارة **ر2=ر1**، والنتيجة تكون القيمة المخزنة في **ر2**، اي ان العبارة **ر4 = ر3= ر2= ر1** تصبح بعد تقييم **ر2 =ر1** على الشكل **ر4 = ر3= ر2**، فقد استبدلت ر2=ر1 بنتيجتها اي **ر2**، ثم تقيم العبارة **ر3=ر2**، والنتيجة هي القيمة المخزنة في **ر3**، فتصبح العبارة الأساسية على الشكل **ر4=ر3** بعد تقييم **ر3=ر2**، وفي الآخر تقيم آخر عملية و هي **ر4 = ر3** ونتيجتها هي القيمة المخزنة في **ر4**.

4 – 2 – 6 تحسين في وصف المدخل 'رس2' لوظيفة النسخ.

لاحظنا فيما سبق ان المدخل **رس2** لوظيفة النسخ قد وصف على انه متغيرة محلية، وفي مثل هذه الحالة، عندما يطلب تنفيذ الوظيفة، كما هو واضح في الكتابة **انسخ_رسم (ر2 ، ر1)** المنتمية الى تعليمة السطر 92 من خطأ: لم يُعثر على مصدر المرجع:

- اولا، في الصلة الممثلة للمدخل **رس1**، تقوم الوظيفة بنسخ معلومات عن موقع المتغيرة **ر2** في الذاكرة ، ونذكر ان صلة مرتبطة بمتغيرة ما، هي نفسها متغيرة بدائية (او اساسية) تحتوي على معلومات تواجد المتغيرة المرتبطة بها، وهكذا تصبح **رس1** صلة للمتغيرة **ر2**، وبفضل **رس1** تتمكن وظيفة النسخ من الوصول الى **ر2** وتغييرها، ومع ان **ر2** هي في حقيقة الأمر متغيرة عامة يمكن لوظيفة النسخ الوصول اليها مباشرة، الا ان مصمم الوظيفة ارادها ان تكون وظيفة قائمة بذاتها محررة من اي متغيرة عامة.
- ثانيا، في المتغيرة المحلية الممثلة للمدخل **رس2** والمسمات ايضا **رس2**، تقوم الوظيفة بنسخ قيم المتغيرة **ر1** التي وصفت في المدخل **رس2.**

وقبل ان تبدأ وظيفة النسخ تنفيذ اول تعليمة فيها، تكون ذاكرة الخوارزم على خطأ: لم يُعثر على مصدر. المرجع.



الشكل 14: حالة الذاكرة بعد تنفيذ التعليمة انسخ_رسم (ر2 ، ر1) و قبل تنفيذ او تعليمة من وظيفة النسخ

حسب ما يظهر في خطأ: لم يُعثر على مصدر المرجع، نرى اولا ان المتغيرة المحلية **رس2** والمتغيرة العامة **ر2** تحتويان على نفس القيم في السمات الطارئة، ومعنى هذا انهما يتشاركان في نفس المتغيرات الطارئة المرتبطة بالصلات الممثلة للسمات، فمع ان عملية النسخ تنجز عملها على الشكل الصحيح الا ان عملية النسخ التي تحدث في المدخل ليست كما اصبحنا نتصوره، اذا نسخت متغيرة مهيكلة فيها سمات طارئة في متغيرة مهيكلة اخرى، يجب ان تكون السمات الطارئة غير مشتركة بين المتغيرتين حتى ولو كانت تحتوي على نفس القيم، وهكذا نرى خللا في انسجام الأفكار، فما تعلمناه حتى الآن هو ان المدخل ان لم يكن صلة، فالمتغيرة التي توصف فيه عند طلب تنفيذ وظيفة او اجراء لا تكون في متناول الإجراء او الوظيفة، فعملية طلب تنفيذ الوظيفة او الإجراء تقوم بنسخ قيم المتغيرة التي وصفت عند المدخل في متغيرة محلية، اما فيما يخص وظيفة النسخ، فنرى ان جانب من المتغيرة لا يمكن الوصول اليه، والجانب الآخر المتمثل في السمات الطارئة فهي متوفرة للوظيفة وهذا عكس ما اردناه في الرسم الظاهر في خطأ: لم يُعثر على مصدر المرجع.

5 – مشكلة نسخ المتغيرات المهيكلة في مداخل الوظائف والإجراءات:

في عبارة السطر 92 من خطأ: لم يُعثر على مصدر المرجع عندما تنفذ العملية **انسخ_رسم (ر2 ، ر1)**، تقوم تعليمة تنفيذ الوظيفة بنسخ ما في المتغير **ر1** في المدخل **رس2** للوظيفة **انسخ_رسم**، وهذه العملية تتطلب اولا انشاء متغيرة محلية تمثل المدخل وحجمها حجم المتغيرة **ر1**، وبعد ذلك تنسخ تباعا كل السمات، فمثلا في حالة **انسخ_رسم (ر2 ، ر1)**، اذا افترضنا ان النمط طبيعي يستهلك من الذاكرة اربع وحدات وان الصلة تستهلك ايضا اربع وحدات، فان حجم المتغيرة المحلية رس2 هو 36 (4 سمات من النمط طبيعي و اربع صلات) (خطأ: لم يُعثر على مصدر المرجع)، واذا افترضنا ان عملية النسخ تكون بنسخ سمة بسمة فان عدد عمليات النسخ التي يقوم بها منفذ الوظيفة هي 8 عمليات.

الأن اذا اردنا ان نرى تكلفة عبارة السطر 92، اي **انسخ_رسم (ر4 ، انسخ_رسم (ر3، انسخ_رسم (ر2 ، ر1)))** و التي تحتوي على ثلاث عمليات طلب تنفيذ الوظيفة **انسخ_رسم**، فان عدد العمليات المخصصة فقط لنسخ السمات هي 24 عملية، وهذا الكم من العمليات المختصة في تحضير مناخ تنفيذ الوظيفة يعتبر نسبة غير تافهة اذا قُرنت بعدد تعليمات الوظيفة نفسها، فاذا كانت النسبة هي مثلا 50% فان هذه النسبة سوف تظهر ايضا في الوقت الذي تأخذه الوظيفة لإتمام عملها، فاذا كان الوقت الصافي لتعليمات الوظيفة هو **س**، فان الوقت الحقيقي يكون **س + س * 50%**، ولذا لا يمكن لنا تجاهل مثل هذه الحالة ووجب علينا البحث عن طريقة من جهة منسجمة تماما مع ما تعلمناه وثانيا غير مستهلكة من وقت الوظيفة الا نسبة تافهة.

في الجانب الأخر، اي على مستوى المدخل **رس1**، نرى امرين هامين:

- عملية انجاز الصلة المحلية الممثلة للمدخل **رس1**، تمكن من الوصول التام للمتغيرة التي توصف صلتها في المدخل، وهذا امر متجانس مع هدف الصلات عندما توصف في المداخل
- عملية نسخ الصلة تتطلب عملية واحدة فقط مهما كان حجم المتغيرة المهيكلة التي توصف صلتها في المدخل و مهما كان عدد سماتها.

6 – مشكلة عدم توافق الأنماط في كتابة العبارات

في السطر 92 من خطأ: لم يُعثر على مصدر المرجع –(أ) وردت **العبارة ر4 = ر3 = ر2 = ر1** وفي السطر 92 من خطأ: لم يُعثر على مصدر المرجع –(ب) وردت العبارة **انسخ_رسم (ر4 ، انسخ_رسم (ر3، انسخ_رسم (ر2 ، ر1)))**، وهاتين العبارتين متساويتين في الهدف، وفيهما كررت اكثر من مرة عملية النسخ، وحسب التعريف بوظيفة النسخ (خطأ: لم يُعثر على مصدر المرجع) نرى جليا ما يلي:

- المعامل الأول **رس1**، والذي يكون على يمين الرمز = صلة لمتغيرة مهيكلة.
- النتيجة (اي ما ترجعه التعليمة **ارجع رس1** في السطر 77 من خطأ: لم يُعثر على مصدر المرجع) هي ايضا صلة.

اذا اخذنا العبارة **ر4 = ر3 = ر2 = ر1**، نرى اولا ان ما ورد فيها هي اسماء لمتغيرات مهيكلة، واذا قمنا بتقييمها، فأول عملية يجب تنفيذها في سياق التقييم هي **ر2 = ر1** (اي تنفيذ الوظيفة **عملية= (ر2 ، ر1)**) ، وما يجلب انتباهنا ان المدخل في التعريف هو على شكل صلة وما وضع في المدخل عند طلب التنفيذ هو اسم متغيرة، **وهنا يجب الحذر الشديد من مثل هذه الكتابات التي لا تتوافق فيها الأنماط.**

فيما يخصنا، ولأننا على مستوى لغة من لغات الخوارزميات، وحسب ما اوردناه في الفصل الأول، فمثل هذه الكتابة يمكن ان ننظر اليها على انها صحيحة، فلغة الخوارزم موجهة للإنسان وهو من ينفذ الخوارزم للتحقق من صحته قبل ترجمته الى لغة البرمجة، ففي لغة الخوارزميات، عندما توصف متغيرة في مدخل عرف على انه صلة، يقوم منفذ الوظيفة بعملية الربط بين الصلة المحلية (الممثلة للمدخل) والمتغيرة التي وصفت في المدخل، اما اذا وصفت صلة في المدخل فيقوم المنفذ بنقل قيمتها الى الصلة المحلية، وهذه القدرة للغة الخوارزم غير مرغوب فيها في لغات البرمجة، فمثلا في لغة "**جافا**" و لغة "**سي++**" لا يمكن ان نضع اسم متغيرة في مدخل عرف انه صلة.

عند انتهاء تنفيذ وظيفة النسخ، تكون النتيجة صلة ظرفية (متغيرة ظرفية من النمط صلة)، نسميها مثلا **كظ1**، فاذا انتقلنا الى المرحلة الثانية من تقييم العبارة ر4 = ر3 = ر2 = ر1 نستبدل ر 2=ر1 بالنتيجة التي تحصلنا عليها و هي **كظ1**، واذا تتبعنا جيدا وظيفة النسخ نرى ان **كظ1** هي في

حقيقتها صلة للمتغيرة **ر2**، و بعد الاستبدال نحصل على العبارة **ر4=ر3=كظ1**، ثم نستمر في التقييم.

لتقييم **ر4=ر3=كظ1**، وحسب سلم الأولويات و آلية تحديد الجهة ذات الأولوية، نبدأ بتقييم العبارة **ر3 = كظ1**، وهنا ايضا نلاحظ **عدم تناسق الأنماط بين مداخل عملية النسخ و بين ما يذكر للعملية في المداخل عند طلب التنفيذ**، فالمعامل الأول **ر3** اسم لمتغيرة وضع في المدخل **رس1** الذي عرف على انه صلة، والمعامل الثاني **كظ1** صلة ذكر في المدخل **رس2** الذي عرف على انه اسم لمتغيرة.

و كما قلنا، فان مثل هذه الكتابات ان كانت متاحة في لغة الخوارزميات وفي بعض لغات البرمجة التي لا تتحقق من توافق الأنماط، الا انها تمثل سببا لكثير من القلق وعدم الفهم عند ترجمة الخوارزم الى لغة برمجة التي لا تحتمل مثل هذه الحرية في الكتابات، وحتى مع اللغات التي تحتمل قسطا كبيرا من الحرية، كلغة "**سي**"، يمكن ان تؤدي الترجمة الى وقوع اخطاء على شكل كتابات صحيحة في الشكل وغير صحيحة في المعنى، و تدارك مثل هذه الأخطاء عملية غير يسيرة.

7 – رفع اللبس الناجم عن عدم توافق الأنماط في كتابة العبارات

بما اننا من المبتدئين، نرى ان عدم وجود اي لبس في كتابة تعليمات الخوارزم مهم جدا، وهذا لجعل الكتابة واضحة لكل من يقرأها ولجعل عملية الترجمة للغة البرمجة عملية سهلة.

للوصول الى مثل هذه الحالة يجب علينا ان نقلص من الحرية في كتابة الخوارزميات باللغة الشرمزية، فنتبع قوانين واضحة في الكتابة، وزيادة على هذا، يجب ان تكون القوانين ذو فائدة على اداء الخوارزم نفسه.

اصل اللبس الذي وقعنا فيه هي الصلات من جهة ومن جهة اخرى المتغيرات المهيكلة، وكل منها مهم جدا في كتابة الخوارزميات، ولا يمكن الاستغناء عنهم، فكما رأينا في الفصل الأول وكذلك في هذا الفصل، كان لزاما علينا ان نشرح المعاني العديدة التي تأخذها العبارات عندما تذكر فيها الصلات، فلا بد لهذه المعاني المتعددة ان تبق حاضرة في ذهن كاتب الخوارزم وان يَحْذر عند استعمالها.

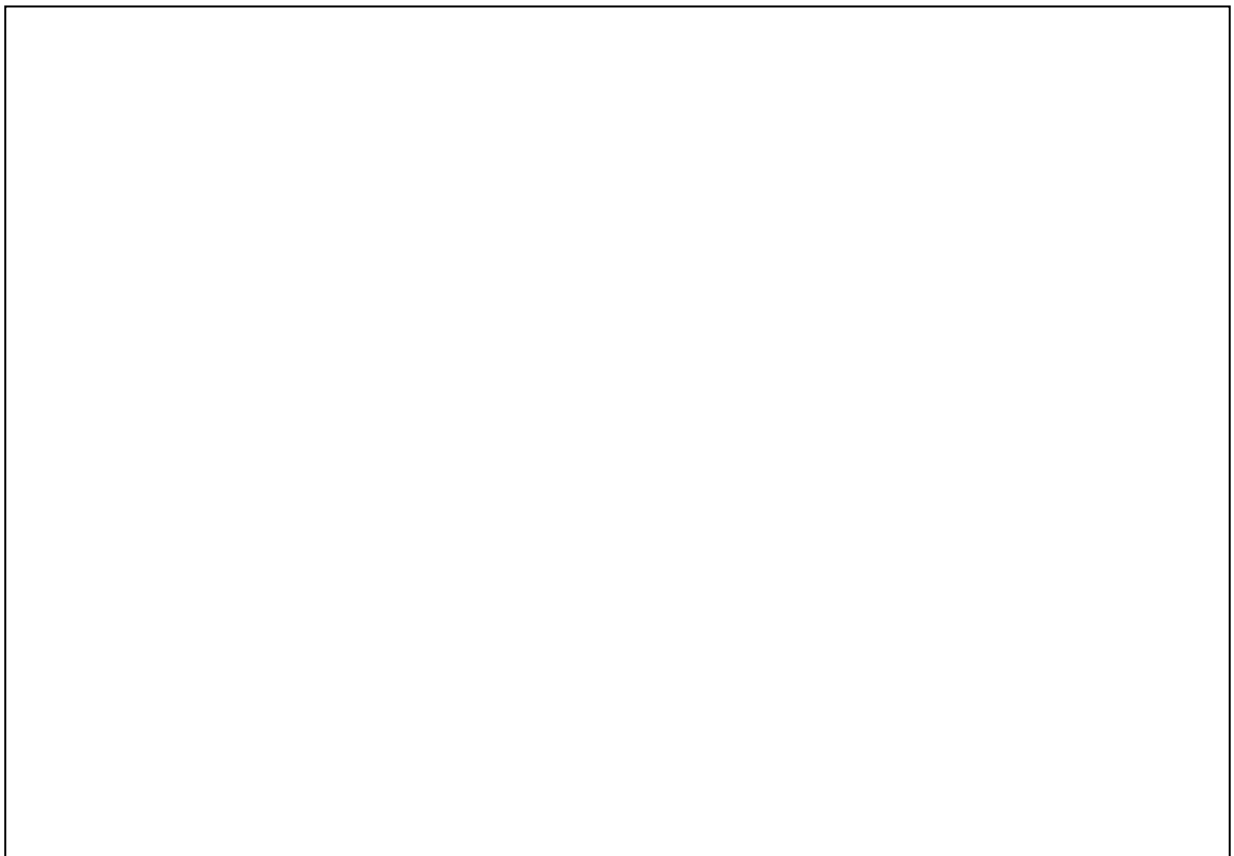
ورأينا ايضا فضل الصلات عندما تكون في منفذ، تدخل منه او تخرج منه متغيرة مهيكلة، فبفضل الصلات يمكن ان نجعل عملية ادخال متغيرة مهيكلة او اخراجا من ووظيفة او اجراء عملية سريعة غير مرتبطة البتة بتعقيد وحجم المتغيرات المهيكلة.

8 – القواعد الجديدة في كتابة نصوص الخوارزميات باللغة الشرمزية

لرفع اللبس المذكور سلفا وما ينجر عليه من متاعب، سوف نتبع من هنا فصاعدا القوانين التالية في كتابة الخوارزميات باللغة **الشرمزية** وهي قوانين تتيح لنا الترجمة المباشرة الى لغة "جافا":

- **عدم استعمال الصلات مع المتغيرات البدائية** (او ما سميناها بالأنماط القاعدية او الأساسية)، ومعنى هذا ان اللغة الشرمزية التي نكتبها لا توجد فيها صلات لمتغيرات من الأنماط الأساسية، فنمتنع اذا ان ننشأ صلة مع متغيرة من احدى الأنماط القاعدية التالية: **حرف، منطقي، طبيعي، حقيقي،** و **صلة** (الصلة في حد ذاتها متغيرة بدائية).
- **عدم استعمال المتغيرات المهيكلة العادية، فكل متغيرة مهيكلة يجب ان تكون متغيرة مهيكلة طارئة**، وفي مثل هذه الحالة يجب اولا ان نصرح بالصلة ثم ننشأ المتغيرة المهيكلة الطارئة باستعمال عملية **شيد** ونربطها بالصلة. فمثلا اذا افترضنا ان الكلمة **شخص** تمثل نمطا مهيكلا، فالكتابة **شخص ش1 ؛** لا تمكن من التصريح **بمتغيرة مهيكلة عادية**، وسوف نرى فيما بعد ان مثل هذه الكتابة لها معنى واحد هو ان **ش1** صلة.
- **عدم استعمال العملية --< لعدم جدواها في اطار ما ذكر من قوانين** (1 و 2)، فهذه العملية تربط متغيرة عادية بدائية او مهيكلة بالصلة، وبما اننا نمتنع من استعمال الصلات مع المتغيرات البدائية ولا نسمح بالمتغيرات المهيكلة العادية، فإنها لن تجد المناخ الذي تستعمل فيه.
- **عدم جدوى من استعمال الرمز & للتدليل على ان الاسم هو لصلة**، فحسب القانون لا نستعمل الصلات مع الأنماط البدائية وكل صلة هي للمتغيرات المهيكلة ولا تستعمل الا معها ولا يمكن استعمال غير الصلات للتعامل مع المتغيرات المهيكلة، وهكذا، في اي مكان من نص الخوارزم، اذا ظهر اسم لمتغيرة تبدو مهيكلة، فالاسم هو في حقيقة الأمر اسم لصلة، ففي اللغة الشرمزية التي تحترم القوانين الجديدة، الكتابة **شخص ش1؛** تفيد ان **ش1** صلة، والكتابة **شخص &ش1؛** لا جدوى منها مع القوانين الجديدة التي علينا اتباعها عند الكتابة باللغة الشرمزية.
- امكانية استعمال العملية **شيد** فور التصريح بصلة لمتغيرة مهيكلة، فمثلا، اذا اردنا انشاء متغيرة من النمط **شخص** نكتب : **شخص ش1= شيد شخص؛**
- عند طلب تنفيذ اجراء او وظيفة، ومن اجل تفادي ظهور اي التباس يمنع منعا باتا عدم احترام توافق الأنماط بين المداخل وما يذكر لها من معطيات.
- كل رموز العمليات التي تستهدف المتغيرات المهيكلة (وهي في الحقيقة تستهدف الصلات)، ما عدا الرمزين = و ==، غير مسموح بها الا بعد تعريفها في اطار وظيفة تحمل اسما مكونا

من الكلمة **عملية** متبوعة برمز العملية (مثلا: **عملية+** او **عملية*** الخ)، فالعملية **=** تنسخ الصلة في يمينها بمحتوى الصلة التي على يسارها، واما العملية **==** فتقوم بمقارنة القيم التي تحتويها الصلات، ونذكر ان محتوى الصلة هي قيمة تبين مكان المتغيرة المرتبطة بالصلة في الذاكرة، فمثلا في السطر 12 من خطأ: لم يُعثر على مصدر المرجع اعيد تعريف سلوك الرمز = بالنسبة للنمط مستطيل، وفي السطر 32 نرى الكتابة **د3 = د1** ، و بما ان **د1** و **د 3** من النمط **دائرة**، و بما ان الرمز = لم يعرف بالنسبة للنمط **دائرة**، فالسلوك الذي يسلكه الرمز = هو السلوك الأصلي، وهنا يقوم بنسخ الصلة **د3** بما في الصلة **د1**، و بما ان **د1** كانت مرتبطة بمتغيرة (السطر 21) فان **د3** تصبح مرتبطة بنفس المتغيرة، ونلاحظ في خطأ: لم يُعثر على مصدر المرجع اننا لم نستعمل الرمز & الذي يستعمل للتصريح بالصلات، فحسب القواعد الجديدة، كل ما يصرح من اسم مرادف لنمط مهيكل هو صلة، اما في السطر 30 و 31 نرى استعمال الرمز = مع متغيرات (في الحقيقة مع صلات المتغيرات) من النمط مستطيل، وهنا لن يسلك الرمز = السلوك الأصلي، بل السلوك الذي عرف في السطر 12، و اذا امعنا النظر في الوظيفة التي تفسر سلوك الرمز = (السطر 12)، نراها تشترط ان يكون المدخلان مرتبطان بمتغيرة، ونلاحظ هنا ايضا عدم استعمال الرمز &، في السطر 30 تنفذ عملية النسخ بنجاح لكون المتغيرتين المرتبطتين بالصلتين **مس3** و **مس2** موجودتين (السطر 19 و السطر 26)، اما بالنسبة للسطر 31، فالكتابة صحيحة شكلا لكنها غير صحيحة في المعنى و لن تنفذ لأن الصلة **س4** حرة، اي انها غير مرتبطة بأي متغيرة ضرورية لنجاح تنفيذ عملية النسخ التي عرفت في السطر 11.



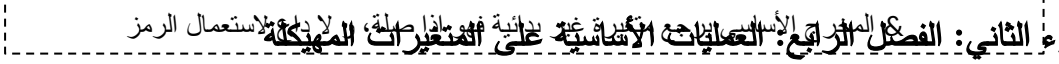
الجزء الثاني: الفصل الرابع: العمليات الأساسية على المتغيرات المهيكلة

وكنتيجة لهذه القوانين:

- في ظل القواعد الجديدة، القولان التاليان متساويان: **متغيرة مهيكلة طارئة** و**متغيرة مهيكلة،** فقولنا متغيرة مهيكلة معناه الحقيقي **متغيرة مهيكلة طارئة** لكون كل صلة تمثل في حقيقتها متغيرة غير بدائية، اي متغيرة مهيكلة وما سواها من المتغيرات الغير البدائية التي سوف نراه في الفصول التالية.
- يمكن ان نقول: متغيرة **كذا** من **نمط كذا**، ونذكر اسم المتغيرة (**كذا** الأول) و اسم النمط(**كذا** الثاني)، فإن كان النمط مهيكل فالمتغيرة مهيكلة والاسم (**كذا** الأول) هو اسم صلة، وان كان النمط اساسي فالمتغيرة اساسية (او بدائية) والاسم هو اسم المتغيرة.
- كلما اردنا ان نَذكر في الكتابة مُتغيرة مُهيكلة فإننا نَذكر في الحقيقة **الصلة** التي من وراءها المتغيرة المهيكلة الطارئة:
  - فعندما يبدو ان احد معاملي العملية = هو متغيرة مهيكلة، ففي الحقيقة، هذا المعامل صلة لمتغيرة مهيكلة، واذا لم يُعاد تعريف سلوك العملية = بالنسبة لنمط معاملها الأيمن ولنمط معاملها الأيسر (كما هو الشأن بالنسبة للنمط مستطيل في خطأ: لم يُعثر على مصدر المرجع)، فإن سلوكها يكون السلوك الأساسي الذي فيه تقوم بنسخ الصلة اليمنى بما في الصلة اليسرى (كما هو الشأن بالنسبة للنمط **دائرة** في خطأ: لم يُعثر على مصدر المرجع).
  - اذا ظهر اسم لمتغيرة من نمط غير اساسي (نمط مهيكل او غيره من الأنماط المعقدة التي سوف نراها في الفصول القادمة) في منفذ من منافذ اجراء او وظيفة ، فالاسم، هو في حقيقة الأمر، اسم **لصلة**، وهذا هو شأن الوظائف والإجراءات في خطأ: لم يُعثر على مصدر المرجع.

 & المدخلين رس1 و رس2 يطلبان متغيرة غير بدائية، فهما اذا صلتين و لا داع لاستعمال الرمز

رسم عملية= (رسم **رس1** ، **رس2**)	**رسم** انسخ_رسم (رسم **رس1** ، **رس2**)



الشكل 16:اعادة كتابة رأس وظيفة نسخ متغيرتين من النمط رسم استنادا الى قواعد كتابة الخوارزميات باللغة الشرمزية

- في اطار متغيرة مهيكلة، تصبح كل السمات التي تنتمي الى نمط مهيكل سمات طارئة، لكوننا، حسب القواعد الجديدة، لا نستعمل المتغيرات المهيكلة العادية (الغير الطارئة)، فكل المتغيرات المهيكلة وغيرها من المتغيرات التي ليست بدائية، متغيرات طارئة تُنشأ في الذاكرة النشطة.

و هكذا، انطلاقا من هذه الأسس الجديدة، يصبح رأس وظيفة نسخ المتغيرات من النمط **رسم** الذي يظهر في خطأ: لم يُعثر على مصدر المرجع على الصفة الجديدة الظاهرة في خطأ: لم يُعثر على مصدر. المرجع.

9 – مقارنة في انجاز خوارزم تهيئة متغيرة مهيكلة بين صيغة الإجراء و صيغة الوظيفة:

مع القواعد الجديدة التي تُمكن من جعل عملية ادخال واخراج المعطيات عبر المنافذ عملية سريعة غير مرتبطة بحجم المتغيرات (أي منفذ من نمط غير بدائي يعتبر **صلة**) ، يتبادر الى الذهن عند الشروع في انجاز خوارزم ما، التساؤل التالي: ما هي الصيغة الأفضل لإنجاز خوارزما ما: اهي صيغة الإجراء ام صيغة الوظيفة، ولكي نلمس الفرق بينهما حتى يتسنى لنا اختيار الصيغة الأنسب لمتطلباتنا، نعيد فيما يلي كتابة الإجراء **هيئ_مستطيل** (خطأ: لم يُعثر على مصدر المرجع) و نرى ما يمكن فعله من كتابة مع كل صيغة.

الوظيفة **مستطيل** (خطأ: لم يُعثر على مصدر المرجع) لها نفس المداخل، الا انها تتميز بمنفذ اضافي الا وهو **منفذ الرجوع**، وتخرج عبر هذا المنفذ صلة في متغيرة ظرفية، وقيمة محتوى الصلة الظرفية هي نفس قيمة الصلة في المدخل الأول، اي **م1** (السطر 5)، ونلاحظ عدم استعمال الرمز & في المدخل **م1** و في مخرج الرجوع (السطر 2) عند تعريف رأس الوظيفة، وكأن الوظيفة لا تعلمنا شيء جديد لأن ما يوضع من صلة في المدخل **م1** يبدو معلوم عند واضع الخوارزم، وهذا في الحقيقة غير صحيح، اذ يمكن لكاتب الخوارزم ان يضع في المدخل صلة لا يعرف اسمها، وهذه الحالة تحدث عند تقييم العبارات التي تضع نتائجها في متغيرات ظرفية، وفيما يخصنا العبارة الشائعة التي ترجع

صلة ظرفية هي العبارة المكونة من طلب تنفيذ العملية **شيد،** وهذا الأمر هو الذي يختلف فيه نوعا ما الإجراء عن الوظيفة (زيادة عن الاختلاف الذي رأيناه من قبل والذي يُمَكن للوظيفة ان تظهر عدة مرات في عبارة ما بينما لا يمكن بناء عبارات باستعمال الإجراءات)، ففي الوظيفة يمكن اعطاء المدخل عبارة تنتج صلة ظرفية كما هو الشأن في السطرين 11 و 12 من خطأ: لم يُعثر على مصدر المرجع ، وهذه الصلة الظرفية سوف تعيدها الوظيفة الى من طلب تنفيذها (السطر 5 من خطأ: لم يُعثر على مصدر المرجع) ، اما الإجراء فلا يمكن استعمال هذه الطريقة معه لأنه سوف يضيع الصلة، والطريقة المشابهة في الإجراء هو اعطاء المدخل عبارة تقوم في آخر تقييمها بحفظ الصلة الظرفية في المتغيرة التي نريد ان نلتقط فيها النتيجة كما يظهر. في السطرين 10 و 11من خطأ: لم يُعثر على مصدر. المرجع.

النص 19: استعمال وظيفة بدل استعمال اجراء لتهيئة متغيرة مهيكلة

النص 20: طريقة عدم ضياع الصلة الظرفية عند اجراء لتهيئة متغيرة مهيكلة

الفصل الخامس

الخولرزميات المنتسبة للأنماط المهيكلة

1 – التعريف:

الخوارزميات (وظائف او اجراءات) المنتسبة لنمط مهيكل، او **الخوارزميات المنتسبة**، هي خوارزميات **تُعَرَّف في سياق تعريف نمط مهيكل**، فهي مكون من مكونات النمط المهيكل، وتنفذ الخوارزميات المنتسبة في اطار المتغيرات المهيكلة، ولا يمكن طلب تنفيذها خارج المتغيرات المهيكلة، وبفضل الخوارزميات المنتسبة تبدو المتغيرات المهيكلة وكأنها تتمتع ببعض الذكاء تجعلها تستقبل طلب تنفيذ خوارزم منتسب لها وترد بما ينتجه الخوارزم المنتسب لها، وطريقة تعريف خوارزم منتسب (اجراء او وظيفة) هي نفسها طريقة تعريف اي خوارزم، والفرق ان التعريف يكون داخل الفقرة التي فيها يعرف النمط المهيكل كما يظهر ذلك في خطأ: لم يُعثر على مصدر المرجع، ففيه يعرف النمط **مستطيل** الذي يحتوي على الإجراء المنتسب **هيئ**.



النص 1 : كيفية تعريف الإجراء المنتسب هيئ المتخصص في تهيئة المتغيرات من النمط مستطيل

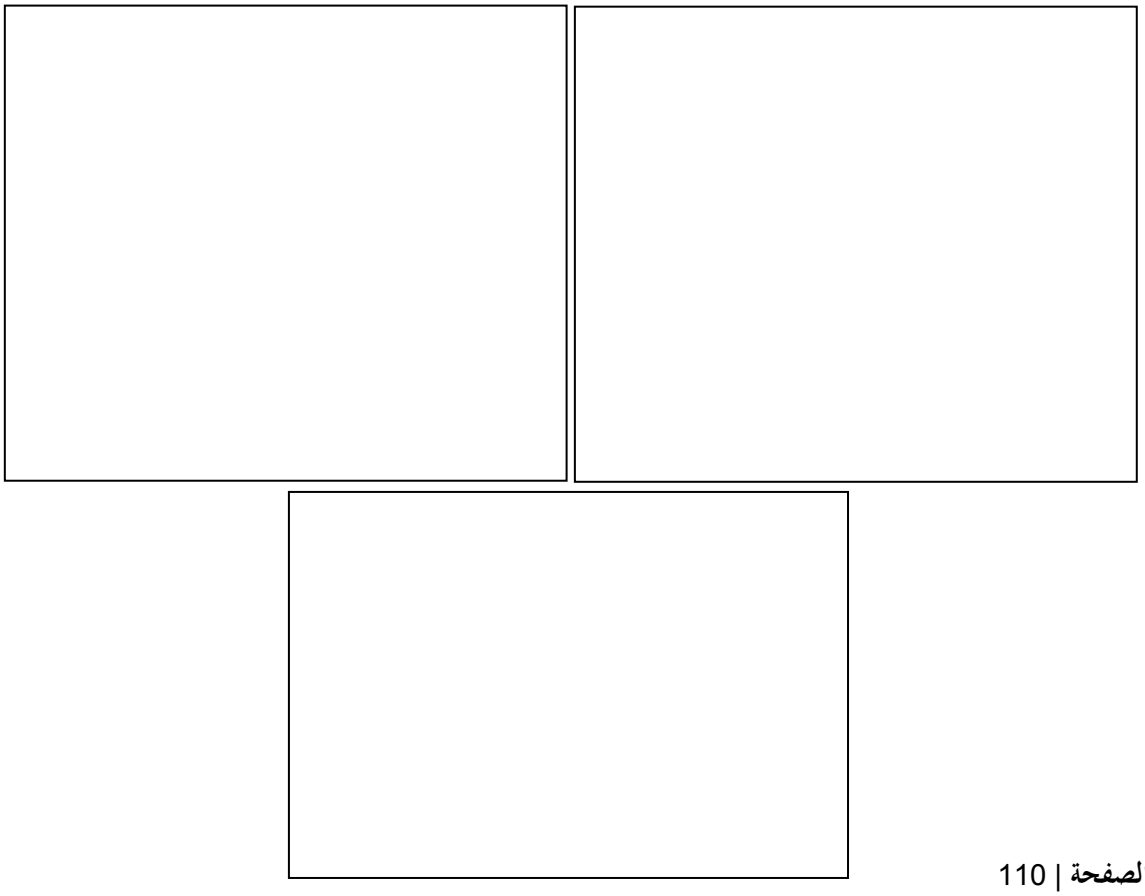
1 – 2 اهمية الخوارزميات المنتسبة في التمثيل الدقيق للواقع المعاش

بفضل الإجراءات والوظائف المنتسبة، يسهل تمثيل الواقع المعاش بصفة ادق واكمل مما يكون عليه الوضع لو استعملنا فقط السمات في تعريف نمط مهيكل، ففي الواقع الذي يستهدفه خوارزما ما، نجد ان اكثر المفاهيم والكائنات التي نريد تمثيلها لها سمات، ولها ايضا سلوكا ودورا فعالا تلعبه في محيطها، فعلى سبيل المثال، اذا اردنا ان نمثل الأشخاص في واقع ما، نعرف النمط **شخص**، وهذا النمط سوف يحتوي على سمات عديدة من بينها **الاسم، اللقب، الطول، الوزن، تاريخ الميلاد، الحالة المدنية** الخ، وهذه السمات لا تمثل الا **الجانب الجامد** في الشخص، فالشخص له كذلك جانب حيوي، نسميه السمات الحيوية ومن بينها: فالشخص يمشي، يجري، يرفع يده، يقبض يده، يضرب برجله هدفا كالكرة، يقود سيارة، يركب حصانا، الخ.. وإذا افترضنا ان الواقع المعاش هو سباق للسيارات والخوارزم هدفه محاكاة سباق السيارات، فالنمط **سيارة** يمثل السيارات، وبنفس التفكير يمكن ان نقول ايضا ان للسيارة سمات جامدة وسمات حيوية، فمن بين السمات الجامدة نجد نوعية السيارة، رقم السيارة، الحجم، اللون، عدد الأبواب، حجم خزان الوقود الخ...، اما السمات الحيوية فهي تشغيل المحرك، تبديل السرعة، التقدم بالسرعة المختارة، الرجوع الى الوراء، الدوران الخ...

1 – 3 استغلال السمات من قبل الخوارزميات المنتسبة

بما ان الخوارزم المنتسب قد عرف في اطار نمط مهيكل، وبما انه يمثل **سمة حيوية** من سمات النمط المهيكل، فعندما ينفذ في اطار متغيرة مهيكلة، لا يحتاج الى عملية الولوج للوصول الى سمات المتغيرة التي ينفذ في اطارها، فهو اصلا في داخل مجال المتغيرة، ولذا يستغل السمة مباشرة باستعمال اسمها دون استعمال اسم المتغيرة التي تحتوي السمة، ونلاحظ هذا الأمر في خطأ: لم يُعثر على مصدر. المرجع، فمثلا على مستوى النمط مستطيل، نرى ان الخوارزميات المنتسبة **هيئ** و**محيط** و**مساحة** تستعمل مباشرة اسماء السمات **س، ج، طول** و **عرض** عندما تحاول ان تعرف محتواها. او تحاول تغييرها ، فالخوارزم المنتسب يُنَفذ في اطار متغيرة ما، فهو سمة من سمات المتغيرة، وعندما ينفذ فهو مباشرة امام كل السمات الأخرى التي هي من نفس مستواه، و لهذا يستعمل مباشرة اسم السمات.

ونشير. هنا الى ان الخوارزم المنتسب عندما يُعرف في سياق تعريف نمط ما، لا يمكنه معرفة اسم المتغيرة التي سينفذ في اطارها، فالنمط ومحتواه عُرِفا قبل انشاء اي متغيرة، ولهذا لا يمكن في الخوارزم المنتسب ان نكتب اسم المتغيرة التي ينفذ في اطارها. متبوعا بعملية الولوج، وزيادة على هذا يمكن ان تكون عديدة تلك المتغيرات التي يمكن ان يُنفذ من خلالها خوارزما منتسبا، واذا تأملنا في خطأ: لم يُعثر على مصدر المرجع، نرى جليا ان اسم المتغيرات **م1** و **م2** التي عرفت في مجال الإجراء اساسي لم تكن متوفرة عند كتابة الخوارزميات المنتسبة للنمط **مستطيل**.

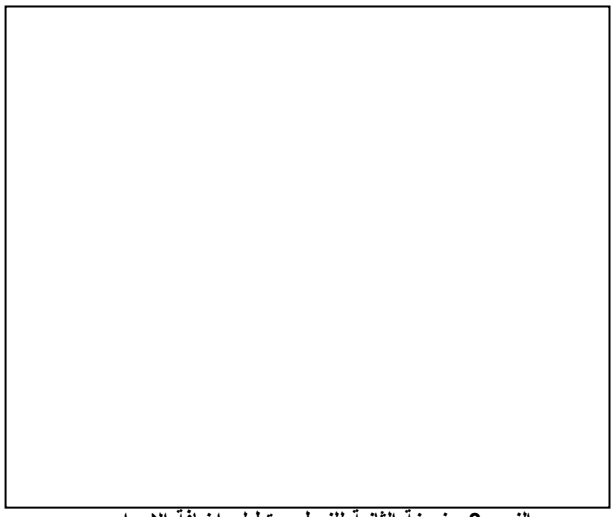


النص 2 : انتظام خوارزميات النص كل في مجالها

1 – 4 كيفية تشغيل الخوارزميات المنتسبة:

هي نفسها الكيفية التي تستعمل للوصول الى السمات، اي باستعمال عملية الولوج الممثلة بالنقطة (.)، فاذا اراد خوارزما ما (اجراء او وظيفة)، نسميه **خ1**، ان يطلب تنفيذ خوارزم منتسب نسميه **خ2**، يكون خ1 في احدى الحالتين التاليتين:

- **خ1** خوارزم منتسب لنفس النمط، اي انه عُرف مع **خ2** في نفس النمط، وفي هذه الحالة يستعمل **خ1** اسم **خ2** ولا يحتاج لعملية الولوج، فهما في نقس المستوى بالنسبة للمتغيرة التي يُنَفذا من خلالها وهذا ما يتبين من خلال خطأ: لم يُعثر على مصدر المرجع، ففيه نرى الإجراء **حجم** وكيفية وصوله الى الإجراء **هيئ** (السطر 15).



النص 3 : نسخة الثانية للنمط مستطيل: اضافة الإجراء حجم

- **خ1** خوارزم غير منتسب للنمط الذي ينتسب اليه **خ2**، اي ان **خ1** يكون غير منتسب لأي نمط او يكون منتسبا لنمط آخر غير النمط خ1، وفي هذه الحالة يجب ان تتوفر للخوارزم **خ1** صلة لمتغيرة مهيكلة من نفس نمط **خ2** يمكن بواسطتها طلب تنفيذ **خ2**، فان توفرت الصلة والمتغيرة المهيكلة، يستعملها **خ1** متبوعة بعملية الولوج ثم بالاسم **خ2**، وهذا ما نراه في خطأ: لم يُعثر على مصدر المرجع على مستوى الإجراء اساسي، فهذا الإجراء لا ينتسب لأي نمط، وعندما يريد الوصول الى الإجراء **هيئ** المنتسب للنمط مستطيل، يستعمل اسم المتغيرة **م1** متبوعة برمز عملية الولوج ثم اسم الإجراء **هيئ**، ونفس الطريقة تتم في خطأ: لم يُعثر على مصدر المرجع مع المتغيرات **م2**، **د1** و **د2**، ونذكر هنا ما يلي:

  - عندما ينفذ الإجراء **هيئ** على المستطيل **م1**، يتعامل الإجراء **هيئ** مع السمات **س**، **ج**، **طول** و **عرض** التابعة للمتغيرة **م1**.

- وعندما ينفذ الإجراء هيئ على المستطيل **م2**، يتعامل الإجراء **هيئ** مع السمات **س، ج، طول** و **عرض** التابعة للمتغيرة **م2**.
- عندما ينفذ الإجراء هيئ على الدائرة **د1**، يتعامل الإجراء هيئ مع السمات **س، ج،** و **شعاع** التابعة للمتغيرة **د1**.
- عندما ينفذ الإجراء هيئ على الدائرة **م2**، يتعامل الإجراء هيئ مع السمات **س، ج** و **شعاع** التابعة للمتغيرة **د2**.

2 – الإجراءات المنتسبة اساس للتحكم الفعال في تنظيم نص الخوارزم

في حقيقة الأمر، لو تأملنا قليلا في الواقع المعاش الذي يستهدفه خوارزما ما، نجد انه بإمكاننا اعطاء كل شيء (مفهوم او كائن) دورا في الخوارزم، وبهذا يكون لكل نمط، يمثل شيئا ما (مفهوما او كائنا)، سمات جامدة وسمات حيوية تبين ما يمكن ان يفعله الشيء في الخوارزم، فعلى سبيل المثال، في الفصول السابقة تعلمنا كيفية انجاز خوارزما. هدفه تهيئة متغيرة من النمط مستطيل، وطبعا كان خوارزم التهيئة غير منتسب، اما الآن كما يظهر في خطأ: لم يُعثر على مصدر المرجع، فقد جعلنا تهيئة متغيرة من النمط **مستطيل** إجراءا منتسبا للنمط **مستطيل**، وللوصول الى مثل هذه الحالة، اي جعل خوارزم التهيئة خوارزما منتسبا، اتبعنا التفكير البسيط والمهم التالي: تهيئة متغيرة من النمط **مستطيل** امر يستهدف فقط المتغيرات من النمط **مستطيل**، بل ينفرد النمط **مستطيل** بهذا الإجراء، ولو جعلنا الإجراء غير منتسب للنمط **مستطيل**، يبق الأمر على حاله، اي ان الإجراء لن يستهدف ابدا المتغيرات من الأنماط الأخرى، اذا، ما الجدوى من جعل الإجراء عاما غير مرتبط بمجاله الخاص اي النمط **مستطيل**، فالموقع الطبيعي للإجراء **هيئ** الذي يهيئ المستطيلات، هو النمط **مستطيل**.

واذا عممنا مثل هذا التفكير، سوف نجد التعريف بمجموعة من الانماط في نص خوارزم، كل نمط يحتوي على الخوارزميات الخاصة به، اي المنتسبة له، وتبق الخوارزميات التي لا تنتسب لأي نمط في المجال العام وهو المجال خارج فقرات تعريف الأنماط، وبهذا نتحصل على نص تنتظم فيه الخوارزميات، كل واحد في مجاله، وهذا ما نراه جليا في خطأ: لم يُعثر على مصدر المرجع الذي يحتوي على ثلاث اجزاء متباينة:

- الجزء الأول وهدفه التعريف بكل ما يخص النمط **مستطيل**.
- الجزء الثاني وهدفه التعريف بكل ما يخص النمط **دائرة**.
- والجزء الثالث يحتوي على خوارزميات غير مرتبطة بنمط ما، كالإجراء **اساسي**.

ونتيجة لهذه الهيكلة الجيدة، نرى امكانية استعمال نفس الرأس (نفس الاسم وحتى نفس المنافذ) للخوارزميات المنتسبة ان كانت تنتمي لأنماط مختلفة، فكل خوارزم يبق سارٍ وفعال في المحال الذي يحدده النمط، اما على مستوى نفس النمط فشكل الرأس (الاسم و المنافذ) يجب ان يكون وحيدا، وهكذا، فالوظيفة **محيط()** التابعة للنمط **مستطيل** مختلفة تماما عن الوظيفة **محيط()** التابعة للنمط دائرة، فالأولى لا يمكن تشغيلها الا في اطار متغيرة من النمط **مستطيل** والثانية لا يمكن تشغيلها الا في اطار متغيرة من النمط **دائرة**.

تذكير هام:

- الأسماء **م1، م2، د1** و **د2** الواردة في السطر 36 و 37 من خطأ: لم يُعثر على مصدر المرجع هي اسماء صِلات، وطبقا للقواعد الجديدة التي عرفت في الفضل 4، فلا نحتاج لاستعمال الرمز &، فكل اسم مرادف لنمط مهيكل هو اسم لصلة ملازمة للنمط.

- لا يمكن كتابة طلب تنفيذ خوارزما منتسبا دون ذكر متغيرة تابعة لنفس النمط اذا كان طلب التنفيذ صادر من خارج النمط الذي عرف فيه الخوارزم المنتسب.

3 – قواعد مستحسنة لاختيار اسماء مداخل الخوارزميات المنتسبة:

في العموم، عندما نُعَرف خوارزما منتسبا لنمط ما، نستعمل المداخل لجلب القيم الجديدة لسمات المتغيرة المهيكلة التي ينفذ الخوارزم عليها، وهنا يجب علينا ان نحذر في تسمية المداخل، فلا نعطيها اسماء السمات بل نعطيها اسماء مختلفة عن اسماء السمات، وعموما لجعل الكتابة تحمل معنى، نعطي للمدخل اسما مشابها لاسم النمط كما يظهر ذلك في الخوارزميات المنتسبة للنمط مستطيل، فمثلا في الإجراء المنتسب **هيئ** (خطأ: لم يُعثر على مصدر المرجع)، استعملنا الأسماء **س1، ج1، طول1** و **عرض1** في المداخل، ومثل هذه الكتابة توحي ان المدخل **س1** يحمل قيمة جديدة للسمة **س** و **ج1** يحمل قيمة للسمة **ج** الخ...

هناك طريقة اخرى للتأكد من اننا نستعمل السمة او متغيرة محلية، وهذه الطريقة تستهدف شكل اسماء السمات، فكل ما هو سمة يجب ان يبدأ اسمه بحرف التسطير(_)، وهكذا اذا كتبنا اسما يبدأ بحرف التسطير فمعنى ذلك اننا نتعامل مع سمة ما(خطأ: لم يُعثر على مصدر المرجع–أ).

الصف	(أ)	(ب)

النص 4 : (أ) قاعدة خاصة باسماء السمات (ب) عواقب الخطأ في اسماء المداخل

3 – 1 اعادة استعمال اسماء السمات في منافذ الخوارزميات المنتسبة:

عند كتابة خوارزما ما، يحدث ان يخطأ الكاتب، فبدل ان يضع في مدخل ما اسما مختلفا ومشابها لاسم سمة ما، يضع نفس الاسم الذي خصص لسمة ما، كما يظهر ذلك في خطأ: لم يُعثر على مصدر المرجع(ب) على مستوى الإجراء **هيئ** (السطر 4)، ونتيجة للخطأ تبرز كتابات غريبة (من السطر 5 الى 8) مفادها نسخ متغيرة في نفسها، وهذه كتابات لا جدوى منها، ومع مثل هذه الكتابات نتساءل : من هي المتغيرات المستهدفة حقيقة من عمليات النسخ؟ اهي المتغيرات الممثلة للسمات؟ ام هي المتغيرات المحلية الممثلة للمداخل؟ في مثل هذه الحالة، الأسبقية تكون دائما للمتغيرات المحلية، فالمتغيرات الممثلة للسمات لا تمس، فتبق على حالها.

في بعض الأحيان، استعمال اسماء السمات في المنافذ لا يكون عن طريق الخطأ، بل هو ما يريده الكاتب، وعلمنا ان مثل هذا التصرف لا يكون له اي جدوى (خطأ: لم يُعثر على مصدر المرجع–ب)، بل هو مصدر هام لظهور عيوب في الخوارزم، ونتساءل هنا: هل يمكن حقا استيعاب مثل هذا التصرف في اللغة الشرمزية؟

الجواب على هذا السؤال هو نعم، فاللغة الشرمزية كما هو الحال في كثير من لغات البرمجة كلغة "**جافا**" ولغة "**سي++**"، تمكن من مثل هذه الكتابة مع اضافة بسيطة تمكن من التفريق، في جسم الخوارزم المنتسب، بين المتغيرة المحلية للخوارزم المنتسب وبين السمة، والتفريق بين ما هو محلي وبين ما هو سمة، هو الإشكال الكبير الذي نريد حله اذا اردنا اعطاء المداخل نفس اسماء السمات.

لتبيان كيفية التفريق بين ما هو محلي وبين ما هو سمة نعرج على المثال التالي، الذي لا نستخدم فيه آلية اللغة الشرمزية للتفريق بين ما هو سمة وبين ما هو محلي(اي المتغيرات المحلية و منها المداخل)، وعدم استعمالنا لهذه الآلية الآن هو في حقيقة الأمر بمثابة مدخل لفهمها.

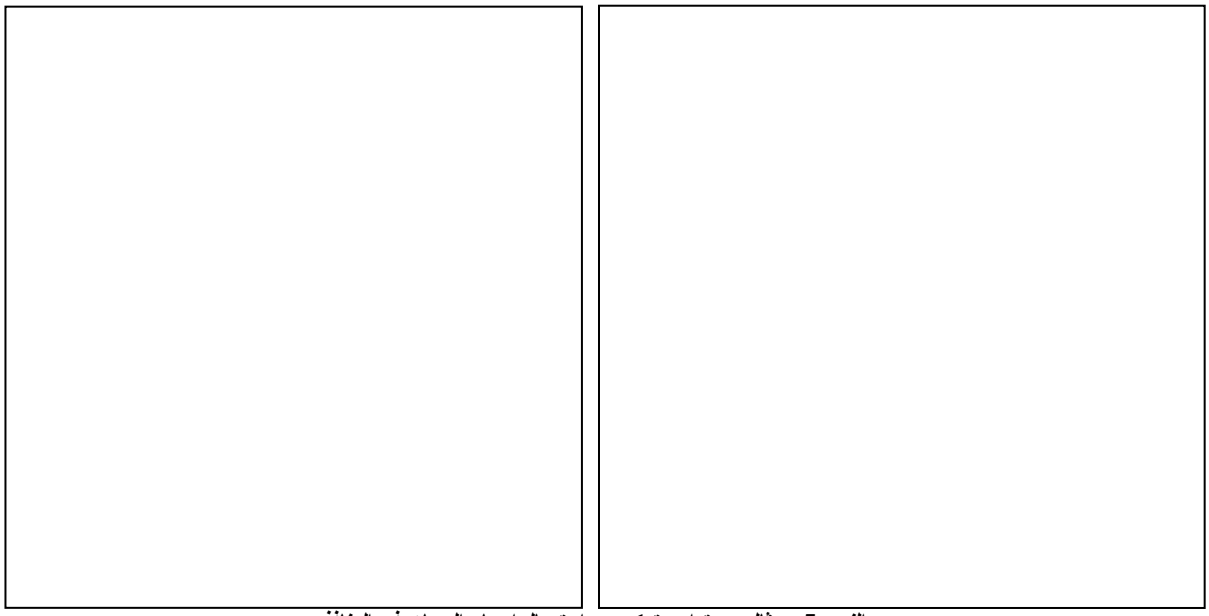
3 – 2 كتابات إضافية لجعل الخوارزم يفرق بين المتغيرات المحلية والسمات الحاملة لنفس الإسم:

في هذا المثال، ومن اجل تمكيننا من اعطاء المداخل نفس اسماء السمات، علينا ان نضيف كتابات هدفها التمكين من التفرقة بين ما هو محلي وبين ما هو سمة، وفي هذا الإطار نزيد ما يلي:

- ندخل نمطا خاصا هدفه وصف ما يجري في المحيط، وفيما يخصنا، يخبرنا المحيط عن المتغيرة المهيكلة التي يتعامل معها الخوارزم في اي وقت، وسمينا هذا النمط "**المحيط**" (السطر 22 من خطأ: لم يُعثر على مصدر المرجع) ، وفيما يخصنا في هذا المثال، فان المحيط يتابع المتغيرات المهيكلة من النمط **مستطيل**(السطر 23 من خطأ: لم يُعثر على مصدر المرجع)، ويمكننا اثراؤه اذا اردنا ان نتابع متغيرات من انماط اخرى.
- انشاء متغيرة عامة من النوع **محيط** (السطر 23 من خطأ: لم يُعثر على مصدر المرجع)
- اضافة سمة من النوع **محيط** الى النمط مستطيل، وسمينا هذه السمة "**انا_المتغيرة_الحالية**" (السطر 2 من خطأ: لم يُعثر على مصدر المرجع) وبفضل هذه السمة تستطيع اي متغيرة من النمط مستطيل ان تعرف صلتها الحقيقية اثناء التنفيذ، وهذه الصلة هي:

"**انا_المتغيرة_الحالية.مس**"، ومعنى هذا انه يمكن لنا الوصول الى سمات متغيرة من النمط مستطيل باستعمال السمة "**انا_المتغيرة_الحالية.مس**" فنكتب مثلا

"**انا_المتغيرة_الحالية.مس.س**" للوصول الى السمة **س** من المتغيرة من النمط **مستطيل،** وللوصول الى السمات الأخرى نكتب: "**انا_المتغيرة_الحالية.مس.ج**" بالنسبة للسمة **ج،** "**انا_المتغيرة_الحالية.مس.طول** و "**انا_المتغيرة_الحالية.مس.عرض**" بالنسبة للسمتين **طول** و **عرض** (من السطر 7 الى السطر 10 من خطأ: لم يُعثر على مصدر المرجع)، ونرى انه بفضل المتغيرة المتابعة للمحيط استطعنا في الإجراء **هيئ** ان نستعمل في مداخلها نفس اسماء السمات وفي جسد الإجراء فرقنا بين السمات والمداخل باستعمال المتغيرة المتابعة للمحيط.



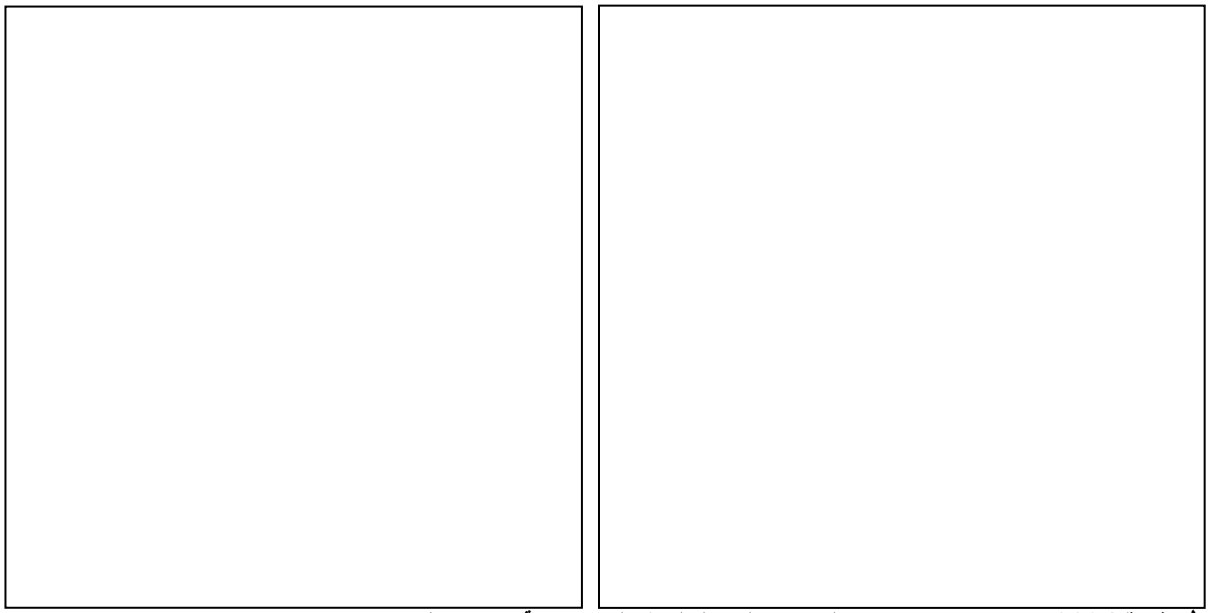
النص 5 : مثال عن تدابير تمكن من استعمال اسماء السمات في المنافذ

- على مستوى الخوارزم الذي تكتب فيه تعليمة طلب تنفيذ الخوارزم المنتسب، وقبل تعليمة طلب تنفيذ الخوارزم المنتسب على متغيرة ما:
  - نخبر المحيط بصلة المتغيرة المهيكلة الحالية، وهذا ما حدث في السطر 33 بالنسبة للمتغيرة **م1** و في السطر 37 بالنسبة للمتغيرة **م2.**
  - نخبر المتغيرة المستهدفة، اي **م1** (السطر 34) و **م2** (السطر 34)، بمتغيرة المحيط لتصل اليها من اجل استخراج صلتها الذاتية من المحيط.

3 – 3 كيف نمكن النمط مستطيل من الوصول مباشرة الى متغيرة عامة:

في خطأ: لم يُعثر على مصدر المرجع وعلى مستوى النمط مستطيل نرى السمة "**انا_المتغيرة_الحالية**" وهي من النمط "**المحيط**"، وبديهي ان هذه السمة ليست في حقيقة الأمر من سمات المستطيل، ومثل هذه السمات الغريبة على متغيرة مهيكلة غير مستحسنة، فالسمة "**انا_المتغيرة_الحالية**" غريبة عن سمات المستطيل، والأفضل ان نتخلص منها في كتابة النمط **مستطيل**، لكن اذا فعلنا هذا فلن نتمكن من اعادة استعمال اسماء السمات كأسماء لمداخل الخوارزميات المنتسبة، والسؤال الذي نطرحه هنا: **الا توجد طريقة من جهة لتخليص النمط مستطيل من السمة الغريبة وفي نفس الوقت نمكن النمط المستطيل من الوصول الى المتغيرة العامة الممثلة للمحيط؟**

الجواب هو نعم، فاللغة الشرمزية تمكن فعلا من الوصول من خارج الخوارزم الى متغيرة عامة تم التصريح بها في الخوارزم، ويتم الوصول الى هذه المتغيرات العامة باستعمال عملية الولوج في الخوارزم ذاته، فمثلا للوصول الى المتغيرة **المتغيرة_الحالية** التي صرح بها في الخوارزم **المثال_3_الفصل5**، نكتب اسم الخوارزم متبوع بعملية الولوج وهي النقطة، ثم اسم المتغيرة العامة كما يظهر في الأسطر من 6 الى 9 من خطأ: لم يُعثر على مصدر المرجع.



3 – 4 الصلة الخاصة "انا":

التدابير السابقة التي تُمَكن من استعمال اسماء السمات في منافذ الخوارزميات المنتسبة فعالة، لكن اتباعها امر غير يسير، مقلق ومصدر كبير لوقوع الأخطاء، وما رأيناه ينطبق فقط على النمط مستطيل، واذا اردنا تعميم التدابير علينا ان نعيد ما فعلناه مع كل نمط مهيكل يحتوي على خوارزميات منتسبة، وهذا ما يزيد في صعوبة اعتماد مثل هذه التدابير الثقيلة والمكلفة من اجل اشباع رغبة خفيفة لكنها مهمة بالنسبة لكاتب الخوارزم وحتى القارئ.

في بعض لغات البرمجة كلغة "**جافا**" و "**سي++**"، نرى ان هذه الرغبات في كتابة البرامج (اي جعل اسماء المداخل كأسماء السمات) متوفرة ولا يحتاج كاتب الخوارزم الا لإضافة طفيفة في الكتابة تمكن الخوارزم المنتسب من التفريق الواضح بين المتغيرة المحلية الممثلة للمدخل والمتغيرة الممثلة للسمة.

الإضافة الطفيفة تكمن في استغلال متغيرة عامة تلعب دور الصلة "**المتغيرة_الحالية.مس**" (خطأ: لم يُعثر على مصدر المرجع و خطأ: لم يُعثر على مصدر المرجع)، ويقوم منفذ البرنامج بتوفيرها آليا وبتسديدها، وهذه المتغيرة العامة متاحة لكل الوظائف والإجراءات، وهي صلة تحمل اسما خاصا، وتحتوي على معلومات تواجد المتغيرة المهيكلة التي ينفذ من خلالها خوارزما منتسبا، وفي اللغة الشرمزية تسمى "**انا**"[2] هذه الصلة العامة.

زيادة على ما سبق، تمتاز الصلة "**انا**" بما يلي:

- الصلة "**انا**" غير متربطة بأي نمط، فهي تسع كل الأنماط، وهذا عكس الصلة "**المتغيرة_الحالية.مس**" المرتبطة بالنمط مستطيل.
- التسديد الآلي لمحتواها من قبل منفذ اللغة الشرمزية، فهي تحتوي دائما على معلومات تواجد المتغيرة المهيكلة التي ينفذ من خلالها خوارزما منتسبا.

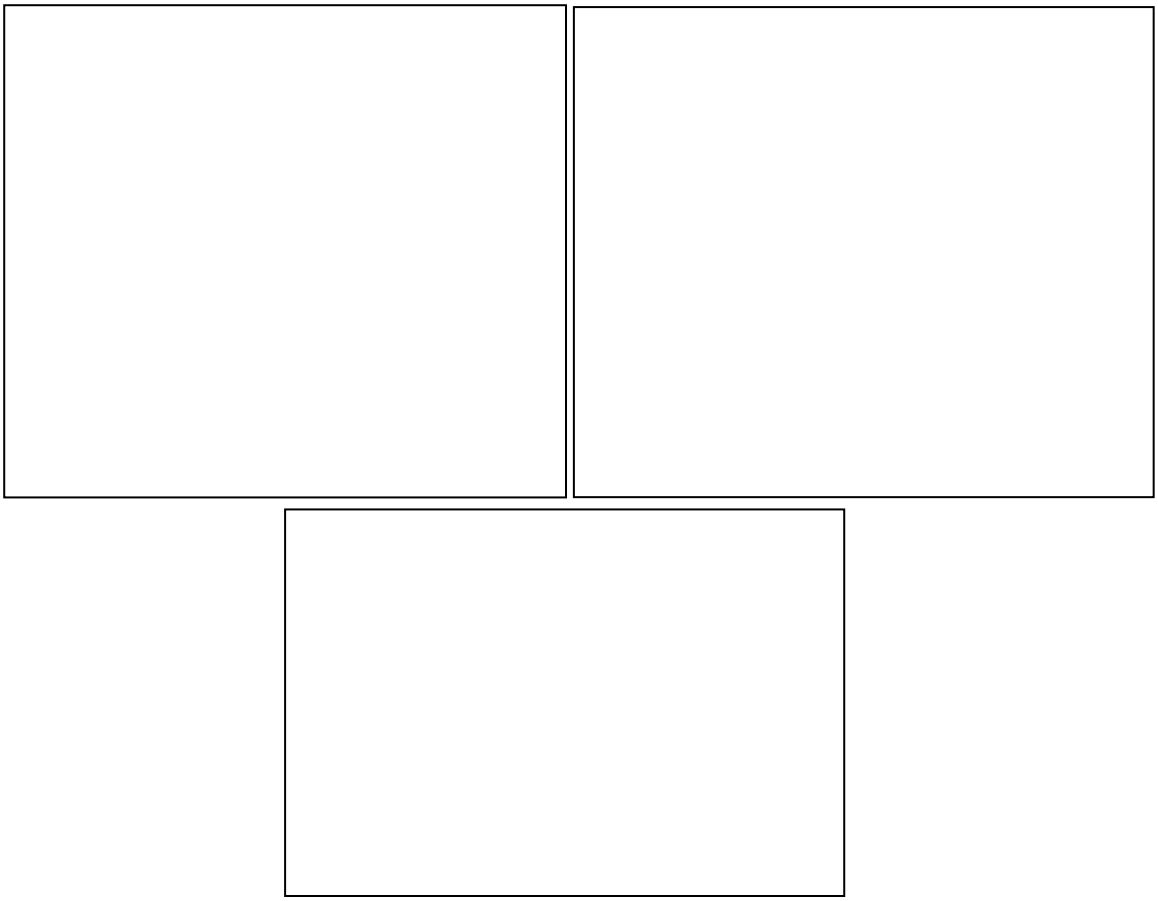
اذا اخذنا مثلا خطأ: لم يُعثر على مصدر المرجع وتأملنا حالة الصلة "**انا**" في مختلف مراحل تنفيذ الإجراء اساسي نرى ما يلي:

- عندما يكون التنفيذ في مستوى السطر 40 ، يكون محتوى الصلة "**انا**" هو محتوى الصلة **م 1**، اي **انا** تساوي **م1**، اي انه في هذه المرحلة يمكن استعمال الصلة **انا** للوصول الى

2 Keyword **this** in Java and C++

سمة من سمات المتغيرة المرتبطة بالصلة **م1** فنكتب مثلا "**انا.ج**"، في هذه المرحلة بالذات (كتابة الإجراء المنتسب) لا يعرف الإجراء المنتسب الاسم **م1**، لكنه يعرف الصلة العامة "**انا**".

- عندما ينتقل التنفيذ الى السطر 41، يكون محتوى الصلة "**انا**" هو محتوى الصلة **م2**، اي "**انا**" تساوي **م2**.
- عندما ينتقل التنفيذ الى السطر 43، يكون محتوى الصلة "**انا**" هو محتوى الصلة **د1**، اي "**انا**" تساوي **د1**.
- عندما ينتقل التنفيذ الى السطر 44، يكون محتوى الصلة "**انا**" هو محتوى الصلة **د2**، اي "**انا**" تساوي د2.



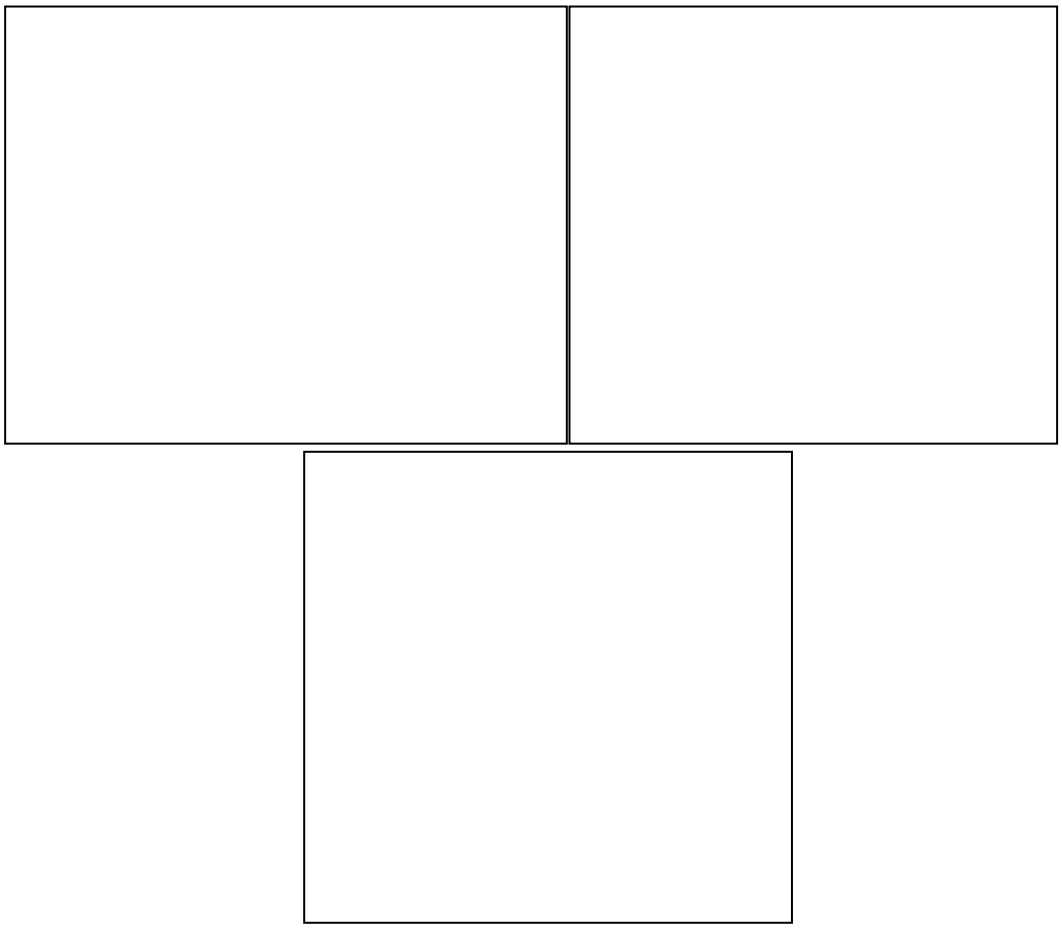
النص 7 : استعمال الصلة الخاصة انا

4 – المشيدات[3]:

هي إجراءات منتسبة **متخصصة فقط في تهيئة المتغيرات المهيكلة**، وخصوصية هذه الإجراءات تكمن فيما يلي:

3 constructors

- **اسم المشيد هو اسم النمط المهيكل** الذي يُعرف فيه، وهذا ما يبينه السطران 4 و 19 من خطأ: لم يُعثر على مصدر المرجع، ففي النمط **مستطيل** يجب ان يكون اسم المشيد هو **مستطيل**، وفي النمط **دائرة** يجب ان يكون اسم المشيد هو **دائرة**، ونلاحظ في كتابة رأس الإجراء المشيد غياب كلمة **إجراء**، فالمشيد لا يمكن ان يكون الا إجراءا، فلا حاجة لذكر الكلمة، وعدم استعمال كلمة اجراء جاءت ايضا لتمييزه عن باق الإجراءات المنتسبة.
- **ينفذ المشيد آليا** في سياق انشاء متغيرة مهيكلة، ولا يمكن استعماله بعد اتمام انشاء المتغيرة، فمهمة المشيد تنته عند انتهاء بناء متغيرة ما، وفيما يخصنا، وحسب قواعد كتابة الخوارزميات التي حددناها في الفصل الرابع، تنشأ المتغيرة المهيكلة، وهي في حقيقة الأمر متغيرة مهيكلة طارئة، باستعمال العملية **شيد**، ومعنى هذا ان العملية الوحيدة المخولة بطلب تنفيذ المشيد هي العملية **شيد**، وهذا ما نراه في السطرين 37 و 38 وفي السطرين 42 و43 من خطأ: لم يُعثر على مصدر المرجع.



النص 8 : استعمال المشيدات في صناعة و تهيئة متغيرات طارئة من النمطين مستطيل و دائرة

4 – 1 المشيد الافتراضي: نلاحظ في السطر 36 من خطأ: لم يُعثر على مصدر المرجع عدم استعمال اي من المشيدات التي صرحت، اذ تقوم عملية **شيد** بإنشاء متغيرة مهيكلة مجهولة المحتوى،

وهذا ما يحصل ايضا في السطر 41. في حقيقة الأمر، الكتابة **شيد (مستطيل)** تستعمل مشيدا خاصا (ليس ضروريا كتابته) توفره اللغة الشرمزية لكل نمط مهيكل، وهذا المشيد ليس له منافذ، والكتابة **شيد (مستطيل())** او **شيد مستطيل()** هي الكتابة الكاملة للكتابة **شيد (مستطيل)**.

في حالة ما اذا لم يوفر كاتب الخوارزم مشيدا دون منافذ، توفر اللغة الشرمزية المشيد الافتراضي على الشكل التالي: **اسم_النمط_المهيكل() { }**، اي ان المشيد الافتراضي اجراء فارغ لا يحتوي على اي تعليمة.

4 – 2 سلوك التعليمة شيد:

يتلخص سلوك العملية **شيد** فيما يلي:

- المرحلة الأولى: انشاء المتغيرة المهيكلة في الذاكرة النشطة، وفي هذه الحالة يكون محتوى كل سمة من سمات المتغيرة المهيكلة تحتوي على قيمة غير معلومة.
- المرحلة الثانية: تهيئ السمات التي صرحت بقيمة اولية.
- المرحلة الثالثة: ينفذ المشيد المذكور لها، وان ذكر فقط اسم النمط، يفهم ان المقصود هو المشيد الذي لا يحتوي على اي منفذ، فهو اما المشيد بدون منافذ الذي كتبه واضع الخوارزم او المشيد الافتراضي ان لم يصرح كاتب الخوارزم بأي مشيد دون منافذ.

في خطأ: لم يُعثر على مصدر المرجع، نرى ان كاتب الخوارزم لم يصرح بأي مشيد ليس له منافذ، ولهذا فان الكتابتين في السطرين 36 و 41 تطلب تنفيذ المشيد الافتراضي الذي توفره اللغة الشرمزية، وكما اشرنا اليه سلفا، فإن المشيد الافتراضي ليس له اي تعليمة، ومعنى هذا ان المتغيرتين التي انشأتا، اي **م1** و **د1**، لا تحتوي اي منهما على اي قيمة معلومة، اي ان المتغيرتين انشأتا ولم تهيء اي منهما، ولتهيئة هاتين المتغيرتين (**م1** و **د1**) نستعمل الإجراء المنتسب **هيئ** (السطرين 11 و 25 من خطأ: لم يُعثر على مصدر المرجع).

في السطرين37 و 38 نرى ان العملية شيد تطلب تنفيذ مشيد النمط **مستطيل** على المتغيرتين قيد الإنشاء المربوطتين بالصلتين **مط2** و **مط3**، وفي السطرين **42** و **43** فإنها تطلب تنفيذ مشيد النمط **دائرة** على المتغيرتين قيد الإنشاء المرتبطتين بالصلتين **د2** و **د3.**

ونلاحظ مما سبق ان تهيئة متغيرة مهيكلة عند الإنشاء باستعمال المشيد يكلف كتابة تعليمة واحدة وهي تعليمة الإنشاء، بينما التهيئة باستعمال الإجراء **هيئ** يكلف كتابة تعليمتين، الإنشاء ثم التهيئة.

4– 3 مقارنة بين المشيد و الخوارزميات المنتسبة العادية

الفرق الكبير بين المشيد واجراء منتسب عادي مثل الإجراء **هيئ** يكمن فيما يلي:

- المشيد يُذكر فقط لعملية **شيد،** وهي الوحيدة المخولة بطلب تنفيذه، ولا يمكن تنفيذه اكثر من مرة على متغيرة، وتنفيذه الوحيد على المتغيرة يكون فقط في سياق انشاء المتغيرة، وفيه تُهيئ المتغيرة.
- الخوارزميات امثال الإجراء **هيئ** لا يمكن تنفيذه الا بعد ان تكون المتغيرة قد انشأت تماما، كما يمكن للمشيد ان يطلب تنفيذها في سياق انشاء متغيرة مهيكلة، وعكس المشيدات، يمكن اعادة تنفيذ الخوارزميات المنتسبة كلما ظهرت الحاجة لذلك بعد اتمام انشاء المتغيرة.

3- 5 استغلال المشيد للخوارزميات المنتسبة العادية

يمكن للخوارزميات المنتسبة العادية امثال الإجراء **هيئ** ان تُنفذ في سياق انشاء المتغيرة، ويتم هذا الأمر من خلال **المشيد،** فمثلا، نلاحظ في خطأ: لم يُعثر على مصدر المرجع ان **المشيد** والإجراء **هيئ** لهما نفس التعليمات، وفي هذه الحالة يمكن اعادة كتابة محتوى المشيد ليحتوي فقط على تعليمة واحدة هي تعليمة طلب تنفيذ الإجراء **هيئ،** كما يظهر ذلك في السطرين 10 و24 من خطأ: لم يُعثر على مصدر المرجع، وخلاصة القول، انه استنادا الى ما اوردناه من خصائص المشيدات والخوارزميات العادية:

- يمكن لمشيد ان يستغل الخوارزميات المنتسبة العادية في اطار تشييد متغيرة مهيكلة.
- لا يمكن ابدا لخوارزم منتسب عادي ان يطلب تنفيذ المشيد مباشرة كطلبه تنفيذ خوارزم آخر.

النص 9 : استغلال الخوارزميات المنتسبة من قبل المشيدات

3 – 6 كيفية التصريح بالمتغيرات المهيكلة باستعمال المشيدات

تُظهر الأسطر 15، 16 و 20 من خطأ: لم يُعثر على مصدر المرجع، كيفية التصريح بالمتغيرات المهيكلة باستعمال المشيدات، ونُذَكر ان المتغيرات المهيكلة كلها متغيرات مهيكلة طارئة، وَجب استعمال العملية **شَيد** لإنشائها، فنرى من خلال هذه الأسطر ان التصريح بالمتغيرات المهيكلة باستعمال المشيدات يكون كالتالي:

- اولا ذكر اسم النمط
- ثانيا ذكر اسم المتغيرة، وهذا الاسم هو في حقيقة الأمر اسم صلة، ومجازا نقول اسم المتغيرة ونعني المتغيرة المتربطة بالصلة.
- ثالثا ذكر عملية الشحن = متبوعة بالكلمة **شيد**
- رابعا ذكر **اسم المشيد** متبوعا بمداخله، ونذكر في كل مدخل القيمة التي يحتاجها المشيد في عمله.

كما يمكن ان نصرح في مرحلة ما من الخوارزم بالصلة (السطرين 17 و 19 من خطأ: لم يُعثر على مصدر المرجع)، ثم بعد ذلك في نص الخوارزم نطلب انشاء متغيرة مهيكلة باستعمال العملية **شيد** متبوعة باسم **المشيد** ومداخله (الأسطر 21، 22 و 23).

4– تعدد المشيدات في نمط مهيكل واحد

يمكن لنمط مهيكل ان يحتوي على اكثر من مشيد، والفرق بين المشيدات التي تحمل نفس الاسم يكون من خلال عدد المنافذ ونمط كل منفذ، ففي خطأ: لم يُعثر على مصدر المرجع، قمنا بتزويد النمط مستطيل بثلاث مشيدات:

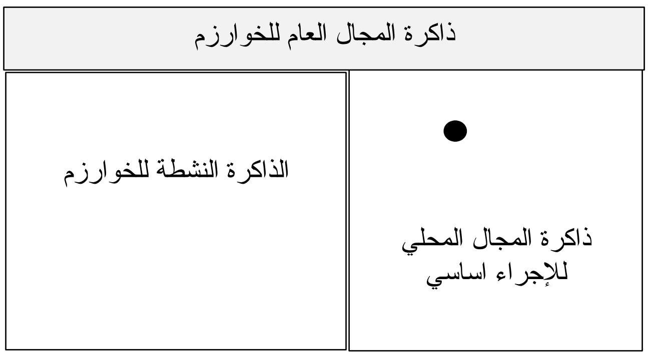
الجزء الثاني:

النص 11 : استعمال وظيفة متخصصة في صناعة و تهيئة متغيرات طارئة من النمط مستطيل

- المشيد الأول (السطر 4)، **مستطيل (طبيعي س1، ج1، ط1، ع1)**: يقوم هذا المشيد بتهيئة كل سمات المستطيل بالقيم التي تعطى له عبر المداخل، فمثلا، كتابة السطر 23:

مستطيل م1 = شيد مستطيل(10، 10، 100، 100) ،

تُنشأ في المجال المحلي للإجراء **اساسي** الصلة **م1** وتُنشأ في الذاكرة النشطة متغيرة طارئة مرتبطة بالصلة **م1** ومحتوية على سمات مَعلومة (خطأ: لم يُعثر على مصدر المرجع).

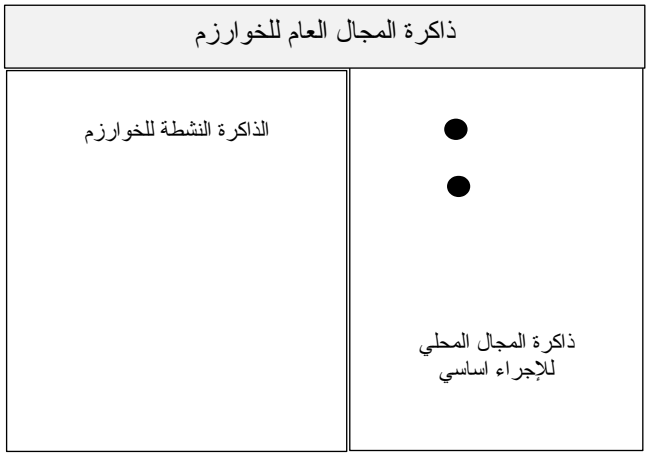


الشكل 1 : ما ينتجه في ذاكرة الخوارزم تصريح ال سطر 23 من النص 11

- المشيد الثاني(السطر 11)، **مستطيل (طبيعي ط1، ع1)**: يقوم بتهيئة سمات الطول والعرض بالقيم التي توفر له عبر المداخل، اما السمات **س** و **ج** فتهيأ بالقيمة 0، فكتابة السطر 24، اي:

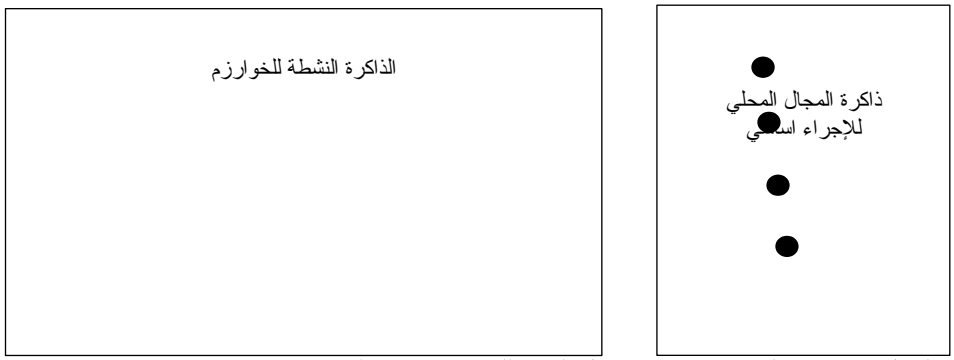
مستطيل م2 = شيد مستطيل (200، 300)

تمكن من انشاء الصلة **م2** في المجال المحلي للإجراء **اساسي** و تُنشأ في الذاكرة النشطة متغيرة طارئة مرتبطة بالصلة **م2** ومحتوية على سمات كلها معلومة (خطأ: لم يُعثر على مصدر المرجع)



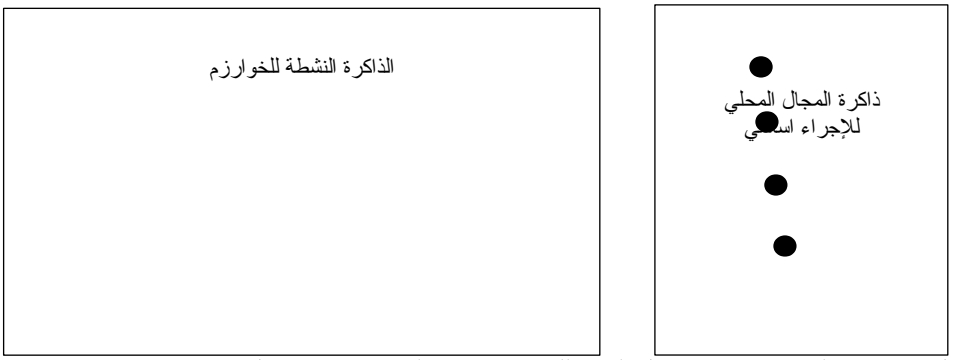
الشكل 2 : ما ينتجه في ذاكرة الخوارزم التصريحين في السطرين 23 و 24 من النص 11

- المشيد الثالث (السطر 17)، و هو **مستطيل()**: لا يحتوي هذا المشيد على اي مدخل، وهذا المشيد يعوض المشيد الافتراضي، وكأننا اعدنا كتابة المشيد الافتراضي وسددنا محتواه، ويقوم المشيد **مستطيل()** بتهيئة كل السمات بالقيمة 0، فمثلا، الكتابة **مستطيل م3 = شيد مستطيل ()** (السطر 25)، او الكتابة **مستطيل م4 = شيد مستطيل ()** (السطر. 26) دون ذكر اقواس المشيد، تُمكن من انشاء الصلتين **م3** و **م4** في المجال المحلي للإجراء **اساسي** وتُنشأ في الذاكرة النشطة متغيرة طارئة مرتبطة بالصلة **م3** ومتغيرة طارئة مرتبطة بالصلة **م4**، وكل متغيرة تحتوي على سمات كلها معلومة (خطأ: لم يُعثر على مصدر المرجع).



الشكل 3: ذاكرة الخوارزم بعد وصول التنفيذ للسطر 26 من النص 11، مع اعادة تعريف محتوى المشيد الافتراضي

لو افترضنا ان المشيد **مستطيل()** لم يصرح به في النمط مستطيل، فان المشيد الافتراضي **مستطيل()** هو الذي يُستعمل في الكتابتين **مستطيل م3 = شيد مستطيل ()** و **مستطيل م4 = شيد مستطيل** (السطر ين25و26)، وفي هذه الحالة تنشأ المتغيرات المهيكلة الطارئة المرتبطة بالصلة **م3** والصلة **م4** بسمات غير معلومة المحتوى كما يظهر في خطأ: لم يُعثر على مصدر المرجع.



الشكل 4: ذاكرة الخوارزم بعد وصول التنفيذ للسطر 26 من النص 11، في حال عدم اعادة تعريف محتوى المشيد الافتراضي

ملاحظة: توفير عدد من المشيدات في نمط ما يوحي و كأن النمط يحتوي على اكثر من طريقة لبناء متغيراته، وهكذا يجد كاتب الخوارزميات اكثر من بَنَاء في نمط مهيكل، وعليه اختيار البناء او المشيد الذي يراه الأنسب لإنشاء المتغيرات التي يصرح بها.

5 – دور الخوارزميات المنتسبة في منهجية كتابة الخوارزميات

تزويد الأنماط المهيكلة بالخوارزميات المنتسبة لن يمر دون ان يكون له اثر على منهجية تنظيم وتصميم الخوارزميات، اما فيما يخص التنظيم فقد تحدثنا عنة آنفا، ولمسنا فائدة هذا التنظيم من خلال النصوص السابقة، واما فيما يخص منهجية تصميم الخوارزميات، فتبرز طريقة جديدة مبنية اساسا على فكرة مفادها جعل المتغيرات المهيكلة الحيوية (التي لها خوارزميات منتسبة) تلعب دورا خاصا بها في الخوارزم، وهذا لكونها تتمتع ببعض الذكاء، ومثل هذه الطريقة المبنية على اساس متغيرات مهيكلة ذكية لها دور ما في الخوارزم، لا يمكن ان تكون مشابهة للطريقة التقليدية في إنجاز الخوارزميات على اساس متغيرات مهيكلة ساكنة، جامدة، لا قدرة لها على التفاعل مع المحيط الخارجي لكونها لا تحتوي على سمات حيوية (اي الخوارزميات المنتسبة)، و فيما يلي نبين الفرق بين الطريقتين (الطريقة التقليدية المبنية على المتغيرات الساكنة والطريقة المبنية على المتغيرات الحيوية) في اطار تصميم خوارزما نسميه "**المستطيل_الوحيد**"، ولجعل الفرق بين الطريقتين اكثر وضوحا، ننجز في اطار كل طريقة نسختين لهذا الخوارزم، النسخة الأولى تكون محدودة قادرة فقط على عمليتين هما: تحريك المستطيل وتغيير حجمه، وفي النسخة الثانية نضيف عمليتين، ثم نرى كيف تتم هذه الإضافة في كل طريقة.

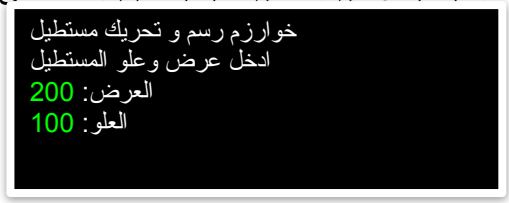


الشكل 5 : اول ما يظهره خوارزم رسم مستطيل على الشاشة

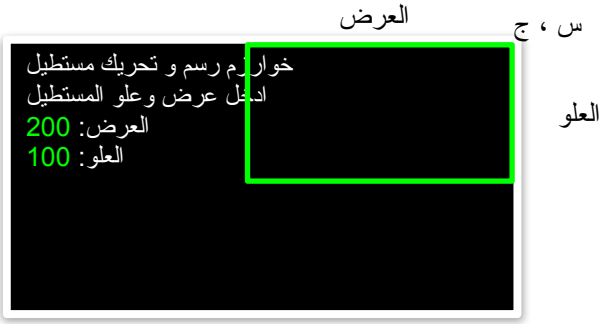
5 – 1 السلوك العام للخوارزم "المستطيل_الوحيد": في نسخته الأولى

يقوم الخوارزم "**المستطيل_الوحيد**" برسم مستطيل على الشاشة ثم يتيح للمستعمل اجراء مختلف العمليات على المستطيل، كتحريكه من موقع الى آخر او تغيير حجمه، ونذكر ان نظام الرسم المستعمل يعتمد على احداثيات طبيعية، وتعتبر القيمة (0 ، 0) هي النقطة في اقصى عُلُو ويمين مساحة الرسم، وكل نقاط الشاشة لها احداثيات اكبر من النقطة (0، 0).

عندما ينطلق تنفيذ الخوارزم، يقوم هذا الأخير في مرحلة اولى، بتقديم نفسه (خطأ: لم يُعثر على مصدر المرجع)، ثم يطلب من المستعمل اعطاءه عرض وعُلو المستطيل الذي سيرسم على الشاشة (خطأ: لم يُعثر على مصدر المرجع). بعد التقاط العرض والعلو، يقوم الخوارزم برسم المستطيل على الشاشة في الإحداثية (0،0) كما يظهر في خطأ: لم يُعثر على مصدر المرجع، و نفترض ان إحداثيات الموقع الذي يرسم فيه المستطيل تساوي دائما قيمة السمات **س** و **ج**، وتمثل **س** و **ج** احداثيات النقطة في اقصى علو و يمين المستطيل ونسميها ايضا بنقطة تعريف المستطيل، ولذا، عند اول رسم للمستطيل، تكون قيمة السمات **س** و **ج** هي (0 ، 0)، ونذكر هنا ان النمط **مستطيل** يُعرف المستطيل بإحداثيات النقطة الموجودة في اقصى علو و يمين المستطيل وبعرضه وعُلوِه(خطأ: لم يُعثر على مصدر المرجع).

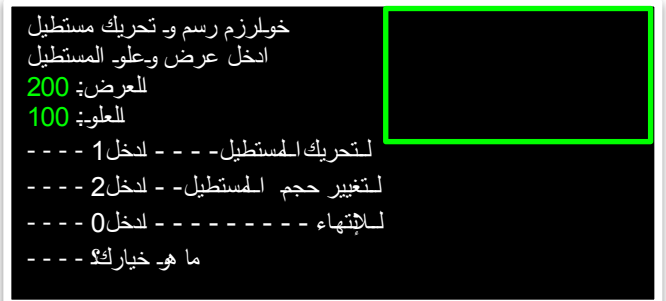


الشكل 6 : الخوارزم يطلب و يلتقط سمات المستطيل



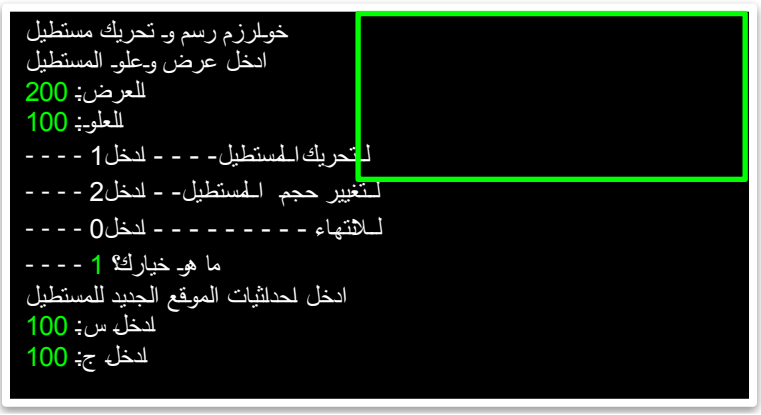
الشكل 7 : الخوارزم يرسم المستطيل في (0،0) بعد التقاطه عرض وعلو المستطيل

بعد الانتهاء من المرحلة الأولى (رسم المستطيل في 0، 0)، يدخل الخوارزم في دوامة يطرح فيها على المستعمل ثلاث خيارات (خطأ: لم يُعثر على مصدر المرجع) و هي: **تحريك** المستطيل من مكان الى آخر، **تغيير حجم** المستطيل، **المغادرة** (انهاء تنفيذ الخوارزم).



و حتى نتمكن من تقديم سلوك عملي، نفترض ان المستعمل يتفاعل مع الخوارزم على اشكل التالي:

- اولا يختار تحريك المستطيل الى الموقع (100، 100).
- ثانيا يختار تغيير حجمه الى 100 على 50.
- ثالثا، يُدخل المستعمل خيارا غير صحيح.
- أخيرا يختار المغادرة.

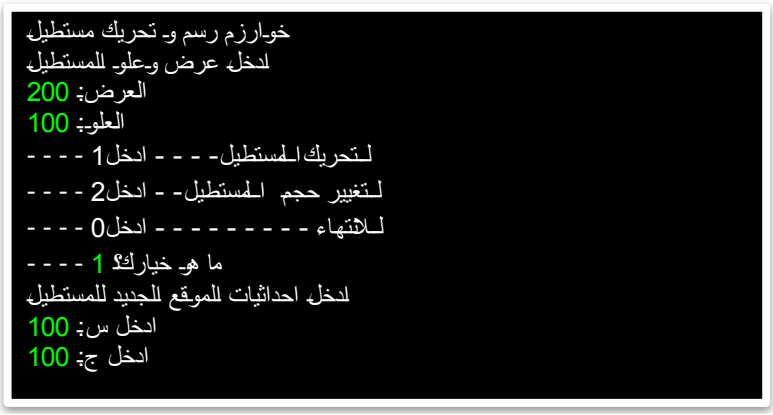


الشكل 9 : الخوارزم يلتقط الخيار الأول و يطلب احداثيات الموقع الجديد

5 – 1 – 2: سلوك الـ

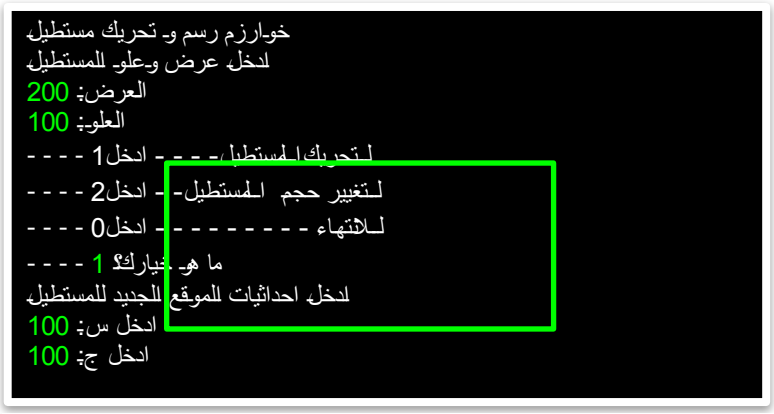
لتحريك المستطيل يدخل المستعمل القيمة 1 (خطأ: لم يُعثر على مصدر المرجع)، وعندها يطلب الخوارزم منه احداثية الموقع الجديد للمستطيل (خطأ: لم يُعثر على مصدر المرجع)، وبعد تلقيه احداثية الموقع الجديد، يقوم الخوارزم بما يلي:

- اولا يمحو المستطيل الموجود على الشاشة (خطأ: لم يُعثر على مصدر المرجع)، والطريقة المتبعة لمحو المستطيل هي اعادة رسمه بلون مساحة الرسم.



الشكل 10 : محو المستطيل بإعادة رسمه بلون مساحة الرسم

- ثانيا يحدث تغيير في سمات **س** و **ج** للمستطيل حتى تكون قيمة هذه السمات تساوي قيمة احداثية الموقع الجديد.
- و أخيرا يقوم برسم المستطيل (خطأ: لم يُعثر على مصدر المرجع).



(100،100)

الشكل 11 : رسم المستطيل في الموقع الجديد (100، 100)

تذكير: استعمال التعليمات الخاصة بالرسم: تعلمنا من الجزء الأول، الفصل الثامن، الفقرة 6، انه من بين الخوارزميات المعتادة التي يمكن استعمالها مباشرة، توجد الخوارزميات المعتادة للرسم وهي التي تظهر في الجدول 4 ونفترض، حسب ما يظهر في الجدول 5، ان كل لون من الوان الرسم يُمَثل برقم يفهمه الإجراء **حدد_اللون**، فمثلا اذا اردنا ان نرسم باللون الأزرق، نكتب **حدد_اللون(4)**.

هدف الخوارزم المعتاد	رأس الخوارزم (اجراء او وظيفة)
اختيار لون الرسم، و نعتبر ان اللون يُمَثل برقم طبيعي	اجراء حدد_اللون (طبيعي اللون)،
رسم خط مستقيم بلون قد اختير من قبل	اجراء ارسم_خط (طبيعي س، ج، الطول، العرض)،
رسم محيط مستطيل بلون قد اختير من قبل	اجراء ارسم_مستطيل (طبيعي س، ج، الطول، العرض)،
رسم مستطيل ممتلئ بلون قد اختير من قبل	اجراء املأ_مستطيل (طبيعي س، ج، الطول، العرض)،
رسم محيط دائرة بلون قد اختير من قبل في اطار مستطيل	اجراء ارسم_دائرة (طبيعي س، ج، الطول، العرض)،
رسم دائرة ممتلئة بلون قد اختير من قبل في اطار مستطيل	اجراء املأ_دائرة (طبيعي س، ج، الطول، العرض)،
رسم محيط دائرة بلون قد اختير من قبل	اجراء ارسم_دائرة (طبيعي س، ج، الشعاع)،
رسم دائرة ممتلئة بلون قد اختير من قبل	اجراء املأ_دائرة (طبيعي س، ج، الشعاع)،

جدول 4 : الخوارزميات المعتادة الموجهة للرسم

القيمة	اللون
0	اسود
1	اصفر
2	بني
3	احمر
4	ازرق

5	اخضر
6	بنفسجي
7	برتقالي
8	ابيض

جدول 5:ارقام الألوان



(100،100)

الشكل 12 :طلب تغيير حجم المستطيل من 200 على 100 الى 100 على 50

5 – 1 – 3 سلوك الخوارزم عند تغيير حجم المستطيل

بعد الانتهاء من تحريك المستطيل الى موقعه الجديد، يعاود الخوارزم عرض الخيارات الثلاثة على المستعمل (خطأ: لم يُعثر على مصدر المرجع)، وهنا نفترض كما اشرنا اليه سابقا، ان المستعمل يختار تغيير حجم المستطيل، اي انه يُدخِل هذه المرة القيمة 2 (خطأ: لم يُعثر على مصدر المرجع)، وهنا يطلب الخوارزم من المستعمل ادخال القيم الجديدة للعرض والعلو (خطأ: لم يُعثر على مصدر المرجع)، وعند تلقيه هذه القيم، يقوم الخوارزم بما يلي: يمحو المستطيل الموجود على الشاشة، ثم يضع القيم التي تحصل عليها من المستعمل في السمات **عرض** و **علو**، وبعدها يقوم برسم المستطيل على حالته الجديدة (خطأ: لم يُعثر على مصدر المرجع)، ثم يعاود عرض الخيارات الثلاثة.



(100،100)

الشكل 13 : تغيير حجم المستطيل الى 100 على 50 و عودة لعرض الخيارات الثلاث

5 – 1 – 4 سلوك الخوارزم في حالة خيار خاطئ ثم في حالة طلب المغادرة

بعد تغيير حجم المستطيل، نفترض ان المستعمل ادخل القيمة 7 (خطأ: لم يُعثر على مصدر المرجع) و هي قيمة مختلفة عن القيم التي تظهر في الخيارات الثلاث، ففي هذه الحالة يقوم الخوارزم بتنبيه المستعمل بأن ما كتبه غير مقبول ثم يعاود عرض الخيارات الثلاث.

اخيرا نفترض ان المستعمل ادخل القيمة 0 (خطأ: لم يُعثر على مصدر المرجع)، في هذه الحالة ينته الخوارزم من التنفيذ بعد ارساله عبر الشاشة النص **مع السلامة...** الى لمستعمل.



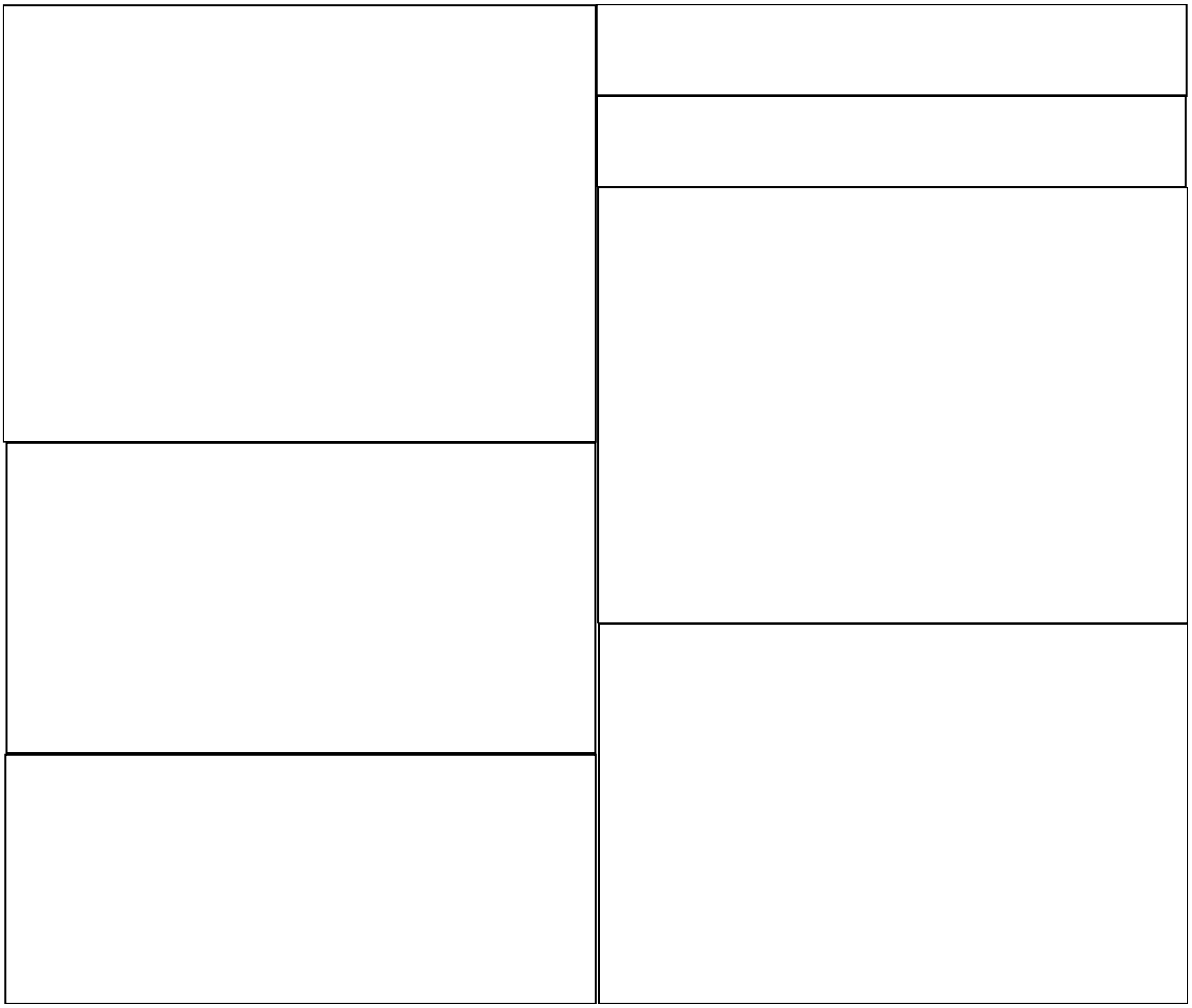
(100،100)

الشكل 14 : سلوك الخوارزم عند ادخال المستعمل لخيار خاطئ و عند رغبة المستعمل الانتهاء من استعمال الخوارزم

5 – 2 كتابة الخوارزم: نص السلوك العام للخوارزم-

انطلاقا من السلوك الذي لمسناه من خلال تفاعل المستعمل مع الخوارزم عبر الشاشة ولوحة الحروف، يمكننا ان نستخلص المراحل الكبرى للخوارزم وهي كالتالي:

1. رسم المستطيل في الموقع (0،0) بعد الحصول على عرضة وعلوه من المستعمل
2. تكرار التعليمات التالية حتى يُعطِ المستعمل الخيار 0
   - 2 . 1 ارسال قائمة الخيارات على الشاشة (3 خيارات: 0، 1 و 2)
   - 2 . 2 التحصل على خيار المستعمل
   - 2 . 3 تنفيذ خيار المستعمل (تحريك او تغيير الحجم او المغادرة)



النص 12 : الخوارزم المستطيل_الوحيد على الطريقة التقليدية : استعمال متغيرات مهيكلة ساكنة

5 – 2 – 1 الطريقة الأولى: استعمال متغيرات مهيكلة ساكنة(بدون خوارزميات منتسبة، خطأ: لم يُعثر على مصدر المرجع)

انطلاقا من السلوك العام المعبر عن المراحل الكبرى للخوارزم، يمكننا ان نستخلص ما يلي لكتابة الخوارزم بشكل منظم:

- اولا التعريف بالنمط الممثل للمستطيل: تحديدا السمات التي تهمنا، اي: **س، ج، عرض** و **علو**
- انجاز الإجراء اساسي الذي يُكلف بما يلي:
  - رسم المستطيل على صفته الأولى

- التفاعل في اطار تعليمة تكرار لتمكين المستعمل من ابداء خياراته وتنفيذ ما يختاره (تحريك او تغيير حجم المستطيل او المغادرة)

- انجاز إجراء متخصص في تحريك المستطيل
- انجاز إجراء آخر متخصص في تغيير حجمه
- انجاز وظيفة تتكفل بما يلي:
  - كتابة الخيارات على الشاشة.
  - التقاط خيار المستعمل.
  - التحقق من صحة الخيار.
  - ارجاع الخيار ان كان صحيحا.

أ – نظرة عامة لمكونات نص الخوارزم (خطأ: لم يُعثر على مصدر المرجع)

في هذه الطريقة كل الخوارزميات (الوظائف والإجراءات) موجودة خارج نطاق النمط **مستطيل**، ولا يحتوي النمط **مستطيل** على اي خوارزم منتسب، ويتكون الخوارزم من:

- التعريف بالنمط **مستطيل** (السطر 1 الى 3)
- التصريح بالثابتتين **لون_مساحة_الرسم** و **لون_المستطيل** (السطر 5 و 6) الأولى تمثل لون المساحة التي يتم فيها رسم المستطيل (الأسود فيما يخصنا) والثانية اللون الذي يتم به رسم المستطيل الذي يظهر في مساحة الرسم.
- ثلاث إجراءات ووظيفة، وهي على التوالي: الإجراء **حرك_المستطيل** (خطأ: لم يُعثر على مصدر المرجع،السطر 7)، الإجراء **غير_حجم_المستطيل** (خطأ: لم يُعثر على مصدر المرجع، السطر 24)، الوظيفة **حصل_الخيار** (من السطر 41 الى السطر 52) والإجراء **اساسي** (من السطر 53الى 73)،

ونلاحظ في بداية الإجراء اساسي التصريح بالمتغيرة المحلية **م (السطر 55)**، وهذه المتغيرة تمثل المستطيل الذي يرسم على الشاشة.

> **تذكير:** نذكر هنا انه طبقا للقواعد الجديدة في كتابة الخوارزميات، كل المتغيرات المهيكلة متغيرات طارئة، فالمتغيرة **م** الممثلة للمستطيل هي في حقيقة أمرها صلة، ومجازا نقول المتغيرة **م** ونعني المتغيرة الطارئة المرتبطة بالصلة **م**.

ب – نظرة تفصيلية لإجراءات ووظائف الخوارزم

- الإجراء **حرك_المستطيل** (خطأ: لم يُعثر على مصدر المرجع،السطر 7): يُمَكِن هذا الإجراء المستعمل من تحريك مستطيل من موقعه الحالي الى موقع جديد يحدده المستعمل، ويرتكز عمل الإجراء على المتغيرة التي توصف في المدخل المسمى **م** والتي تمثل المستطيل المرسوم على الشاشة، وليتمكن هذا الإجراء من تحريك المستطيل الى موقع جديد، يطلب اولا احداثيات الموقع الجديد (خطأ: لم يُعثر على مصدر المرجع، من السطر 9 الى السطر 13)، و قبل ان يثبت القيم الجديدة في السمات **س** و **ج** للمتغيرة **م**، يقوم الإجراء بمحو المستطيل، وكما علمنا مما سبق، فعملية المحو في حقيقتها هي عملية رسم المستطيل بلون مساحة الرسم (السطرين 15 و 16)، وبعد هذه العملية يثبت الخوارزم القيم الجديدة للسمتين **س** و **ج** (السطرين 18 و 19)، ثم يرسم المستطيل على حالته الجديدة باستعمال لون مختلف عن لون مساحة الرسم (السطرين 21 و 22).
- **غير_حجم_المستطيل** (خطأ: لم يُعثر على مصدر المرجع، السطر 24)، هدف هذا الإجراء هو تكبير او تصغير المستطيل ، ويتم هذا الأمر بتغيير السمات **علو** و**عرض**، و يبق المستطيل في مكانه، اي ان السمات **س** و **ج** لا تتغير، ويرتكز عمل الإجراء على المتغيرة التي توصف في المدخل المسمى **م**، ولإنجاز عمله، يطلب الإجراء من المستعمل ان يوافيه بالقيم الجديدة للسمات **علو** و**عرض**، ويحتفظ بهما في المتغيرتين المحليتين **ط1** و **ع1** (الأسطر من 26 الى 30) ، وقبل ان يثبت القيم الجديدة في سمات المتغيرة **م** الممثلة للمستطيل، يقوم الإجراء بمحو المستطيل (السطرين 32 و 33)، وبعد اختفاء المستطيل من الشاشة، يثبت الإجراء القيم الجديدة للسمتين **علو** و **عرض** (السطرين 35 و 36) ثم يرسم المستطيل على حالته الجديدة باستعمال لون مختلف عن لون مساحة الرسم (السطرين 38 و 39).
- **الوظيفة حصل_الخيار:** تقوم هذه الوظيفة بالحصول على الخيار الذي يريده المستعمل من بين 3 خيارات تخرجها الوظيفة في الشاشة على شكل قائمة من الخيارات (الأسطر من 44 الى 47)، ولا تنته الوظيفة من عملها الا بعد الحصول على خيار صحيح، ولذا صممت الوظيفة على شكل حلقة غير متناهية (القيمة **صحيح** هي قيمة العبارة المتحكمة في التكرار)، فعندما لا تتحصل الوظيفة على خيار صحيح (0 ، 1 او 2)، تنبه المستعمل بخطئه ثم تعاود الكرة بنشر قائمة الخيارات، وعندما تتحقق الوظيفة ان الخيار الذي ادخله المستعمل خيارا مقبولا تنته من التنفيذ فورا وترجع الخيار الذي ادخله المستعمل (السطر 49).
- **الإجراء أساسي:** زيادة على التصريح بالمتغيرة **م** (السطر 55) التي تمثل المستطيل الذي يرسم على الشاشة، يحتوي هذا الإجراء على ثلاث مراحل كبيرة:

- في المرحلة الأولى (الأسطر من 57 الى 65) يقوم الخوارزم بالتفاعل مع المستعمل لالتقاط **علو** و **عرض** المستطيل (الأسطر من 59 الى 62) الذي يرسم في نهاية هذه المرحلة في الموقع (0، 0)، ثم يحدد الخوارزم الموقع الذي يرسم فيه الخوارزم بتثبيت قيم السمات **س** و **ج** (السطر 63)، و تنته هذه المرحلة برسم المستطيل، فيتم اولا تحديد لون الرسم (السطر 64) ثم الرسم الفعلي (السطر 65) باللون الذي اختير، ونلاحظ ان لون الرسم مختلف عن لو مساحة الرسم.
- في المرحلة الثانية يتفاعل الخوارزم مع المستعمل من اجل اما تحريك المستطيل او اعادة تحجيمه، و يبق هذا التفاعل يتكرر ما دام المستعمل لم يفصح عن رغبته في الانتهاء من استعمال الخوارزم، وفي النص تمثل تعليمة التكرار وما تحتويه من تعليمات هذه المرحلة الثانية (الأسطر من 66 الى 71)، ويحتوي التفاعل على مرحلتين: في المرحلة الاولى يطلب الإجراء اساسي من الوظيفة **حصل_خيار** ان توافيه بما يريده المستعمل(السطر 67)، وهذه الوظيفة لا ترجع الا الخيارات الصحيحة وهي 0، 1 او 2، وبعد الحصول على رد الوظيفة **حصل_خيار** في المتغيرة المسماة **الخيار** والتي صُرح بها في السطر 56، يتعرف الإجراء اساسي في الأسطر من 68 الى 70 على رغبة المستعمل ثم ينفذ له ما اراد، ونلاحظ ان عبارة التحكم في التكرار (السطر 66) هي القيمة **صحيح**، اي ان التكرار لن ينته ابد من خلال هذه العبارة، والخروج من التكرار يتم من خلال التعليمة "**غادر**" (السطر 68) بعد ان تعرف الإجراء على رغبة المستعمل في المغادرة.
- في هذه المرحلة الأخيرة ينته الإجراء من التنفيذ (السطر 72) ، وقبل الانتهاء الفعلي يرسل الى المستعمل الرسالة **مع السلامة...** ثم ينته الإجراء اساسي من التنفيذ وبانتهائه ينته الخوارزم من التنفيذ.

5 – 2 – 2 الطريقة الثانية: استعمال متغيرات مهيكلة حيوية(فيها خوارزميات منتسبة)

كما كان ذلك في الطريقة الأولى، فالتعريف بالنمط مستطيل يمثل الخطوة الأولى في سياق انجاز نص الخوارزم، وفي هذه الطريقة الثانية (خطأ: لم يُعثر على مصدر المرجع)، عند التعريف بالنمط مستطيل، لا نكتف فقط بالتعريف بالسمات الساكنة التي تهمنا بل ايضا بالسمات الحيوية التي ينفرد بها المستطيل عما سواه من الكائنات في الخوارزم، وتعتبر المشيدات اول هذه السمات الحيوية،

فهي التي تقوم بدور التهيئة عند انشاء متغيرة، والسؤال المطروح في هذه المرحلة هو كالتالي: ما هي الطريقة التي نتبعها في استخراج السمات الساكنة والحيوية التي يوفرها النمط مستطيل؟

للجواب على هذا التساؤل ننطلق من الفكرة القائلة ان السمات الساكنة والحيوية خاصة بالمستطيل، والنمط مستطيل في حقيقته غير مرتبط حيويا بالخوارزم **المستطيل_الوحيد** الذي نحن بصدد انجازه، ولذا يكون من الأفضل ان ننجز النمط مستطيل دون التأثر بالخوارزم الذي سوف يستعمله، بل اكثر من ذلك، يمكن في تعريف النمط مستطيل ان نضع شروطا وقواعد يجب على الخوارزم المستغل للنمط ان يحترمها ويقف عندها، فالنمط بفضل حيويته يمكنه ان يعرف هل المستعمل احترم شروط وقواعد الاستغلال، ولترسيخ استقلالية النمط من مستغليه، علينا ان نركز التفكير فيما يمكن للخوارزم ان يتوفر عليه من سمات ساكنة وما يمكن ان يقدم من خدمات للعالم الخارجي عبر سماته الحيوية.

في هذه الطريقة الثانية، وبعد ان نتحصل على النمط مستطيل، نركز التفكير في ما يجب ان نكتبه على ضوء ما يقدمه الخوارزم من قدرات، فمثلا علينا ان نحدد:

- ما هي الإجراءات او الوظائف المتبقية التي ليست من سمات المستطيل لكنها ضرورية للخوارزم.
- كيف يكون عمل الإجراء الأساسي الذي به ينطلق الخوارزم، وغالبا ما يكون عمل الخوارزم عملا ينسق به بين مكونات الخوارزم.

أ – السمات الساكنة للمستطيل:

هذه المرة، لا نكتف فقط بالسمات التي تُعرِف موقع المستطيل وحجمه (**س** و **ج**، **علو**، **عرض**)، والتي كانت كافية في اطار الطريقة الأولى، بل نبحث عن كل السمات الساكنة دون التأثر بالخوارزميات التي يمكن ان تستغل النمط مستطيل، فمثلا لكل مستطيل **لون** يمكن ان يكون خاصا به، ولمحو المستطيل يجب ان يرسم بلون خاص، هو في حقيقة الأمر لون المساحة التي يرسم فيها، وفي حالنا نكتف بهاتين السمتين الإضافيتين، وهكذا يصبح النمط مستطيل محتويا على 6 سمات كما يظهر ذلك في الأسطر من 2 الى 4 من خطأ: لم يُعثر على مصدر المرجع.

النص 13 : نسخة الثالثة للنمط مستطيل

ب – السمات الحيوية للمستطيل:

فيما سبق (خطأ: لم يُعثر على مصدر المرجع) رأينا بعض السمات الحيوية التي يتميز بها المستطيل وهي على التوالي: الوظيفة **محيط**، الوظيفة **مساحة** والإجراء **حجم**، و نلاحظ ان الوظيفة **حَجِم** يمكن استعمالها لكون تحجيم المستطيل من بين اهداف الخوارزم **المستطيل_الوحيد**، وهنا نتساءل هل هذا ما يمكن للمستطيل ان يوفره للخوارزميات التي سوف تستغله، والإجابة انه يمكن تزويد المستطيل بأكثر مما ذكر في (خطأ: لم يُعثر على مصدر المرجع)، فمثلا يمكن اضافة عمليات التحريك والتدوير ومختلف العمليات الهندسية التي نجدها في العالم الخاص بالمستطيل، وهكذا دون ان ندفع بالتفكير بعيدا يمكن ان نزود النمط **مستطيل** بالقدرات التالية: حساب المحيط، حساب المساحة، الظهور على مساحة الرسم (او الرسم)، الاختفاء من مساحة الرسم (او المحو) تغيير المكان (او التحرك)، تغيير الحجم و الدوران.

ج – كيفية انجاز السمات الحيوية:

اصبح من البديهي لدينا الآن، انه لتزويد نمط ما بسمة حيوية، علينا انجاز وظيفة او إجراء، والسؤال الذي يتبادر الى الذهن هو: هل يكف خوارزم واحد لتمثيل سمة حيوية ما؟ و الجواب مرتبط في الحقيقة بعدد الرؤى (او الزوايا التي ينظر من خلالها) التي تحتملها السمة الحيوية، ففي العموم اذا كانت السمة لا يمكن ان ينظر اليها الا من زاوية واحدة (رؤية واحدة)، فَيَكفيها خوارزما واحدا يكون على شكل اجراء او وظيفة، وهذا هو شأن السمات المتعلقة بحساب المحيط والمساحة، اما اذا تعددت الرؤى لسمة حيوية ما، فلكل رؤية خوارزما خاصا بها، وهذا هو شأن السمات مثل التحريك، والتدوير والتحجيم، فيمكن لكل واحدة ان ترى من زوايا مختلفة.

إذا اخذنا مثلا الإجراء المنتسب **حَجِم** (خطأ: لم يُعثر على مصدر المرجع)، نراه مناسبا للخوارزم **المستطيل_الوحيد** اذ من بين اهداف هذا الخوارزم تمكين المستعمل من تغيير حجم المستطيل، ونذكر انه حسب السلوك العام للخوارزم **المستطيل_الوحيد،** على المستعمل ان يُوَفِر **العلو** و **العرض** للحصول على الحجم الجديد، وعندما نلق نظرة على رأس الإجراء **حجم** نراه يحتوي على مدخل واحد، والمدخل يمثل المعامل الذي به يعاد تحجيم المستطيل، فهنا نرى على الأقل وجود زاويتين ينظر منهما الى سمة التحجيم :

- الزاوية الأولى ويتم من خلالها تغيير الحجم انطلاقا من توفير قيم جديدة للعلو و العرض.
- الزاوية الثانية ويتم من خلالها تغيير الحجم انطلاقا من توفير معامل يحدد المدى الذي يزيد او ينقص به حجم المستطيل، اي علوه و عرضه

و يمكن ايضا ايجاد رؤى اخرى، فمثلا يمكن ان نرى تغيير الحجم بتوفير معاملين، المعامل الأول ينطبق على العلو و المعامل الثاني ينطبق على العرض.

د – كيفية تجسيد الرؤى المختلفة لسمات حيوية

في غالب الأحيان نخصص اسما واحدا للخوارزميات المنتسبة الممثلة لرؤية ما من رؤى سمة حيوية ما، وتختلف الخوارزميات المنتسبة التي لها نفس الاسم بمكونات رأسها، وبالتحديد بعدد المنافذ ونمط كل منفذ، فمثلا، الإجراء الممثل لسمة التحجيم المنظور اليها من الزاوية التي ترى تغيير الحجم انطلاقا من معامل التحجيم، يكون رأسها محتو على مدخل واحد من النمط حقيقي، ويكتب كالتالي: **اجراء حجم (حقيقي معامل_حجم)**، اما الرؤية التي ترى ان تغيير الحجم يكون على اساس تغيير العلو والعرض، فان رأس الخوارزم الذي يمثلها يكون محتويا على مدخلين، و يكتب هذا الرأس كالتالي: **اجراء حجم (طبيعي علو، عرض)**، واذا اردنا ان نأخذ بالرؤية الثالثة التي تفرق بين معامل التحجيم الخاص بالعلو ومعامل التحجيم الخاص بالعرض، فعلينا ان ننجز خوارزما يكون رأسه على الشكل التالي: **اجراء حجم (حقيقي مع_علو، مع_عرض)**

هـ – ماذا لو كان لرؤيتين رأس واحد:

و نعني هنا ان مداخل كلا الخوارزميان متساويان في العدد وكذلك في النمط، فمثلا، فيما يخص الإجراء **حجم،** اذا اعتمدنا على الرؤية الثانية التي ترى ان تغيير الحجم يكون على اساس تغيير العلو والعرض، وعلى رؤية رابعة تفرق بين معامل التحجيم الخاص بالعلو ومعامل التحجيم الخاص بالعرض، وكلا المعاملين من النمط طبيعي (في الرؤية الثالثة المعاملين من النمط حقيقي)، فإننا نكون امام حالة تَساوي رأسان كما يظهر في السطرين التاليين:

- رأس الرؤية الثانية: **اجراء حجم (طبيعي علو، عرض)**

- رأس الرؤية الرابعة: **اجراء حجم (طبيعي مع_علو، مع_عرض)**

والتساوي بين الرأسين لا يعتمد على اسماء المنافذ، فهذه اسماء محلية لا يعرفها الا الإجراء، اما من المنظور الخارجي، فحقيقة الرأس تعتمد فقط على أنماط المنافذ، فمن المنظور الخارجي يكون كلا الإجراءين على الشكل التالي: **اجراء حجم (طبيعي ، طبيعي).**

ففي مثل هذه الحالة، يجب علينا ادخال تغيير طفيف على اسم الإجراء المنتسب، فمثلا نسمي الإجراء الثاني **حجم1**، فيكون رأسه **اجراء حجم1 (طبيعي مع_علو، مع_عرض)**، او الاستغناء عن هذا الإجراء والاحتفاظ بإجراء الرؤية الثالثة، اي : **اجراء حجم (حقيقي مع_علو، مع_عرض).**

و – نص النمط مستطيل

يحتوي خطأ: لم يُعثر على مصدر المرجع على النسخة الثالثة للنمط **مستطيل**، وفيه تظهر السمات منظمة على الشكل التالي:

- اولا كتبت السمات الساكنة، ونلاحظ هذه المرة ان السمات المتعلقة بلون الرسم والمحو قد ادخلتا في النمط.
- ثانيا كتبت المشيدات وهي الإجراءات الحيوية المرتبطة بتهيئة المتغيرات فور انشاءها، وفي النسخة الثالثة احتفظنا فقط بمشيد واحد.
- ثالثا كتبت مختلف الخوارزميات المنتسبة المرتبطة بالسمات الحيوية للنمط **مستطيل**.

ز – سمة الدوران في النص

و في هذا النص اكتفينا بإجراء واحد فقط لتحمل سمة الدوران، مع ان هذه السمة تحتمل اكثر من اجراء منتسب، فالإجراء الذي يظهر في النص يقوم بتدوير المستطيل بتسعين درجة فقط، ويمكن مثلا ان نتصور إجراءا يقوم بتدوير المستطيل حسب خيار ما، وهنا يجب ان يوصف للإجراء هذا الخيار، ويمكن ايضا ان نتصور اجراء يقوم بتدوير المستطيل حسب خيار ما وعلى اساس نقطة معينة، وفي هذه الحالة يجب على رأس الإجراء المنتسب ان يحتوي على الخيار واحداثية النقطة التي يتم عليها الدوران، فإذا افترضنا ان الخيار معلومة من النط طبيعي يكون الرأسين الذين لم يذكرا في خطأ: لم يُعثر على مصدر المرجع على الشكلين التالين: **اجراء دور (طبيعي الخيار)** و **اجراء دور. (طبيعي الخيار، س، ج)**

ح – الظهور في مساحة الرسم والاختفاء منها

كما تعلمنا مما سبق، فان الفرق بين رسم مستطيل ومحوه يكمن في قيمة اللون المستعمل في الرسم، فاذا كان اللون مختلف من لون مساحة الرسم، فعملية الرسم تظهر الشكل، اما اذا كان اللون المستعمل في الرسم هو لون مساحة الرسم، فعملية الرسم تخفي الشكل ان كان ظاهرا، وانطلاقا من هذه الملاحظة يمكن ان نعرف بإجرائين: واحدا للرسم وواحدا للمحو، كما يمكن ان نستعمل فقط إجراءا واحدا للظهور والاختفاء، و يجب ان يحتوي الإجراء على مدخل فيه توضع قيمة لون الرسم، وفي النص اعتمدنا الطريقتين.

ط – ملاحظة وتذكير: تطابق الأنماط في العبارات

نلاحظ في السطر 21 من خطأ: لم يُعثر على مصدر المرجع ذكر العبارة **طول=علو* معامل_حجم**، ونتيجة تقييم **علو* معامل_حجم** يكون من النوع حقيقي لكون احد المعاملين من النمط حقيقي، لكن عندما نشحن قيمة حقيقية في متغيرة طبيعية، لن يدخل من العدد الحقيقي في المتغيرة الطبيعية الا جزئه الصحيح.

في بعض لغات البرمجة لا يمكن كتابة مثل هذه العبارات الا بزيادة كتابة تفيد بنوعية النتيجة، اي ان الكتابة المضافة تبين ان كاتب الخوارزم يريد فقط الجزء الصحيح من القيمة الحقيقية، وهذه الزيادة في حقيقة الأمر عملية تحويل قيمة من نمط ما الى نمط آخر، فمثلا في اللغات "سي"، "سي++" و"جافا"، يعبر عن عملية تحويل قيمة من نمط اول الى نمط ثان بوضع اسم النمط الثان بين قوسين قبل القيمة الأولى التي هي من النمط الأول، وهكذا تكتب عبارة السطر 20 على الشكل التالي:

طول= (طبيعي) علو* معامل_حجم ؛

اما فيما يخصنا فالتحويل يكون دون ذكر عملية تحويل قيمة من نمط الى آخر.

يتكرر هذا الأمر في السطرين 31 و32، فإذا اخذنا ما في السطر 31 كمثل، تُقَيم اولا العبارة **علو* مع_علو**، ونتيجة هذا التقييم تكون من النمط حقيقي لكون المتغيرة **مع_علو** من النمط حقيقي، ثم بعد ذلك يُشرع في وضع القيمة الحقيقية في وعاء من النمط طبيعي المتمثل بالمتغيرة **انا.علو**، ولن يستطيع مثل هذا الوعاء الا استقبال الجزء الصحيح من القيمة الحقيقية، ففيما يخصنا الكتابة صحيحة والنتيجة معلومة، اما ان كنا في اطار لغة برمجة شديدة التحقق من توافق الأنماط نكون مضطرين لإضافة عملية التحويل من النمط حقيقي الى النمط طبيعي، فتكتب العبارة على الشكل:

انا.عرض= (طبيعي) عرض* مع_ عرض ؛

ي – ملاحظة : خلو الخوارزميات المنتسبة للنمط مستطيل من تعليمات التفاعل مع المستعمل

نلاحظ في نص النمط مستطيل (خطأ: لم يُعثر على مصدر المرجع) عدم استعمال تعليمات التفاعل لجلب المعطيات اللازمة لبعض الخوارزميات المنتسبة، وهذا الأمر راجع الى الفكرة المهمة التي ارتكزنا عليها في تصميم النمط مستطيل والقائلة بان النمط عليه ان يحتوي على كل ما تنفرد به المستطيلات من سمات ساكنة او حيوية، واذا تأملنا في تعليمات التفاعل لجلب المعطيات نراها لا تخص المستطيلات، ولا توجد علاقة بين مفهوم المستطيل ومفهوم التفاعل مع المستعمل، وتماشيا مع هذه الفكرة لم نستعمل عمليات التفاعل مع المستعمل لطلب المعطيات بل استعملنا المنافذ العادية التابعة للخوارزميات المنتسبة.

ك- هل يجب استعمال كل ما ذكر في النمط:

في الطريقة الثانية، عندما نركز على نمط ما لنزوده بمختلف الإمكانيات الخاصة به، يتبادر الى الاذهان، خاصة عند المبتدئين، التساؤلين التالين: لما كل هذه السمات ؟ وهل يجب علينا استعمالها كلها؟

اما فيما يخص التساؤل الأول فهو متعلق بروح هذه الطريقة الثانية، فالنمط الذي نكتب ليس مرتبطا بأي خوارزم، ويجب عليه ان يكون فعالا مع مختلف الخوارزميات، فكل من يريد استغلالها سيجد فيه لا محالة السمات الساكنة والحيوية التي تهمه، اما عن التساؤل الثاني فكل خوارزم يستغل ما يهمه، بل عليه ايجاد السمات والخوارزميات المنتسبة الأنسب له، وسوف نرى فيما بعد كيف تمكن هذه الطريقة خوارزما ما، استغل نمطا فيه سمات حيوية، من إضافة كفاءات اخرى بشكل خفيف وسريع.

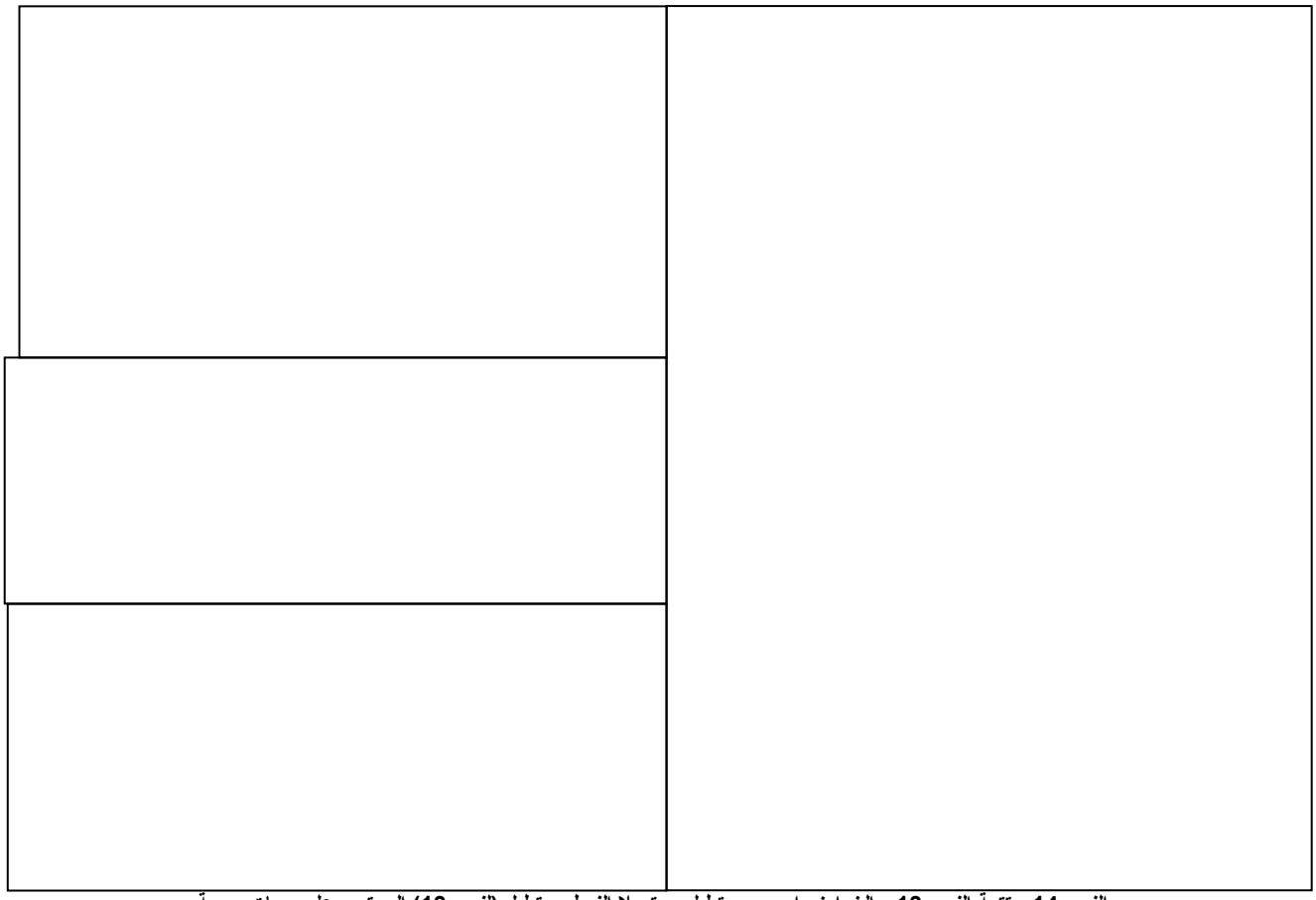
ل - إجراءات و وظائف الخوارزم المستطيل_الوحيد.

هذه الإجراءات او الوظائف غير منتسبة للنمط مستطيل، وتقوم باستغلال النمط المستطيل للوصول الى تجسيد سلوك **المستطيل_الوحيد** الذي تعرفنا عليه من قبل، وكانت ملامحه الكبرى كالتالي:

1. رسم المستطيل على هيئته الأصلية بعد اخذ عرضة و علوه من المستعمل
2. تكرار التعليمات التالية حتى يعط المستعمل الخيار 0
2 . 1 ارسال قائمة الخيارات على الشاشة (3 خيارات: 0، 1 و 2)
2 . 2 التحصل على خيار المستعمل
2 . 3 تنفيذ خيار المستعمل : تحريك او تغيير الحجم او المغادرة

انطلاقا من ملامحه الكبرى ومما يحتويه الآن النمط مستطيل من قدرات، يُنجز الخوارزم **المستطيل_الوحيد** على النحو التالي (خطأ: لم يُعثر على مصدر المرجع)

- التعليمات المتعلقة بالمرحلة الأولى تكون في الإجراء اساسي، وهي التي تظهر في الأسطر. 62 الى 67 من خطأ: لم يُعثر على مصدر المرجع ونلاحظ في السطر 66 استعمال مشيد عند انشاء المتغيرة المرتبطة بالصلة **م**، ونلاحظ في السطر 67 طلب تنفيذ الإجراء المنتسب **ارسم** على المتغيرة **م**.
- التعليمات المتعلقة بالمرحلة الثانية هي تلك التي تظهر في الأسطر. من 68 الى 73 من خطأ: لم يُعثر على مصدر المرجع، ونلاحظ ان هذه التعليمات هي نفسها التي استعملت في الطريقة الأولى.



النص 14 : تتمة النص 13 : الخوارزم ارسم_مستطيل مستعملا النمط مستطيل (النص 13) المحتوي على سمات حيوية

- بما ان العمليات على المستطيل (التحريك و تغيير الحجم) تتم على مستوى النمط مستطيل، فان الإجراءات الغير المنتسبة تتكفل فقط بالتفاعل مع المستعمل ان لزم الأمر لجلب المعطيات الضرورية للعمليات ثم تطلب من المتغيرة **م** ان تنجز العملية المنتسبة المطلوبة، فعلى سبيل المثال، في الأسطر من 90 الى 92 امن لإجراء **غير_حجم_المستطيل** (خطأ: لم يُعثر على مصدر المرجع)، يتم التفاعل، وفي سطر واحد هو السطر 94 وبتعليمة واحدة تتم عملية تحجيم المستطيل **م**، بطلب تنفيذ الإجراء المنتسب **حجم** على المتغيرة **م**، ونلاحظ ايضا ان الإجراء **غير_حجم_المستطيل** في خطأ: لم يُعثر على مصدر المرجع اصبح اقصر بكثير

(تصريحين و5 تعليمات) من نفس الإجراء في خطأ: لم يُعثر على مصدر المرجع (تصريحين و11 تعليمة)، ونفس هذه الملاحظات قائمة بالنسبة للإجراء **حرك_المستطيل**.

6- النسخة الثانية للخوارم ارسم_مستطيل: اضافة كفاءة اخرى:

بعد انجاز النسخة الأولى بكلتا الطريقتين، نشرع الآن في انجاز النسخة الثانية التي تحتوي على كفاءة جديدة وهي قدرة الخوارزم على اعطاء المستعمل خيار آخر يتمكن المستعمل بموجبه بتدوير المستطيل.

6- 1 النسخة الثانية على الطريقة الأولى (خطأ: لم يُعثر على مصدر المرجع)

في هذه الطريقة الأولى، علينا ان ندخل التعديلات التالية على النسخة الأولى:

- انجاز اجراء مكلف بعملية الدوران على نفس الطريقة التي انجزنا بها الإجراءين **غير_حجم_المستطيل** و**حرك_المستطيل**، وفي خطأ: لم يُعثر على مصدر المرجع سُمي **دور_مستطيل** هذا الإجراء (الأسطر من 76 الى 88).
- اضافة خيار خاص بالدوران في الوظيفة **حصل_الخيار**(الأسطر 46، 50 و51)
- اضافة خيار الدوران على مستوى تعليمة التكرار في الإجراء اساسي (السطر 72).

النص 15 : النسخة الثانية من الخوارزم ارسم مستطيل على الطريقة التقليدية : استعمال متغيرات ساكنة

6- 2 النسخة الثانية على الطريقة الثانية (خطأ: لم يُعثر على مصدر المرجع)

في هذه الطريقة الثانية، علينا ان ندخل تعديلين خفيفين فقط على النسخة الأولى:

- اضافة خيار خاص بالدوران في الوظيفة **حصل_الخيار** (الأسطر 82، 86 و87).
- اضافة خيار الدوران على مستوى تعليمة التكرار في الإجراء اساسي (السطر 73).

ونلاحظ هنا اننا لم ننشأ اجراءا آخرا متخصص في الدوران، كون هذا الإجراء قد انجز من قبل كإجراء منتسب للنمط مستطيل، وهو الذي يطلب تنفيذه في السطر 73.

سبب خفة وسرعة التعديلات في الطريقة الثانية يرجع الى الطريقة نفسها التي جعلت النمط مستطيل يتوفر مسبقا على خدمة الدوران، وكان بالإمكان جعله يتوفر على اكثر مما هو عليه في خطأ: لم يُعثر على مصدر المرجع.

الجزء الثاني: الفصل الخامس: الخوارزميات المنتسبة للأنماط المهيكلة

6 – 3 الفرق بين الطريقتين في هيكلة الخوارزم:

يظهر مما سبق ان الطريقتين مختلفتين اختلافا هاما في تنظيم كتابة الخوارزم، ففي الطريقة الأولى كل الإجراءات والوظائف وضعت في نفس المستوى، اي مستوى الخوارزم المكون من هذه الإجراءات والوظائف، فعمل هذه الإجراءات والوظائف تتم في اطار واحد هو الخوارزم، وتتميز مثل هذه الهيكلة بمركزية العلاج للإشكال الذي من اجله وضع الخوارزم، اما الطريقة الثانية فتتميز بعدم مركزية العلاج، فالإجراءات والوظائف موزعة في أُطُر عدة، منها ما يوجد في الإطار المركزي الممثل بالخوارزم، كالإجراء **اساسي** والإجراءات الموجودة في نفس مستواه، ومنها الأطر الأخرى المتمثلة في الأنماط المهيكلة المحتوية على السمات الحيوية، فكل اطار مكلف بعمل يخصه.

الفصل السادس

الأوعية: الجداول

الجزء الثاني: الفصل السادس : الأوعية: الجداول

1 – التعريف:

الأوعية متغيرات مهيكلة، تحتوي على معلومات تنتمي الى نمط واحد فقط، فمثلا يمكن ان نجد في وعاء اول المعلومات التي تُمثل معدلات الطلبة في مادة ما، وفي وعاء ثان أسماء الناجحين في مسابقة ما، وفي وعاء ثالث السيارات المشاركة في مسابقة ما، ولا يمكن ان نجد في وعاء معلومتين لا تنتميان الى نمط واحد، فلا يمكن مثلا ان نجد سيارة في الوعاء الذي يحتوي على معدلات الطلبة في مادة ما.

ابرز خصائص الأوعية:

- عدم وجود اسما خاصا بأي معلومة منتمية للوعاء، كما هو الشأن مع المعلومات المهيكلة التي نصل اليها عبر صلة لا عبر اسم خاص بها.
- يوفر الوعاء آلية خاصة به تمكن المستغل من الوصول الى اي معلومة فيه.

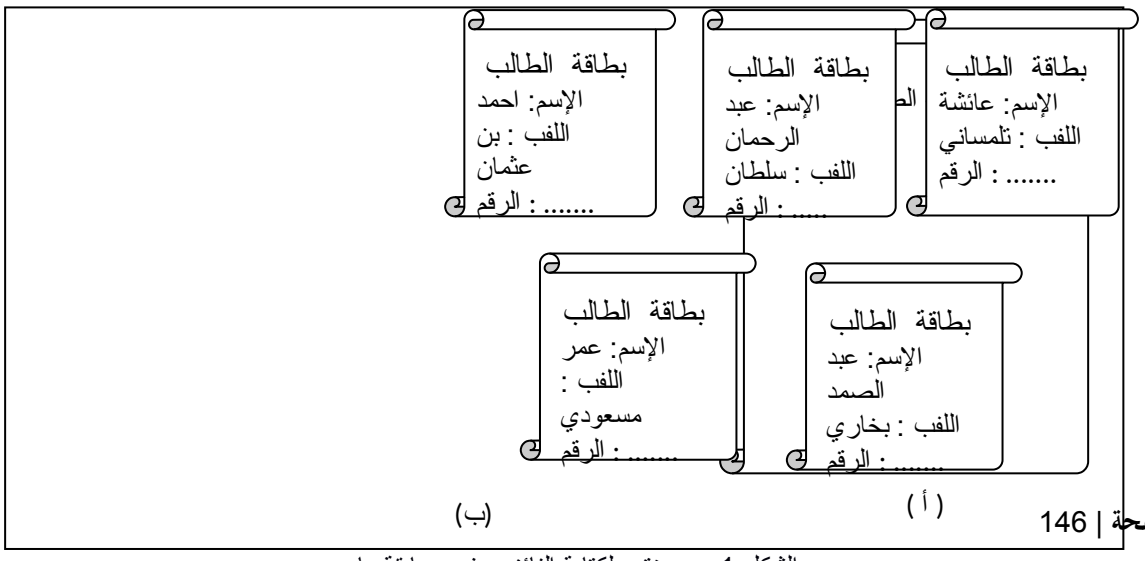
2– الأوعية في الواقع المعاش

اذا تأملنا في الواقع المعاش، نرى ان الأوعية كثيرة الاستعمال وتظهر على شكلين اساسين:

- الجداول
- السلاسل

فمثلا، يمكن كتابة الفائزين في مسابقة ما على احد الشكلين:

- **جدول** من الفائزين (خطأ: لم يُعثر على مصدر المرجع، أ)، كل سطر يحتوي على معلومات الفائز.
- **سلسلة** من البطاقات، كل بطاقة تحتوي على معلومات الفائز (خطأ: لم يُعثر على مصدر المرجع، ب).



الشكل 1 : صيغتين لكتابة الفائزين في مسابقة ما

3 – 3 الفروق الأساسية بين طرق التجميع في الأوعية:

يجب علينا ان نَعلم ان الاوعية توفر لكل معلومة مكان لتوضع فيه، فمثلا فيما يخص الجدول، يرسم الجدول اولا لتحديد المواقع (او الخانات او الأسطر) التي ستستضيف معلومة ما، وبعد اتمام انشاء الجدول، تُعَبأ خانات الجدول بالمعلومات.

في بعض الحالات يمكن ان تبق بعض الخانات (او الأسطر) غير مستعملة (شاغرة ، فارغة)، اما في حالات اخرى فتستعمل كل الخانات، واذا بقيت معلومات لم تدرج في الجدول، فعلينا ايجاد حل لهذا المشكل، وفي العموم هناك حلين بديهيين:

- في الحل الأول يُرسم جدول جديد لتوضع فيه المعلومات المتبقية، فتصبح المعلومات في اكثر من جدول، كل جدول في بطاقة خاصة به، وكأننا نستعمل طريقة التسلسل لمعلومات مهيكلة هي في حد ذاتها اوعية.
- في الحل الثاني، يُنشأ جدول اكبر حجما من الأول، يُنقل اليه ما في الجدول الأول وتضاف اليه المعلومات التي لم يتسع لها الجدول الأول، و يُتلف الجدول الأول.

الفرق الأول، المواقع التي توضع فيها المعلومات: الفرق الأول بين تجميع المعلومات في جدول وبين تجميعها على شكل سلسلة (معلومات متسلسلة) يكون على مستوى المواقع التي توضع فيها المعلومات، ففي حالة الجدول يجب وضع المعلومة في المواقع (اي الخانات) التي عينت مسبقا، وعددها محدود، ولا يمكن وضع المعلومات خارج الخانات، فهناك فرق بين الورقة التي رُسِم فيها الجدول والجدول نفسه، فالورقة هي مكان للجدول، والجدول فيه مواقع للمعلومات (اي الخانات)، و عندما تُستَنفذ (تُستعمل) كل الخانات لا يمكن اضافة معلومة اخرى للجدول، بل يجب اعادة انشاء جدول اكبر حجما، يُنقل اليه ما في الجدول الأول وتُضاف اليه المعلومات التي لم يتسع لها الجدول الأول، ثم يتلف الجدول الأول ، ونلاحظ ان خانات الجدول محاذية بعضها البعض، فيمكن التنقل من خانة الى اخري بسهولة كبيرة.

اما فيما يخص التجميع بالترابط (المعلومات المتسلسلة)، فالمواقع ليست بالضرورة متجاورة، ولفهم التجميع بالترابط نورد المثال التالي المتعلق بطريقة تسجيل الطلبة في بداية السنة الدراسية باستعمال البطاقات. ففي مثل هذه الطريقة توضع معلومات الطالب في بطاقة ملونة خاصة به عندما يتم تسجيله، ويدل لون البطاقة على مستوى الطالب في المؤسسة التعليمية، ونفترض ان اللون الأزرق مخصص لطلبة السنة الأولى، واللون الأخضر لطلبة السنة الثانية، والأصفر للسنة الثالثة الخ، ففي بداية السنة عندما تبدأ عملية تسجيلات الطلبة، يُخصص صندوق لتوضع فيه بطاقات كل الطلبة

المسجلين مهما كان مستواهم، ونلاحظ هنا ان الصندوق هو مكان للبطاقات وليس مكانا للمعلومات، فالبطاقة هي مكان المعلومة، ففي بداية التسجيلات يكون الصندوق فارغا، وكلما جاء طالب للتسجيل، أُخذت ورقة باللون الذي يناسب مستوى الطالب، ثم تكتب المعلومات عن الطالب، وبعدها توضع البطاقة في الصندوق، وتعاد هذه العملية مع كل طالب، وبعد فترة من الزمن نرى في الصندوق عدد كبيرا من البطاقات بألوان مختلفة، ونفرق بين مستوى الطلبة انطلاقا من لون البطاقة، فاذا اردنا من العامل المكلف بالتسجيل ان يتصفح بطاقات طلبة السنة الأولى بحثا على طالب ما، فإننا نلاحظ ما يلي:

- يتصفح فقط البطاقات الزرقاء.
- البطاقات الزرقاء ليست متجاورة بالضرورة، لكنها وضعت في مواقع غير متجاورة، والرابط بينها هو اللون.
- يستعمل المتصفح اللون الأزرق عندما ينتقل من بطاقة الى التي تليها، وكأن اللون يمثل الوعاء الذي تجتمع فيه البطاقات.
- لم يخصص في الصندوق مكانا خاصا للبطاقات الزرقاء او الخضراء الخ، وهذا الأمر يختلف كليا عن الجدول الذي تخصص له منطقة في الورقة التي يرسم عليها، ولا يمكن وضع المعلومات خارج الجدول، اي في مناطق الورقة التي لا يشملها الجدول.
- الوعاء الأزرق (والأخضر والأصفر) يكبر مع كل اضافة، فليس له حجم محدد مسبقا لا يمكن تجاوزه، وهذا عكس الجدول الذي يرسم بحجم محدد لا يمكن تغييره، وفي بدايته يكون الوعاء فارغا.

الفرق الثاني، حجم الوعاء: كما رأيناه الآن يظهر على مستوى حجم الأوعية، ففي حالة الجدول يكون الحجم محددا لا يمكن توسيعه الا بإنشاء جدول جديد اكبر حجما، اما فيما يخص الأوعية المرتكزة على الترابط، فحجمها يكبر ويصغر مع عدد المعلومات التي توضع فيه، والحَجم الذي لا يمكن تجاوزه مرتبط بالمكان الذي يوجد فيه الوعاء المترابط.

الفرق الثالث، سرعة الوصول الى معلومات الوعاء: يكمن في طريقة وسرعة الوصول الى معلومة في موقع ما، ففيما يخص الجداول، تكون اسطر الجدول مرقمة، فاذا طلب من احد ان يقرأ السطر 52، فانه يذهب مباشرة الى السطر(او الخانة) الذي يحمل الرقم 52، اما في حالة الترابط، اذا طلب من احد ان يقرأ ما في البطاقة رقم 69، فتراه يبدأ التعداد انطلاقا من البطاقة الأولى في الترابط، فيكون وقت الوصول الى المعلومة اطول من وقت الوصول الى المعلومة في الجدول.

ملاحظة: الفرق الأساسي بين المعلومات المهيكلة و الأوعية المجمعة:

– لا يمكن اضافة أو حذف سمة من الهيكل بعد تعريفه واستغلاله، فالمعلومة المهيكلة تحتاج الى وجود كل سماتها، حتى ولو كانت السمة لا تستعمل في الخوارزم، اما الوعاء المجمع فيمكن معه اضافة او حذف معلومة ، ويمكن ان يكون الوعاء فارغا، لا يحتوي على اي معلومة، ولا يمكن ان يحتوي الوعاء على معلومات من انماط مختلفة، ويمكن ان يحتوي الوعاء على معلومات من نمط مهيكل ما.

– كل سمة في هيكل المعلومة المهيكلة لها اسم خاص بها، اما في الأوعية فلا توجد اسماء للمعلومات، وتُعرف المعلومات بمواقعها في الوعاء.

4 – الأوعية في عالم الخوارزميات:

كما لحظنا، فان الواقع المعاش يستعمل بكثافة الأوعية، ولتسهيل تمثيل الواقع المعاش في نص الخوارزم، يجب علينا توفير آليات فعالة للتمثيل الفعال للأوعية.

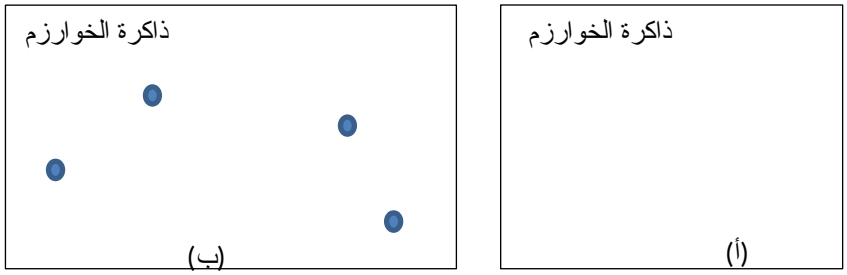
الأوعية في عالم الخوارزميات وفي عالم لغات البرمجة ليست بمتغيرات بدائية، بل هي متغيرات معقدة ومهيكلة على شكل خاص، فهي تَجمعٌ لعدد من المتغيرات، كلها من نمط واحد، والأوعية في عالم الخوارزميات على شكلين: **الجداول** و **السلاسل** (او الخلايا المتسلسلة او المترابطة) .

الجدول مكون من عدد من المتغيرات المتجاورة في الذاكرة (خطأ: لم يُعثر على مصدر المرجع–أ) ، وكل متغيرة تمثل خانة من خانات جدولٍ حقيقي من الواقع المعاش ، وتكون المتغيرات الممثلة للخانات متواجدة في منطقة واحدة من الذاكرة، ومتجاورة بعضها البعض، اي ان الجدول تخصص له منطقة واحدة في الذاكرة، ويظهر خطأ: لم يُعثر على مصدر المرجع–أ جدولين في ذاكرة الخوارزم.

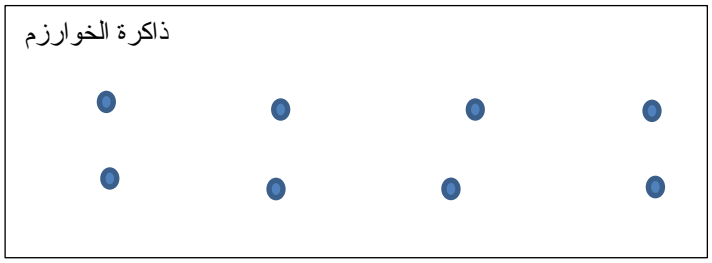
الخلايا المتسلسلة (او المترابطة) مكونة من عدد من الخلايا، وكل خلية تحتوي على:

- معلومات ربط الخلية بغيرها من الخلايا.
- المعلومات التي تحملها الخلية.

و يمكن ان تكون معلومات ربط الخلايا على صور عدة كما يظهر ذلك في خطأ: لم يُعثر على مصدر المرجع–أ وفي خطأ: لم يُعثر على مصدر المرجع، و غالبا ما تكون الخلايا مبعثرة في الذاكرة (اي انها غير متجاورة).



الشكل 2 : المتغيرات الممثلة للأوعية المجمعة: (ا) الجداول (ب) السلسلة



الشكل 3 :الوعاء المجمع بالتسلسل او المتغيرات المتسلسلة

5 – الجداول في لغة الخوارزميات:

كما اوردناه سابقا، فإن الجدول في حد ذاته نمط غير بدائي، وحسب القواعد الجديدة التي ادخلناها واعتمدناها في الفصل الرابع، ولكونه متغيرة معقدة غير بدائية، يُنشأ الجدول في الذاكرة النشطة ويُستغل عبر صلة، ولإنشاء جدول (او التصريح بجدول)، علينا ان نوفر المعلومات التالية :

- نمط المعلومة التي يحتويها الجدول.
- عدد ابعاد الجدول.
- اسم الجدول، وهو في حقيقة الأمر اسم لصلة تستعمل للوصول الى المنطقة التي تحتضن فعليا الجدول في الذاكرة النشطة.
- حجم كل بُعدٍ في الجدول

كيفية كتابة هذه المعلومات تختلف في حقيقة الأمر من لغة الى اخرى، وفيما يخصنا نتبع الكتابات التالية للتعبير عن الأبعاد من جهة وعن نمط الجدول من جهة اخرى:

التعبير عن ابعاد الجدول: نستعمل العارضتين [ ] للتدليل على ابعاد الجدول، فالرمزين [ ] لوحدهما يدلان على بعد واحد، فإذا اردنا اكثر من بعد اضفنا الرمزين لكل بعد ، فكتابة بعدين يكون على الشكل [ ] [ ] و ثلاث ابعاد على الشكل [ ] [ ] [ ] الخ...

التعبير عن نمط جدول: علمنا ان اي جدول يكون مرتبطا ارتباطا حيويا بنمط المعلومات التي يحتويها، فلا يمكن ان نتصور في عالم الخوارزميات جدولا دون الإفصاح عن نمط المعلومات التي يمكنه احتواؤها، والجدول نمط كباقي الأنماط، فهو نمط غير بدائي، بل مهيكل بطريقة خاصة، بل مركب من امرين اساسيين:

- الجدول بِبُعدٍ ما، المعَبَر عنه بذكر عارضتين لكل بعد.
- نمط المعلومة التي يحتويها الجدول.

و نُعبر عن النمط **جدول** بكتابة نمط المعلومات التي يحتويها الجدول، متبوعا بعارضتين او اكثر حسب البعد الذي نريد، ويبين الجدول 6 بعض الأمثلة حول كتابة النمط الممثل لجدول ما.

نمط المحتوى	نمط الجدول	تعليق

حقيقي	حقيقي[ ]	جدول ببعد واحد للأعداد الحقيقية
طبيعي	طبيعي[ ][ ]	جدول ببعدين للأعداد الطبيعية
سلسلة	سلسلة[ ]	جدول ببعد واحد لسلاسل الحروف
حرف	حرف[ ] [ ] [ ]	جدول بثلاثة ابعاد للحروف
مستطيل	مستطيل[ ]	جدول ببعد واحد للمستطيلات
تاريخ	تاريخ[ ]	جدول ببعد واحد للتواريخ
سيارة	سيارة[ ]	جدول ببعد واحد للسيارات
عجلة	عجلة[ ]	جدول ببعد واحد للعجلات

جدول 6 : امثلة عن كيفية كتابة الأنماط المختلفة للجداول

5 – 1 كيفية انشاء الجداول:

تنشأ الجداول باستعمال احدى الطريقتين التاليتين:

- انشاء الجدول عند التصريح به بسرد محتواه، اي التصريح بجدول مهيأ بقيم أولية.
- انشاء الجدول باستعمال التعليمة **شيد**.

تذكير: الجدول متغيرة غير بدائية، فهو متغيرة مهيكلة، لها هيكل خاص يمثل مفهوم الجدول في الواقع المعاش، وهذه الملاحظة تؤدي بنا الى القول بان الجداول تنشأ في الذاكرة النشطة كما هو الشأن بالنسبة للمتغيرات المهيكلة، والمتغيرات التي تنشأ في الذاكرة النشطة لا بد لها من صلة تمكن من الوصول اليها، ومعنى هذا ان الاسم الذي يعطى للجدول انما هو في حقيقته صلة.

الرقم	النص	تعليق
1	طبيعي[ ] جط1 = { 10، 15، 20، 25 } ؛	انشاء جدول طبيعي ذو بعد واحد وبقيم اولية، حجم الجدول 4، جط1 صلة تمكن من الوصول الى الجداول الطبيعية ذات بعد واحد
2	طبيعي[ ] [ ] جط2 ={	انشاء جدول طبيعي ذو بعدين و بقيم اولية، عدد الأسطر 3 و عدد الأعمدة 6، جط2 صلة تمكن من الوصول الى الجداول الطبيعية ذات بعدين
3	{ 10، 15، 20، 25 ، 30، 35}،	
4	{ 11، 16، 21، 26 ، 31، 36}،	
5	{ 12، 17، 22، 27، 32، 37}	
6	} ؛	
7	منطقي[ ] [ ] جم2 ={	انشاء جدول منطقي ذو بعدين، 3 اسطر×3 اعمدة، جم2 صلة تمكن من الوصول الى الجداول المنطقية ذات بعدين
8	{ صحيح، صحيح، خطأ}،	
9	{ صحيح، خطأ، خطأ }،	
10	{ خطأ ، خطأ، خطأ }	
11	} ؛	

جدول 7: التصريح بجداول مهيأة بقيم اولية

5 – 1 – 1 انشاء الجداول بالتصريح بها مهيأة بقيم اولية:

كما هو الشأن بالنسبة للمتغيرات البدائية والمتغيرات المهيكلة، يمكن ان نصرح بجدول يحتوي على قيم اولية، وهذا ما نراه في الجدول 7، ففي مثل هذه الكتابات لا نذكر حجم الأبعاد، **فعدد المعلومات التي تظهر في التصريح هي التي يُستنبط منها حجم كل بعد**، فمثلا :

- في السطر 1، يكون حجم الجدول الطبيعي **جط1** هو 4.
- في الأسطر من 2 الى 6، نرى التصريح بالجدول الطبيعي ذو بعدين، **جط2**، عدد اسطره 3 وعدد اعمدته 6.
- في الأسطر من 7 الى 11، نرى التصريح بالجدول المنطقي ذو بعدين، **جم2**، عدد اسطره 3 وعدد اعمدته 3.

وكما يظهر جليا في الجدول 7، كلما اضفنا بعد، استعملنا العارضتين [ ]، فالعارضة المفتوحة الأولى تمثل ما في البعد الأول، وان وجدت عارضة ثانية مفتوحة فهي تمثل البعد الثاني، وان اتبعت بحاضنة مفتوحة ثالثة فهي للبعد الثالث وهكذا.

اخيرا، لا ننس ان **جط1** (السطر. 1) و **جط2** (السطر. 3) و **جم2** (السطر. 8) هي في حقيقتها صلات، ومجازا نقول انها أسماء جداول، فنقول الجدول **جط1** او الجدول **جط2** الخ..

5 – 1– 2 انشاء الجداول باستعمال العملية شيد.

علمنا مما سبق ان الجدول يُنشأ في الذاكرة النشطة لكونه متغيرة غير بدائية، فمثلا الكتابة **حقيقي[ ] جط؛** تكتف بإنشاء الصلة **جط** فقط، والتي نسميها مجازا اسم الجدول، وهذه الصلة تمكن من الوصول لجدول ذو بعد واحد قادر على احتواء الأعداد الحقيقية، واذا اردنا ان ننشأ فعليا الجدول، فيجب علينا استعمال العملية **شيد** وعلينا ايضا ان نوفر لهذه العملية **حجم** كل بعد حتى تتمكن العملية **شيد** من حجز المساحة الكافية من الذاكرة النشطة للجدول، ويبين الجدول 8 كيفية التصريح بجداول تُنشأ فعليا في الذاكرة النشطة بواسطة التعليمة **شيد**.

التصريح	تعليق
طبيعي[ ] جط1	التصريح بالصلة **جط1** التي تمكن من الوصول لجدول ذو بعد واجد قادر على احتواء الأعداد الطبيعية، الجدول **جط1** لم ينشأ بعد، ومحتوى **جط1** غير معلوم
سلسلة [ ] جس1	التصريح بالصلة جس1 التي تمكن من الوصول لجدول ذو بعد واجد قادر على احتواء سلاسل الحروف، الجدول جس1 لم ينشأ بعد، ومحتوى **جس1** غير معلوم
حقيقي[ ] ج1 = شيد حقيقي[12]	إنشاء جدول ببعد واحد فيه 12 خانة، وكل خانقة في حقيقتها متغيرة من النمط حقيقي، و تمكن الصلة **ج1** من الوصول الى هذا الجدول
طبيعي[ ] [ ] ج2 = شيد طبيعي[5][5]	إنشاء جدول ببعدين، حجم كل بعد 5، ويحتوي الجدول على 25 خانة (او متغيرة من النمط طبيعي)، و في الجدول نجد 5 سطور و 5 اعمدة
منطقي[ ] [ ] ج2 = شيد طبيعي[12][4]	إنشاء جدول ببعدين، حجم البعد الأول 12، ويمثل عدد الأسطر، وحجم البعد الثاني

	هو 4، و يمثل عدد الأعمدة و هو نفسه عدد الخانات في كل سطر، ويحتوي الجدول على44 خانة (او متغيرة من النمط منطقي)
مستطيل[ ] جمس1 = شيد مستطيل[4]	إنشاء جدول ببعد واحد فيه 4 خانات، وكل خانقة قادرة على احتواء المعلومات المتعلقة بمستطيل ما

جدول 8 : بعض التصاريح بإنشاء جداول

5 – 2 كيفية الوصول لخانات الجدول واستغلالها:

علمنا مما سبق ان المتغيرات التي يحتويها اي وعاء ليس لها اسم خاص بها، ويتم الوصول اليها بمعرفة موقعها (او عنوانها)، وعلمنا مما سبق، ان لكل بُعد في الجدول حجم قد حدد مرة واحدة، فمثلا:

- ينتج عن التصريح **حقيقي[ ] ج1 = شيد حقيقي[12]** انشاء جدول ببعد واحد، وحجم البعد هو 12، اي ان الجدول يحتوي على 12 موقع (او خانة أو خلية).
- و ينتج عن التصريح **طبيعي[ ] [ ] ج2 = شيد طبيعي[7][5]** انشاء جدول ببعدين، وحجم البعد الأول هو 7 ويمثل عدد اسطر الجدول، وحجم البعد الثاني هو 5 ويمثل عدد اعمدة الجدول، اي ان الجدول يحتوي على 7×5 موقعا، اي 35 موقعا (او خانة أو خلية).

بما ان خانات الجدول مرتبة، يمكننا استعمال رقم ترتيب الخانة للتدليل عليها، وهذا ما يحدث في لغات الخوارزميات ولغات البرمجة، وعموما، حسب اللغة المستعملة، يمكن للترقيم ان يبدأ بالرقم 1، كما يمكنه ان يبدأ بالرقم 0، وفيما يخصنا، نبدأ الترقيم من 0 كما يظهر في (خطأ: لم يُعثر على مصدر المرجع)، وينجر عن هذا الاختيار ان اول رقم في بعد ما هو 0 وآخر رقم يمكن استعماله في بعد ما هو حجم البعد ناقص واحد.

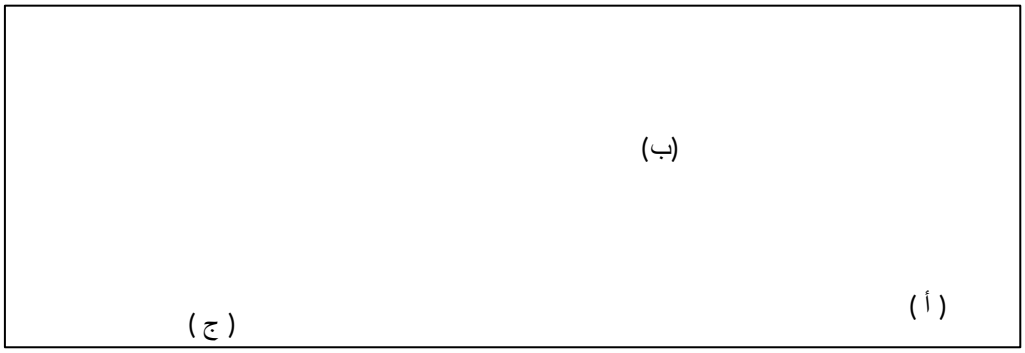
للتدليل على موقع (او ما يمكن ان نسميه عنوان الخانة) ما في الجدول، نُوَفِر رقم ترتيب الخانة في كل بعد من الجدول.

- فإن كان الجدول ذو بعد واحد نكتب اسم الجدول متبوعا بعارضتي البعد الأول وداخل العرضتين نكتب رقم الخانة في البعد، فمثلا، واذا ارتكزنا على التصريح **حقيقي[ ] ج1=شيد حقيقي[6]**
  - الكتابة **ج1[0]** تمثل الخانة الأولى (لا ننس ان الترقيم يبدأ من 0)، والكتابة تدل على عنوان الخانة (او موقع الخانة)، ومجازا نقول ان الكتابة **ج1[0]** هي اسم المتغيرة الممثلة للخانة الأولى في الجدول.
  - الكتابة **ج1[5]** تمثل الخانة الأخيرة،(لا ننس ان الترقيم يبدأ من 0، و بهذا يكون آخر رقم يمكن استعماله في بُعدٍ ما هو حجم البُعد ناقص واحد، أي 6 – 1).

- الكتابة **ج1**[2] تمثل الخانة الثالثة، ومجازا هي اسم المتغيرة الممثلة للخانة الثالثة في الجدول.
- الكتابة **ج1**[6] غير صحيحة، فلا وجود في الجدول للخانة السابعة (لا ننس ان الترقيم يبدأ من 0)
- الكتابة **ج1**[−1] غير صحيحة، فالترقيم يبدأ من 0، وهو موجب غير سالب.

- فان كان الجدول ذو بعدين نكتب اسم الجدول متبوعا بعارضتي البعد الأول وداخل العرضتين نكتب رقم الخانة في البعد الأول (اي رقم السطر الذي يحتوي على الخانة) ثم عارضتي البعد الثاني وداخل العرضتين نكتب رقم الخانة في البعد الثاني (اي رقم العمود الذي يحتوي على الخانة)، وكما اشرنا اليه يمثل البعد الأول الأسطر والبعد الثاني الأعمدة ، فمثلا، اذا ارتكزنا على التصريح **حقيقي[ ] [ ] ج2 = شيد حقيقي[6] [7] (خطأ: لم يُعثر على مصدر المرجع-ج)**

- الكتابة **ج2**[0] [0] تمثل الخانة الأولى الموجودة في السطر الأول (الرقم 0) والعمود الأول (الرقم 0)، ومجازا نقول ان الكتابة **ج2**[0] [0] هي اسم المتغيرة الممثلة للخانة الموجودة في السطر الأول والعمود الأول.
- الكتابة **ج2**[0] [3] هي الخانة في السطر الأول (الرقم 0) والعمود الرابع (الرقم 3).
- الكتابة **ج2**[5] [6] تمثل الخانة الأخيرة الموجودة في السطر السادس (الرقم 5) والعمود السابع (الرقم 6)
- الكتابة **ج2**[0] [7] خاطئة، العمود الثامن غير موجود، أرقام الأعمدة بين 0 و 6.
- الكتابة **ج2**[6] [6] خاطئة، السطر السابع غير موجود، أرقام الأسطر بين 0 و 5



الشكل 4 :ترقيم خانات الجداول، (أ)، (ب): جدول ببعد واحد، (ج) جدول ببعدين

الرمز	الأولوية	الاتجاه	ملاحظة
( )	0	اليمين	طلب تنفيذ خوارزم او طلب تقييم عبارة
[ ]	0	اليمين	تحديد المواقع في الجداول
.	0	اليمين	الولوج والوصول الى سمة في متغيرة مهيكلة
!	1	اليسار	العملية المنطقية لا،

* / %	2	اليمين	الضرب، القسمة وبقية القسمة بين عددين طبيعيين
+ -	3	اليمين	جمع و طرح الأعداد
+	4	اليمين	جمع سلاسل الحروف
> >= =< <	5	اليمين	اصغر، اصغر او يساوي، اكبر، اكبر او يساوي،
== =!	6	اليمين	يساوي، لا يساوي
&&	7	اليمين	العطف (واو)
||	8	اليمين	التخيير (أو)
=	9	اليسار	تعيين (وضع، حمل، شحن)

جدول 9 : سلم الاولويات و آلية اختيار العمليات ذات الأولوية المتساوية

5 – 3 الجداول في العبارات

العريضتان **[ ]** اللتان تلازمان اسم الجدول (اي الصلة) تمثل في حقيقة امرها عملية من العمليات التي تُبنى عليها العبارات، وهدف هذه العملية هو تحديد عنوان (او موقع) الخانة في الجدول والتمكين من الوصول الى خانة ما لاستغلالها في العبارات، وترتكز. هذه العملية على معاملين:

- المعامل الأول هو اسم الجدول (اي الصلة وهي تحتوي على موقع الجدول في الذاكرة).
- و المعامل الثاني هو رقم ترتيب الخانة في البعد، وبشكل ادق، فالمعامل الثاني هو اي عبارة ينته تقييمها الى عدد طبيعي اكبر او يساوي 0.

في سلم أولويات العمليات (الجدول 9)، حُدِد لعملية تحديد مواقع الخانات الممثلة بالرمزين [ ] ، نفس الاولوية التي حددت لعملية طلب تنفيذ وظيفة الممثلة بالقوسين ولعملية الولوج في متغيرة مهيكلة الممثلة بالنقطة، ويظهر. النص 23، خوارزم ذكرت فيه بعض العبارات الخفيفة التي تظهر عملية تحديد مواقع الخانات في العبارات، وندرس فيما يلي بعض منها.

	نص الخوارزم	تعليق
	خوارزم المثال_1_الفصل_6 {	
1	اجراء اساسي(){	
2	طبيعي س=2، ع=0	التصريح بقيمة اولية لمتغيرتين من النمط طبيعي
3	طبيعي[ ] جط1_1 = شيد طبيعي[20] ؛	إنشاء جدول ببعد واحد يتسع لعشرين معلومة من النمط طبيعي
4	طبيعي[ ] [ ] جط2_1 = شيد طبيعي[10] [15]	إنشاء جدول ببعدين يتسع لمئة وخمسين معلومة (10 اسطر على 15 عمود) من النمط طبيعي
5	طبيعي[ ] جط1_2 ؛	التصريح بصلة تستعمل مع الجداول من النمط طبيعي ببعد واحد
6	طبيعي[ ][ ] جط2_2 ؛	التصريح بصلة تستعمل مع الجداول من النمط طبيعي ببعدين
7	جط1_1[0] = 10 ؛	شحن الخانة الأولى من الجدول جط1_1 بالقيمة 10،
8	جط1_1[23%5] = 40 ؛	شحن الخانة الرابعة من الجدول جط1_1 بالقيمة 40، (رقم الخانة الرابعة هو 3)
9	جط1_1[س] = 30 ؛	شحن الخانة الثالثة من الجدول جط1_1 بالقيمة 30، (قيمة س هي 2 و رقم الخانة الثالثة هو 2)
10	جط1_1[س - 1] = جط1_1[س -2] * 2 ؛	شحن الخانة الثانية بقيمة الخانة الأولى مضروبة في 2 اي 10 * 2، اي 20
11	جط1_1[19] = 100 ؛	شحن الخانة الأخيرة بالقيمة 100(الترقيم يبدأ من 0 و ينته في 19، اي حجم البعد - 1)
12	س = جط1_1 [س]*2	تغيير قيمة س لتصبح 60 (ما في جط1_1[س] هو ما في جط1_1[2] اي 30)
13	جط1_1[س] = 0	خطأ خطير: محاولة الوصول الى الخانة الواحدة و الستون (الرقم 60)، و هي خانة غير موجودة
14	جط1_2[0] = 5	خطأ خطير: محاولة الوصول لخانة من جدول غير معلوم، الصلة جط1_2 لم تربط باي جدول
15	جط1_2 = جط2_1	خطأ خطير: عدم توافق الأبعاد، جط1_2 صلة لجدول ببعد واحد، وجط2_1 صلة لجدول ببعدين
16	جط1_2 = جط1_1	الجدول جط1_1 له الأن صلتين: جط1_1 و جط1_2
17	جط1_2[س/6] =16	شحن الخانة الحادية عشر بالقيمة 16 (س/6 تساوي 10 كون المتغيرة س تحتوي على القيمة 60)
18	جط1_2[4][6] = 12	خطأ خطير: عدم توافق الأبعاد، الجدول جط1_2 ذو بعد واحد و ليس ذو بعدين

19	جط2_2[4][6]	خطأ خطير: الكتابة صحيحة لكنها غير واقعية، الصلة جط2_2 لم تربط بأي جدول حقيقي معلوم
20	جط2_2 = جط2_1	الجدول الموجود جط2_1 له الآن صلتين: جط2_1 و جط2_2
21	جط2_2[4][6] = 12	الوصول الى خانة السطر 5 و العمود 7 من الجدول جط2_1 عبر جط2_2 و شحنها بالقيمة 12
22	ع = س = جط2_1[4][6]	شحن المتغيرة س بما في الخانة (سطر رقم 4، عمود رقم 6) اي القيمة 12، ثم شحن ع بما في س
23	جط2_1[س][ع] = جط2_1[س][ع]*2	شحن الخانة (سطر رقم س اي 12، عمود رقم ع اي 12) بما فيها في 2، اي 12*2 اي 24
24	جط2_1[س+1][ع-1] = جط2_1[س][ع]*س- 10	شحن الخانة (سطر رقم س+1 اي 13، عمود رقم ع -1، اي 11) بما في الخانة (سطر رقم س اي 12، عمود رقم ع اي 12) اي 12، في س، اي 12*12، اي 144 ثم نقص 10، اي ان النتيجة هي 134 في الخانة (سطر 13، عمود 11)
25	جط2_2[س*3][6] = 12	خطأ خطير: السطر س*3 اي 12*3 اي 36 غير موجود في الجدول جط2_2 الذي هو ايضا الجدول جط2_1
26	} // نهاية الإجراء اساسي	
} // نهاية الخوارزم المثال_1_الفصل_6		

النص 23: عبارات خفيفة لكيفية استغلال العملية [ ]

أ – عبارة السطر 8: في هذه العبارة تُقيم اولا عملية تحديد المواقع، وفي اطارها تُقيم العبارة 23%5، والتي تفضي الى النتيجة 3، اي ان الكتابة **جط1_1[23%5]=40** تصبح في المرحلة التالية **جط 1_1[3]=40**، وفي المرحلة الثانية تقيم العبارة **جط1_1[3]** فنتحصل على متغيرة ظرفية تمثل الخانة الرابعة في الجدول، ونسميها **مظ1،** فتصبح العبارة على الشكل مظ1=40، وفي هذه المرحلة، يمكننا الاكتفاء باستعمال **جط1_1[3]** وكأنها اسم الخانة الرابعة في الجدول ولا حاجة للغوص في تقييمها لاستنتاج المتغيرة الظرفية **مظ1**، اي اننا لا نستعمل المتغيرة الظرفية **مظ1،** وتنته عملية تحديد المواقع في الجداول عندما تصبح العبارة داخل الحاضنتين قيمة طبيعية، وبهذه العقلية تكون العبارة **جط1_1[3]=40** هي عبارة المرحلة الثانية، وفيها تُنتخب عملية الشحن = للتنفيذ، وتكون نتيجة التقييم هي القيمة 40 في المتغيرة (او الخانة الرابعة او الخلية الرابعة) **جط1_1[3]**.

ب – عبارة السطر 10:في هذه العبارة، نلاحظ وجود عمليات عديدة في المرحلة الأولى من مراحل تقييمها (الجدول 10) وهي على التوالي: عمليتين لتحديد المواقع في الجداول، عملية الضرب، عمليتين للطرح وعملية الشحن، وفي مثل هذه الحالة، وحسب سلم الأولويات (جدول 9) تكون العملية الأولى التي يُشرع في تنفيذها هي عملية تحديد المواقع في الجداول، لكن وجود عمليتين من هذا النوع يجبرنا ان نرجع الى سلم الأولويات لنرى الاتجاه الذي نسلكه عندما تكون اكثر من عملية بنفس الأولوية مرشحة للتنفيذ، وحسب الجدول 9، نرى ان العملية التي نبدأ بها هي التي على اليمين، وهكذا تكون اول عملية يُشرع في تقييمها هي العملية **جط1_1[س – 1]،**

و نلاحظ ان عملية تحديد المواقع في الجداول التي نشرع في تنفيذها تشتمل على معامل ثان هو نفسه عبارة، وهنا، في اطار تنفيذ عملية تحديد المواقع في الجداول نشرع اولا في تقييم العبارة **س – 1** (الجدول 11)، و دون ان نطيل الشرح (انظر الجزء الأول، الفصل الثاني عشر) يفضي تقييم العبارة **س – 1** الى القيمة 1 مخزنة في متغيرة ظرفية، ونذكر أن المتغيرة **س** تحتوي على القيمة 2 (السطر 2)، وهكذا بعد تقييم العبارة **س – 1** تصبح الكتابة **جط1_1[س – 1]** تساوي الكتابة **جط1_1[1]،** وهنا ينته التقييم كما اشرنا الى ذلك من قبل، لكوننا نعتبر ان الكتابة **جط1_1[1]** هي

مجازا اسم المتغيرة الثانية والتي تحتوي على نتيجة التقييم، ونتيجة التقييم كما تظهر في السطر 2 من الجدول 10 هي القيمة 40 مخزنة في المتغيرة **جط1_1[1].**

تذكير من الجزء الأول الفصل 12: عندما نتتبع عملية تقييم عبارة باستعمال جداول التقييم، نتحصل في مرحلة ما على قيمة تقييم عبارة جزئية (مثلا نتيجة تقييم **جط1_1[س – 2]** في السطر. 2 من الجدول 10) ، ففي المرحلة التالية عندما نريد ان نكتب الصيغة الجديدة للعبارة، نستبدل العبارة الجزئية التي قيمت في المرحلة السابقة بنتيجتها، ونكتب النتيجة اما باستعمال اسم المتغيرة الحاملة للنتيجة او قيمة النتيجة، فمثلا في السطر. 3:

– اذا استعملنا اسم المتغيرة تكون الكتابة على الشكل: جط1_1[1] = جط1_1[0] * 2

– اذا استعملنا القيمة تكون الكتابة على الشكل: جط1_1[1] = 10 * 2

وعندما يمكن تغيير القيمة في المراحل التالية (مثلا، القيمة تظهر. على يمين العملية =)، فهنا لا نستعمل القيمة بل نستعمل فقط اسم المتغيرة، ففي السطر 2 او السطر 3 لا يمكن كتابة القيمة 40 في مكان اسم المتغيرة جط1_1[1]

بعد الانتهاء من تقييم العبارة الجزئية **جط1_1[س – 1]** ننتقل الى تقييم العبارة الجزئية **جط1_1[س –2]**، و نتيجة تقييم هذه العبارة الأخيرة يفضي الى القيمة 10 مخزنة في المتغيرة **جط1_1[0].**

في المرحلة الثالثة نعيد كتابة عبارة المرحلة الثانية ونعوض العبارة الجزئية التي قيمت اما بقيمة تقييمها او باسم المتغيرة، ولا نستعمل القيمة ان كانت ستغير في المتغيرة الحاملة لها، وكما يظهر في السطر الثالث عوضنا الكتابة جط1_1[س –2] بقيمتها، اي 10، فأصبحت العبارة في المرحلة الثالثة على الشكل الذي يظهر. في السطر الثالث، وفي هذه المرحلة تنتخب عملية الضرب فتقيم العبارة الجزئية **10 * 2** فتكون النتيجة هي القيمة 20 مخزنة هذه المرة في متغيرة ظرفية، وفي المرحلة الرابعة نعوض الكتابة **10 * 2** بقيمتها لتصبح عبارة المرحلة الرابعة هي **جط1_1[1] =20**، وتقييم عبارة المرحلة الرابعة يفضي الى القيمة 20 مخزنة في المتغيرة **جط1_1[1]** ، ونلاحظ ان القيمة التي كانت مخزنة في المتغيرة **جط1_1[1]** في المرحلة الأولى، اي 40، قد غيرت في المرحلة الرابعة.

مرحلة	العبارة قيد التقييم	المرشحة	العبارة	القيمة	الوعاء
1	جط1_1[س – 1] = جط1_1[س –2] * 2	[ ]	جط1_1[س – 1]	40	جط1_1[1]
2	جط1_1[1] = جط1_1[س –2] * 2	[ ]	جط1_1[س – 2]	10	جط1_1[0]
3	جط1_1[1] = 10 * 2	*	10 * 2	20	مظ3
4	جط1_1[1] =20	=	جط1_1[1] =20	20	جط1_1[1]
النتيجة	**20**				**جط1_1[1]**

جدول 10 : مراحل تقييم عبارة السطر 10 من النص 23

المرحلة	العبارة قيد التقييم	العملية المرشحة	العبارة	القيمة	الوعاء
1 : 1	س – 1	–	س – 1	1	مظ1
النتيجة	1				مظ1

جدول 11 : تقييم العبارة س–1 الموجود داخل عملية تحديد المواقع في عبارة 10 من النص 23

المرحلة	العبارة قيد التقييم	العملية المرشحة	العبارة	القيمة	الوعاء
1 : 1	س – 2	–	س – 2	0	مظ2
النتيجة	0				مظ2

جدول 12 : تقييم العبارة س–2 الموجود داخل عملية تحديد المواقع في عبارة 10 من النص 23

5 – 4 استعمال العبارات في تحديد حجم كل بعد عند انشاء الجدول.

في سياق انشاء جدول باستعمال عملية **شيد** ، يمكن ان نصف كمعامل ثان لعملية تحديد الخانات، عبارات يمكن ان تكون معقدة جدا، واهم شيء في هذه العبارات ان تقييمها يجب ان يفضي الى عدد طبيعي اكبر من 0، و يحتوي النص 24 والنص 25 والنص 26 على بعض الأمثلة التي ندرسها. فيما يلي.

5 – 4 – 1 دراسة النص 24

يحتوي النص 24 على الإجراء اساسي و الوظيفة **حصل_حجم_بُعد_جدول** (من السطر 1 الى السطر 9)، و هدف هذه الأخيرة هو التفاعل مع المستعمل للحصول على قيمة اكبر من 0 تكون حجما لبعد ما في جدول ما.

	نص الخوارزم	تعليق
	خوارزم المثال_2_الفصل_6 {	
1	طبيعي حصل_حجم_بعد_جدول (سلسلة رسالة){	هذه الوظيفة تتعامل مع المستعمل ليعطيها عددا اكبر من 0، و معنى العدد الذي يطلب من المستعمل يحمله المدخل رسالة الذي تمثله المتغيرة المحلية رسالة التي هي من النمط سلسة (سلسلة الحروف)، و لا تنته هذه الوظيفة حتى يعط المستعمل قيمة مقبولة، اي اكبر من 0
2	طبيعي س؛	
3	ما دام (صحيح) {	
4	اكتب رسالة	
5	اقرأ س	
6	اذا كان (س < 0) ارجع س؛	
7	اكتب "خطأ، الحجم يجب ان يكون اكبر من 0"	
8	}	
9	}	
10	اجراء اساسي(){	
11	طبيعي س=10، ع =20 ؛	التصريح بمتغيرتين معلومتين القيمة
12	طبيعي س1، ع1؛	التصريح بمتغيرتين غير معلومتين القيمة
13	طبيعي[ ] جط1_1 = شيد طبيعي[س]؛	إنشاء جدول ببعد واحد يتسع لعشر قيم طبيعية (قيمة س هي 10)
14	طبيعي[ ] [ ] جط2_1 = شيد طبيعي[س] [ع]	إنشاء جدول ببعدين يتسع لمئتي قيمة طبيعية (قيمة س 10 و ع 20)
15	طبيعي[ ] جط1_2 ، جط1_3 ؛	التصريح بصلتين غير مرتبطتين بأي جدول ببعد واحد
16	طبيعي[ ] [ ] جط2_2 ، جط2_3 ؛	التصريح بصلتين غير مرتبطتين بأي جدول ببعدين
17	س = حصل_حجم_بعد_جدول ("اعطني حجم الجدول ")	
18	جط1_2 = شيد طبيعي[س]	إنشاء جدول ببعد واحد حجمه يكون حسب القيمة المخزنة في المتغيرة س
19	س = حصل_حجم_بعد_جدول("اعطني حجم البعد الأول")	
20	ع = حصل_حجم_بعد_جدول("اعطني حجم البعد الثاني")	
21	جط2_2 = شيد طبيعي[س] [ع]	إنشاء جدول ببعدين حجم كل بعد يكون حسب القيم المخزنة في س و ع
22	جط1_3 = شيد طبيعي[س1]	كتابة خطيرة جدا: حجم البعد غير معلوم، يمكن ان يكون سالبا
23	س1 = حصل_حجم_بعد_جدول("اعطني حجم البعد الأول")	
24	جط2_2 = شيد طبيعي[س1] [ع1]	كتابة خطيرة جدا: حجم البعد الثاني غير معلوم، يمكن ان يكون سالبا
25	} // نهاية الإجراء اساسي	

} // نهاية الخوارزم المثال_2_الفصل_6	

النص 24: عبارات يمكن ان تكون معقدة في المعامل الثاني للعملية [ ]

من بين الأمثلة التي تبين كيفية استعمال العبارات كمعامل ثان لعملية تحديد الخانات في سياق انشاء جدول ما، تعليمة السطر. 18، وفيها نرى ان العبارة داخل العارضتين عبارة سهلة متمثلة في المتغيرة **س**، والكتابة الظاهرة في السطرين 17 و 18 واضحة وسهلة الاستيعاب، فقيمة **س** تحدد في السطر 17 وتستعمل لإنشاء الجدول في السطر 18، فمثلا اذا ادخل المستعمل القيمة 134، تصبح الكتابة **شيد طبيعي[س]** بعد تقييم العبارة البسيطة **س** على الشكل **شيد طبيعي[134].**

لكون عملية تحديد الخانات ([ ]) تحتمل عبارات معقدة كمعامل ثان، يمكن ان نجعل السطرين 17 و 18 سطرا واحدا فيه يطلب تنفيذ الوظيفة **حصل_حجم_بعد_جدول**، ويكون هذا السطر على الشكل التالي:

جط1_2 = شيد طبيعي[حصل_حجم_بعد_جدول ("اعطني حجم الجدول ببعد واحد")]

و نذكر انه عندما نستعمل العبارات داخل عملية تحديد عدد الخانات في البعد، يجب على العبارة ان تفضي الى قيمة طبيعية معلومة، و إلا، فالكتابة، مهما كانت صحيحة لفظيا، يكون معناها غير متحكم فيه، فتصبح خطيرة جدا ووَبَالًا على سلوك الخوارزم، وهذا ما نبينه في الأسطر 22 و 24، فنرى استعمال المتغيرة ع1 التي لم تهيأ بأي قيمة بعد، فمحتوى ع1 مجهول.

	النص	تعليق
	خوارزم المثال_3_الفصل_6	
1	طبيعي اختر_خانة(سلسلة رسالة؛ طبيعي حجم){	تتفاعل هذه الوظيفة مع المستعمل لتطلب منه رقم خانة في حالة بعد واحد، و نلاحظ ان معنى الرقم تفسره الرسالة التي توضع في المدخل الأول، فعلى المستعمل ان يحدد عبر الرسالة التي تكتب للمستعمل نوعية الرقم اذي يدخله استجابة لطلب الوظيفة
2	طبيعي رقم ؛	
3	ما دام (صحيح){	
4	اكتب رسالة ؛	
5	اقرأ رقم ؛	
6	اذا كان (رقم <= 0 && رقم > حجم) ارجع رقم؛	
7	اكتب("خطأ، الرقم يجب ان يكون من 0 الى "+(حجم – 1))	
8	}	
9	} // نهاية الوظيفة اختر_خانة	
10		
11	اجراء اساسي(){	
12	طبيعي س=10، خيار، قيمة؛	التصريح بمتغيرات ثلاث
13	طبيعي[ ] جط1 = شيد طبيعي[س]؛	التصريح بجدول طبيعي ذو بعد واحد،
14		
15	اكتب ("هل تريد ان تملأ جدولا ذو بعد واحد حجمه "+ س)	بداية التفاعل مع المستعمل
16	اكتب ("نعم: ادخل 1، لا: ادخل 0 ")	
17	اقرأ (خيار)	
18	اذاكان (خيار == 1) {	
19	مادام (صحيح){	
20	اكتب "ادخل القيمة التي تريد وضعها في الجدول"	
21	اقرأ قيمة ؛	
22	رقم = اختر_خانة("ادخل رقم الخانة"، س)	
23	جط1[رقم] = قيمة	
24	اكتب ("هل تريد ان تتابع العملية ")	
25	اكتب ("نعم: ادخل 1، لا: ادخل 0 ")	
26	اقرأ (خيار)	
27	اذاكان (خيار == 0) غادر	
28	} // نهاية مادام	
29	} // نهاية اذاكان	
30	اكتب ("شكرا على الزيارة... الى اللقاء..") ؛	

31	} // نهاية الإجراء اساسي
} // نهاية الخوارزم المثال_3_الفصل_6	

النص 25: خوارزم تعمير انتقائي لجدول ذو بعد واحد

5 – 4 – 2 دراسة النص 25

يحتوي النص 25 على خوارزم هدفه التعمير الانتقائي لجدول ذو بعد واحد صُرح به في السطر 13، وحُدد حجمه عن طريق القيمة التي وضعت في المتغيرة **س** (السطر 12) ، والتعمير الانتقائي الذي نقصده هو ان المستعمل يختار الخانة التي يَضَع فيها القيمة التي يريدها، وفي مثل هذا الخوارزم يصبح استعمال العبارات داخل عملية تحديد خانات الجدول امرا حتميا لبلوغ هدف الخوارزم.

تحتوي الأسطر من 1 الى 9 على الوظيفة **اختر_خانة** وهدفها التفاعل مع المستعمل للحصول على رقم الخانة التي يريد المستعمل ان يعمرها، وتتكفل هذه الوظيفة بالتأكد من ان الرقم الذي يدخله المستعمل يكون اكبر او يساوي 0 وكذلك يكون اصغر من حجم البعد لأن الأرقام تبدأ من 0 وتنته في الرقم الذي يساوي حجم البعد ناقص واحد، فمثلا اذا كان حجم البعد هو 100، فأكبر رقم يمكن استعماله في هذا البعد هو 100 – 1 اي 99، وهو رقم آخر خانة في البعد، وآخر خانة في الجدول اذا كان ذو بعد واحد، ولتتمكن الوظيفة من التأكد من صحة الرقم، تطلب هذه الوظيفة من مستغليها توفير حجم الجدول (اي حجم البعد الوحيد).

يقوم الإجراء **اساسي** (من السطر 11 الى السطر 31) من تسيير التفاعل مع المستعمل للحصول على القيمة التي يريد ان يضعها في الجدول ثم الموقع في الجدول الذي يريد ان توضع فيه القيمة، والسلوك العام للإجراء اساسي هو كما يلي:

- اولا يتحقق الإجراء من رغبة المستعمل في تعمير الجدول (الأسطر من 15 الى 18)
- ان اراد المستعمل تعمير الجدول، يدخل الإجراء في تكرار للعمليات التالية
  - يطلب الإجراء القيمة التي يريد المستعمل وضعها في الجدول (السطرين 20 و 21)
  - يطلب الإجراء عبر الوظيفة **اختر_خانة** افادته برقم الخانة التي سوف تُخزن فيها القيمة التي تحصل عليها الإجراء من قبل المستعمل، ونلاحظ ان الإجراء اساسي يوفر للوظيفة **اختر_خانة** معلومتين:
    - الرسالة التي سوف تظهر على شاشة المستعمل، والتي تبين للمستعمل معنى القيمة المطلوبة منه.

- القيمة الأقصى التي لا يجب على المستعمل تجاوزها، وفيما يخصنا هذه القيمة هي القيمة المخزنة في المتغيرة **س** (السطر 12)، وهذه القيمة هي نفسها القيمة التي بها انشأ الجدول (السطر 13) الذي يتم تعميره.

- بعد الحصول على القيمة ورقم الموقع، يتم عبر تعليمة السطر 23 تخزين القيمة في الجدول وفي الموقع المحدد سلفا.
- عند الانتهاء من تخزين القيمة في الخانة التي حددت، يطلب الخوارزم من المستعمل رغبته: هل يريد الاستمرار في عملية التعمير او المغادرة، وهنا اذا اختار المستعمل المغادرة ينته الإجراء اساسي من التنفيذ بعد ارساله النص **شكرا على الزيارة... الى اللقاء..** الى المستعمل عبر الشاشة.

5 – 4 – 3 دراسة النص 26

يحتوي النص 26 على خوارزم هدفه التعمير الانتقائي لجدول ذو بعدين، وقد صُرح بالجدول ذو بعدين في السطر 24، ونلاحظ ان حجم كل بعد يستنتج من قيمة في متغيرة ما، فعدد اسطر الجدول تحدده قيمة المتغيرة "**س**" التي صُرح بها بقيمة اولية في السطر 22، وعدد اعمدة الجدول تحدده قيمة المتغيرة "**ع**" التي صرح بها بقيمة اولية في السطر 22,

لكي يتمكن المستعمل من تعمير جدول ذو بعدين يجب عليه توفير ما يلي:

- القيمة التي ستوضع في خانة ما من خانات الجدول.
- موقع (او عنوان) الخانة المستهدف من عملية التعمير، وفي حالة جدول ذو بعدين يتكون عنوان الخانة من عددين: رقم السطر الذي تقع فيه الخانة ورقم العمود الذي تقع فيه الخانة ، اي ان المعلومة الممثلة لعنوان الخانة معلومة غير بدائية، كما هو الشأن مع جدول ذو بعد واحد، فالمعلومة الممثلة لعنوان خانة في جدول ذو بعدين معلومة مهيكلة في حقيقتها، وهذا هو الذي يفسر وجود التعريف بالنمط **خانة** في الأسطر الأولى من النص 26.

مهمة الحصول على موقع خانة في جدول ذو بعدين اوكلت الى الوظيفة **اختر_خانة2** (من السطر 14 الى السطر 19)، ونرى جليا في رأس هذه الوظيفة انها تشترط لعملها قيمتين طبيعيتين هما :

- القيمة الأولى هي حجم البعد الأول (عدد الأسطر) و تعطى للوظيفة عبر المدخل حجم_ب1
- القيمة الثانية هي حجم البعد الثاني (عدد الأعمدة) و تعطى للوظيفة عبر المدخل حجم_ب2

و بفضل هاتين المعلومتين تتحقق الوظيفة من صحة ما يدخله المستعمل، فاذا ادخل قيمة اقل من 0 او قيمة اكبر او تساوي حجم البعد، فلن تقبل الوظيفة القيمة المدخلة وتطلب اعادة ادخال

قيمة صحيحة، ونلاحظ ان مثل هذا السلوك نجده في الوظيفة **اختر_خانة** التي درسناها في النص 25، و لذا نرى انه في الوظيفة **اختر_خانة2** لم نعاود كتابة السلوك الموجود في الوظيفة **اختر_خانة** بل قامت الوظيفة **اختر_خانة2** باستغلال الوظيفة **اختر_خانة** في كل بعد للحصول على قيمة كل معلومة متعلقة بموقع الخانة، ففي السطر 16 تطلب **اختر_خانة2** من **اختر_خانة** ان تجلب لها رقم السطر، وفي السطر 17 تطلب **اختر_خانة2** من **اختر_خانة** ان تجلب لها رقم العمود، وفي كل طلب توفر الوظيفة **اختر_خانة2** للوظيفة **اختر_خانة** المعلومات المناسبة.

حتى تتمكن الوظيفة **اختر_خانة2** من ارجاع المعلومة المهيكلة التي تحتوي على موقع الخانة في الجدول، تقوم الوظيفة **اختر_خانة2** في السطر 15 بإنشاء المتغيرة **خ** من النمط **خانة** (السطر 1)، و تضع في السمتين **س** و **ع** (السطرين 16 ز 17) رقم السطر ورقم العمود اللذين ادخلهما المستعمل، وعند انتهاء عمل **اختر_خانة2** ترجع هذه الأخيرة المتغيرة **خ،** وبشكل ادق ترجع قيمة الصلة **خ.**

سلوك الإجراء الأساسي في هذا النص شبيه بسلوك الإجراء اساسي في النص 25، ففي اول الأمر ينشأ جدولا ذو بعدين(السطر 24)، وحجم البعدين تتحكم فيه القيم التي وضعت في المتغيرتين "**س**" و "**ع**" (السطر 22)، وما هو جديد في هذا النص هو استعمال الصلة "**خ**" (السطر 23) من النمط **خانة** لالتقاط ما ترجعه الوظيفة **اختر_خانة2** عند طلب تشغيلها (السطر 32)، وبفضل سمات المتغيرة "خ"، يتمكن الخوارزم من الوصول الى خانة في الجدول **جط2** (السطر 33) ليضع فيها القيمة التي اراد المستعمل ان توضع في الجدول.

	النص	تعليق
1	عرف نمط خانة {	التعريف بالنمط خانة الذي تستعمله الوظيفة اختر_خانة2 للحصول على رقم السطر و العمود في حالة جدول ذو بعدين
2	طبيعي س ، ع	
3	}	
خوارزم المثال_4_الفصل_6 {		
4	طبيعي اختر_خانة(سلسلة رسالة؛ طبيعي حجم){	تتفاعل هذه الوظيفة مع المستعمل لتطلب منه رقم خانة في حالة بعد واحد، و نلاحظ ان معنى الرقم تفسره الرسالة التي توضع في المدخل الأول، فعلى المستعمل ان يحدد عبر الرسالة التي تكتب للمستعمل نوعية الرقم اذي يدخله استجابة لطلب الوظيفة
5	طبيعي رقم ؛	
6	ما دام (ضحيح){	
7	اكتب رسالة ؛	
8	اقرأ رقم ؛	
9	اذا كان (رقم <= 0 &&& رقم > حجم) ارجع رقم؛	
10	اكتب("خطأ، الرقم يجب ان يكون من 0 الى "+ اكبر_رقم)	
11	}	
12	} // نهاية الوظيفة اختر_خانة	
13		
14	خانة اختر_خانة2(طبيعي حجم_ب1 ، حجم_ب2){	
15	خانة خ = شيد (خانة) ؛	
16	خ.س = اختر_خانة("اعطني رقم السطر"، حجم_ب1) ؛	
17	خ.ع = اختر_خانة("اعطني رقم العمود"، حجم_ب2) ؛	
18	ارجع خ ؛	
19	} // نهاية الوظيفة اختر_خانة2	
20		
21	اجراء اساسي(){	
22	طبيعي س=10، ع =20 ، خيار، قيمة ؛	
23	خانة خ ؛	
24	طبيعي[ ] [ ] جط2 = شيد طبيعي[س] [ع]	

25	اكتب ("هل تريد ان تملأ جدولا ذو بعدين: "+ س+ " على " + ع)	
26	اكتب ("نعم: ادخل 1، لا: ادخل 0 ")	
27	اقرأ (خيار)	
28	اذاكان (خيار == 1) {	
29	مادام (صحيح){	
30	اكتب "ادخل القيمة التي تريد وضعها في الجدول"	
31	اقرأ قيمة ؛	
32	خ = اختر_خانة2(س، ع)	
33	جط2[خ.س] [خ.ع] = قيمة	العبارة داخل العريضتين تمكن من الوصول الى السمات س و ع
34	اكتب ("هل تريد ان تتابع العملية ")	
35	اكتب ("نعم: ادخل 1، لا: ادخل 0 ")	
36	اقرأ (خيار)	
37	اذاكان (خيار == 0) غادر	
38	} // نهاية مادام	
39	} // نهاية اذاكان	
40	اكتب ("شكرا على الزيارة... الى اللقاء..") ؛	
41	} // نهاية الإجراء اساسي	
	} // نهاية الخوارزم المثال_4_الفصل_6	

النص 26: خوارزم تعمير انتقائي لجدول ذو بعدين

6 – استغلال السمات في جداول المعلومات المهيكلة.

كما اشرنا اليه من قبل، يمكن ان يكون النمط المرتبط بالجدول نمطا مهيكلا، فمثلا الكتابة **مستطيل [ ]** تعبر عن الجداول ذو بعد واحد والقادرة على احتواء المعلومات الممثلة للمستطيلات.

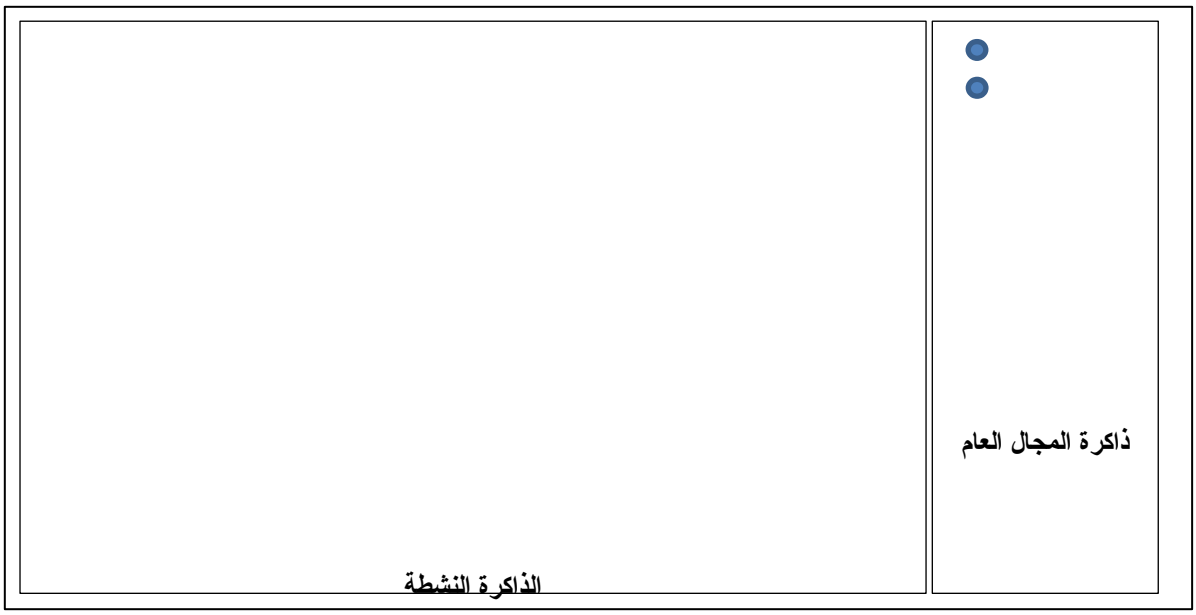
يحتوي السطر 4 من النص 27 على التصريح بجدول للمستطيلات، ذو بعد واحد ، وحجم الجدول هو 6، اي ان هذا الجدول يمكنه احتواء 6 مستطيلات، وكل خانة تمثل مستطيلا واحدا.

كما اشرنا اليه سابقا، فان كل متغيرة من نمط غير بدائي، لا بد ان تنشأ باستعمال التعليمة **شيد**، فالجدول متغيرة غير بدائية ينشأ بالعملية **شيد**، كما يظهر ذلك من خلال السطر 4، وفي نفس السطر او الأسطر التي تلي لا نرى ان المستطيلات الست للجدول قد انشأت لعدم وجود تعليمة تشييد لأي مستطيل، ونفس الملاحظة تنطبق على السطر 5 الذي ينشأ فيه جدول مستطيلات ذو بعدين، ولا نرى اي تعليمات لأنشاء المستطيلات التابعة للجدول ذو بعدين، وهنا نجد انفسنا امام الطريقتين التاليتين:

- التعليمة شيد التي تقوم بإنشاء الجدول هي التي تقوم آليا بإنشاء المستطيلات التابعة للجدول، وفي هذه الحالة لا يلزمنا كتابة تعليمات تشييد المستطيلات، وهذا الأمر ينطبق على جداول المتغيرات مهيكلة.
- التعليمة شيد تقوم فقط بإنشاء الجدول، وعلى كاتب الخوارزم ان يكتب صراحة كل التعليمات التي تقوم بتشييد المستطيلات وتربط كل مستطيل مع خانة من خانات الجدول، اي ان الجدول الذي انشأ بفعل تعليمة التشييد هو جدول تحتوي خاناته على صلات تُربط فيما بعد، بالمتغيرات الطارئة الممثلة للمستطيلات، وهذا يعني ان جدول المستطيلات يُنشأ في مرحلتين: مرحلة انشاء جدول الصلات ومرحلة انشاء المستطيلات وربطها بخانات الجدول.

هام جدا: في هذا المستوى، وقبل الدراسة الدقيقة للطريقتين لمعرفة ايهما انفع، ثم تقديم الحالة التي تتبنها اللغة الشرمزية، نفترض ان انشاء جدول يتم حسب الطريقة الأولى، اي ان التعليمة شيد تقوم في مرحلة واحدة بتشييد الجدول وما يحتويه من مستطيلات، وبهذا فتنفيذ التعليمة **مستطيل[ ] جد_مس = شيد مستطيل[6]** تقوم بتشييد 6 مستطيلات، كل مستطيل في خانة من خانات الجدول، كما يظهر في خطأ: لم يُعثر على مصدر المرجع، والتعليمة مستطيل[ ] [ ] جد_مس2 = شيد مستطيل [20] [10] تقوم بتشييد 200 مستطيل، كل مستطيل في خانة من خانات الجدول ذو بعدين، كما يظهر في خطأ: لم يُعثر على مصدر المرجع.

من خلال التصريحين (السطرين 4 و5 من نص 27)، لا نرى اي اشارة تفيد بأن سمات المستطيلات التي يحتويها الجدولين قد هيئت بقيمة ما، فكل السمات التابعة لكل المستطيلات التي أُنشأت في الجداول لها قيم اذا غير معلومة، وهذا ما يبينه خطأ: لم يُعثر على مصدر المرجع.



الشكل 5: حالة ذاكرة خوارزم النص 5 مباشرة بعد التصريح بالمتغيرتين العامتين الممثلتين للجدولين جد_مس و جد_مس2

الرقم	النص	تعليق
1	عرف نمط مستطيل {	
2	طبيعي س، ج ، طول، عرض ؛	
3	}	
خوارزم المثال_5_الفصل_6 {		
4	مستطيل[ ] جد_مس = شيد مستطيل[6] ؛	
5	مستطيل[ ] [ ] جد_مس2 = شيد مستطيل [20] [10] ؛	
6	اجراء اساسي(){	
7	طبيعي مساحة، مساحة2؛	
8		
9	جد_مس[4].س = 100 ؛	
10	جد_مس[4].ج = 40 ؛	
11	جد_مس[4].طول = 210 ؛	
12	جد_مس[4].عرض = 70 ؛	
13	مساحة = جد_مس[4].طول * جد_مس[4].عرض ؛	

14		
15	جد_مس2[5][9].س = جد_مس[4].س	
16	جد_مس2[5][9].ج = جد_مس[4].ج	
17	جد_مس2[5][9].طول = جد_مس[4].طول * 2	
18	جد_مس2[5][9].عرض = جد_مس[4].عرض*3 + جد_مس2[5][9].طول	
19	مساحة2 = جد_مس2[5][9].طول * جد_مس2[5][9].عرض	
20	} // نهاية الإجراء اساسي	
	} // نهاية الخوارزم المثال_5_الفصل_6	

نص 27: كيفية استغلال السمات في الجداول المحتوية على معلومات من الأنماط المهيكلة

بعد انشاء الجدولين **جد_مس** و**جد_مس**2، الهدف هو تحديد قيم سمات مستطيلات الجدولين، وهنا يطرح السؤال التالي: ما هي الكتابة التي تمكننا من الوصول الى السمة المستهدفة لنحدد قيمتها؟

كما اشرنا اليه من قبل فان المتغيرات التي يحتويها الجدول والتي تمثل خاناته، ليس لها اسم خاص بها، بل نستعمل عملية تحديد المواقع في الجدول لنصل الى المتغيرة الممثلة للخانة، واشرنا ايضا انه في اطار جدول ما، نعتبر ان اسم الجدول متبوعا بعملية تحديد المواقع يمثلان مجازا اسم متغيرة في الجدول، فمثلا، اذا كان لدينا جدولا ببعد واحد، اسمه ج1، و حجمه يساوي 10، فان الكتابة ج1[2] تمثل مجازا اسم المتغيرة الثالثة في الجدول، وهكذا، اذا ارتكزنا على الجدول **جد_مس** (السطر 4 من نص 27)، فان الكتابة **جد_مس[4]** تمثل المتغيرة الخامسة من الجدول **جد_مس.**

بما ان الكتابة **جد_مس[4]** تمثل اسم متغيرة من النمط **مستطيل**، فالوصول الى سمات المتغيرة يقتضي فقط استعمال عملية الولوج في المتغيرات المهيكلة على اسم المتغيرة، وبهذه الطريقة، يمكننا ان نكتب ما يلي:

- الكتابة **جد_مس[4].س** تمثل اسم السمة **س** التابعة للمستطيل الموجود في الخانة الخامسة من الجدول **جد_مس.**
- الكتابة **جد_مس[0].ج** تمثل اسم السمة ج التابعة للمستطيل الموجود في الخانة الاولى من الجدول **جد_مس.**
- الكتابة **جد_مس2[1][8]. طول** تمثل اسم السمة **طول** التابعة للمستطيل الموجود في الخانة التابعة للسطر الثاني (رقم 1) و العمود التاسع (الرقم 8) من الجدول **جد_مس2.**
- العبارة **جد_مس[2].س = 120** تفضي الى وضع القيمة 120 في السمة **س** للمستطيل الموجود في الخانة الثالثة من الجدول **جد_مس.**
- العبارة **جد_مس[5].طول * جد_مس[5].عرض** تفضي الى قيمة هي مساحة المستطيل الموجود في الخانة الأخيرة من الجدول **جد_مس.**

ويحتوي نص 27 على عبارات عديدة تبين كيفية الوصول الى السمات في جداول الانماط المهيكلة، وفيما يلي بعض من هذه الأمثلة:

- **عبارة السطر 15:** يحتوي السطر 15 على العبارة: **جد_مس2[5][9].س=جد_مس[4].س**؛ ويتم من خلالها تحديد السمة **س** للمستطيل الموجود في السطر السادس (الرقم 5) و العمود العاشر (الرقم 9) للجدول **جد_مس2**، بقيمة السمة **س** للمستطيل الموجود في الخانة الخامسة (الرقم 4) من الجدول **جد_مس.**
- **عبارة السطر 18:جد_مس2[5][9].عرض=جد_مس[4].عرض*3 +جد_مس2[5][9].طول؛** هذه العبارة اعقد نوعا ما من عبارة السطر 15، ففيها تحدد قيمة السمة **عرض** للمستطيل جد_مس2[5][9] ، فقيمة هذه السمة تحسب بزيادة طول نفس المستطيل، اي **جد_مس2[5][9]** ، الى قيمة عرض المستطيل الموجود في الخانة الخامسة من الجدول **جد_مس**، اي جد_مس[4]، مضروبة في ثلاثة (**جد_مس[4].عرض*3**).

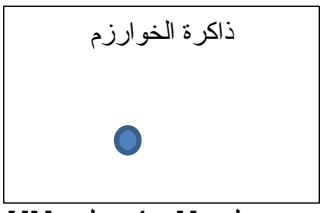
الفصل السابع

الأوعية: تمثيل الجداول والخانات في الذاكرة

1 – تذكير:

قبل ان نشرع في تبيان الطرق المختلفة لتمثيل الجداول في ذاكرة الخوارزم نُذكر بما يلي:

- حسب القواعد التي نتبعها فيما يخص الأنماط غير البدائية، فان كل متغيرة من نمط **غير بدائي** تنشأ في **الذاكرة النشطة** وتُستغل عن طريق **الصلات**، وهذا هو حال كل متغيرة مهيكلة وكل جدول، لأن الجدول كما اشرنا الى ذلك من قبل، ليس بنمط بدائي، بل هو نمط مهيكل خاص.
- الكتابة **طبيعي[ ] ج1** تُنتج الصلة **ج1** فقط، والتي يُفترض ان تُوَصِل الى جدول طبيعي ذو بعد واحد، وبما ان محتوى **ج1** غير معلوم، فلا نعلم شيئا عن الجدول المرتبط بالصلة **ج1**، وبما ان اسم الجدول في حقيقته صلة، يمكننا ايضا ان نكتب **طبيعي[ ][ ] ج2 = صلة_حرة**، فقيمة **ج2** في هذه الحالة معلومة وتفيد بعدم ارتباط الصلة **ج2** بأي جدول، ونذكر ان الكتابتين السالفتين لا يَنتُجُ عنهما انشاء اي جدول (الشكل 1).



الشكل 1: حالة الذاكرة بعد التصريحين: طبيعي[ ] ج1 ؛ طبيعي[ ][ ] ج2 = صلة_حرة

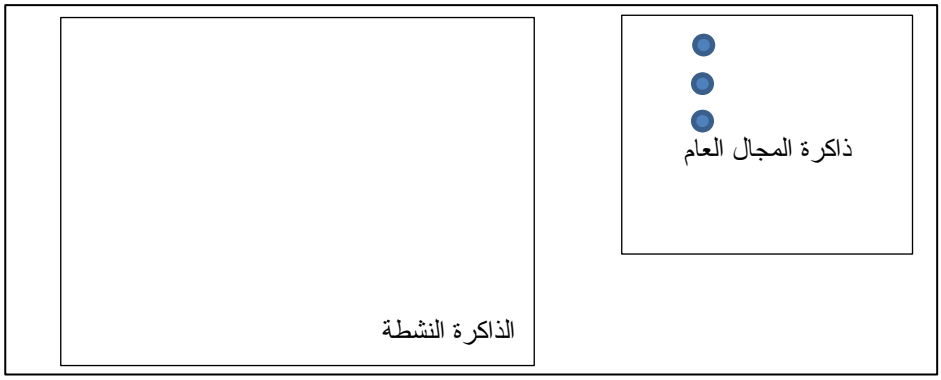
2 – المستوى الأول من التمثيل: الجدول متغيرة طارئة

كما عَلِمناه من قبل، يتم انشاء الجداول بإحدى الطريقتين: التصريح بالجدول بقيم اولية، او استعمال العملية **شيد**، وفي كلتا الطريقتين يُنشأ الجدول في الذاكرة النشطة، اي انه متغيرة طارئة غير بدائية، ولذا فانه يُستغل عن طريق صلة نسميها مجازا اسم الجدول، ولمزيد من الشرح نرتكز على النص 28 ونرى ماذا يحدث في الذاكرة بعد إنشاء الجداول واجراء بعض العمليات عليها.

خوارزم المثال_1_الفصل_7 {	
حقيقي[ ] جد_حق = {1.5 ، 2.7، 3.11 } ؛	1
طبيعي[ ] جد_طب = شيد طبيعي[5] ؛	2
طبيعي[ ] [ ] جد_طب2 = شيد طبيعي[6] [3] ؛	3
اجراء اساسي(){	4
طبيعي[ ] ج1؛	5
طبيعي[ ] [ ] ج2	6
جد_طب[2] = 8 ؛	12
جد_طب2[0][0] = 12 ؛	13
ج1 = جد_طب	14
ج2 = جد_طب2	15
ج2[1][1] = ج1[2]*2	16
} // نهاية الإجراء اساسي	17
} // نهاية الخوارزم المثال_1_الفصل_7	

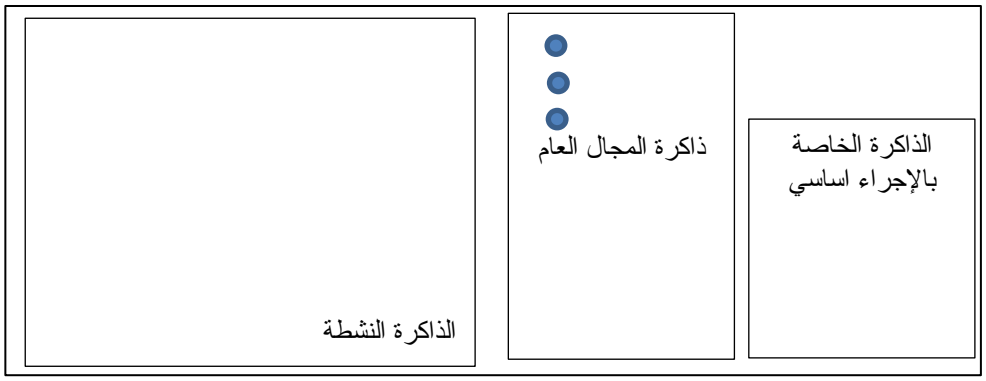
النص 28: خوارزم تعمير انتقائي لجدول ذو بعدين

يمثل خطأ: لم يُعثر على مصدر المرجع حالة الذاكرة مباشرة قبل بداية تنفيذ الإجراء اساسي (النص 28)، و في مثل هذه الحالة تقتصر ذاكرة الخوارزم فقط على الذاكرتين العامة والذاكرة النشطة، وهنا نرى ان الصلات متواجدة في الذاكرة العامة والجداول في الذاكرة النشطة، ونرى ان السبيل الوحيد للوصول الى الجداول هو استعمال الصلات، ونلاحظ ايضا ان محتوى الجدولين المرتبطين بالصلتين **جد_طب** و **جد_طب2** غير معروف، فالجدولين لم يصرح بهما بقيم اولية كما هو الشأن بالنسبة للجدول المرتبط بالصلة **جد_حق** .



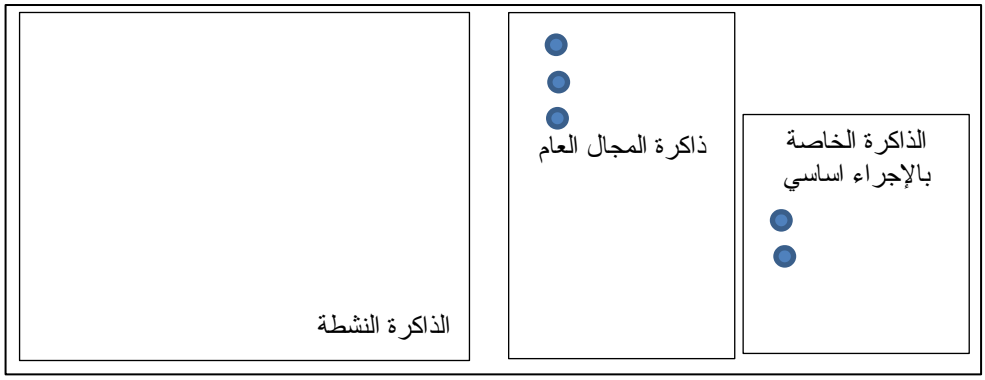
الشكل 2: حالة ذاكرة الخوارزم مباشرة قبل انطلاق تنفيذ الإجراء اساسي من النص 7

بعد ان يبدأ تنفيذ الإجراء الأساسي، تضاف الى ذاكرة الخوارزم، الذاكرة الخاصة بالإجراء اساسي، وعند الانتهاء من تنفيذ تعليمة السطر 13، تكون الذاكرة على الحالة التي يبينها خطأ: لم يُعثر على مصدر المرجع، ونرى فيه ان:



الشكل 3: حالة ذاكرة الخوارزم مباشرة بعد تنفيذ تعليمة السطر 13 من الإجراء اساسي (النص 7)

- بفعل تعليمة السطر 12 (**جد_طب**[2] = 8)، اصبحت الخانة الثالثة من الجدول **جد_طب** تحتوي على القيمة 8.
- بفعل تعليمة السطر 13 (**جد_طب2[0][0] = 12**)، اصبحت خانة السطر الأول والعمود الأول من الجدول **جد_طب2** تحتوي على القيمة 12.
- الذاكرة الخاصة بالإجراء **اساسي** تحتوي على صلتين مجهولتين المحتوى بفعل التصريحين في السطرين 5 و 6.



الشكل 4: حالة ذاكرة الخوارزم مباشرة بعد تنفيذ تعليمة السطر 16 من الإجراء اساسي (النص 1)

بعد اتمام تنفيذ تعليمة السطر. 16 تصبح الذاكرة على الهيئة الظاهرة في خطأ: لم يُعثر على مصدر. المرجع، و فيها نلاحظ ما يلي:

- الصلة **ج1** اصبحت مرتبطة بالجدول **جد_طب** كنتيجة لتنفيذ تعليمة السطر. 14(**ج1= جد_طب**)، وهكذا يصبح الجدول مرتبطا بصلتين: **جد_طب** و **ج1**، ومجازا. فللجدول اسمان: الجدول **جد_طب** هو نفسه الجدول **ج1.**
- الصلة **ج2** اصبحت مرتبطة بالجدول **جد_طب2** بعد تنفيذ تعليمة السطر 15(**ج2=جد_طب2**)، وهكذا يصبح الجدول مرتبطا بصلتين: **جد_طب2** و**ج2**، و مجازا له اسمان : الجدول **جد_طب 2** هو نفسه الجدول **ج2.**
- بفعل تعليمة السطر 16 تغير محتوى خانة السطر الثاني (رقم 1) والعمود الثاني (رقم 1) في الجدول **ج2** الذي هو نفسه الجدول **جد_طب2**، والقيمة التي وضعت في هذه الخانة هي ضعف قيمة الخانة الثالثة من الجدول **ج1**، و هو نفسه الجدول **جد_طب**، اي 8 * 2، اي 16.

2 – المستوى الثاني: تمثيل الخانات في الذاكرة.

هناك طريقتين لتمثيل خانات الجداول في الذاكرة:

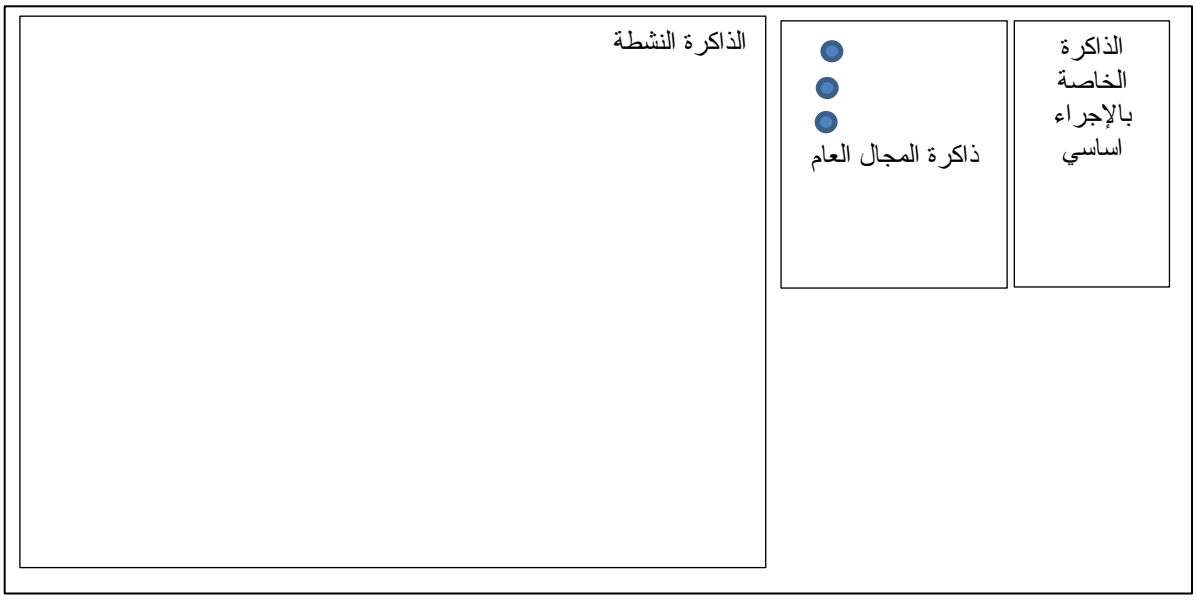
- الطريقة الأولى: الخانة تُمثل بمتغيرة من النمط المرتبط بالجدول مهما كان النمط، بدائي او غير بدائي، فكل المعلومات التابعة للخانة موجودة في مكان واحد داخل المساحة التي خصصت للجدول (الشكل 5).
- الطريقة الثانية: هناك حالتين:
  - الحالة الأولى: الجدول مرتبط بنمط بدائي (جدول طبيعي، جدول حقيقي، جدول من النمط حرف، جدول منطقي، جدول من الصلات) في هذه الحالة، تحتوي خانة

الجدول متغيرة من النمط البدائي المرتبط بالجدول، وبشكل ادق الخانة هي متغيرة من النمط البدائي المرتبط بالجدول.

- الحالة الثانية: الجدول مرتبط بنمط مهيكل، وفي هذه الحالة الأخيرة، فان الخانات تحتوي على صِلات مرتبطة بمتغيرات مهيكلة، فالمعلومات التابعة للخانة لن تكون في مكان واحد من الذاكرة، بل في مكانين مختلفين.

1	عرف نمط نقطة {
2	طبيعي س ، ج
3	}
4	عرف نمط مستطيل {
5	طبيعيس، ج ، طول، عرض
6	}
خوارزم المثال_2_الفصل_7 {	
7	مستطيل[ ] جد_مس = شيد مستطيل[5] ؛
8	نقطة[ ] [ ] جد_نق = شيد نقطة[4] [3] ؛
9	حقيقي [ ] جد_حق شيد حقيقي[4]
10	
11	اجراء اساسي(){
12	جد_مس[1].س = 9 ؛
13	جد_مس[1].طول = 6؛
14	جد_نق[1][0].س = 2 ؛
15	جد_نق[1][2].ج = 7 ؛
16	جد_حق[1] = جد_حق[2] = 6.8 ؛
17	} // نهاية الإجراء اساسي
} // نهاية الخوارزم المثال_2_الفصل_7	

النص 29: خوارزم لتبيان كيفية تمثيل الخانات في الذاكرة



الشكل 5: حالة ذاكرة خوارزم النص 3 مباشرة بعد تنفيذ آخر تعليمة من الإجراء اساسي

2 – 1 الطريقة الأولى:

في الطريقة الأولى تكون الخانة على شكل متغيرة من النمط المرتبط بالجدول، فاذا كان الجدول من النمط طبيعي، تكون كل خانة ممثلة بمتغيرة من النمط طبيعي، واذا كان الجدول من نمط

مهيكل، مثلا جدول المستطيلات، تكون كل خانة ممثلة بمتغيرة من النمط مستطيل، اي اننا نجد سمات المتغيرة المهيكلة في الخانة، وهذا ما يظهره خطأ: لم يُعثر على مصدر المرجع الذي يبين حالة الذاكرة بعد تنفيذ آخر تعليمة من الإجراء اساسي(النص 29)، ففي هذا النص نرى في اوله التعريف بنمطين: النمط **مستطيل** و النمط **نقطة**، وبعدهما التصريح في المجال العام للخوارزم بثلاثة جداول: جدول المستطيلات ذو بعد واحد **(جد_مس)**، جدول النقاط ذو بعدين **(جد_نق)** وجدول حقيقي ذو بعد واحد **(جد_حق)**، وفي هذه التصريحات تكون الصلات موجودة في الذاكرة الممثلة للمجال الذي فيه صرح بها، لأن الصلات في حد ذاتها متغيرات بدائية، يمكن ان توجد في اي ذاكرة، اما الجداول فمكانها الذاكرة النشطة لكونها متغيرات غير بدائية.

2 – 1 – 1 عيوب طريقة تمثيل الخانات بمتغيرات مهيكلة:

أ – العيب الأول: تبذير مفرط الذاكرة

في العموم، عند استعمال الجداول يحاول كاتب الخوارزم او البرنامج ان يصرح في البداية بجدول يتسع لكل الحالات التي يعالجها الخوارزم، فمثلا اذا انجز برنامج لتسيير شؤون الطلبة في الجامعات، واصبح من الضروري في الخوارزم استعمال جدول، فان مصمم الخوارزم يجعله صالح لكل الحالات، فمثلا يكون صالحا لمدرسة فيها 100 طالب، ويكون ايضا صالحا لقسم في الجامعة فيه 2000 طالب، بل اكثر من هذا يحاول كاتب الخوارزم ان يجعله صالحا حتى مع الحالات الشاذة جدا، بل حالات غير واقعية، ولذا يصرح بجدول يكون حجم بُعدِه كافيا للتكيف مع اي حالة، فمثلا، في حال خوارزم موجه لتسيير شؤون الطلبة في قسم من اقسام الجامعات، يصرح المصمم بجدول من النمط طالب ذو بعد واحد وبحجم 2500 (النص 30)، وهكذا اذا افترضنا أن:

- متغيرة من النمط سلسلة تستهلك 10 وحدات من الذاكرة.
- متغيرة من النمط طبيعي او النمط حقيقي تستهلك 4 وحدات من الذاكرة.

فان متغيرة من النمط شخص سوف تستهلك 3× 10 (اللقب، الاسم، التخصص) + 5×4 (اليوم، الشهر، السنة، الوزن، السنة_الدراسية) + 1×4 (الطول)، اي 54 وحدة من الذاكرة.

انطلاقا من حجم متغيرة من النمط شخص، نستطيع بسهولة معرفة المساحة التي سوف يستحوذ عليها الجدول **جد_الطلبة** (السطر 9 من النص 30) بعد انشاءه، اي 2500 × 54، اي 135000 وحدة من الذاكرة.

المشكل الكبير ليس في الحجم المستحوذ عليه بل في التبذير الذي اضطر اليه مصمم الخوارزم بسبب الطريقة الأولى في تمثيل خانات الخوارزم، ففي اغلب الحالات يكون عدد الطلبة اقل بكثير من حجم الجدول **جد_الطلبة**، ونادرا ما يكون العدد يقترب من حجم الجدول، ولا يبلغه حتى في الحالات

الشاذة جدا، فمثلا اذا افترضنا ان البرنامج المنبثق عن الخوارزم يستعمل في قسم فيه 500 طالب، تكون المساحة الضائعة (اي التي لا تستخدم) من الجدول **جد_الطلبة** هي 108000 وحدة ذاكرة (اي 2000 *54)

الرقم	النص
1	عرف نمط طالب {
2	سلسلة اللقب، الاسم
3	طبيعي اليوم، الشهر، السنة // معلومات الميلاد
4	طبيعي الوزن
5	حقيقي الطول
6	طبيعي السنة_الدراسية
7	سلسلة التخصص
8	}
خوارزم المثال_3_الفصل_7 {	
9	طالب[ ] جد_الطلبة = شيد طالب[2500] ؛
} // نهاية الخوارزم المثال_3_الفصل_7	

النص 30: نص لبيان العيب الأول في طريقة تمثيل الخانات بمتغيرات مهيكلة

ب – العيب الثاني: عدم التجانس مع الطريقة المعتمدة في استغلال المتغيرات المهيكلة

نذكر اننا في الفصل الرابع ادخلنا القواعد الجديدة التي يجب علينا اتباعها في كتابة نصوص الخوارزميات، ومن الأمور المهمة التي تحتويها هذه القواعد كيفية التعامل مع المتغيرات الغير البدائية كالمتغيرات المهيكلة والأوعية المجمعة كالجداول، فكل متغيرة غير بدائية تنشأ في الذاكرة النشطة وتستغل عن طريق الصلات.

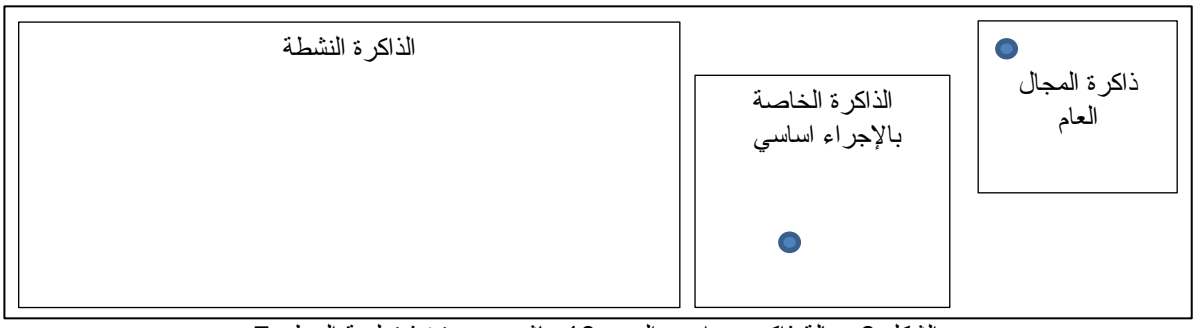
الطريقة الأولى في تمثيل الخانات بالنسبة لجداول المعلومات المهيكلة غير متجانسة مع ما ادخلناه من قواعد في الفصل الرابع، ولتبيان هذا الأمر نرتكز على ما في النص 31.

في السطر 7، وفي المجال العام، يُصرح بجدول المستطيلات **جد_مس**، وفي الإجراء اساسي يُصرح في مجاله الخاص بالصلة **س1** (السطر 9)، وهي صلة تمكن من الوصول الى متغيرة من النمط مستطيل، وبعدها، في السطر 10، تُشَيد متغيرة من النمط **مستطيل** مرتبطة بالصلة **مس2**، وتصبح الذاكرة بعد الانتهاء من تشييد المتغيرة **مس2** على الحالة الظاهرة في خطأ: لم يُعثر على مصدر المرجع.

الرقم	النص
1	عرف نمط مستطيل {
2	طبيعي س، ج، طول، عرض
3	مستطيل(طبيعي س، ج، طول، عرض){
4	انا.س = س؛ انا.ج = ج؛
5	انا.طول = طول ؛ انا.عرض = عرض ؛
6	}
خوارزم المثال_4_الفصل_7 {	
7	مستطيل[ ] جد_مس = شيد مستطيل[6] ؛
8	اجراء اساسي(){
9	مستطيل مس1 ؛
10	مستطيل مس2 = شيد مستطيل(5، 10، 30، 60) ؛

11	جد_مس[4] = مس2 ؛
12	مس1 = جد_مس[2]
13	} // نهاية الإجراء اساسي
} // نهاية الخوارزم المثال_4_الفصل_7	

النص 31: نص لبيان العيب ثاني في طريقة تمثيل الخانات بمتغيرات مهيكلة



.الشكل 6: حالة ذاكرة خوارزم النص 10مباشرة بعد تنفيذ تعليمة السطر 7

في السطر 11، نرى التعليمة **جد_مس[4] = مس2** ، وكأن الكاتب يريد ان ينقل ما في المتغيرة المرتبطة بالصلة **مس2** الى الخانة الخامسة، لكن هذه الكتابة، حسب القواعد الجديدة، غير متجانسة، فيسار عملية الشحن صلة، ولابد في هذه الحالة ان تكون صلة على يمين عملية، وهذا ما لا نراه، فعلى يمين عملية الشحن خانة من خانات الجدول، وكما افترضنا من قبل، فان الخانة متغيرة من النمط مستطيل، والكتابة **جد_مس[4]** ليست بصلة بل اسم لمتغيرة من النمط مستطيل، وحتى اذا افترضنا ان عملية الشحن (=) تقبل مثل هذه الكتابة، فهنا يطرح اشكال آخر لا يمكن حله، وهو كما يلي: اذا كانت مساحة الصلة هي 4 وحدات من الذاكرة، واذا كانت مساحة متغيرة من النمط طبيعي هي 4 وحدات من الذاكرة، فكيف نشحن محتوى 16 وحدة من الذاكرة الممثلة لخانة من خانات الجدول **جد_مس** بمحتوى أربع وحدات من الذاكرة الممثلة للصلة **س2**.

في السطر 12، يمكن ان نُفهم التعليمة **مس1 = جد_مس[2]** على احد الشكلين التاليين:

- كاتب الخوارزم يريد ان يربط المتغيرة **جد_مس[2]** بالصلة **س1**، وكما اسلفنا في الفصل الرابع، وحسب القواعد الجديدة التي نتبعها في كتابة الخوارزميات باللغة الشرمزية، فعملية الربط غير موجودة في هذه اللغة، وعملية الشحن (=) ليست بعملية ربط وانما شحن متغيرة بمحتوى متغيرة من نفس النمط، فاذا اردنا ان تكون تعليمة السطر 12 صحيحة فلا بد ان تكون **جد_مس[2]** اسم **لصلة** من النمط **مستطيل** لا اسم لمتغيرة من النمط **مستطيل** كما هو الحال الآن حسب ما افترضناه سابقا في كيفية تمثيل جداول الأنماط المهيكلة في الذاكرة.
- تكرار المعنى الذي رأيناه سابقا فيما يخص السطر 11، وهنا اذا افترضنا ان العملية = تقبل مثل كتابة السطر 11، فكيف السبيل لشحن محتوى 4 وحدة من الذاكرة الممثلة للصلة س1،

لخانة من خانات الجدول **جد_مس** بمحتوى 16 وحدة من الذاكرة للخانة الثالثة من الجدول **جد_مس،** و هذا امر غير ممكن.

اذا اردنا ان تكون الكتابة صحيحة فلا بد لمعاملي عملية الشحن ان يكونا صلتين، فمثلا نُصرح بصلة ثم نربطها بالمتغيرة **جد_مس[4]** باستعمال عملية الربط و رمزها --< (السطر 11 من النص 32) و بعد ذلك بدل استعمال اسم المتغيرة **جد_مس[4]** نستعمل الصلة(السطر 13)، ونفس الطريقة نتبعها مع المتغيرة **جد_مس[2]** كما يظهر ذلك في السطرين 12 و 14 من النص 32.

مع ان الكتابتين في السطر 13 و 14 من النص 32 لن يكون لهما المعنى الذي كان يريده كاتب الخوارزم في الكتابات الخاطئة في السطرين 11 و 12 من النص 31، **فان هاتين الكتابتين غير مقبولتين في اطار القواعد الجديدة التي نتبعها في كتابة الخوارزميات باللغة الشرمزية،** ونذكر ان عملية الربط و رمزها <-- اصبحت غير معترف بها في اللغة الشرمزية بسبب القاعدتين التاليين:

- القاعدة الأولى: الصلات لا تستعمل مع الأنماط البدائية.
- القاعدة الثانية: الأنماط المهيكلة لا يمكن استغلالها الا عن طريق الصلات.

و نرى جليا، ان القول بأن **جد_مس[2]** هو اسم لمتغيرة من النمط مستطيل لا يتوافق البتة مع القاعدة الثانية، اي ان الافتراض نفسه حول كيفية تمثيل جداول الأنماط المهيكلة لا يتوافق تماما مع قواعد كتابة الخوارزميات باللغة الشرمزية.

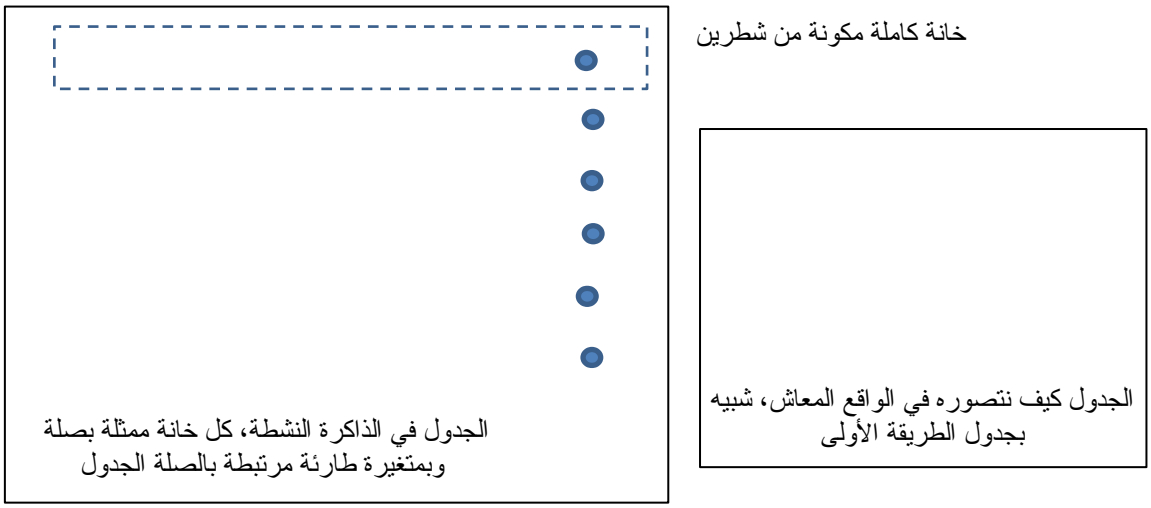
الرقم	النص
1	عرف نمط مستطيل {
2	طبيعي س، ج، طول، عرض
3	مستطيل(طبيعي س، ج، طول، عرض){
4	انا.س = س؛ انا.ج = ج؛
5	انا.طول = طول ؛ انا.عرض = عرض ؛
6	}
خوارزم المثال_5_الفصل_7 {	
7	مستطيل[ ] جد_مس = شيد مستطيل[6] ؛
8	اجراء اساسي(){
9	مستطيل مس1 ؛
10	مستطيل مس2 = شيد مستطيل(5، 10، 30، 60) ؛
11	مستطيل مس3 --< جد_مس[4]
12	مستطيل مس4 --< جد_مس[2]
13	مس3 = مس2 ؛
14	مس1 = مس4
15	} // نهاية الإجراء اساسي
} // نهاية الخوارزم المثال_5_الفصل_7	

النص 32: نص لبيان العيب ثاني في طريقة تمثيل الخانات بمتغيرات مهيكلة

2 – 1 الطريقة الثانية:

هذه الطريقة تعالج عيب الطريقة الأولى فيما يخص تبذير الذاكرة وفي آن واحد هي متوافقة تماما مع القواعد التي نتبعها في كتابة الخوارزميات باللغة الشرمزية، ففي هذه الطريقة عندما يُنشأ جدولا من نمط مهيكل، فهذا الجدول لن يستحوذ مباشرة على حجم المساحة التي يستهلكها جدول الطريقة الأولى، لأن الجدول الذي يُنشأ هو في الحقيقة جدولا للصلاة وليس بجدول للمتغيرات من

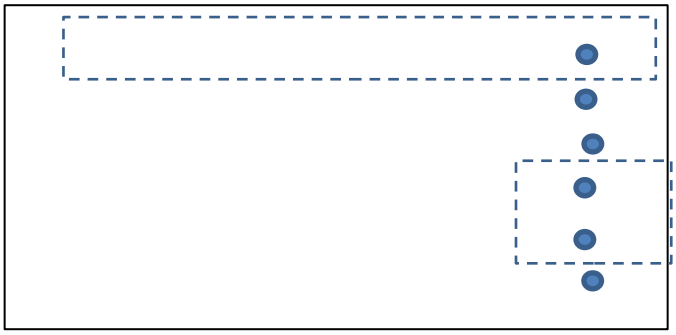
النمط المهيكل المرتبط بالجدول، فالمعلومات الممثلة لخانة ما ليست في مكان واحد، بل مكانين مختلقين من الذاكرة، فتكون كل خانة مُمثلة بشطرين (خطأ: لم يُعثر على مصدر المرجع):



الشكل 7: تمثيل الجدول في الذاكرة على الطريقة الثانية

- الشطر الأول: هو صلة، وكل الصلات الممثلة للشطر الأول موجودة في مكان واحد من الذاكرة، في جدول فيه فقط الصلات، **فالجدول الحقيقي هو جدول للصلات.**
- الشطر الثاني: هو متغيرة طارئة محتوية على سمات الخانة، وهذه المتغيرة متواجدة في مكان من الذاكرة النشطة غير المكان الذي يوجد فيه الجدول الذي يحتوي على الشطر الأول، وهي ايضا متواجدة في مكان من الذاكرة لا علاقة له بأماكن تواجد المتغيرات الطارئة المرتبطة بصلات باقي خانات الجدول.

و معنى هذا ان **جدول الطريقة الثانية في حقيقة أمره، هو جدول للصلات**، وعند انشائه لن تكون فيه الا الصلات ويكون الشطر الثاني لكل خانة غير موجود، وارتكازا على هذه الطريقة الثانية في تمثيل خانات الجداول، يمكننا ان نجد في الجدول خانات كاملة، اي فيها الشطرين، او نجد خانات غير مكتملة، محتوية فقط على الشطر الأول، اي ان الشطر الثاني لن يكون موجودا الا في الحالة الذي يكون وجوده ضروري، كما نراه من خلال خطأ: لم يُعثر على مصدر المرجع،



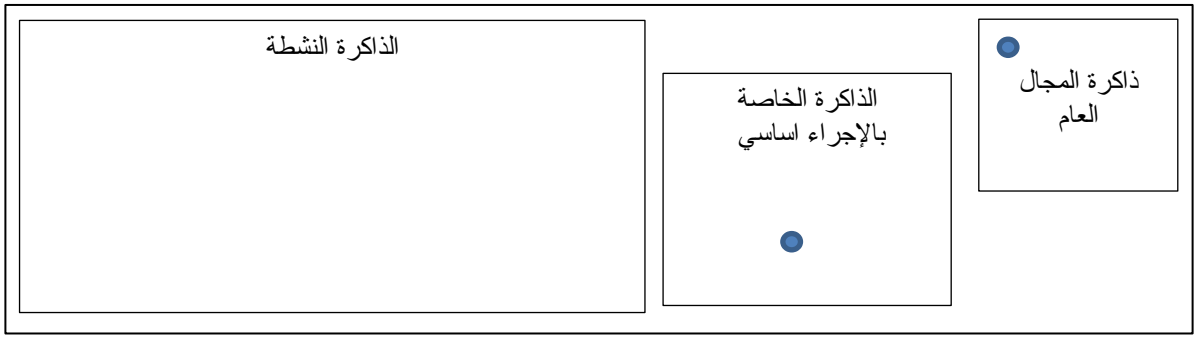
الشكل 8: جدول فيه خانات كاملة و خانات غير مكتملة

2 – 2 مراحل انشاء جدول الطريقة الثانية

في سياق إنشائه، يمر جدول الطريقة الثانية بمرحلتين:

- في المرحلة الاولى ينشأ الجدول الحقيقي، وهذا ما نشاهده في خطأ: لم يُعثر على مصدر المرجع بخصوص انشاء الجدول **جد_مس** (السطر 7 من النص 31) .
- في المرحلة الثانية، وحسب الاحتياج او الضرورة يتم إنشاء المتغيرات التي تُربط بصلات الجدول الحقيقي.

توضيح و تذكير: لماذا نستعمل الكلمة الجدول الحقيقي: اذا رجعنا الى تعريف الجدول فهو مجموعة من المتغيرات موجودة في نفس المنطقة من الذاكرة، كل متغيرة بجانبيه متغيرة اخرى من نفس النمط الا المتغيرتين الأولى و الأخيرة، فالأولى لا توجد متغيرة قبلها والأخيرة لا توجد متغيرة بعدها.



الشكل 9: حالة ذاكرة خوارزم النص 4 بعد تنفيذ تعليمة السطر 7 باستعمال الطريقة الثانية في تمثيل خانات الجدول في الذاكرة

2 – 3 هل تمثل المرحلة الثانية في انشاء جدول عيبا غير موجود في الطريقة الأولى؟

عكس الطريقة الأولى التي فيها تُنشأ المتغيرات الممثلة للخانات عند التصريح بالجدول، ففي الطريقة الثانية، زيادة على انشاء الجدول الحقيقي الذي هو جدول يحتوي فقط على الصلات، علينا ان نكتب التعليمات لإنشاء المتغيرات من النمط المهيكل المرتبط بالجدول، اي ما سميناه من قبل الشطر الثاني لكل خانة، فمثلا، اذا استعملنا خوارزم موجه لتسيير شؤون الطلبة في الجامعات في قسم ذو 500 طالب، كما فعلناه عند دراسة الطريقة الأولى، فعلينا ان نكتب التعليمات التي تمكننا من انشاء 500 متغيرة مرتبطة بصلات الجدول الحقيقي (او ما سميناه للشطر الثاني للخانات)، والأفضل ان تنشأ المتغيرات بقيمة اولية باستعمال المشيدات.

في حقيقة الأمر هذه الكتابة تكون ايضا ضرورية مع الطريقة الأولى، ففي الطريقة الأولى عندما تنشأ المتغيرات الممثلة للخانات فإنها تُنشأ بقيم مجهولة، وبما ان كل خانة تمثل طالبا واحدا، يصبح لزاما علينا كتابة عملية تهيئة لكل خانة من الخانات الخمس مئة بالقيم المناسبة.

وبهذا نرى ان المرحلة الثانية من الطريقة الأولى، اذا استُغلت فيها المشيدات، وهذا هو المفروض، فسوف لن يكون هناك فرق في عدد التعليمات التي هدفها النهائي هو الحصول على خانات مهيأة بمعلومات الطلبة.

2 – 3 مزايا الطريقة الثانية بالنسبة للطريقة الأولى:

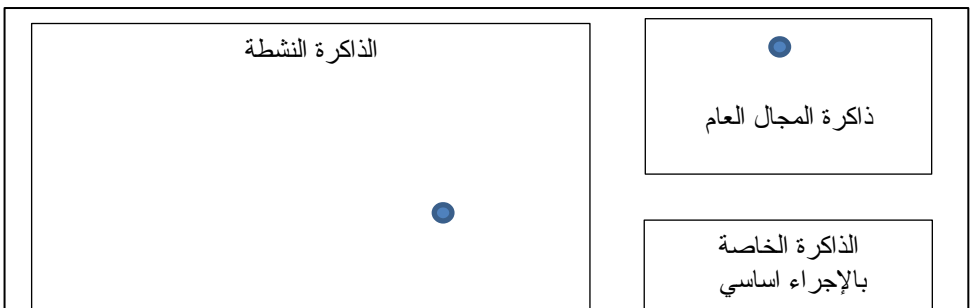
2 – 3 –1 تقليص كبير في كمية الذاكرة الضائعة:

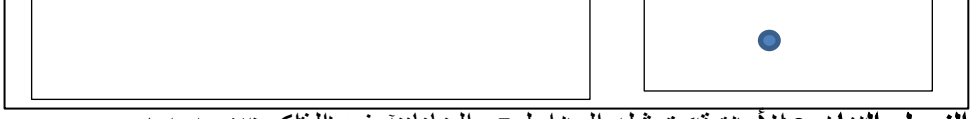
نرى جليا من خلال خطأ: لم يُعثر على مصدر المرجع، ان جدول الطريقة الثانية يحتل مساحة اقل من جدول الطريقة الأولى (خطأ: لم يُعثر على مصدر المرجع)، وحتى نلمس الفرق بين الطريقتين نقارن بالأرقام ما يضيع من ذاكرة في كل طريقة.

اذا افترضنا كما سبق ان الصلة تستحوذ على أربع وحدات من الذاكرة، فجدول الطريقة الثانية سوف يستحوذ على مساحة 4× عدد الخانات، فمثلا اذا رجعنا الى النص 30، فان الجدول **جد_الطلبة** سوف يستحوذ في الطريقة الثانية على مساحة 2500 × 4 اي 10000 و حدة من الذاكرة، و هذا رقم اصغر بكثير من الرقم 135000 الذي يمثل المساحة التي يستحوذ عليها الجدول **جد_الطلبة** اذا استعملنا الطريقة الأولى(انظر دراسة الطريقة الأولى) وبعد انشاء الشطر الثاني للخانات الخمس مئة الممثلة للطلبة، ما هي المساحة الضائعة؟ في الطريقة الثانية تكون الذاكرة الضائعة هي : (2500 – 500) × 4 اي 8000 وحدة و هذا رقم جد بسيط اذا قورن بالرقم 108000 الذي يمثل الضياع في الطريقة الأولى.

2 – 3 – 2 تجانس الطريقة الثانية مع قواعد كتابة الخوارزميات

بما ان محتوى جدول الطريقة الثانية هي في حقيقتها صلات، فان الكتابات الظاهرة في النص 31 تتماشى تماما مع القواعد الجديدة لكتابة الخوارزميات والتي ادخلناها في الفصل الرابع، فـخطأ: لم يُعثر على مصدر المرجع الذي يمثل النص 31 حسب الطريقة الثانية يبين مثلا ان الكتابة **جد_مس[4]** هو اسم لمتغيرة تمثل **صلة**، اي ان **جد_مس[4]** يمثل اسم **صلة**، وبهذا يكون كلا جانبي عملية الشحن في السطرين 11 و 12 من النص 31 صلة، فتعليمة السطر. 11 تشحن الصلة **جد_مس[4]** بما في الصلة **س2**، وتعليمة السطر. 12 تشحن الصلة **س1** بما في الصلة **جد_مس[2]**، وبعد تنفيذ هذه التعليمات تصبح ذاكرة خوارزم النص 31 على خطأ: لم يُعثر على مصدر المرجع.





2 – 4 مزايا اخرى

من مزايا الطريقة الثانية، انها تمكننا من:

- تحرير الشطر الأول لأي خانة من شطرها الثاني ليصبح حرا.
- استبدال الشطر الثاني بمتغيرة طارئة اخرى بكتابة واحدة، ويمكن ان تكون هذه المتغيرة تمثل الشطر الثاني لخانة اخرى من الجدول نفسه او جدولا آخر من نفس النمط.

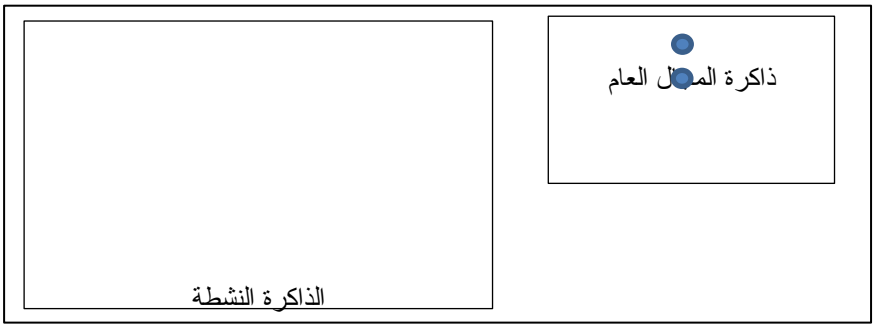
الرقم	النص
1	عرف نمط مستطيل {
2	طبيعي س، ج، طول، عرض
3	مستطيل(طبيعي س، ج، طول، عرض){
4	انا.س = س؛ انا.ج = ج؛
5	انا.طول = طول ؛ انا.عرض = عرض ؛
6	}
خوارزم المثال_6_الفصل_7 {	
7	مستطيل[ ] جد_مس = شيد مستطيل[5] ؛
8	مستطيل[ ] جد_مس1 = شيد مستطيل[4] ؛
9	اجراء اساسي(){
10	مستطيل مس1 = شيد مستطيل(40، 16، 90، 80) ؛
11	مستطيل مس2 = شيد مستطيل(5، 10، 30، 60) ؛
12	جد_مس[0] = شيد مستطيل(0، 0، 10، 10) ؛
13	جد_مس[1] = شيد مستطيل(10، 10، 20، 20) ؛
14	جد_مس[3] = س1 ؛
15	جد_مس[4] = س2 ؛
16	جد_مس1[0] = جد_مس [4]؛
17	جد_مس1[1] = س1؛
18	جد_مس1[2] = س2؛
19	س1 = س2 = جد_مس1[3] = شيد مستطيل(30، 30، 60، 60) ؛
20	} // نهاية الإجراء اساسي
} // نهاية الخوارزم المثال_6_الفصل_7	

النص 33: نص لبيان مزايا الطريقة الأولى في تمثيل خانات الجداول

للفهم العملي لهذه الميزات الأخيرة نستعرض ما في النص 33 وما في الأشكال من خطأ: لم يُعثر على مصدر المرجع الى خطأ: لم يُعثر على مصدر المرجع من.

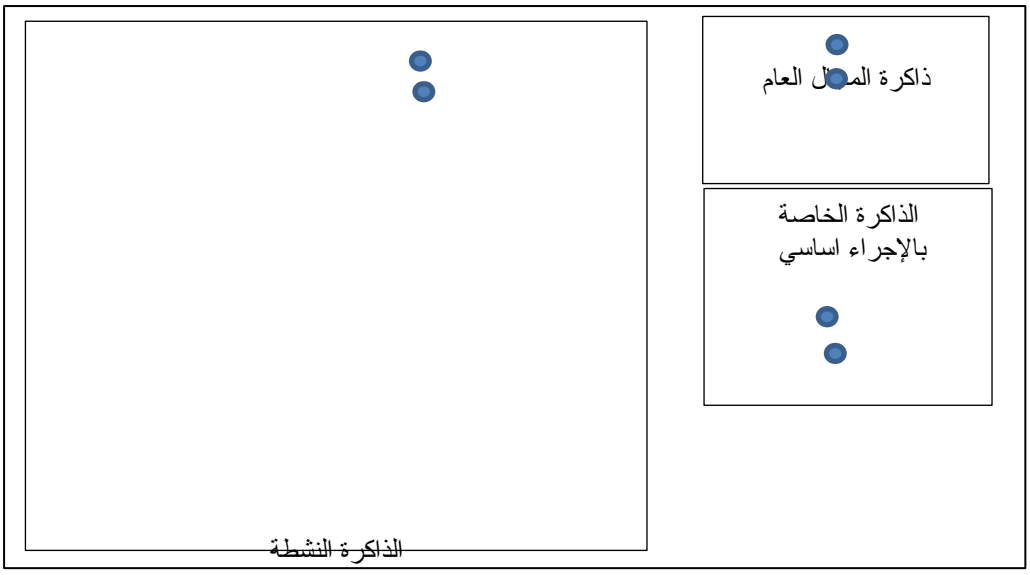
يمثل خطأ: لم يُعثر على مصدر المرجع الذاكرة بعد انشاء الجدولين **جد_مس** و **جد_مس1** (السطرين 7 و 8) في الذاكرة العامة والنشطة: فقبل ان يبدأ تنفيذ الإجراء اساسي، أُنشئت الصلتين **جد_مس** و **جد_مس1** في المجال العام اما الجدولين المرتبطين بهما فقد تم انشاؤهما في الذاكرة

النشطة، وحسب نوعية التصريح فان محتوى الجدولين غير معلوم، اي ان قيم الخانات التي هي صلات غير معلومة.



الشكل 11: حالة ذاكرة خوارزم النص 6 قبل بداية تنفيذ الإجراء اساسي

بعد انشاء الجدولين **جد_مس** و **جد_مس1**، ينطلق تنفيذ الإجراء اساسي، وتقوم التعليمات في الأسطر من 10 الى 13 من تشييد اربع مستطيلات، اثنان مرتبطين بالصلتين **مس1** و **مس2** اللتان صرح بهما في نفس سطر عملية التشييد، واثنان مرتبطتين بالخانتين الأولى والثانية التابعتين للجدول **جد_مس**، ويبين خطأ: لم يُعثر على مصدر المرجع حالة الذاكرة بعد تنفيذ التعليمات من السطر 10 الى السطر 13.

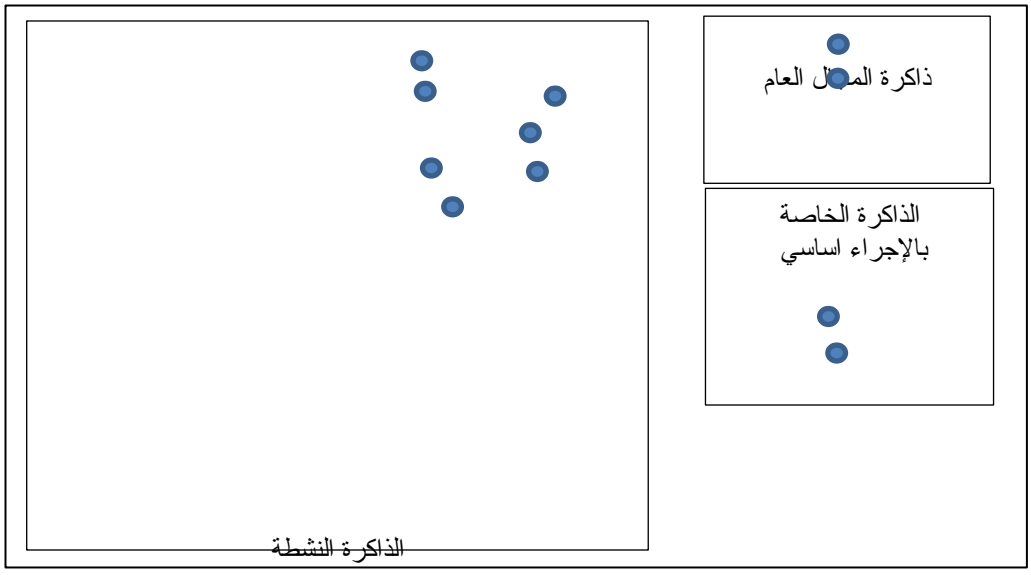


الشكل 12: حالة ذاكرة خوارزم النص 6 بعد تنفيذ التعليمات من السطر 10 الى السطر 13

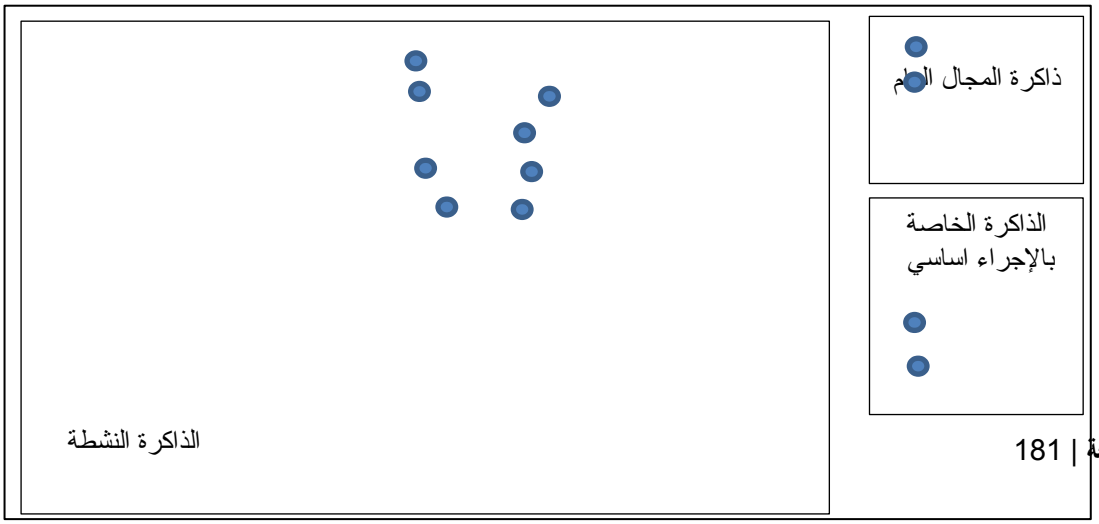
التعليمات من السطر 14 الى السطر 18 تقوم بشحن بعض الصلات بقيم اخرى، ففي السطر 14 (تعليمة مشابهة توجد في السطر 15) تشحن الصلة **جد_مس[3]** الممثلة للخانة الرابعة من الجدول **جد_مس** بقيمة الصلة **س1**، ومعنى هذا ان المتغيرة المرتبطة بالصلة **س1** اصبحت مرتبطة ايضا بالصلة **جد_مس[3]**، ونتيجة مشابهة تتم في السطر 15 إذ تصبح المتغيرة المرتبطة بالصلة **س2** مرتبطة ايضا بالصلة **جد_مس[4]**، ونفس الأمر يتم مع المتغيرات **جد_مس1[1]** و**جد_مس1[2]**

في السطرين 17 و 18، اما في السطر 16 فنرى عملية شحن الخانة الأولى للجدول **جد_مس1**، اي **جد_مس1[0]**، بما في الخانة الخامسة للجدول **جد_مس**، اي **جد_مس1[4]**، و يبين خطأ: لم يُعثر على مصدر المرجع نتيجة تنفيذ هذه التعليمات، وهذه النتيجة تبين ما يلي:

- المتغيرة المرتبطة بالصلة **س1** لها ثلاث صلات (**س1**، **جد_مس[3]** و **جد_مس1[1]**) .
- المتغيرة المرتبطة بالصلة **س2** لها اربع صلات (**س2**، **جد_مس[4]** ، **جد_مس1[0]** و **جد_مس1[2]**) .
- يمكن لصلتين تمثلان الشطر الأول لخانتين موجودتان في نفس الجدول او جدولين مختلفين، ان تشتركا في شطر ثان واحد، اي ان المتغيرة الممثلة للشطر الثاني لخانة من خانات الجدول يمكن ان تكون ايضا شطرا ثان لخانة اخرى من نفس الجدول، وهذه حالة لا يمكن تصورها في جداول الطريقة الأولى، فمثلا نرى ان الخانة الأولى والثالثة من الجدول **جد_مس1 (اي جد_مس1[0] و جد_مس1[2])** و **الخانة الخامسة** من الجدول **جد_مس (اي جد_مس[4])** تشترك في نفس الشطر الثاني، ويحدث هذا ايضا مع الخانة الرابعة من الجدول **جد_مس (اي جد_مس[3])** والخانة الثانية من الجدول **جد_مس1 (اي جد_مس1[1])**



الشكل 13: حالة ذاكرة خوارزم النص 6 بعد تنفيذ التعليمات من السطر 14 الى السطر 18

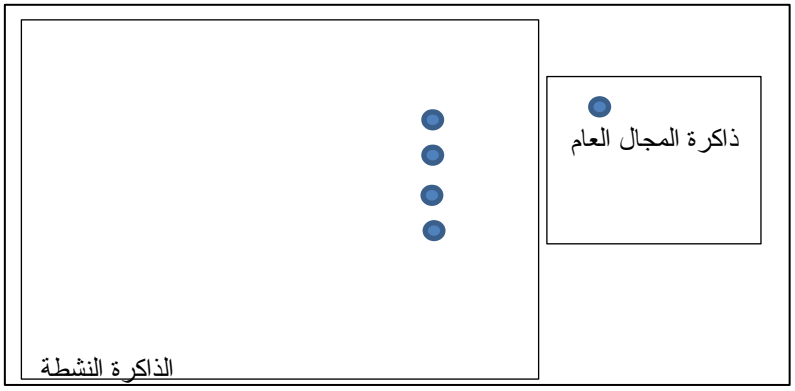


الشكل 14: حالة ذاكرة خوارزم النص 7 بعد تنفيذ تعليمة السطر 19

آخر تعليمة من الإجراء اساسي هي عبارة فيها ثلاث عمليات شحن (=) وعملية **شيد** التي تقوم بإنشاء مستطيل باستعمال المشيد الوحيد الذي عُرف في النمط مستطيل، فيبدأ اولا تقييم العبارة **شيد مستطيل(30، 30، 60، 60)** التي ترجع قيمة لصلة ظرفية تدل على موقع المتغيرة التي انشأت بفضل العملية **شيد** وهيأت بفضل المشيد الوحيد للنمط مستطيل، ثم تشحن هذه القيمة تباعا في الصلة **جد_مس1[3]** ، ثم في الصلة **س2**، ثم في الصلة **س1**، وبعد تنفيذ هذه العملية تُفك ارتباطات **س1**، و**س2** وتربط مع الصلة **جد_مس1[3]** بالمتغيرة التي تم انشاؤها بفضل العملية شيد، وهذا ما يبينه خطأ: لم يُعثر على مصدر المرجع.

3 – التصريح بقيم اولية لجدول الطريقة الثانية

القيم الأولية للجدول في حالنا هي القيم التي توضع في سمات المتغيرات الطارئة المرتبطة بخانات الجدول، فمثلا، بِفعِل تصريحٍ بقيم أولية، كيف نتحصل على الجدول الظاهر في خطأ: لم يُعثر على مصدر المرجع وهو جدول من النمط **نقطة** (النص 34).



الشكل 15: جدول الطريقة الثانية من النمط نقطة

الرقم	النص
1	عرف نمط نقطة {
2	طبيعي س ، ج
3	نقطة(طبيعي س ، ج){
4	انا.س = س ؛ انا.ج = ج ؛
5	}
6	}
خوارزم المثال_7_الفصل_7 {	
7	نقطة ن1 = شيد نقطة(12، 24) ؛
8	نقطة ن2 = شيد نقطة(36، 63) ؛
9	نقطة ن3 = شيد نقطة(54، 84) ؛
10	نقطة ن4 = شيد نقطة(77، 15) ؛
11	
12	
13	نقطة[ ] جد_نق = {ن1، ن2، ن3، ن4} ؛
14	نقطة[ ] [ ] جد_نق2 = {
15	{ن1 ، ن3}،

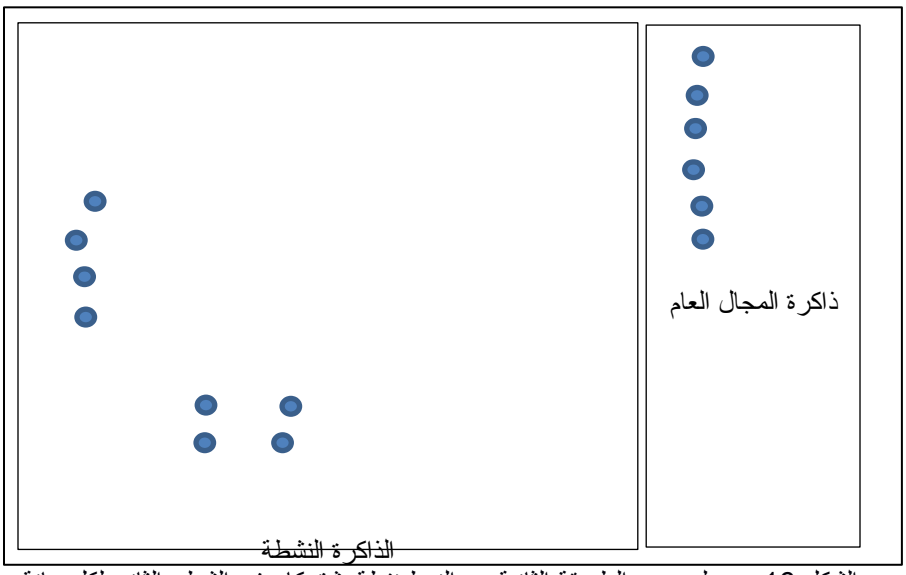
```
16  {ن2 ، ن4 }
17  } ؛
} // نهاية الخوارزم المثال_7_الفصل_7
```

النص 34: التصريح بقيوم اولية لجدولين من النمط نقط

بما ان خانة جدول الطريقة الثانية مكونة من شطرين (الشطر الأول يمثل الجدول الحقيقي وهو جدول من الصلات، والشطر الثاني تمثله مختلف المتغيرات الطارئة المرتبطة بصلات الجدول الحقيقي)، فلا بد اذا من قيمة للشطر الأول وهي صلة، وقيم لسمات المتغيرة المهيكلة الطارئة التي تمثل الشطر الثاني والمرتبطة بالصلة.

الكتابة التي بها نصرح بجدول من نمط مهيكل مشابهة تماما للكتابة في حال جدول من نمط بدائي، فبدل ان نذكر القيمة في حالة الأنماط البدائية، نذكر قيمة الصلة التي تمثل الشطر الأول لكل خانة، ومعنى هذا، ان وضع قيمة الصلة في الخانة لابد ان تسبقه عملية انشاء المتغيرة الطارئة المرتبطة بالخانة وتهيئتها بقيم اولية، ويبين النص 34 كيفية ساذجة للتصريح بجدولين من نمط مهيكل وبقيم اولية.

ففي الأسطر من 7 الى 10، ننشأ المتغيرات الطارئة الممثلة لأربع نقاط، ثم في السطر 13 نصرح بالجدول **جد_نق** ونذكر الصلات التي تأخذ قيمها وتوضع في الجدول، فنحصل على ما يبينه خطأ: لم يُعثر على مصدر المرجع، وفي الأسطر من 14 الى 17، نصرح بجدول ذو بعدين و بقيم اولية، و يبين خطأ: لم يُعثر على مصدر المرجع حالة الذاكرة بعد انشاء الجدولين، ونلاحظ ان خانات الجدول **جد_نق** و **جد_نق3** يشتركان في نفس المتغيرات الطارئة الممثلة للشطر الثاني لخانات الجدولين، ونلاحظ ايضا ان كل المتغيرات الطارئة من النمط نقطة الممثلة للشطر الثاني في الجدولين لها ثلاث صلات.



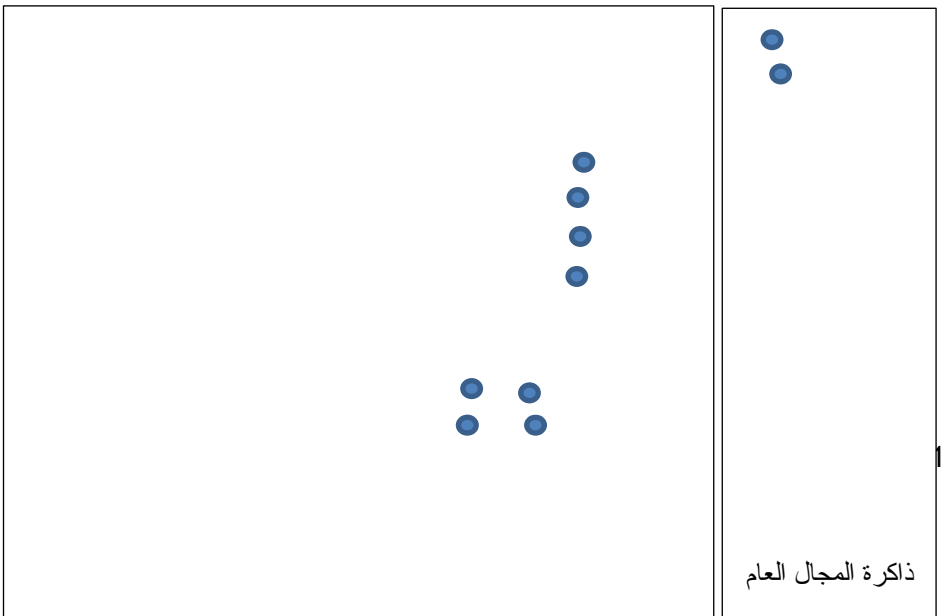
الشكل 16: جدولين من الطريقة الثانية من النمط نفطة يشتركان في الشطر الثاني لكل خانة

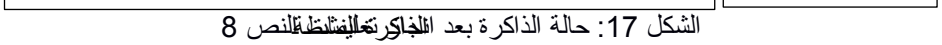
من خلال النص 34، وبعد انشاء الجدولين، يظهر لنا انه يمكن لنا ان نستغني عن اسماء الصلات في الخوارزم فلا نستعملها ونستعمل بدلا من ذلك خانات الجدولين، فمثلا، في عبارة ما، بدل ان نكتب **ن1**، نكتب **جد_نق[0]**، او **جد_نق[0] [0]**، و بدل ان نكتب **ن2** نكتب **جد_نق[1]** او **جد_نق[1] [0]**، اي باختصار، فإن وجود الأسماء **ن1**، **ن 2** الخ.. ليست ضرورية للخوارزم، ومعنى هذا انه بدل ذكر **ن1** , **ن2** الخ... عند التصريح بالجدولين، نستعمل مباشرة قيمة هذه المتغيرات، وقيمة هذه المتغيرات قد جاءتهم من العملية **شيد** في الأسطر 7 الى 10، ويبين النص 35 (ونتائج تنفيذه في خطأ: لم يُعثر على مصدر المرجع) الطريقة المختصرة في كتابة التصريح بقيم اولية لخانات جداول الأنماط المهيكلة ، اي نذكر العملية شيد بدل من اسم متغيرة مهيكلة قد انشأت من قبل، فالعملية **شيد** ترجع قيمة صلة توضع في الخانة التي وصفت في مكانها العملية **شيد**.

ملاحظة هامة: في هذه الطريقة المختصرة التي لا تعتمد على اسماء الصلات مثل ن1 و ن2 الخ، فان الجدولين **جد_نق** و**جد_نق2 لا يشتركان** في المتغيرات الطارئة التي انشأت في سياق تهيئة الجدولين كما يظهر ذلك في خطأ: لم يُعثر على مصدر المرجع، فإذا اردنا من كلا الجدولين ان يشتركا في متغيرة طارئة، فلا مفر من انشاء متغيرة الطارئة بصلة ثم نذكر الصلة عند تهيئة الجدولين كما فعلنا سابقا.

الرقم	النص
1	عرف نمط نقطة {
2	طبيعي س ، ج
3	نقطة(طبيعي س ، ج){
4	انا.س = س ؛ انا.ج = ج ؛
5	}
6	}
خوارزم المثال_8_الفصل_7 {	
7	نقطة[] جد_نق = { شيد نقطة(12، 24)، شيد نقطة(36، 63)، شيد نقطة(54، 84)، شيد نقطة(77، 15) } ؛
8	نقطة[][] جد_نق2 = {
9	{ شيد نقطة(12، 24) ، شيد نقطة(54، 84) }،
10	{ شيد نقطة(36، 63) ، شيد نقطة(77، 15) }
11	} ؛
} // نهاية الخوارزم المثال_8_الفصل_7	

النص 35: الاستغناء عن اسماء الصلات عند التصريح بقيم اولية لجدولين من النمط نقط





الشكل 17: حالة الذاكرة بعد الخطوة النص 8

الجزء الثاني: الفصل السابع: الأوعية: تمثيل الجداول و الخانات في الذاكرة

الفصل الثامن

الأوعية: تمثيل الجدول ذو بعدين او اكثر في ذاكرة الخوارزم

1 – كيفية تمثيل الجدول ذو بعدين او اكثر في ذاكرة الخوارزم:

بما ان الجدول في حد ذاته متغيرة مهيكلة، فان الجدول ذو ابعاد عدة، هو في حقيقته جدول مكون من جداول اخرى، فمثلا "**الجدول الطبيعي ذو بعدين**" هو جدول ذو بعد واحد يحتوي على صلات لجداول من النمط طبيعي، اي ان نمط خانة الجدول في البعد الأول هو "**جدول طبيعي ذو بعد واحد**" ونمط خانة الجداول في البعد الثاني هو **طبيعي**، وهذه هي الطريقة المعتمدة في كثير من لغات البرمجة كلغة "**سي**"، و "**سي++**" و "**جافا**".

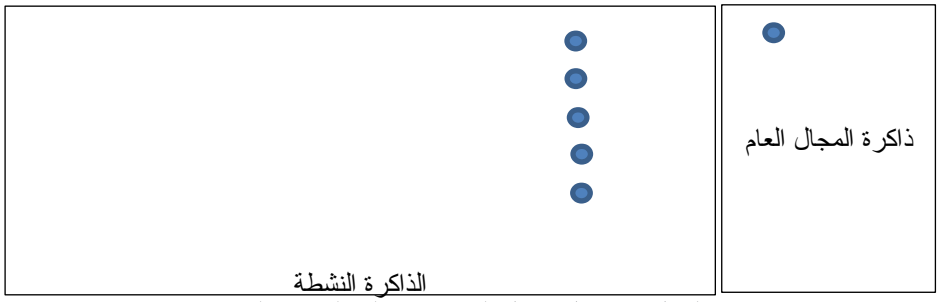
1 – 1 مثال توضيحي للطريقة المعتمدة في تمثيل الجدول ذو بعدين

يبين خطأ: لم يُعثر على مصدر المرجع الجدول **جد_طب** الذي صرح به في السطر 1 من النص 36، ومن خلاله نرى:

- الخانات التي توجد فعلا جنبا الى جنب هي فقط خانات الجداول ذو بعد واحد فقط.
- لتمثيل جدول ذو بعدين نحتاج اولا الى جدول يبين لنا السطور والى جداول يمثل السطور نفسها، وكل خانة من السطر تنتمي الى عمود من اعمدة الجدول.
- في الجداول التي تمثل الأسطر، **نمط الخانة** هو **طبيعي**، اي ان نمط الجداول التي تمثل السطور هو **طبيعي[ ]**.
- في الجدول الأول الذي يُمكننا من الوصول الى السطور، تحتوي كل خانة على صلة تمكن من الوصول الى سطر وهو **جدول طبيعي ذو بعد واحد**، اي ان نمط خانات الجدول الأول هو **طبيعي[ ]**، فالجدول الأول هو اذا جدول لقيم من النمط **طبيعي[ ]**، فنمطه اذا: **طبيعي[ ][ ]**، اي نكتب نمط الخانة وهو **طبيعي[ ]** متبوعا بالعارضتين [ ] الممثلتين للنمط جدول، ومن هنا نرى ان جدول طبيعي ذو بعدين هو في حقيقته جدول لجداول طبيعية ذو بعد واحد.

الرقم	النص
	خوارزم المثال_1_الفصل_8 {
1	طبيعي[ ][ ] جد_طب2 = {
2	{12، 24، 48، 96 }،
3	{11، 22، 44، 88 }
4	{10، 20، 40، 80 }
5	{9، 18، 36، 72 }
6	{8، 16، 32، 64 }،
7	} ؛
	} // نهاية الخوارزم المثال_1_الفصل_8

النص 36: التصريح بجدول ذو بعدين من النمط طبيعي



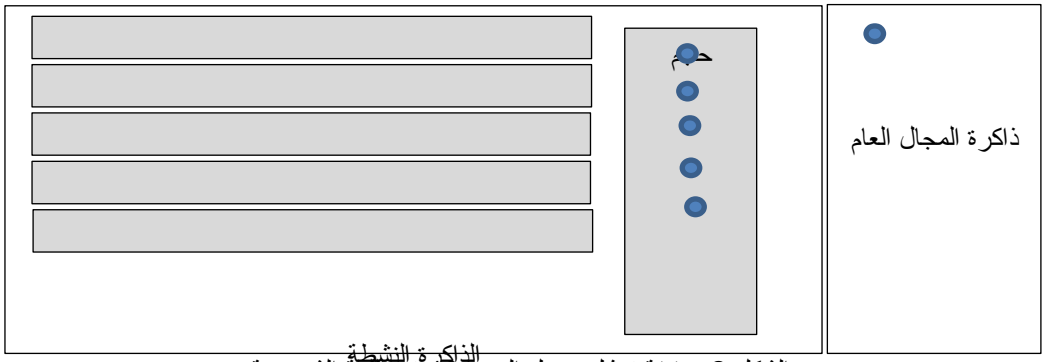
الشكل 1: تمثيل جدول النص 10 على الطريقة الثانية

1 – 2 الجدول ذو بعد واحد في اللغة الشرمزية

الجدول ذو بعد واحد في الطريقة التي تعتمدها اللغة الشرمزية، هو خليط بين الأنماط المهيكلة والجداول، فالجدول ذو بعد واحد يحتوي في مساحة واحدة من الذاكرة على معلومتين (او قُل على سيمتين):

- المعلومة الأولى هي الخانات، وكل خانة اما ان تكون متغيرة من نمط بدائي اذا كان الجدول جدولا لنمط بدائي او **صلة** ان كان الجدول لنمط مهيكل، ونذكر ان الصلة في ما يخصنا تحتوي على معلومة بدائية تدل على موقع في الذاكرة، فالصلة نفسها، نمطها بدائي.
- المعلومة الثانية هي **متغيرة ثابتة** اسمها **حجم**، وتحتوي على **قيمة ثابتة** لا يمكن تغييرها، وهذه القيمة الثابتة هي **حجم** الجدول، اي عدد خانات الجدول، وتحدد هذه القيمة عند انشاء الجدول ولا يمكن تغييرها بعد ذلك.

و يبين خطأ: لم يُعثر على مصدر المرجع حقيقة الجدول **جد_طب2** الذي صرح به في النص 36، ويمكن الوصول الى السمة **حجم** من جدول ما باستعمال عملية الولوج في المتغيرة المهيكلة الممثلة لجدول ذو بعد واحد، ويبين النص 37 بعض الأمثلة في استغلال السمة **حجم**.



الشكل 2: حقيقة تمثيل جدول النص 10 في اللغة الشرمزية

الرقم	النص
خوارزم المثال_2_الفصل_8	
1	طبيعي[ ] [ ] جد_طب2 = {
2	{12، 24، 48، 96 }،
3	{11، 22، 44، 88 }
4	{10، 20، 40، 80 }
5	{9، 18، 36، 72 }
6	{8، 16، 32، 64 }،
7	} ؛
9	اجراء اساسي(){
10	طبيعي عدد_الاسطر = جد_طب2.حجم ؛
11	طبيعي عدد_خانات_السطر= جد_طب2[1].حجم ؛
12	طبيعي عدد_الكلي_للخانات = جد_طب2.حجم * عدد_خانات_السطر ؛
13	
14	جد_طب2[1] = حرة ؛
15	جد_طب2[3] = حرة ؛
16	عدد_خانات_السطر = جد_طب2[1].حجم // كتابة خاطئة: الجدول جد_طب2[1] غير موجود
17	عدد_خانات_السطر = جد_طب2[3].حجم // كتابة خاطئة: الجدول جد_طب2[3] غير موجود
18	} // نهاية الإجراء اساسي

الجزء الثاني: الفصل الثامن : الأوعية: تمثيل الجدول ذو بعدين او اكثر في ذاكرة الخوارزم

```
} // نهاية الخوارزم المثال_2_الفصل_8
```

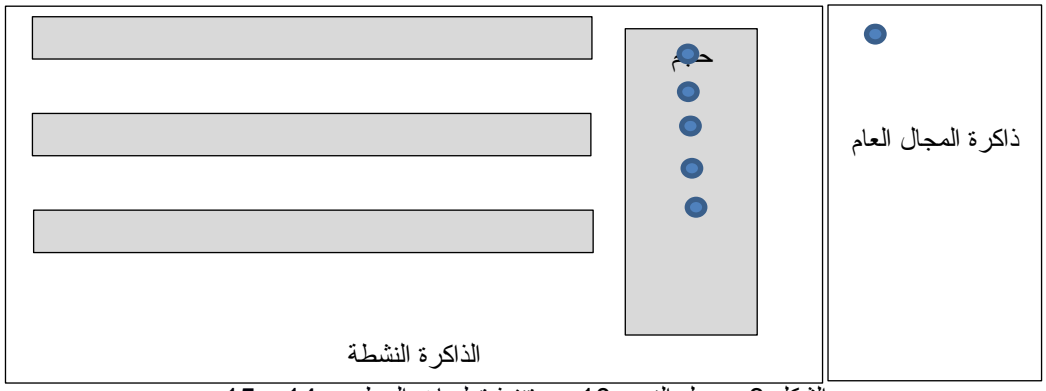
النص 37: كيفية استغلال السمة حجم في الجداول و كيفية حذف الأسطر

فمثلا اذا اردنا ان نعرف عدد أسطر الجدول ذو بعدين **جد_طب2**، فهذا العدد هو **حجم** الجدول الأول، واسم هذا الجدول هو مجازا **جد_طب2**، والوصول الم المتغيرة **حجم** من الجدول الأول يقتضي استعمال عملية الولوج على اسم الجدول، اي **جد_طب2**، فنكتب اذا: **جد_طب2.حجم**، وهذا ما يظهر في السطر 10 من النص 37، واذا اردنا ان نعرف عدد الخانات في السطر الثاني، علينا اولا ايجاد اسم السطر الثاني، وهو **جد_طب[1]**، وللوصول الى السمة حجم علينا فقط استعمال عملية الولوج على اسم الجدول الذي يمثل السطر الثاني، فنكتب: **جد_طب[1].حجم**.

1 – 3 مزايا الطريقة المعتمدة في تمثيل الجداول في الذاكرة:

الطريقة التي تعتمدها اللغة الشرمزية هي الطريقة المتبعة في لغة **جافا** ولها مزايا عديدة منها:

- **امكانية عدم وجود بعض الأسطر او كلها:** في الطريقة المعتمدة، تمثل الأسطر في حقيقتها الشطر الثاني من الجدول الأول، وكما سبق وان اشرنا اليه، يمكن ان يكون الشطر الثاني لخانة ما غائبا، وهذا ما يبينه خطأ: لم يُعثر على مصدر المرجع، فبعد تنفيذ التعليمتين الموجودتين في السطرين 14 و 15 من النص 37، يصبح الجدول **جد_طب2** بثلاث اسطر، ونرى جليا ان قيمة السمة **حجم** التابعة للجدول الأول لا علاقة لها بالوجود الفعلي للسطر او لا.



الشكل 3: جدول النص 10 بعد تنفيذ تعليمات السطرين 14 و 15

ملاحظه هامة: الكتابتين في السطرين 16 و 17 صحيحتين في الشكل، خطيرتين جدا في المضمون، فبعد ان ازيح السطر الثاني والرابع من الجدول **جد_طب2** (خطأ: لم يُعثر على مصدر المرجع)، تصبح الصلتين **جد_طب2[1]** و **جد_طب2[1]** حرتين، ومعنى هذا ان الصلتين لا تؤديان لأي جدول ذو بعد واحد، فالكتابة **جد_طب2[1].حجم** يكون معناها: الولوج في جدول غير موجود للوصول الى

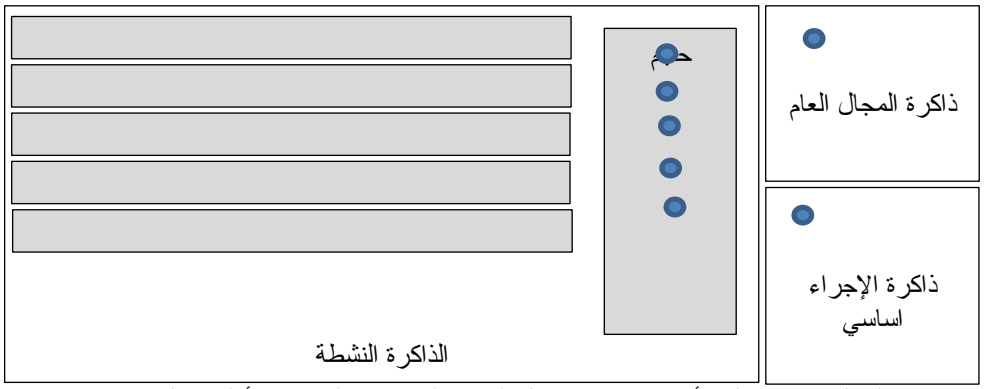
السمة **حجم**، ومع مثل هذه الكتابات لا يمكن لمنفذ الخوارزم الاستمرار في التنفيذ، وينته فورا. وبشكل غير عادي تنفيذ الخوارزم.

- **الاستبدال السهل للأسطر:** بعملية واحدة يمكن نقل سطر من خانة الى اخرى او استبدال سطر. بسطر. جديد لم يكن موجودا. في الجدول ثنائي الأبعاد من قبل، وهذا ما نشاهده في خطأ: لم يُعثر على مصدر المرجع، في خطأ: لم يُعثر على مصدر المرجع و في خطأ: لم يُعثر على مصدر. المرجع .

ففي خطأ: لم يُعثر على مصدر المرجع، نرى حالة ذاكرة الخوارزم بعد تنفيذ تعليمات الأسطر من 11 الى 13 من النص 38، فالسطر الأول يصبح السطر. الخامس والسطر. الخامس يصبح الأول، وهذه العملية تتم فقط بشحن قيم الصلات بعضها. في بعض، ففي السطر. 10 انشأت صلة لجدول طبيعي. ذو بعد واحد وهي **جط**، ونستعمل. هذه الصلة للحفاظ مؤقتا على قيمة الصلة **جد_طب2[0]** (السطر 11) التي ستغير في التعليمة التالية (السطر 12) بقيمة الصلة **جد_طب2[4]** ، وفي. الأخير تُشحن الصلة **جد_طب2[4]** بما كان في الصلة **جد_طب2[0]** قبل تغييرها، اي بما في الصلة **جط** (السطر 13).

الرقم	النص
	خوارزم المثال_3_الفصل_8 {
1	طبيعي[ ] [ ] جد_طب2 = {
2	{12، 24، 48، 96 }،
3	{11، 22، 44، 88 }
4	{10، 20، 40، 80 }
5	{9، 18، 36، 72 }
6	{8، 16، 32، 64 }،
7	} ؛
9	اجراء اساسي(){
10	طبيعي[ ] جط،
11	جط = جد_طب2[0]
12	جد_طب2[0] = جد_طب2[4]
13	جد_طب2[4] = جط
14	} // نهاية الإجراء اساسي
	} // نهاية الخوارزم المثال_3_الفصل_8

النص 38: كيفية استبدال الأسطر



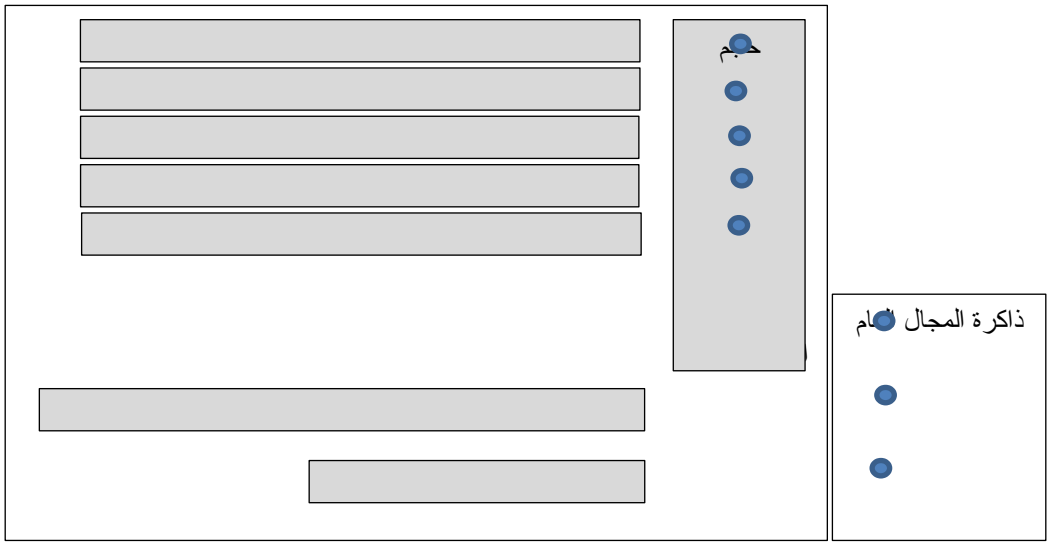
الشكل 4: استبدال الأسطر و هو استبدال الشطر الثاني بين الخانتين الأولى و الرابعة

في خطأ: لم يُعثر على مصدر المرجع نرى حالة الذاكرة قبل بداية تنفيذ الإجراء اساسي من النص 39، وفي خطأ: لم يُعثر على مصدر المرجع نرى ماذا حدث في الذاكرة بعد تنفيذ تعليمات الإجراء اساسي، ونلاحظ من خلال الرسم ما يلي:

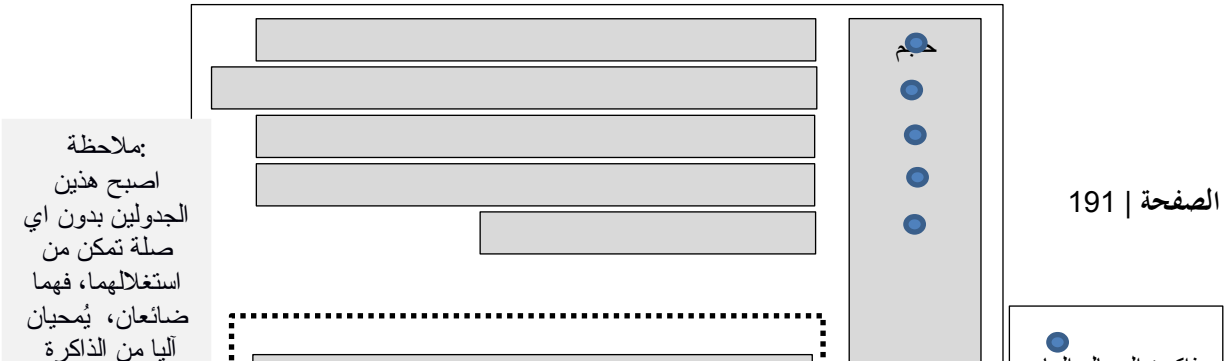
- الجدولين الذين كانا يمثلان السطر الثاني والخامس في الجدول **جد_طب2** اصبحا دون صلة تمكن من استغلالهما، ففي هذه الحالة تُخلى الذاكرة منهما حتى لا تبق المساحة المستعملة من قبلهما ضائعة.
- حجم الأسطر غير متساوية، وهذه ميزة لا يمكن الوصول اليها اذا اتبعنا الطرق الأخرى في تمثيل خانات الجداول والجداول نفسها.

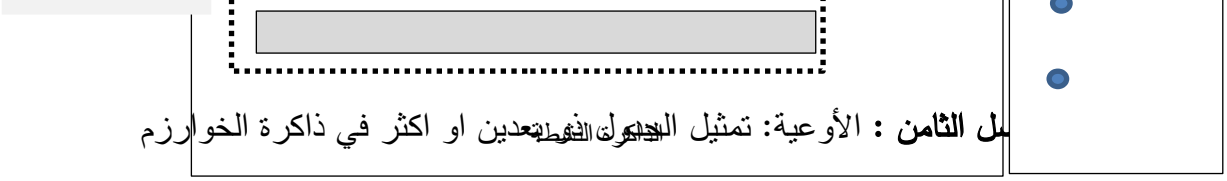
لرقم	النص
	خوارزم المثال_4_الفصل_8 {
1	طبيعي[ ] [ ] جد_طب2 = {
2	{12، 24، 48، 96 }،
3	{11، 22، 44، 88 }
4	{10، 20، 40، 80 }
5	{9، 18، 36، 72 }
6	{8، 16، 32، 64 }،
7	} ؛
9	طبيعي[ ] [ ] جط_1 = {19، 20، 21، 22، 23، 24}
10	طبيعي[ ] [ ] جط_2 = {1 ، 3 }
11	اجراء اساسي(){
12	جد_طب2[1] = جط_1 ؛
13	جد_طب2[4] = جط_2 ؛
14	} // نهاية الإجراء اساسي
	} // نهاية الخوارزم المثال_4_الفصل_8

النص 39: استبدال السطر الثاني و الخامس بأسطر اخرى لم تكن في الجدول ثنائي الأبعاد



الشكل 5: حالة الذاكرة قبل تنفيذ تعليمات الإجراء اساسي من النص 4





الشكل 6: حالة الذاكرة بعد تنفيذ تعليمات الإجراء اساسي من النص 4

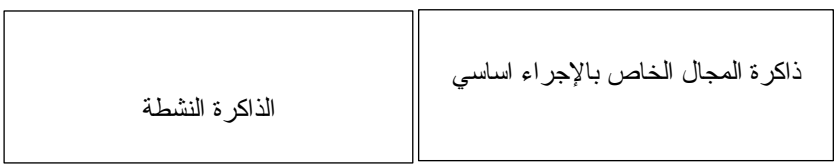
2 – مراحل انشاء جدول ذو بعدين:

اول ما يُنشأ من الجدول هي الصلة التي يُسمى بها الجدول مجازا، ثم يُنشأ الجدول الذي يمثل البعد الأول، وعدد أسطر الجدول ذو بعدين يحدده حجم هذا الجدول الأول، وبعد انشاء الجدول الأول، تُنشأ حسب الحاجة الجداول الممثلة للأسطر، وكل سطر يُنشأ بصفة مستقلة عن الأسطر الأخرى، ويمكن ان يكون له سطر حجم خاص به، ويبين النص 40 والأشكال من خطأ: لم يُعثر على مصدر المرجع الى خطأ: لم يُعثر على مصدر المرجع هذه المراحل.

في بداية تنفيذ الإجراء **اساسي** من النص 40، تُنشأ اولا، في السطر 3، الصلة **جط_2**، وهي صلة لجدول طبيعي ذو بعدين، وكما يبينه خطأ: لم يُعثر على مصدر المرجع، لا وجود في هذه المرحلة لأي جدول، المتغيرة الوحيدة الموجودة في الذاكرة هي **جط_2** التي انشأت في المجال الخاص بالإجراء اساسي.

لرقم	النص
خوارزم المثال_5_الفصل_8 {	
1	اجراء اساسي(){
2	طبيعي[ ] [ ] جط_2 ؛
3	جط_2 = شيد طبيعي[4] [ ] ؛
4	جط_2[0] = شيد طبيعي[6] ؛
5	جط_2[1] = شيد طبيعي[5] ؛
6	جط_2[2] = شيد طبيعي[4] ؛
7	جط_2[3] = شيد طبيعي[3] ؛
8	
9	جط_2[0][0] = 0 ؛
10	جط_2[0][1] = 1 ؛
11	جط_2[1][0] = 1 ؛
12	جط_2[3][5] = 8 ؛ // كتابة غير صحيحة، السطر الرابع يحتوي فقط على 3 خانات
13	} // نهاية الإجراء اساسي
} // نهاية الخوارزم المثال_5_الفصل_8	

النص 40: خطوات انشاء جدول دو بعدين



الشكل 7: حالة الذاكرة في بداية الإجراء اساسي، بعد تصريح السطر 2 من النص 5

الجزء الثاني: الفصل الثامن : الأوعية: تمثيل الجدول ذو بعدين او اكثر في ذاكرة الخوارزم

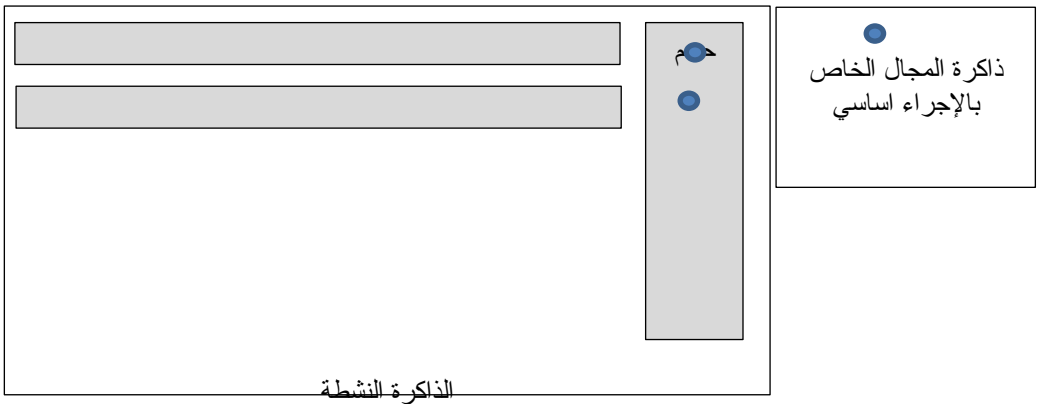
بعد هذه المرحلة الأولى، تنفذ التعليمة الثانية، ونلاحظ في الكتابة ان كاتب الخوارزم اراد فقط انشاء البعد الأول الذي يحدد عدد اسطر الجدول **جط_2،** وبمثل الكتابة **شيد طبيعي[4] [ ]** يُنشأ فقط الجدول الأول (خطأ: لم يُعثر على مصدر المرجع).

تنبيه هام: الكتابة **شيد طبيعي[] [6]** غير معقولة اذ لا يمكن انشاء الأعمدة قبل الأسطر، فخلايا الأسطر فيما بينها هي التي تبين الأعمدة، فلا بد من انشاء الأسطر لنستنتج منها الأعمدة، ولا يمكن الكلام عن اسطر دون وجود الجدول الأول.

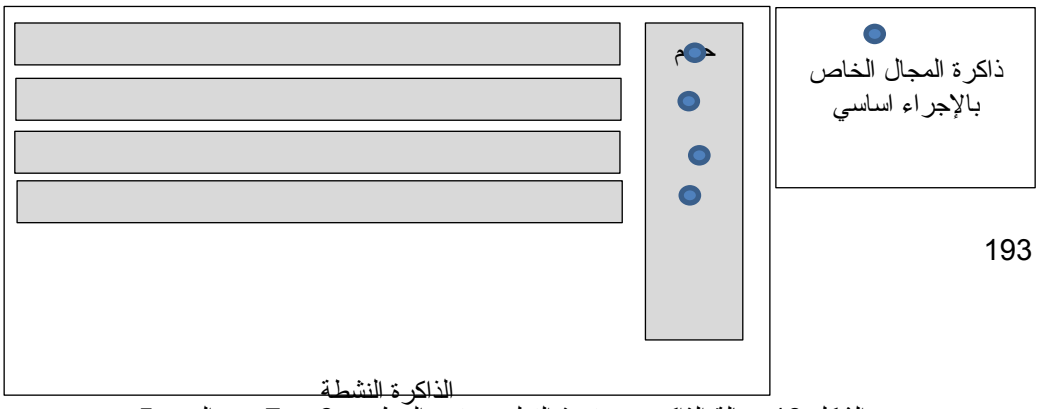


الشكل 8: حالة الذاكرة تنفيذ تعليمة السطر 3 من النص 5

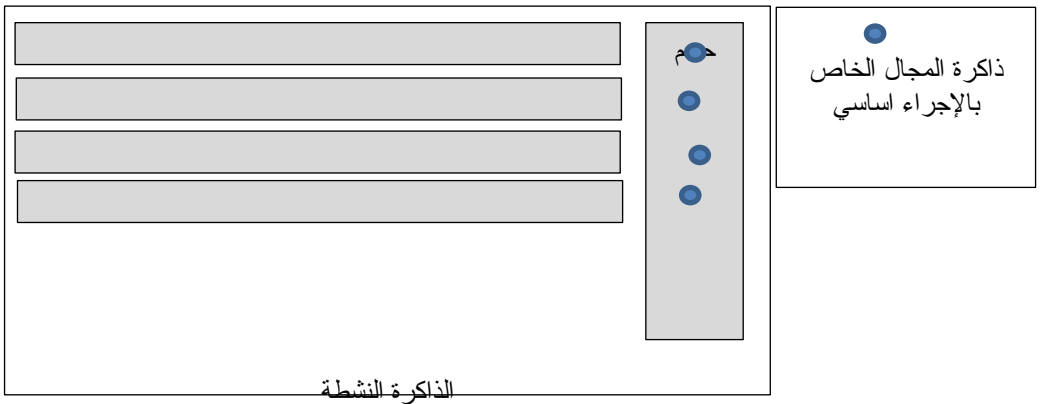
بعد انشاء جدول البعد الأول الطي يحدد عدد الأسطر، يُشرع في انجاز الأسطر الممثلة للبعد الثاني، ويتم هذا من خلال تعليمات الأسطر 4 الى 7، و نلاحظ ان لكل سطر حجم مختلف عن الأحجام الأخرى، ونرى هذه المرحلة من خلال خطأ: لم يُعثر على مصدر المرجع و خطأ: لم يُعثر على مصدر المرجع.



الشكل 9: حالة الذاكرة بعد تنفيذ التعليمين في السطرين 4 و 5 من النص 5



اخيرا نقوم باستغلال الجدول، فمثلا توضع في خاناته بعض القيم، كما يظهر ذلك من خلال تعليمات الأسطر من 9 الى 11 ونلاحظ خطأ جسيما في السطر 12 اذ تطلب الكتابة شحن خانة غير موجودة بالقيمة 8.



الشكل 11: حالة الذاكرة بعد تنفيذ تعليمات الاسطر من 9 الى 11 من النص 5

الفصل التاسع

الأوعية: التجوال في الجداول (1)

الجزء الثاني: الفصل التاسع: الأوعية: التجوال في الجداول (1)

1 – التجوال في الجداول

و نعني بكلمة التجوال، التنقل من خانة الى اخرى لاستغلالها، وهناك طبعا كيفيات عديدة يحددها كاتب الخوارزم، ويصبح التجوال ضرورة عندما يقوم الخوارزم بعملية تشمل كل القيم الموجودة في الجدول او بعضها، فمثلا، اذا كان لدينا جدول من النمط حقيقي ذو بعد واحد، وكل خانة تمثل معدلا لتلميذ في مادة التاريخ، فمن بين العمليات التي تشمل كل الخانات، اي ان هذه العمليات لا بد لها ان تتجول في الجدول فتنتقل من خانة الى اخرى، نجد ما يلي:

- تعمير الجدول بمعدلات التلاميذ في مادة التاريخ: وهنا على الخوارزم ان ينتقل من خانه الى اخرى ليضع فيه المعدل المناسب، فيبدأ من الخانة رقم 0 الممثلة للتلميذ رقم 1، فتوضع فيه معدل التلميذ رقم 1، ثم ينتقل الخوارزم الى الخانة رقم 1 ويضع فيها معدل التلميذ رقم 2، ثم ينتقل الخوارزم الى الخانة رقم 2 وهكذا.
- حساب معدل التلاميذ في مادة التاريخ: لحساب معدل التلاميذ في مادة التاريخ يجب علينا جمع كل القيم الموجودة في الجول ثم تقسيمها بعدد الخانات.
- معرفة اكبر معدل في مادة التاريخ، ولمعرفة اكبر معدل يجب ان نتصفح كل قيم الجدول لاستخراج اكبر قيمة فيه.
- معرفة عدد التلاميذ الذين نجحوا في المادة، اي ان علامتهم اكبر او تساوي 10 / 20، ولمعرفة عدد التلاميذ الذين نجحوا، علينا ان نتخذ عدادا بقيمة اولية تساوي 0، ثم نتصفح القيم واحدة تلو الأخرى، وكلما وجدنا قيمة اكبر او تساوي 10، اضفنا القيمة 1 للعداد.

فكل هذه العمليات تستلزم زيارة الخانات اما لشحنها بقيمة او استغلال ما فيها، وكما علما فان لكل خانة اسم مكون باسم الجدول (وهذا الاسم صلة) متبوع برقم الخانة بين حاضنتين، فمثلا اذا كان **جط** جدول طبيعي دو بعد واحد وحجمه 10، فالكتابة جط[3] تمثل الاسم الثابت للخانة الرابعة، وهنا نتساءل: في اطار عملية يتم من خلالها زيارة كل الخانات لاستغلال محتواها او لشحنها ببعض القيم، هل علينا استعمال الاسم الثابت لكل خانة ام ان استعمال الاسماء الثابتة للخانات لن يكون فعالا؟ وان كانت الاسماء الثابتة غير فعالة، فما هي الطريقة التي علينا سلوكها حتى نتمكن من الزيارة الفعالة لكل خانات الجداول؟

لتبيان فعالية او محدودية استعمال الاسماء الثابتة عندما ينجز الخوارزم عملية تستدعي منه التنقل من خانة لأخرى لاستغلال قيمها، نكتب فيما يلي الخوارزميات التالية:

- خوارزم يعمر كل خانات جدول ما بنفس القيمة.
- خوارزم يعرض على الشاشة قيم كل خانات الجدول.
- خوارزم يعمر جدولا بمعدلات التلاميذ.
- خوارزم يعمر جدولا ذو بعدين بنقاط التلاميذ في مادة ما.

2 – تعمير خانات الجداول ذو بعد واحد بنفس القيمة:

2 – 1 النسخة الأولى: في هذه النسخة الأولى نفترض ان حجم الجدول هو 100، ونريد ان نضع في كل خاناته القيمة 0، فاذا استعملنا الاسماء الثابتة للخانات، علينا ذكر كل خانة في العبارة التي من خلالها توضع القيمة 0 في الخانة، كما يظهر ذلك من خلال النص 41.

لرقم	نص خوارزم تعمير جدول لقيم طبيعية
خوارزم_1_فصل_9_تعمير_جدول_طبيعي {	
1	اجراء اساسي(){
2	طبيعي[ ] جد_طب = شيد طبيعي[100]؛
3	جد_طب[0] = 0 ؛
4	جد_طب[1] = 0 ؛
5	جد_طب[2] = 0 ؛
6	جد_طب[3] = 0 ؛
	.....
	.....
101	جد_طب[98] = 0 ؛
102	جد_طب[99] = 0 ؛
103	} // نهاية الإجراء اساسي
} // نهاية الخوارزم_1_فصل_9_تعمير_جدول_طبيعي	

النص 41: خوارزم تعمير جدول دو بعد واحد بنفس القيمة باستعمال الاسماء الثابتة للخانات

نلاحظ ان النص 41 خاص لتعمير جداول حجمها 100 خانة فقط، واذا اردنا من الجدول ان يعمر جدولا فيه 1000 خانة، يستلزم علينا كتابة جدول آخر يحتوي على ما في النص 41 و إضافة 900 تعليمة اخرى، كل واحدة متخصصة في تعمير خانة واحدة، وهكذا اذا اردنا ان نعمر 100 جدول، كل واحد بحجم خاص به، علينا كتابة 100 خوارزم، وهنا نرى جليا محدودية طريقة استعمال الأسماء الثابتة، و بشكل ادق استعمال القيم الثابتة في النصوص.

علمنا من قبل (الجزء الأول) انه من الأفضل استعمال المتغيرات في مكان القيم الثابتة لجعل الخوارزم مرن، يمكن ان يتأقلم مع تغيرات المحيط، وفي حالنا يمكن مثلا :

- بدل استعمال القيمة ثابتة 100، ننشئ متغيرة تكون قيمتها هي حجم الجدول ونستعمل المتغيرة في عملية انشاء الجدول.
- نستعمل متغيرة تحتوي على رقم الخانة (او اي عبارة يفضي تقييمها الى قيمة طبيعية تمثل رقم خانة) عندما نكتب اسم الخانة في العبارات التي تشحن فيها الخانة بقيمة 0.

انطلاقا من هذه الملاحظات نقوم بإعادة كتابة النص 41 باستعمال المتغيرة **حجم_الجدول** عوض القيمة الثابتة 100 في تعليمة انشاء الجدول واستعمال المتغيرة **رقم_الخانة** التي تحتوي فعلا على قيمة هي رقم الخانة، و نتحصل بذلك على النص 42.

اذا تأملنا في النص 42، نرى اعادة تكرار 100 مرة، كتابة التعليمات التالية

```
جد_طب[رقم_الخانة] = 0 ؛
رقم_الخانة++ ؛
```

وفي آخر تكرار نلاحظ ان محتوى المتغيرة **رقم_الخانة** ينتقل الى القيمة 100، وبعدها لا يعاد التكرار، ومعنى هذا ان التكرار يستمر ما دامت قيمة المتغيرة **رقم_الخانة لم** تصل الى القيمة 100،

وبشكل افضل نقول : ان التكرار يستمر ما دامت قيمة المتغيرة **رقم_الخانة لم** تصل الى القيمة الموجودة في المتغيرة **حجم_الجدول.**

لرقم	نص خوارزم تعمير جدول لقيم طبيعية
خوارزم_2_فصل_9_تعمير_جدول_طبيعي {	
1	اجراء اساسي(){
2	طبيعي حجم_الجدول = 100؛
3	طبيعي[ ] جد_طب = شيد طبيعي[حجم_الجدول]؛
4	طبيعي رقم_الخانة = 0
5	جد_طب[رقم_الخانة] = 0 ؛
6	رقم_الخانة++ ؛
7	
8	جد_طب[رقم_الخانة] = 0 ؛
9	رقم_الخانة++ ؛
10	
11	جد_طب[رقم_الخانة] = 0 ؛
12	رقم_الخانة++ ؛
.....	
.....	
302	جد_طب[رقم_الخانة] = 0 ؛
303	رقم_الخانة++ ؛
304	
305	} // نهاية الإجراء اساسي
} // نهاية الخوارزم_2_فصل_9_تعمير_جدول_طبيعي	

النص 42: خوارزم تعمير جدول ذو بعد واحد بنفس القيمة باستعمال متغيرات في اسماء الخانات

انطلاقا من هذه الملاحظات يمكننا ان نعيد كتابة النص 42 باستعمال تعليمة من تعليمات التكرار، و تكون عبارة التحكم في التكرار هي: **رقم_الخانة > 100،** اوبشكل افضل: **رقم_الخانة< حجم_الجدول،** وبهذه النظرة الجديدة نتحصل على النص 43، ولا ننس ان محتوى المتغيرة **رقم_الخانة** يبدأ من القيمة 0 (السطر 4 من النص 42 و من النص 43).

لرقم	نص خوارزم تعمير جدول لقيم طبيعية
خوارزم_3_فصل_9_تعمير_جدول_طبيعي {	
1	اجراء اساسي(){
2	طبيعي حجم_الجدول = 100؛
3	طبيعي[ ] جد_طب = شيد طبيعي[حجم_الجدول]؛
4	طبيعي رقم_الخانة = 0
5	مادام (رقم_الخانة > حجم_الجدول){
6	جد_طب[رقم_الخانة] = 0 ؛
7	رقم_الخانة++ ؛
8	}
9	} // نهاية الإجراء اساسي
} // نهاية الخوارزم_3_فصل_9_تعمير_جدول_طبيعي	

النص 43: خوارزم تعمير جدول ذو بعد واحد بنفس القيمة باستعمال تعليمة التكرار

نرى جليا نتيجة استعمال تعليمة التكرار في النص 43، إذ تقلص بشكل كبير حجم الخوارزم، لكن لم تحل كل المعضلات، فالخوارزم ما زال رهينا بالقيم الثابتة التي حددت في النص ووضعت في المتغيرات، فلا يمكنه التعاطي مع جداول حجمها مختلف عن 100.

مع هذا التحسن الكبير يبق الخوارزم رهينة القيمة 100 المثبتة في النص للمتغيرة **حجم_الجدول**، فاذا بقي الخوارزم على هذا الحال فلا يمكنه التعاطي الا مع جداول بحجم 100 خانة(النص 43)، فتقييد قيم بعض المتغيرات كالمتغيرة **حجم_الجدول** في نص الخوارزم يحد من مرونة الخوارزم بشكل كبير.

2 – 2 – 1 ما هو الحل لجعل الخوارزم مرن بشكل كاف

راينا في الجزء الأول، انه لجعل خوارزم ما مرنا، يجب ان لا تثبت في النص المتغيرات المتعلقة بالمحيط الذي يستهدفه الخوارزم، بل يجب على الخوارزم ان يتحصل على القيمة من محيطه الخارجي ثم يثبها في المتغيرة، فالتثبيت يتم وقت تنفيذ الخوارزم وليس عند كتابة الخوارزم، وفي حالنا تأتي القيمة من خارج الإجراء اساسي (النص 43).

كما تعلمناه في الجزء الأول، تثبيت قيمة متغيرة من المحيط الخارجي لإجراء او وظيفة يتم باستعمال احدى الطريقتين التاليتين:

- استعمال تعليمات التفاعل مع المحيط الخارجي، وفي غالب الأحيان يكون المستعمل هو مصدر معلومات المحيط الخارجي وتُستعمل التعليمة **اقرأ** لجلب القيمة من المستعمل ووضعها في المتغيرة المستهدفة (المتغيرة **حجم_الجدول** فيما يخصنا).
- استعمال منافذ الإجراء او الوظيفة للحصول على القيمة، وفي هذه الحالة تكون الإجراءات والوظائف الاخرى هي مصدر القيمة التي توضع في المتغيرة المستهدفة (المتغيرة **حجم_الجدول** فيما يخصنا)، وترسل قيمة المحيط الخارجي عندما يطلب تنفيذ الوظيفة (او الإجراء) الذي يحتوي على المتغيرة المستهدفة.

لرقم	نص خوارزم تعمير جدول لقيم طبيعية
خوارزم_4_فصل_9_تعمير_جدول_طبيعي {	
1	اجراء اساسي(){
2	طبيعي حجم_الجدول ، رقم_الخانة ؛
3	طبيعي القيمة_الموحدة ؛
4	طبيعي[ ] جد_طب ؛
5	اكتب "اهلا و سهلا في خوارزم تعمير جدول طبيعي ذو بعد واحد بقيمة طبيعية واحدة"
6	اكتب "فضلا، حدد القيمة التي تريد وضعها في كل خانات الجدول:"
7	اقرأ (القيمة_الموحدة) ؛
8	اكتب "فضلا، حدد حجم الجدول الذي تريد تعميره بالقيمة " + القيمة_الموحدة
9	اقرأ (حجم_الجدول) ؛
10	جد_طب = شيد طبيعي[حجم_الجدول]؛
11	رقم_الخانة = 0 // انطلاق التجوال من الخانة رقم 0
12	مادام (رقم_الخانة < حجم_الجدول){
13	جد_طب[رقم_الخانة] = القيمة_الموحدة ؛
14	رقم_الخانة++ ؛
15	}
16	} // نهاية الإجراء اساسي
} // نهاية الخوارزم_4_فصل_9_تعمير_جدول_طبيعي	

النص 44: خوارزم تعمير جدول ذو بعد واحد :حجم الجدول و القيمة التي توضع في الخانات

2 – 2 – 2 تفعيل الحل الأول: استعمال تعليمات التفاعل مع المحيط الخارجي

التفاعل مع المحيط الخارجي يجلب للخوارزم مرونة كافية للتعاطي مع مختلف الوضعيات التي يكون عليها المحيط الخارجي، وحتى نجعل الخوارزم اكثر مرونة، فلا نكتف بجلب قيمة حجم الجدول الذي نريد ان نحدد كل خاناته بالقيمة 0، بل نزيد، فنجعل المحيط هو الذي يحدد القيمة التي يريد وضعها في كل الخانات، وهذا ما يحتويه النص 44.

هذه المرة لم تحدد اي قيمة ثابتة للمتغيرة **حجم_الجدول**(السطر 2)، واضيفت متغيرة اخرى للنص وهدفها احتواء القيمة التي ستوضع في خانات الجدول، وهي المتغيرة **القيمة_الموحدة**، ونلاحظ ايضا ان المتغيرة **القيمة_الموحدة** لم تحدد لها قيمة مسبقة، فقيمة المتغيرة **القيمة_الموحدة** تحدد بالتفاعل مع المستعمل (السطرين 6 و7)، ونفس السلوك نراه مع المتغيرة **حجم_الجدول** (السطرين 8 و 9).

الجدول الحقيقي يتم انشاؤه هذه المرة بعد ما يتحصل الخوارزم على قيمة المتغيرة **حجم_الجدول** التي نراها تُستعمل في السطر 10 في تعليمة انشاء الجدول، ومعنى هذا ان نفس الخوارزم اذا نفذ عدة مرات، يمكن له ان ينشأ في كل مرة الجدول المناسب ويضع فيه القيمة التي حددها المستعمل، اي ان النص 44 اصبح قادرا على ان يتكيف بسهولة تامة مع رغبات مستعمليه.

2 – 2 – 3 تفعيل الحل الثاني: قيم المتغيرات تأخذ من مداخل الإجراء او الوظيفة

في خضم هذه الطريقة، يستقي الخوارزم (اجراء او وظيفة) قيمة المتغيرتين **حجم_الجدول** و **القيمة_الموحدة** من مداخله، فعلينا اذا اضافة مدخل للإجراء اساسي، لكن هذا الحل غير مجدي، فلقد علمنا مما سبق في الجزء الأول ان:

- الاجراء اساسي هو اول ما ينفذ عندما يشرع في تنفيذ اي خوارزم، فلا ينفذ قبله اي اجراء او وظيفة من اجراءات او وظائف الخوارزم.
- الإجراء (او الوظيفة) الذي يحتوي على منافذ، ونسميه مثلا خ1، يستغله إجراء (او الوظيفة) آخر نسميه مثلا خ2، ومعنى هذا ان خ2 يسبق في التنفيذ خ1، ومعنى هذا، ان الإجراء (او الوظيفة) الذي يحتوي على منافذ وتستغله اجراءات (او وظائف) اخرى من الخوارزم لا يمكن ان يكون هو اول اجراء (او وظيفة) تنفذ في الخوارزم، و بهذا نرى ان مداخل الإجراء اساسي لا جدوى منها، ولا يمكن استغلالها من قبل الإجراءات (او الوظائف) الأخرى.

انطلاقا من هذه الملاحظات، ولكي نتمكن من استعمال المداخل لجلب قيمة المتغيرتين **حجم_الجدول** و**القيمة_الموحدة** ، علينا انشاء اجراء آخر (او وظيفة)، نسميه مثلا **عمر_جدول_بقيمة**،

و يكون الإجراء **اساسي** هو الذي يضع في مداخل الإجراء (او الوظيفة) **عمر_جدول_بقيمة** قيم المتغيرتين **حجم_الجدول** و **القيمة_الموحدة.**

2 – 2 – 3 تفعيل الحل الثاني عبر انجاز اجراء:

اذا اردنا انجاز اجراء، فإن انشاء الجدول يكون بالضرورة على مستعمل الإجراء **عمر_جدول_بقيمة**، اي الإجراء **اساسي** فيما يخصنا، وعند طلب تنفيذ الإجراء **عمر_جدول_بقيمة** يقوم الإجراء **اساسي** بتوفير الجدول والقيمة التي على الإجراء **عمر_جدول_بقيمة** ان يملأ بها الجدول، اما حجم الجدول فلا ضرورة لجعل مدخلا خاصا به في الإجراء **عمر_جدول_بقيمة،** فحجم الجدول متوفر مسبقا في السمة **حجم** للجدول، وعلى الإجراء **عمر_جدول_بقيمة** استغلال هذه السمة في تعليمة التكرار (السطر 3 من النص 45) التي فيها يقوم الإجراء **عمر_جدول_بقيمة** بشحن خانات الجدول بالقيمة التي وصفت له في المدخل.

و هكذا، كما يظهر ذلك في النص 45، يحتوي الإجراء **عمر_جدول_بقيمة** على مدخلين:

- في المدخل الأول نجد الجدول، او بصفة ادق صلة الجدول، ومع الجدول تكون السمة التي تبين حجم الجدول.
- في المدخل الثاني، نجد القيمة التي بها يُعمر الجدول.

لرقم	نص خوارزم تعمير جدول لقيم طبيعية
	} // نهاية الخوارزم_5_فصل_9_تعمير_جدول_طبيعي
1	إجراء عمر_جدول_بقيمة (طبيعي[ ] جد_طب ؛ طبيعي القيمة_الموحدة){
2	طبيعي رقم_الخانة = 0 // انطلاق التجوال من الخانة رقم 0
3	مادام (رقم_الخانة < جد_طب.حجم){ // استغلال السمة حجم للجدول جد_طب
4	جد_طب[رقم_الخانة] = القيمة_الموحدة ؛
5	رقم_الخانة++ ؛
6	}
7	}
8	
9	اجراء اساسي(){
10	طبيعي حجم_الجدول ، رقم_الخانة ؛
11	طبيعي القيمة_الموحدة ؛
12	طبيعي[ ] جد_طب ؛
13	اكتب "اهلا و سهلا في خوارزم تعميرطبيعي جدول ذو بعد واحد بقيمة طبيعية واحدة"
14	اكتب "فضلا، حدد القيمة التي تريد وضعها في كل خانات الجدول:"
15	اقرأ (القيمة_الموحدة) ؛
16	اكتب "فضلا، حدد حجم الجدول الذي تريد تعميره بالقيمة " + القيمة_الموحدة
17	اقرأ (حجم_الجدول) ؛
18	جد_طب = شيد طبيعي[حجم_الجدول]؛
19	عمر_جدول_بقيمة(جد_طب ، القيمة_الموحدة) ؛
20	
21	} // نهاية الإجراء اساسي
	} // نهاية الخوارزم_5_فصل_9_تعمير_جدول_طبيعي

النص 45: خوارزم تعمير جدول ذو بعد واحد :استعمال اجرا متخصص في تعمير الجدول بقيمة واحدة

2 – 2 – 4 تفعيل الحل الثاني عبر انجاز وظيفة:

كما درسناه من قبل، للوظيفة ميزات اضافية، ففيها كل ما في الإجراء وزيادة، فمع الوظيفة يمكن استغلال جدولا أُنشأ خارج الوظيفة، كما يمكن للوظيفة ان تُنشأ جدولا ثم ترجع صلة هذا الجدول الى من طلب تنفيذه (إجراء او وظيفة)، وفيما يخصنا نتبع الطريقة التي تقوم فيها الوظيفة بإنشاء الجدول، وفي هذه الحالة، لا بد للوظيفة ان تتحصل على حجم الجدول الذي سوف تُنشئه والقيمة التي سوف توضع في الجدول، وعند اتمام مهمتها عليها ان ترجع صلة الجدول الذي انشأته، ومعنى هذا ان للوظيفة مدخلين، هما حجم الجدول والقيمة التي توضع في الجدول، اما صلة الجدول الذي تُنشئه الوظيفة، فسوف ترجعه عبر المخرج الرئيسي للوظيفة، وهذا ما نراه من خلال الوظيفة **عمر_جدول_بقيمة** في النص 46 (من السطر 1 الى السطر 9).

لرقم	نص خوارزم تعمير جدول لقيم طبيعية
خوارزم_6_فصل_9_تعمير_جدول_طبيعي {	
1	طبيعي[ ] عمر_جدول_بقيمة (طبيعي حجم، القيمة_الموحدة){
2	طبيعي[ ] جد_طب = شيد طبيعي[حجم]
3	طبيعي رقم_الخانة = 0 // انطلاق التجوال من الخانة رقم 0
4	مادام (رقم_الخانة < حجم){
5	جد_طب[رقم_الخانة] = القيمة_الموحدة ؛
6	رقم_الخانة++ ؛
7	}
8	ارجع جد_طب ؛
9	}
10	
11	اجراء اساسي(){
12	طبيعي حجم_الجدول ، رقم_الخانة ؛
13	طبيعي القيمة_الموحدة ؛
14	طبيعي[ ] جد_طب ؛
15	اكتب "اهلا و سهلا في خوارزم تعميرطبيعي جدول ذو بعد واحد بقيمة طبيعية واحدة"
16	اكتب "فضلا، حدد القيمة التي تريد وضعها في كل خانات الجدول:"
17	اقرأ (القيمة_الموحدة) ؛
18	اكتب "فضلا، حدد حجم الجدول الذي تريد تعميره بالقيمة " + القيمة_الموحدة
19	اقرأ (حجم_الجدول) ؛
20	جد_طب = عمر_جدول_بقيمة(حجم_الجدول ، القيمة_الموحدة) ؛
21	} // نهاية الإجراء اساسي
} // نهاية الخوارزم_6_فصل_9_تعمير_جدول_طبيعي	

النص 46: خوارزم تعمير جدول ذو بعد :استعمال وظيفة متخصصة في تعمير الجدول بقيمة واحدة

وفي النص 46 نلاحظ ما يلي:

- الإجراء اساسي يهتم فقط بجلب حجم الجدول والقيمة التي توضع في الجدول، فلا توجد اي تعليمة لإنشاء جدول في الإجراء اساسي، وبعد الحصول على القيم المطلوبة يقوم الإجراء اساسي بتنفيذ الوظيفة **عمر_جدول_بقيمة** ويلتقط اجابتها، اي صلة الجدول الذي انشأ، في المتغيرة **جد_طب** (السطر 20)،
- نلاحظ في السطر 1 ان نمط الوظيفة **عمر_جدول_بقيمة** هو "**جدول طبيعي ذو بعد واحد**"، اي ان الوظيفة عند انتهاءها ترجع لمستعملها **صلة** تدل على جدول من النمط طبيعي.

اول عمل تقوم به الوظيفة **عمر_جدول_بقيمة** هو انشاء جدول انطلاقا من القيمة التي تتحصل عليها من خلال المدخل **حجم** (السطر 2)، و بعدها، وفي اطار تعليمة التكرار تقوم الوظيفة بملء الجدول بالقيمة التي تحصلت عليها عبر المدخل **القيمة_الموحدة**، وبعد الانتهاء من تعمير الجدول تقوم الوظيفة **عمر_جدول_بقيمة** بإرجاع صلة الجدول (السطر 8).

3 – خوارزم تعمير جدول معدلات التلاميذ في مادة ما:

في هذا المثال الثاني نكرس مرة ثانية عدم جدوى استعمال الاسماء الثابتة للخانات في خوارزم يقوم بالتجوال داخل الجدول للوصول الى هدفه، وهدف هذا الخوارزم هو تعمير جدول سميناه بجدول المعدلات، وهو جدول ذو بعد واحد من النمط **حقيقي** مهمته احتواء معدلات التلاميذ في مادة ما.

النسخة الأولى لهذا الخوارزم والمبنية على الاسماء الثابتة للخانات تظهر في النص 47، ومن خلال هذا النص نرى ان الخوارزم يتفاعل مع المستعمل لجلب معدلات التلاميذ في مادة ما، وعدد المعدلات التي يمكن جلبها هي 25، اي ان هذا الخوارزم موجه لقسم فيه 25 تلميذ، واذا تأملنا في هذا النص نلاحظ ملاحظات شبيهة بتلك التي لوحظت في المثال السابق:

- وجود عدد الكبير من التعليمات مع ان القسم فيه فقط 25 تلميذ، فما بالك اذا كان عدد التلاميذ اكبر.
- لكل خانة (او تلميذ) تُعاد كتابة نفس التعليمات، والفرق بينها طفيف جدا ويتمثل في رقم التلميذ ورقم الخانة مع العلم ان رقم الخانة يساوي **(رقم التلميذ – 1)**.

الرقم	نص خوارزم تعمير جدول المعدلات
	لخوارزم_7_فصل_9_معدلات_التلاميذ {
1	اجراء اساسي(){
2	حقيقي[ ] جد_المعدلات = شيد حقيقي[25]؛
3	
4	اكتب "أهلا و سهلا في خوارزم تعمير جدول المعدلات"
5	اكتب "أدخل معدل التلميذ رقم 1"
6	اقرأ (جد_المعدلات[0])
7	اكتب "أدخل معدل التلميذ رقم 2"
8	اقرأ (جد_المعدلات[1])
9	اكتب "أدخل معدل التلميذ رقم 3"
10	اقرأ (جد_المعدلات[0])
	.....
	.....
53	اكتب "أدخل معدل التلميذ رقم 25"
54	اقرأ (جد_المعدلات[24])
56	} // نهاية الإجراء اساسي
	} // نهاية الخوارزم_7_فصل_9_معدلات_التلاميذ

النص 47: خوارزم تعمير جدول المعدلات باستعمال الأسماء الثابتة للخانات

- خوارزم النص 47 خاص فقط بالإقسام التي فيها 25 تلميذ، فاذا اتبعنا نفس الطريقة، واردنا من الخوارزم ان يحسب معدل قسم فيه 20 تلميذ، علينا بكتابة خوارزم آخر شبيه جدا بالأول، واذا كان لدينا 50 قسما، وكان عدد التلاميذ في كل قسم مختلف عن عدد التلاميذ في الاقسام

الاخرى، سيكون علينا كتابة 50 خوارزما، كل واحد موجه لقسم خاص، وهذه الطريقة تنتج خوارزميات خاصة جدا لا تتكيف مع محيطها، فاذا غادر تلميذ واحد القسم او التحق تلميذ واحد بالقسم علينا اعادة كتابة الخوارزم.

من خلال هذا المثال نرى مجددا ان استعمال الاسماء الثابتة للخانات ينتج خوارزميات يمكن ان تكون ضخمة جدا ولا تتأقلم مع تغير طفيف جدا للمحيط.

3 – 1 البحث عن طريقة لا تستعمل فيها الأسماء الثابتة للخانات

بعد ان لمسنا ثقل وعدم جدوى استعمال الأسماء الثابتة للخانات في عمليات التجوال في الجداول، نتساءل: هل هناك طريقة اخرى، بها يمكننا كتابة خوارزما يتأقلم بفعالية مع تغيرات المحيط؟ للإجابة على السؤال نحاول معرفة السبب الذي جعل الخوارزم يستعمل فقط في حالة خاصة جدا.

اذا تأملنا النص 47، نرى ان السبب يعود الى كثرة استعمال القيم الثابتة، اما عند انشاء جدول المعدلات او عند ذكر اسماء الخانات، وعلمنا من قبل ما يلي:

- من الأفضل استعمال المتغيرات في مكان القيم الثابتة لجعل الخوارزم مرن، يمكن ان يتأقلم مع تغيرات المحيط.
- عندما نريد انشاء جدول يمكن استعمال متغيرة تحتوي على حجم الجدول.
- عندما نريد الوصول الى خانة ما، بدل استعمال قيمة ثابتة، يمكن استعمال متغيرة او اي عبارة يفضي تقييمها الى قيمة طبيعية هي مكان الخانة في الجدول.

انطلاقا من هذه الملاحظات نقوم بإعادة كتابة النص 47 باستعمال المتغيرة **عدد_التلاميذ** عوض القيمة الثابتة 25 في تعليمة انشاء جدول المعدلات، واستعمال المتغيرة **رقم_التلميذ** التي تدل على التلميذ وكذلك على رقم الخانة، فرقم الخانة يساوي رقم **(التلميذ – 1)**، و نتحصل على النص 48.

لرقم	نص خوارزم تعمير جدول المعدلات
خوارزم_8_فصل_9_معدلات_التلاميذ {	
1	اجراء اساسي(){
	طبيعي عدد_التلاميذ = 25؛
2	حقيقي[ ] جد_المعدلات = شيد حقيقي[عدد_التلاميذ]؛
3	طبيعي رقم_التلميذ =1؛
4	اكتب "أهلا و سهلا في خوارزم تعمير جدول المعدلات"
5	اكتب "أدخل معدل التلميذ رقم " + رقم_التلميذ
6	اقرأ (جد_المعدلات[رقم_التلميذ - 1])
7	رقم_التلميذ++
8	اكتب "أدخل معدل التلميذ رقم " + رقم_التلميذ
9	اقرأ (جد_المعدلات[رقم_التلميذ - 1])
10	رقم_التلميذ++
	.....
77	اكتب "أدخل معدل التلميذ رقم " + رقم_التلميذ
78	اقرأ (جد_المعدلات[رقم_التلميذ - 1])
79	رقم_التلميذ++
80	} // نهاية الإجراء اساسي
} // نهاية الخوارزم_8_فصل_9_معدلات_التلاميذ	

النص 48: خوارزم تعمير جدول المعدلات : استعمال متغيرة بدل رقم التلميذ و رقم الخانة

هذه المرة، نلاحظ في النص 48 تكرار نفس التعليمات لجلب معدل التلميذ الى الخانة التي تمثل معدله في مادة التاريخ، وهذه التعليمات هي:

```
اكتب "أدخل معدل التلميذ رقم " + رقم_التلميذ
اقرأ ( جد_المعدلات[رقم_التلميذ - 1] )
رقم_التلميذ++
```

و تكرر هذه التعليمات الثلاث 25 مرة في النص 48، وبشكل ادق تكرر حسب القيمة التي توجد في المتغيرة **عدد_التلاميذ**.

في مثل هذه الحالة، كما رأيناه في المثال السابق، من المفيد جدا ان نستعمل تعليمة من تعليمات التكرار، وتكتب مرة واحدة التعليمات التي تكرر وتتكفل العبارة المتحكمة في عملية التكرار. من تكرار التعليمات العدد الكافي من المرات، وفي ما يخصنا تكون العبارة المتحكمة في تعليمة التكرار. هي **رقم_التلميذ >= عدد_التلاميذ**، فحسب النص تكون القيمة الأولى للمتغيرة **رقم_التلميذ** هي 1 و بعد التنفيذ الأول تنتقل قيمة هذه المتغيرة الى 2، وهكذا حتى تصبح 25، واذا اصبحت 26 ينته التكرار، وهذا ما يبينه النص 49، ونرى بشكل واضح تراجع حجم الخوارزم الى 11 سطرا بعد ان كان 80 سطرا في النص 48.

لرقم	نص خوارزم تعمير جدول المعدلات
خوارزم_9_فصل_9_معدلات_التلاميذ {	
1	اجراء اساسي(){
2	طبيعي عدد_التلاميذ = 25؛
3	حقيقي[ ] جد_المعدلات = شيد حقيقي[عدد_التلاميذ]؛
4	طبيعي رقم_التلميذ =1؛
5	اكتب "أهلا و سهلا في خوارزم تعمير جدول المعدلات"
6	مادام (رقم_التلميذ >= عدد_التلاميذ){
7	اكتب "أدخل معدل التلميذ رقم " + رقم_التلميذ
8	اقرأ (جد_المعدلات[رقم_التلميذ - 1])
9	رقم_التلميذ++
10	}
11	} // نهاية الإجراء اساسي
} // نهاية الخوارزم_9_فصل_9_معدلات_التلاميذ	

النص 49: خوارزم تعمير جدول المعدلات : استعمال تعليمة التكرار

مع هذا التحسن الكبير يبق الخوارزم رهينة القيمة المثبتة في النص للمتغيرة **عدد_التلاميذ**، فاذا بقي الخوارزم على هذا الشكل فسوف لن يكون مرنا ولن يتكيف مع محيطه حتى ولو كان التغيير الذي يطرأ على المحيط خفيفا جدا كالتحاق تلميذ جديد واحد بالقسم او مغادرة تلميذ واحد للقسم.

رأينا في الجزء الأول، انه لجعل خوارزم ما مرنا، يجب ان لا تثبت المتغيرات في النص، بل يجب ان تثبت المتغيرة من المحيط الخارجي الخوارزم (او من خارج الإجراء اساسي)، وكما تعلمناه في الجزء الأول، تثبيت قيمة متغيرة من المحيط الخارجي يتم باستعمال احدى الطريقتين التاليتين:

- استعمال تعليمات التفاعل مع المحيط الخارجي، وفي غالب الأحيان يمثل المستعمل هذا المحيط الخارجي وتُستعمل التعليمة **اقرأ** لجلب القيمة الى المتغيرة، اي ان القيمة تحدد عندما ينفذ الخوارزم ولا تحدد في نص الخوارزم.

- استعمال منافذ الإجراء او الوظيفة للحصول على القيمة، وفي هذه الحالة تجلب القيمة من وظيفة او اجراء يطلب تنفيذ الإجراء او الوظيفة المحتوية على المنافذ التي تأتي القيمة من خلالها.

لرقم	نص خوارزم تعمير جدول المعدلات
	خوارزم_10_فصل_9_معدلات_التلاميذ {
1	اجراء اساسي(){
2	طبيعي عدد_التلاميذ؛
3	حقيقي[ ] جد_المعدلات ؛
4	طبيعي رقم_التلميذ =1؛
5	اكتب "أهلا و سهلا في خوارزم تعمير جدول المعدلات"
6	اكتب " فضلا، وفني بعدد التلاميذ في القسم"
7	اقرأ (عدد_التلاميذ)
8	جد_المعدلات = شيد حقيقي[عدد_التلاميذ]؛
9	مادام (رقم_التلميذ >= عدد_التلاميذ){
10	اكتب "أدخل معدل التلميذ رقم " + رقم_التلميذ
11	اقرأ (جد_المعدلات[رقم_التلميذ - 1])
12	رقم_التلميذ++
13	}
14	} // نهاية الإجراء اساسي
	} // نهاية الخوارزم_10_فصل_9_معدلات_التلاميذ

النص 50: خوارزم تعمير جدول المعدلات : استعمال التفاعل لجلب عدد التلاميذ

3 – 2 استعمال تعليمات التفاعل مع المحيط الخارجي:

في هذه الحالة علينا كتابة التعليمات التي تتفاعل مع المستعمل للحصول على عدد التلاميذ، وبعدها يقوم الخوارزم بإنشاء جدول المعدلات بحجم عدد التلاميذ، ثم، في اطار تعليمة التكرار يقوم بتعمير الجدول، وهذا ما نشاهده في النص 50، ففي هذا النص نلاحظ اولا ان قيمة المتغيرة لم تثبت في النص (السطر 2)، وأن إنشاء الجدول يتم بعد الحصول على عدد التلاميذ، فيمكن ان يكون العدد هو 25 كما يمكن ان يكون 30 او 17 او اي عدد، ومعنى هذا ان الخوارزم اصبح الآن مرن ويتأقلم مع اي عدد من التلاميذ في القسم، فمثلا اذا نفذنا الخوارزم اليوم وكان عدد التلاميذ 30، وفي الغد غادر 8 تلاميذ القسم، فعند التنفيذ نعط فقط الرقم الجديد، اي 22، ولا يُمس نص الخوارزم.

3 – 3 استعمال منافذ الإجراء او الوظيفة للحصول على القيمة:

لكي نتمكن من استعمال المداخل لجلب قيمة ما كقيمة عدد التلاميذ التي تستعمل في انشاء جدول المعدلات، علينا انشاء وظيفة او اجراء آخر، وهذا ما نراه في النص 51، ففيه أُنجزت وظيفة لتعمير جدول المعدلات، وهي الوظيفة **عمر_جدول** التي تحتوي على مدخل واحد يمثل عدد التلاميذ، ويتم فيها انشاء جدول المعدلات انطلاقا من القيمة التي توضع في المدخل **عدد_التلاميذ**، وفي اطار الوظيفة **عمر_جدول** يتم كذلك تعمير الجدول بمعدلات التلاميذ.

نلاحظ في السطر الأول ان نمط الوظيفة هو جدول **حقيقي** ذو بعد واحد، اي ان الوظيفة عند انتهاءها ترجع لمستعملها (السطر 9) صلة تدل على جدول المعدلات الذي أُنشأ (السطر 2) في سياق تنفيذها.

لرقم	نص خوارزم تعمير جدول المعدلات
خوارزم_11_فصل_9_معدلات_التلاميذ {	
1	حقيقي[ ] عمر_جدول(طبيعي عدد_التلاميذ){
2	حقيقي[ ] جد_المعدلات = شيد حقيقي[عدد_التلاميذ]؛
3	طبيعي رقم_التلميذ =1؛
4	مادام (رقم_التلميذ >= عدد_التلاميذ){
5	اكتب "أدخل معدل التلميذ رقم " + رقم_التلميذ
6	اقرأ (جد_المعدلات[رقم_التلميذ - 1])
7	رقم_التلميذ++
8	}
9	ارجع جد_المعدلات
10	}
11	
12	اجراء اساسي(){
13	طبيعي عدد_التلاميذ؛
14	حقيقي[ ] جد_المعدلات ؛
15	اكتب "أهلا و سهلا في خوارزم تعمير جدول المعدلات"
16	اكتب " فضلا، وفني بعدد التلاميذ في القسم"
17	اقرأ (عدد_التلاميذ)
18	جد_المعدلات = عمر_جدول(عدد_التلاميذ)
19	}
20	} // نهاية الإجراء اساسي
} // نهاية الخوارزم_11_فصل_9_معدلات_التلاميذ	

النص 51: خوارزم تعمير جدول المعدلات : استعمال التفاعل لجلب عدد التلاميذ

4 – خوارزم تعمير جدول ذو بعدين بقيمة واحدة:

يحتوي النص 52 على خوارزم يعمر جدول طبيعي ذو بعدين بقيمة واحدة، واستعملت فيه المتغيرات بدل القيم الثابتة، لكن قيم المتغيرات ثبتت في النص، مما يجعل النص خاص بالمحيط الذي تمثله القيم التي ثبتت في المتغيرات.

اذا تأملنا في هذا الخوارزم، نرى تكرار نفس التعليمات، لكن هذه المرة على مستويين:

- المستوى الأول هو مستوى السطر، ففي اطار سطر ما نرى مجموعة من التعليمات يعاد كتابتها في الأسطر الأخرى، ومعنى هذا انه يمكن لنا استعمال تعليمة التكرار وكتابة مرة واحدة في جسم تعليمة التكرار التعليمات التي تعاد على مستوى كل سطر، ويبدأ التكرار من السطر رقم 0 و يستمر التكرار حتى تتم زيارة كل الأسطر، اي عندما تصبح قيمة المتغيرة **رقم_السطر** تساوي 19، اي ان تعليمات جسم تعليمة التكرار نفذت عندما كانت قيمة المتغيرة **رقم_السطر** هي 0، ثم 1، ثم 2 الى القيمة 19، اما عندما اصبحت القيمة هي 20، فإن التعليمات لم يُعد تنفيذها لعدم وجود السطر رقم 20 (اي السطر 21)، واستعمال تعليمة التكرار على مستوى السطر جعلنا نتحصل على النص 53.

- المستوى الثاني هو مستوى الأعمدة: التأمل في النص 53 يجعلنا نلاحظ في جسم تعليمة التكرار، تكرار كتابة نفس التعليمات، وهذه التعليمات هي التي تتولى زيارة كل الخانات الموجودة في كل سطر من اسطر الجدول، و هذا ما يجعلنا نفكر بإمكانية اعادة كتابة جسم التكرار على شكل تعليمة تكرار، فنتحصل في الأخير على تكرار داخل تكرار، التكرار الخارجي يخص الأسطر. والتكرار الداخلي يخص خانات السطر.

لرقم	نص خوارزم تعميز جدول المعدلات
خوارزم_12_فصل_9_تعمير_جدول_ذو_بعدين_بقيمة_واحدة {	
1	اجراء اساسي(){
2	طبيعي عدد_الأسطر=20، عدد_ألأعمدة = 9
3	طبيعي [ ] [ ] جد_طب2 = شيد طبيعي [عدد_الأسطر] [عدد_ألأعمدة] ؛
4	طبيعي القيمة_الموحدة = 100 ؛
5	طبيعي رقم_السطر، رقم_العمود؛
6	
7	رقم_السطر=0؛ // البداية من السطر رقم 0، اي السطر الأول
8	رقم_العمود=0؛ // في اطار السطر الحالي تكون البداية من الخانة رقم 0 (اي الخانة الأولى او العمود الأول)
9	جد_طب2 [رقم_السطر] [رقم_العمود] = القيمة_الموحدة
10	رقم_العمود++ ؛ // الانتقال الى العمود (او الخانة) التالي، اي العمود رقم 1
11	
12	جد_طب2 [رقم_السطر] [رقم_العمود] = القيمة_الموحدة
13	رقم_العمود++ ؛ // الانتقال الى العمود (او الخانة) التالي ، اي العمود رقم 2
.....	
33	جد_طب2 [رقم_السطر] [رقم_العمود] = القيمة_الموحدة
34	رقم_العمود++ ؛ // الانتقال الى العمود رقم 9، وهو غير موجود، أي ان كل اعمدة السطر قد تم زيارتها، ننتقل الى السطر التالي
35	
36	رقم_السطر++ ؛ الانتقال الى السطر رقم 1، اي السطر الثاني
37	رقم_العمود=0؛ // في اطار السطر الحالي تكون البداية من الخانة رقم 0 (اي الخانة الأولى او العمود الأول)
38	جد_طب2 [رقم_السطر] [رقم_العمود] = القيمة_الموحدة
39	رقم_العمود++ ؛ // الانتقال الى العمود (او الخانة) التالي، اي العمود رقم 1
.....	
62	جد_طب2 [رقم_السطر] [رقم_العمود] = القيمة_الموحدة
63	رقم_العمود++ ؛ // الانتقال الى العمود رقم 9، وهو غير موجود، أي ان كل اعمدة السطر قد تم زيارتها، ننتقل الى السطر التالي
.....	
581	رقم_السطر++ ؛ الانتقال الى السطر رقم 20، وهو غير موجود، أي ان كل ا الأسطر قد تم زيارتها. انتهت زيارة الجدول.
582	} // نهاية الإجراء اساسي
} // نهاية الخوارزم_12_فصل_9_تعمير_جدول_ذو_بعدين_بقيمة_واحدة	

النص 52: خوارزم تعمير جدول طبيعي ذو بعدين بقيمة واحدة

لرقم	نص خوارزم تعمير جدول المعدلات
خوارزم_13_فصل_9_تعمير_جدول_ذو_بعدين_بقيمة_واحدة {	
1	اجراء اساسي(){
2	طبيعي عدد_الأسطر=20، عدد_ألأعمدة = 9
3	طبيعي [ ] [ ] جد_طب2 = شيد طبيعي [ ] [ ] ؛
4	طبيعي القيمة_الموحدة = 100 ؛
5	طبيعي رقم_السطر، رقم_العمود؛
6	
7	رقم_السطر=0؛ // البداية من السطر رقم 0، اي السطر الأول
8	مادام(رقم_السطر > عدد_الأسطر){
9	رقم_العمود=0؛ // في اطار السطر الحالي تكون البداية من الخانة رقم 0 (اي الخانة الأولى او العمود الأول)
10	جد_طب2 [رقم_السطر] [رقم_العمود] = القيمة_الموحدة
11	رقم_العمود++ ؛ // الانتقال الى العمود (او الخانة) التالي، اي العمود رقم 1
12	
13	جد_طب2 [رقم_السطر] [رقم_العمود] = القيمة_الموحدة
14	رقم_العمود++ ؛ // الانتقال الى العمود (او الخانة) التالي ، اي العمود رقم 2

```
.....
31        جد_طب2 [رقم_السطر] [رقم_العمود] = القيمة_الموحدة
32        رقم_العمود++ ؛   // الانتقال الى العمود (او الخانة) التالي ، اي العمود رقم 9
33
34        رقم_السطر++  ؛ الانتقال الى السطر التالي
35    }
36  } // نهاية الإجراء اساسي
} // نهاية الخوارزم_13_فصل_9_تعمير_جدول_ذو_بعدين_بقيمة_واحدة
```

النص 53: خوارزم تعمير جدول طبيعي ذو بعدين بقيمة واحدة باستعمال تعليمة التكرار على مستوى الأسطر

التكرار الداخلي الذي يخص خانات السطر. يبدأ من الخانة رقم 0 الى الخانة رقم 8، فاذا اصبح رقم الخانة هو 9، فمعنى هذا ان كل خانات السطر قد تمت زيارتها، و بذلك تنته تعليمة التكرار، و هذا ما نده في النص 54.

```
لرقم  نص خوارزم تعمير جدول المعدلات
خوارزم_14_فصل_9_تعمير_جدول_ذو_بعدين_بقيمة_واحدة {
1   اجراء اساسي(){
2       طبيعي عدد_الأسطر=20، عدد_ألأعمدة = 9
3       طبيعي [ ] [ ]  جد_طب2 = شيد طبيعي [عدد_الأسطر] [عدد_ألأعمدة]  ؛
4       طبيعي القيمة_الموحدة = 100 ؛
5       طبيعي رقم_السطر، رقم_العمود؛
6
7       رقم_السطر=0؛ // البداية من السطر رقم 0، اي السطر الأول
8       مادام(رقم_السطر  >  عدد_الأسطر){
9           رقم_العمود=0؛  // في اطار السطر الحالي تكون  البداية من الخانة رقم 0 (اي الخانة  الأولى او العمود الأول)
10          مادام(رقم_العمود  >  عدد_الأعمدة){
11              جد_طب2 [رقم_السطر] [رقم_العمود] = القيمة_الموحدة
12              رقم_العمود++ ؛   // الانتقال الى العمود (او الخانة) التالي،
13          }
14          رقم_السطر++  ؛ الانتقال الى السطر التالي
15      }
16  } // نهاية الإجراء اساسي
} // نهاية الخوارزم_14_فصل_9_تعمير_جدول_ذو_بعدين_بقيمة_واحدة
```

النص 54: خوارزم تعمير جدول طبيعي ذو بعدين بقيمة واحدة باستعمال تعليمة التكرار على مستوى الأسطر والأعمدة

```
لرقم  نص خوارزم تعمير جدول المعدلات
خوارزم_15_فصل_9_تعمير_جدول_ذو_بعدين_بقيمة_واحدة {
1   طبيعي [ ] [ ] عمر_جدول_ط2 (طبيعي عدد_الأسطر، عدد_الأعمدة ، القيمة_الموحدة){
2       طبيعي [ ] [ ]  جد_طب2 = شيد طبيعي [عدد_الأسطر] [عدد_الأعمدة ] ؛
3       طبيعي رقم_السطر=0؛  // البداية من السطر رقم 0، اي السطر الأول
4       مادام(رقم_السطر  >  عدد_الأسطر){
5           رقم_العمود=0؛  // في اطار السطر الحالي تكون  البداية من الخانة رقم 0 (اي الخانة  الأولى او العمود الأول)
6           مادام(رقم_العمود  >  عدد_الأعمدة){
7               جد_طب2 [رقم_السطر] [رقم_العمود] = القيمة_الموحدة
8               رقم_العمود++ ؛   // الانتقال الى العمود (او الخانة) التالي،
9           }
10          رقم_السطر++  ؛ الانتقال الى السطر التالي
11      }
12      ارجع جد_طب2 ؛
13  } // نهاية الوظيفة عمر_جدول_ط2
14  اجراء اساسي(){
15      طبيعي [ ] [ ]  جد_طب2 ؛
16      طبيعي عدد_الأسطر، عدد_ألأعمدة، القيمة_الموحدة ؛
17      اكتب " فضلا، ادخل على التوالي: عدد الأسطر، عدد الأعمدة و القيمة التي تريد ان تعمر بها الجدول"
18      اقرأ (عدد_الأسطر، عدد_ألأعمدة، القيمة_الموحدة)
19      جد_طب2 = عمر_جدول_ط2 (عدد_الأسطر، عدد_ألأعمدة، القيمة_الموحدة) ؛
20  } // نهاية الإجراء اساسي
} // نهاية الخوارزم_15_فصل_9_تعمير_جدول_ذو_بعدين_بقيمة_واحدة
```

النص 55: خوارزم مرن لتعمير جدول طبيعي ذو بعدين بقيمة واحدة و هوغير مرتبط بمحيط خاص

4 – 1 وظيفة تعمير جدول طبيعي ذو بعدين

خوارزم النص 54، برغم التحسن الذي ظهر عليه، يبق خاصا بحالة واحدة، وهي تعمير جدول ذو بعدين فيه 20 سطرا و 9 اعمدة، ولجعله غير مرتبط بأي محيط، علينا ان لا نثبت القيم المتعلقة بالمحيط في النص، بل علينا ان نجعل الخوارزم يتحصل عليها من محيطه الخارجي، فلكل محيط قيم فيما يخص عدد الأسطر وعدد الأعمدة، وللوصول الى هذه الحالة نعيد كتابة النص 54 فنتحصل على النص 55 الذي يحتوي على الإجراء اساسي والوظيفة **عمر_جدول_ط2**.

تقوم الوظيفة **عمر_جدول_ط2** بإنشاء جدول ذو بعدين وتعميره بنفس القيمة، ويرتكز عمل الوظيفة على ثلاث معطيات تجلب من مداخلها الثلاث: **عدد_الأسطر**، **عدد_الأعمدة** و **القيمة_الموحدة**، فالمدخلين الأولين يحددان حجم البعد الأول والثاني من الجدول، اما المدخل الثالث فيحدد القيمة التي توضع في الجدول، وبعد الانتهاء من عملها، ترجع الوظيفة صلة الجدول الذي انشأته.

4– 2 خوارزم تعمير جدول النقاط:

في هذا الخوارزم نريد تعمير جدول ذو بعدين بالنقاط التي تحصل عليها التلميذ في امتحانات مادة ما، ولنفرض ان المادة هي التاريخ، ونفترض ان عدد الامتحانات التي عقدت في مادة التاريخ هي 4، اي ان عدد النقاط هي 4، و نفترض ان عدد التلاميذ هو 25، و هكذا يكون في الجدول 25 سطرا (الجدول 13)، كل سطر يمثل تلميذ من التلاميذ، فالسطر رقم 0 يمثل التلميذ رقم 1، و السطر رقم 1 يمثل التلميذ رقم 2، و هكذا، وفي سطر ما، تحتوي كل خانة على نقطة في امتحان ما، فالخانة رقم 0 تحتوي على نقطة الامتحان الأول، والخانة رقم 1 تحتوي على نقطة الامتحان الثاني الخ... (الجدول 13).

	نقطة الامتحان 1	نقطة الامتحان 2	نقطة الامتحان 3	نقطة الامتحان 4
نقاط التلميذ رقم 1	12.5	13.5	8.5	15.00
نقاط التلميذ رقم 2	10.5	14.00	18.5	12.00
نقاط التلميذ رقم 3	7.5	12.00	8.00	4.5
.....				
نقاط التلميذ رقم 24	6.5	10.00	11.00	10.5
نقاط التلميذ رقم 25	18.5	17.00	15.00	19.00

جدول 13: نموذج لجدول النقاط في امتحانات مادة ما

4– 3 سلوك الخوارزم:

انطلاقا من القيم الثابتة التي تمثل عدد الطلبة وعدد الامتحانات، يقوم الخوارزم في بداية تنفيذه بإنشاء جدول ذو بعدين من النمط حقيقي، فيه 25 سطرا و في كل سطر 4 خانات (السطر 2 من النص 56)، وبعدها يقوم الخوارزم بالتفاعل مع المستعمل لإعمار الجدول.

في هذا الخوارزم يتم اعمار الجدول سطرا بعد سطر، ونبدأ بالسطر الأول، اي السطر رقم 0، و في كل سطر تعمر الخانات الواحدة تلو الأخرى، و نبدأ بالخانة الأولى، اي الخانة رقم 0، و كما فعلنا في المثال السابق، ونرى ان عدد من التعليمات المتشابهة تكرر عدة مرات، وفي هذا الخوارزم نلاحظ مستويين في التكرار:

- التكرار على مستوى الأسطر، وهو التكرار الأول (او الخارجي)، فعند الانتهاء من سطر نعيد نفس العمليات مع السطر التالي
- التكرار على مستوى الأعمدة (او خانات السطر)، وهذا تكرار يقع داخل التكرار السابق، ففي سطر ما، كلما انتقلنا من خانة الى الخانة التالية تعاد نفس التعليمات.

في النص 56، التعليمات التي تعاد، تختلف فقط على مستوى رقم السطر ورقم الخانة في السطر، واذا استعملنا المتغيرات بدل القيم الثابتة للأسطر والخانات، سوف نرى ان نفس العمليات تردد عدة مرات، وهذا ما يبينه النص 57.

لرقم	نص خوارزم تعمير جدول المعدلات
خوارزم_16_فصل_9_تعمير_جدول_النقاط_ذو_بعدين {	
1	اجراء اساسي(){
2	حقيقي[ ] [ ] جد_النقاط = شيد حقيقي[25][4]؛
3	اكتب "أهلا و سهلا في خوارزم تعمير جدول النقاط"
4	اكتب " فضلا، وفني بنقاط التلميذ رقم 1"
5	اكتب " وفني بالنقطة الأولى"
6	اقرأ (جد_النقاط[0][0])
7	اكتب " وفني بالنقطة الثانية "
8	اقرأ (جد_النقاط[0][1])
9	اكتب " وفني بالنقطة الثالثة"
10	اقرأ (جد_النقاط[0][2])
11	اكتب " وفني بالنقطة الرابعة"
12	اقرأ (جد_النقاط[0][3])
13	اكتب " فضلا، وفني بنقاط التلميذ رقم 2"
14	اكتب " وفني بالنقطة الأولى"
15	اقرأ (جد_النقاط[1][0])
16	اكتب " وفني بالنقطة الثانية "
17	اقرأ (جد_النقاط[1][1])
18	اكتب " وفني بالنقطة الثالثة"
19	اقرأ (جد_النقاط[1][2])
20	اكتب " وفني بالنقطة الرابعة"
21	اقرأ (جد_النقاط[1][3])
.....	
220	اكتب " فضلا، وفني بنقاط التلميذ رقم 25"
221	اكتب " وفني بالنقطة الأولى"
222	اقرأ (جد_النقاط[24][0])
223	اكتب " وفني بالنقطة الثانية "
224	اقرأ (جد_النقاط[24][1])
225	اكتب " وفني بالنقطة الثالثة"
226	اقرأ (جد_النقاط[24][2])
227	اكتب " وفني بالنقطة الرابعة"
228	اقرأ (جد_النقاط[24][3])

```
    } // نهاية الإجراء اساسي
} // نهاية الخوارزم_16_فصل_9_تعمير_جدول_النقاط_ذو_بعدين
```

النص 56: خوارزم تعمير جدول النقاط ز هو جدول ذو بعدين

نلاحظ من خلال النص 57، ان نفس التعليمات الموجودة في الأسطر من 8 الى 21، يتم اعادة كتابتها 25 مرة، فتكتب أولا عندما تكون المتغيرة **رقم_السطر** تحتوي على القيمة 0، و هي قيمة رقم السطر الأول، ثم تكرر. هذه التعليمات عندما تصبح قيمة المتغيرة **رقم_السطر.** هي 1، ثم 2 حتى تصبح القيمة تساوي 24، وهذه القيمة هي رقم آخر سطر، وعندما تصبح قيمة المتغيرة **رقم_السطر** هي 25، ينته تكرار الكتابة وينته الخوارزم، وننبه ان القيمة 25 هي القيمة الموجودة في المتغيرة **عدد_الاسطر،** اي انه في مثل هذه الحالة من الأفضل استعمال المتغيرة **عدد_الاسطر** بدل القيمة الثابتة 25.

لرقم	نص خوارزم تعمير جدول المعدلات
خوارزم_17_فصل_9_تعمير_جدول_النقاط_ذو_بعدين {	
1	اجراء اساسي(){
2	طبيعي رقم_السطر ، رقم_العمود ؛
3	طبيعي عدد_الاسطر = 25، عدد_الاعمدة = 4 ؛
4	حقيقي[ ] [ ] جد_النقاط = شيد حقيقي[عدد_الاسطر][عدد_الاعمدة]؛
5	اكتب "أهلا و سهلا في خوارزم تعمير جدول النقاط"
6	رقم_السطر = 0 ؛ // نبدأ بالسطر الأول و رقمه 0
7	
8	اكتب " فضلا، وفني بنقاط التلميذ رقم " + (رقم_السطر+1)
9	رقم_العمود = 0 ؛ // في السطر // نبدأ بالعمود الأول
10	اكتب " وفني بالنقطة رقم" + (رقم_العمود+1)
11	اقرأ (جد_النقاط[رقم_السطر][رقم_العمود])
12	رقم_العمود++ ؛
13	اكتب " وفني بالنقطة رقم" + (رقم_العمود+1)
14	اقرأ (جد_النقاط[رقم_السطر][رقم_العمود])
15	رقم_العمود++ ؛
16	اكتب " وفني بالنقطة رقم" + (رقم_العمود+1)
17	اقرأ (جد_النقاط[رقم_السطر][رقم_العمود])
18	رقم_العمود++ ؛
19	اكتب " وفني بالنقطة رقم" + (رقم_العمود+1)
20	اقرأ (جد_النقاط[رقم_السطر][رقم_العمود])
21	رقم_السطر++ ؛ // ننتقل الى السطر التالي ، هنا السطرالثاني و رقمه 1
22	
23	اكتب " فضلا، وفني بنقاط التلميذ رقم " + (رقم_السطر+1)
24	رقم_العمود = 0 ؛ // في السطر // نبدأ بالعمود الأول
25	اكتب " وفني بالنقطة رقم" + (رقم_العمود+1)
26	اقرأ (جد_النقاط[رقم_السطر][رقم_العمود])
27	رقم_العمود++ ؛
28	اكتب " وفني بالنقطة رقم" + (رقم_العمود+1)
29	اقرأ (جد_النقاط[رقم_السطر][رقم_العمود])
30	رقم_العمود++ ؛
31	اكتب " وفني بالنقطة رقم" + (رقم_العمود+1)
32	اقرأ (جد_النقاط[رقم_السطر][رقم_العمود])
33	رقم_العمود++ ؛
34	اكتب " وفني بالنقطة رقم" + (رقم_العمود+1)
35	اقرأ (جد_النقاط[رقم_السطر][رقم_العمود])
.....	
362	اكتب " وفني بالنقطة رقم" + (رقم_العمود+1) // تعليمات خاصة بالسطر 23
363	اقرأ (جد_النقاط[رقم_السطر][رقم_العمود])
364	رقم_السطر++ ؛ // ننتقل الى السطر التالي ، هنا السطر الخامس و العشرون و رقمه 24
265	
366	اكتب " فضلا، وفني بنقاط التلميذ رقم " + (رقم_السطر+1)

```
367         رقم_العمود = 0 ؛  // في السطر // نبدأ بالعمود الأول
368         اكتب " وفني بالنقطة رقم" + (رقم_العمود+1)
369         اقرأ (جد_النقاط[رقم_السطر][ رقم_العمود])
370         رقم_العمود++ ؛
371         اكتب " وفني بالنقطة رقم" + (رقم_العمود+1)
372         اقرأ (جد_النقاط[رقم_السطر][ رقم_العمود])
373         رقم_العمود++ ؛
374         اكتب " وفني بالنقطة رقم" + (رقم_العمود+1)
376         اقرأ (جد_النقاط[رقم_السطر][ رقم_العمود])
377         رقم_العمود++ ؛
378         اكتب " وفني بالنقطة رقم" + (رقم_العمود+1)
379         اقرأ (جد_النقاط[رقم_السطر][ رقم_العمود])
380         رقم_السطر++ ؛ // ننتقل الى السطر التالي ، هنا السطر الخامس و العشرون و رقمه 25
381     } // نهاية الإجراء اساسي
} // نهاية الخوارزم_17_فصل_9_تعمير_جدول_النقاط_ذو_بعدين
```

النص 57: خوارزم تعمير جدول النقاط باستعمال متغيرات بدل القيم الثابتة التي استعملت في النص 56

في مثل هذه الحالة، بدل اعادة كتابة التعليمات 25 مرة، نكتبها مرة واحدة في اطار تعليمة تكرار وتكون قيمة المتغيرة **رقم_السطر.** هي المتحكمة في التكرار، فما دامت هذه القيمة اصغر من 25، اي قيمة المتغيرة **عدد_الاسطر**، يكرر. تنفيذ التعليمات الموجودة في جسم تعليمة التكرار، وهذا ما نراه في النص 58، ففي السطر 8 نرى رأس تعليمة التكرار والعبارة المنطقية المتحكمة في التكرار، وفي السطر 27 نرى التعليمة التي تجعل التكرار ينتقل من سطر الى السطر الذي يليه في جدول النقاط **جد_النقاط.**

```
لرقم    نص خوارزم  تعمير جدول المعدلات
خوارزم_18_فصل_9_تعمير_جدول_النقاط_ذو_بعدين {
1       اجراء اساسي(){
2           طبيعي رقم_السطر ، رقم_العمود  ؛
3           طبيعي عدد_الاسطر = 25، عدد_الاعمدة = 4 ؛
4           حقيقي[ ] [ ] جد_النقاط = شيد  حقيقي[عدد_الاسطر][ عدد_الاعمدة]؛
5           اكتب "أهلا و سهلا في خوارزم تعمير جدول النقاط"
6           رقم_السطر = 0 ؛   // نبدأ بالسطر الأول و رقمه 0
7
8           مادام (رقم_السطر >  عدد_الاسطر){
9               اكتب " فضلا، وفني بنقاط التلميذ رقم " + (رقم_السطر+1)
10              رقم_العمود = 0 ؛  // في السطر // نبدأ بالعمود الأول
11
12              اكتب " وفني بالنقطة رقم" + (رقم_العمود+1)
13              اقرأ (جد_النقاط[رقم_السطر][ رقم_العمود])
14              رقم_العمود++ ؛
15
16              اكتب " وفني بالنقطة رقم" + (رقم_العمود+1)
17              اقرأ (جد_النقاط[رقم_السطر][ رقم_العمود])
18              رقم_العمود++ ؛
19
20              اكتب " وفني بالنقطة رقم" + (رقم_العمود+1)
21              اقرأ (جد_النقاط[رقم_السطر][ رقم_العمود])
22              رقم_العمود++ ؛
23
24              اكتب " وفني بالنقطة رقم" + (رقم_العمود+1)
25              اقرأ (جد_النقاط[رقم_السطر][ رقم_العمود])
26
27              رقم_السطر++ ؛ // ننتقل الى السطر التالي ، هنا السطرالثاني و رقمه 1
28          }
29      } // نهاية الإجراء اساسي
} // نهاية الخوارزم_18_فصل_9_تعمير_جدول_النقاط_ذو_بعدين
```

النص 58: خوارزم تعمير جدول النقاط باستعمال تعليمة التكرار على مستوى الأسطر

في النص 58، نلاحظ ان جسم تعليمة التكرار. (من السطر 8 الى السطر 28) يحتوي ايضا على تعليمات الأسطر من 12 الى 14 تكررت كتابتها اربع مرات، ونلاحظ انه في المرة الأخيرة لم تكتب التعليمة **رقم_العمود++**، فلو كتبت لأًصبح رقم العمود هو 4، هذه القيمة خارجة عن نطاق ارقام الخانات التي تبدأ من 0 و تنته عند 3، وكما فعلنا مع التكرار على مستوى الأسطر، يمكننا ان نستعمل تعليمة التكرار لتفادي تكرار كتابة نفس التعليمات 4 مرات (النص 59)، و تكون عبارة التحكم في هذا التكرار على مستوى الخانات مبنية على القيمة الموجودة في المتغيرة **رقم_العمود**، فما دامت هذه القيمة اقل من 4، اي اقل من قيمة المتغيرة **عدد_الاعمدة** (السطر. 12 من النص 59)، يكرر تنفيذ التعليمات، اما اذا تجاوزت قيمة المتغيرة **رقم_العمود** القيمة 3، ينته التكرار فورا.

لرقم	نص خوارزم تعمير جدول المعدلات
خوارزم_19_فصل_9_تعمير_جدول_النقاط_ذو_بعدين {	
1	اجراء اساسي(){
2	طبيعي رقم_السطر ، رقم_العمود ؛
3	طبيعي عدد_الاسطر = 25، عدد_الاعمدة = 4 ؛
4	حقيقي[ ] [ ] جد_النقاط = شيد حقيقي[عدد_الاسطر][عدد_الاعمدة]؛
5	اكتب "أهلا و سهلا في خوارزم تعمير جدول النقاط"
6	رقم_السطر = 0 ؛ // نبدأ بالسطر الأول و رقمه 0
7	
8	مادام (رقم_السطر < عدد_الاسطر){
9	اكتب " فضلا، وفني بنقاط التلميذ رقم " + (رقم_السطر+1)
10	
11	رقم_العمود = 0 ؛ // نبدأ بالعمود الأول
12	مادام (رقم_العمود < عدد_الاعمدة){ // التحقق من ان رقم_العمود < 4
13	اكتب " وفني بالنقطة رقم" + (رقم_العمود+1)
14	اقرأ (جد_النقاط[رقم_السطر][رقم_العمود])
15	رقم_العمود++ ؛
16	}
17	رقم_السطر++ ؛ // ننتقل الى السطر التالي
18	}
19	} // نهاية الإجراء اساسي
} // نهاية الخوارزم_19_فصل_9_تعمير_جدول_النقاط_ذو_بعدين	

النص 59: خوارزم تعمير جدول النقاط باستعمال تعليمة التكرار على مستوى الأسطر و الخانات

4- 4 استعمال اجراء لتعمير أسطر الجدول

بدل من كتابة التعليمات التي تعمر خانات سطرا ما في جسم التكرار الأول الخاص بأسطر جدول النقاط (النص 59، الأسطر. من 11 الى 16)، يمكننا ان نتبع طريقة اخرى في تنظيم الخوارزم، فننشأ إجراءا متخصصا في تعمير جدول ذو بعد واحد، وبعد ذلك نستعمل هذا الإجراء في تعمير اسطر جدول النقاط، ونذكر. ان كل سطر من اسطر. جدول النقاط هو في حقيقته **جدول من النمط حقيقي ذو بعد واحد.**

الإجراء الذي سميناه **عمر_جدول_ن2** في النص 60، يشبه بشكل كبير الوظيفة **عمر_جدول** من النص 51، والفرق بينهما هو ان جدول الوظيفة **عمر_جدول** يُنشأ في الوظيفة، بينما جدول الإجراء **عمر_جدول_ن2** يُنشأ خارج الإجراء، و يوصف له في مدخله (السطر 1 من النص 60).

في رأس الإجراء **عمر_جدول_ن2**، نلاحظ وجود مدخل واحد، وهو المدخل الذي من خلاله يتحصل الإجراء على صلة الجدول، وبها يتمكن من تعمير الجدول، وهنا نتساءل : كيف يعرف الإجراء عدد النقاط التي يجب عليه ان يتحصل عليها من المستعمل، فالمتغيرة التي تحتوي على هذا العدد، ونعني المتغيرة **عدد_الاعمدة**، متغيرة محلية للإجراء اساسي، ولا يمكن للإجراء **عمر_جدول_ن2** ان يتحصل على محتواها، والجواب ان الإجراء **عمر_جدول_ن2** في حقيقته لا يحتاج لهذه المتغيرة لمعرفة عدد الأعمدة (اي عدد النقاط، او عدد الخانات في سطر واحد)، لأن عدد أعمدة السطر (هو ايضا عدد الخانات) موجود في السمة **حجم** التي يحتويها كل جدول، وهذا ما نراه في عبارة التحكم في التكرار (السطر 3)، وكنتيجة لهذا التنظيم الجديد للجدول، نلاحظ ان لإجراء اساسي اصبح في النص 60 اقصر و اوضح واسهل للفهم مما كان عليه في النص 59.

لرقم	نص خوارزم تعمير جدول المعدلات
خوارزم_20_فصل_9_تعمير_جدول_النقاط_ذو_بعدين {	
1	اجراء عمر_جدول_ن2 (حقيقي[ ] جد_نقاط_التلميذ){
2	طبيعي رقم_العمود = 0
3	مادام (رقم_العمود > جد_نقاط_التلميذ.حجم){
4	اكتب " وفني بالنقطة رقم" + (رقم_العمود+1)
5	اقرأ (جد_نقاط_التلميذ [رقم_العمود])
6	رقم_العمود++ ؛
7	}
8	اجراء اساسي(){
9	طبيعي رقم_السطر ؛
10	طبيعي عدد_الاسطر = 25، عدد_الاعمدة = 4 ؛
11	حقيقي[ ] [ ] جد_النقاط = شيد حقيقي[عدد_الاسطر][عدد_الاعمدة]؛
12	اكتب "أهلا و سهلا في خوارزم تعمير جدول النقاط"
13	
14	رقم_السطر = 0 ؛ // نبدأ بالسطر الأول و رقمه 0
15	مادام (رقم_السطر > عدد_الاسطر){
16	اكتب " فضلا، وفني بنقاط التلميذ رقم " + (رقم_السطر+1)
17	عمر_جدول_ن2(جد_النقاط[رقم_السطر])
18	رقم_السطر++ ؛ // ننتقل الى السطر التالي
19	}
20	} // نهاية الإجراء اساسي
} // نهاية الخوارزم_20_فصل_9_تعمير_جدول_النقاط_ذو_بعدين	

النص 60: خوارزم تعمير جدول النقاط: استعمال وظيفة تعمير سطر في جسم تعليمة التكرار المتعلقة بأسطر جدول النقاط

لرقم	نص خوارزم تعمير جدول المعدلات
خوارزم_21_فصل_9_تعمير_جدول_النقاط_ذو_بعدين {	
1	اجراء عمر_جدول_ن2 (حقيقي[ ] جد_نقاط_التلميذ){
2	طبيعي رقم_العمود = 0
3	مادام (رقم_العمود > جد_نقاط_التلميذ.حجم){
4	اكتب " وفني بالنقطة رقم" + (رقم_العمود+1)
5	اقرأ (جد_النقاط[رقم_السطر][رقم_العمود])
6	رقم_العمود++ ؛
7	}
8	حقيقي[ ] [ ] عمر_جدول_النقاط (طبيعي عدد_التلاميذ، عدد_النقاط){
9	حقيقي[ ] [ ] جد_النقاط = شيد حقيقي[عدد_التلاميذ][عدد_النقاط]؛
10	طبيعي رقم_السطر = 0 ؛ // نبدأ بالسطر الأول و رقمه 0
11	مادام (رقم_السطر > عدد_التلاميذ){
12	اكتب " فضلا، وفني بنقاط التلميذ رقم " + (رقم_السطر+1)
13	عمر_جدول_ن2(جد_النقاط[رقم_السطر])

```
14      |   رقم_السطر++ ؛ // ننتقل الى السطر التالي
15      | }
16      | ارجع جد_النقاط ؛
17  }
18  اجراء اساسي(){
19      | طبيعي عدد_التلاميذ، عدد_النقاط ؛
20      | حقيقي[ ] [ ] جد_النقاط ؛
21      | اكتب "أهلا و سهلا في خوارزم تعمير جدول النقاط"
22      | اكتب " فضلا، وفني بعدد تلاميذ القسم "
23      | اقرأ (عدد_التلاميذ)
24      | اكتب " فضلا، وفني بعدد النقاط في المادة "
25      | اقرأ (عدد_النقاط)
26      | جد_النقاط = عمر_جدول_النقاط (عدد_التلاميذ، عدد_النقاط) ؛
27  } // نهاية الإجراء اساسي
} // نهاية الخوارزم_21_فصل_9_تعمير_جدول_النقاط_ذو_بعدين
```

النص 61: خوارزم تعمير جدول النقاط مستقل عن عدد التلاميذ و عدد النقاط

4- 5 تعمير الجدول ذو بعدين باستعمال وظيفة

من خلال دراسة النص 60 و النص 59، يتبين لنا ان هذين النصين لا يمكنهما التعامل مع عدد من التلاميذ غير العدد 25، ومع عدد النقاط غير 4 نقاط، و السبب الوحيد لهذه المحدودية يرجع الى تحديد عدد التلاميذ وعدد النقاط في النص ذاته (السطر 10 من النص 60 والسطر. 3 من النص 59)، ولجعل الخوارزم مستقل تماما عن عدد التلاميذ وعدد النقاط، يكون من الأفضل ان لا تحدد هذه القيم في النص، بل تأخذ من خارج النص، اما عبر التفاعل مع المستعمل او عبر مداخل الإجراءات او الوظائف، وهذا ما يهدف اليه النص 61، فنراه مكونا من ثلاث قطع: الإجراء **اساسي**، الوظيفة **عمر_جدول_النقاط** والإجراء **عمر_جدول_ن2**.

نرى في السطر 19 من النص 61 ان قيم المتغيرات **عدد_التلاميذ** و **عدد_النقاط** لم تحدد في النص، بل يتم تحديد قيم هذه المتغيرات في سياق تنفيذ الخوارزم باستعمال تعليمات التفاعل مع المستعمل (الأسطر. من 22 الى 25)، وبعد الحصول عليها يطلب الإجراء **اساسي** من الوظيفة **عمر_جدول_النقاط** تعمير الجدول **جد_النقاط**، ونلاحظ ان الجدول **جد_النقاط** لم ينشأ في السطر. 20، بل صرح فقط **بصلة لجدول من النمط حقيقي ذو بعدين**,

يُنشأ جدول النقاط في الوظيفة **عمر_جدول_النقاط** باستعمال القيم التي تعطى للوظيفة عبر مداخلها، وهذه القيم تدل على حجم كل بعد في جدول النقاط، وبعد الانتهاء من تعميره، ترجع الوظيفة **عمر_جدول_النقاط** صلة الجدول التي يلتقطها الإجراء اساسي في اطار تنفيذ تعليمة السطر **26**، ونلاحظ ان الوظيفة **عمر_جدول_النقاط** تتكفل بالانتقال من سطر الى الذي يليه، اما من يعمر فعليا الخانات فهو الإجراء **عمر_جدول_ن2** الذي عرف اولا في النص 60 واعيد استعماله في النص 61.

4 – 6 تعمير جدول يحتوي على اسطر بأحجام مختلفة

نفترض ان عدد النقاط التي يتحصل عليها طالب ما يمكن ان تختلف عن عدد النقاط التي يتحصل عليها طالب أخر، فيمكن ان يكون للطالب رقم 1 خمس نقاط، وللطالب 2 ست نقاط وللطالب 3 اربع نقاط وهكذا، فمع مثل هذه الجداول، يكون استعمال الأسماء الثابتة للخانات أمرا معقدا جدا، اذ يجب ان نراعي في كتابة التعليمات حجم كل سطر.

للتجوال في مثل هذه الجداول باستعمال تعليمات التكرار، يجب على كل تكرار متعلق بسطر ما ان يعرف عدد خانات السطر، فالمتغيرة **عدد_الأعمدة** التي استعملت في الأمثلة السابقة لم تصبح ذات جدوى لكون كل سطر له حجمه الخاص به.

للحصول على عدد خانات السطر المستهدف في تكرار ما، نستغل السمة **حجم** المتوفرة في كل جدول، فمثلا اذا كان **جد_النقاط** جدول ذو بعدين فالكتابة **جد_النقاط.حجم** تمثل عدد اسطر الجدول، والكتابة **جد_النقاط[0].حجم** تمثل حجم السطر الأول والكتابة **جد_النقاط[1].حجم** تمثل حجم السطر الثاني الخ...

استعمال السمة **حجم** قد حدث في الإجراء **عمر_جدول_ن2** من النص 61، ففي السطر 3 نرى ان عبارة التحكم في التكرار مبنية على السمة **حجم** للجدول الذي يوصف في المدخل **جد_نقاط_التلميذ،** ونلاحظ ان ارتكاز الإجراء **عمر_جدول_ن2** على السمة **حجم** يجعله يتعاطى مع اي جدول حقيقي ذو بعد واحد، مهما كان حجمه، وهذه القدرة تجعل الإجراء **عمر_جدول_ن2** مرشحا للاستعمال في اطار تعمير جدول ذو بعدين يحتوي على اسطر بأحجام مختلفة.

يتكفل النص 62 بتعمير جدول نقاط التلاميذ في مادة ما، وعدد النقاط من تلميذ لآخر يمكن ان تكون مختلفة، وفي هذا النص يتكفل الإجراء اساسي بمعرفة عدد التلاميذ من خلال التفاعل مع المستعمل (السطرين 26 و 27)، ثم يطلب من الوظيفة **عمر_جدول_النقاط_ن2 (السطر 28)** ان تنشأ وتعمر جدول النقاط، ويلتقط الإجراء اساسي في المتغيرة **جد_النقاط** صلة الجدول الذي تنشئه الوظيفة عمر_**جدول_النقاط_ن2 (السطر 28).**

ينطلق عمل الوظيفة **عمر_جدول_النقاط_ن2** فقط من معرفة عدد التلاميذ الذي تتحصل عليه من المدخل **عدد_التلاميذ** (السطر 8)، و بعد انشاء الجزء الأول من الجدول ذو بعدين الذي سوف توضع فيه نقاط التلاميذ، وهو ما نسميه بجدول الأسطر (السطر 9)، تقوم الوظيفة بزيارة كل خانات جدول الأسطر ابتداءا من الخانة رقم 0 حتى الخانة رقم **عدد_التلاميذ -1**، اي آخر خانة في جدول الأسطر.

في اطار تعليمة التكرار (من السطر 13 الى السطر 19)، كلما زارت الوظيفة خانة من جدول الأسطر تقوم بإنشاء الجدول الممثل للسطر وتربطه بالخانة التي تمكن من الوصول الى السطر (السطر 16) ، و يحمل السطر نقاط التلميذ الذي يمثله هذا السطر، وقبل انشاء السطر الممثل لنقاط

التلميذ، تتفاعل الوظيفة مع المستعمل فتطلب منه عدد نقاط التلميذ، وتلتقط الوظيفة في المتغيرة **عدد_النقاط** العدد الذي يدخله المستعمل(السطرين 14 و 15)، ونلاحظ ان تعليمة انشاء السطر ترتكز على ما تحتويه المتغيرة **عدد_النقاط** لتحديد حجم الجدول الذي تطلب انشاءه من العملية **شيد (السطر 16)**.

بعد الانتهاء من انشاء سطر بحجم عدد نقاط التلميذ، تطلب الوظيفة **عمر_جدول_النقاط_ن2** المساعدة من الإجراء **عمر_جدول_ن2** (السطر. 17) لتعمير السطر. بعلامات التلميذ، وبما ان الإجراء **عمر_جدول_ن2** قد استَعمَل السِمة **حجم** من الجدول الذي يوصف له في مدخله(السطر 3)، فانه سوف ينجز عمله مهما كان حجم السطر الذي يوصف في مدخله.

لرقم	نص خوارزم تعمير جدول المعدلات
خوارزم_22_فصل_9_تعمير_جدول_النقاط_ذو_بعدين {	
1	اجراء عمر_جدول_ن2 (حقيقي[ ] جد_نقاط_التلميذ){
2	طبيعي رقم_العمود = 0
3	مادام (رقم_العمود < جد_نقاط_التلميذ.حجم){
4	اكتب " وفني بالنقطة رقم" + (رقم_العمود+1)
5	اقرأ (جد_النقاط[رقم_السطر][رقم_العمود])
6	رقم_العمود++ ؛
7	}
8	حقيقي[ ] [ ] عمر_جدول_النقاط_ن2 (طبيعي عدد_التلاميذ){
9	حقيقي[ ] [ ] جد_النقاط = شيد حقيقي[عدد_التلاميذ][] ؛ // انشاء جدول الأسطر
10	طبيعي عدد_النقاط ، // قيمة هذه المتغيرة يحددها المستعمل لكل تلميذ، اي لكل سطر
11	// فيما يلي الإنشاء التدريجي لكل الأسطر، كل سطر ينشأ بحجمه الخاص
12	طبيعي رقم_التلميذ = 0 ؛ // نبدأ بإنشاء السطر الأول و رقمه 0 الذي يمثل التلميذ الأول
13	مادام (رقم_التلميذ < عدد_التلاميذ){
14	اكتب " فضلا، وفني بعدد نقاط التلميذ رقم " + (رقم_التلميذ +1)
15	اقرأ عدد_النقاط ؛ // عدد النقاط يمثل حجم السطر الذي سينشأ في التعليمة التالية
16	جد_النقاط[رقم_التلميذ] = شيد حقيقي [عدد_النقاط]؛
17	عمر_جدول_ن2(جد_النقاط[رقم_التلميذ])
18	رقم_التلميذ++ ؛ // ننتقل الى التلميذ التالي
19	}
20	ارجع جد_النقاط ؛
21	}
22	اجراء اساسي(){
23	طبيعي عدد_التلاميذ ؛
24	حقيقي[ ] [ ] جد_النقاط ؛
25	اكتب "أهلا و سهلا في خوارزم تعمير جدول النقاط"
26	اكتب " فضلا، وفني بعدد تلاميذ القسم "
27	اقرأ (عدد_التلاميذ)
28	جد_النقاط = عمر_جدول_النقاط_ن2 (عدد_التلاميذ) ؛
29	} // نهاية الإجراء اساسي
} // نهاية الخوارزم_22_فصل_9_تعمير_جدول_النقاط_ذو_بعدين	

النص 62: خوارزم تعمير جدول النقاط مستقل عن عدد التلاميذ و عدد النقاط

الفصل العاشر

الأوعية: التجوال في الجداول (2)

1 – خوارزميات عرض محتوى الجدول على الشاشة

1 – 1 تعليمات كتابة البيانات على الشاشة:

لإرسال اي نص الى الشاشة نستعمل تعليمتين اساسيتين: التعليمة **اكتب** والتعليمة **اكتب_سطر**.

تقوم التعليمة **اكتب** بإرسال النص التي حدد لها، ولا تزيد اي حرف، اما التعليمة **اكتب_سطر**. فإنها تكتب النص الذي حدد لها وتزيد حرف يظهر فعله وهو حرف الرجوع الى السطر والذي تمثله القيمة الثابتة الرمزية **نهاية_سطر**.

تكتب النصوص التي ترسل الى الشاشة في موقع تحدده اشارة، تكون عادة متحركة (تظهر وتختف) في مكانها، فالحرف الأول يكتب في مكان الإشارة وبعد كتابة النص تنتقل الإشارة مباشرة بعد آخر حرف كتب من نص ما.

لفهم الفرق بين التعليمة **اكتب** والتعليمة **اكتب_سطر**، نعتمد على النص 63 وخطأ: لم يُعثر على مصدر المرجع(من **أ** الى ع) الذي يبين حالة الشاشة اثر تنفيذ تعليمات النص 63.

لرقم	نص خوارزم تعمير جدول المعدلات
خوارزم_1_الفصل_10_تعليمات_الإخراج_على_الشاشة {	
1	اجراء اساسي(){
2	طبيعي رقم1، رقم2، رقم3 ؛
3	اكتب " فضلا، ادخل رقما ما بين 1 و 100:"
4	اقرأ رقم1
5	اكتب_سطر " فضلا، ادخل رقما ما بين 101 و 200:"
6	اقرأ رقم1
7	اكتب " فضلا، ادخل رقما ما بين 201 و 300: " + نهاية_سطر
8	اقرأ رقم3
9	اكتب رقم1 + "،" + رقم2 + "،" + رقم3 +"."
10	اكتب_سطر "--" + رقم1 + "--" + رقم2 + "--" + رقم3 + "."
11	اكتب "سطر 1: "+ رقم1 + "،" + رقم2 + "-----"+ نهاية_سطر+ "سطر2:" + 500 + "،" + رقم3 + "-----"
12	} // نهاية الإجراء اساسي
} //نهاية الخوارزم_1_الفصل_10_تعليمات_الإخراج_على_الشاشة	

النص 63: مختلف الاستعمالات لتعليمة اكتب و اكتب_سطر

في بداية الأمر تكون الشاشة بدون اي نص، وتكون الإشارة موجودة في اعلى يمين الشاشة كما يظهر ذلك في خطأ: لم يُعثر على مصدر المرجع–(أ)، وبعد تنفيذ تعليمة السطر 3 من النص 63، تصبح الشاشة على صورة خطأ: لم يُعثر على مصدر المرجع–(ب)، ونلاحظ في هذا الشكل ان الإشارة بقيت في آخر السطر مباشرة بعد آخر حرف من النص.

عند تنفيذ تعليمة السطر 4، تكتب الإجابة في نفس السطر(خطأ: لم يُعثر على مصدر المرجع–(ج))، وبعد ان يضغط المستعمل على المفتاح **ادخل** (او الحرف **ادخل**)، يرجع المؤشر الى بداية السطر التالي(خطأ: لم يُعثر على مصدر المرجع–(ج))، و نذكر ان المفتاح **ادخل** هو حرف يُرى

مفعوله على الشاشة، ومفعوله هو الرجوع الى بداية السطر. التالي في الشاشة ورمزه في كتابة الخوارزميات هو الرمز **نهاية_سطر.**



الشكل 1: حالة الشاشة اثر تنفيذ تعليمات النص 1

في السطر الخامس تنفذ التعليمة **اكتب_سطر.**، ونرى من خلال (خطأ: لم يُعثر على مصدر المرجع–(د))، ما تحدثه هده التعليمة على الشاشة، فهي تكتب النص الذي ذكر لها، وتجعل المؤشر يرجع الى بداية السطر التالي، وعندما تنفذ تعليمة السطر 6، نرى من خلال خطأ: لم يُعثر على مصدر. المرجع–(س)، ان اجابة المستعمل تُكتب في السطر الذي يلي السطر. الذي كتبته التعليمة **اكتب_سطر.** على الشاشة.

تعليمة **اكتب** في السطر. 7 تحدث ما تحدثه التعليمة **اكتب_سطر. (**خطأ: لم يُعثر على مصدر المرجع–(ص))، والسبب هو وجود الحرف **نهاية_سطر.** كآخر حرف ترسله التعليمة الى الشاشة، فاذا كانت التعليمة اكتب لا تُحدِث الانتقال الآلي الى السطر التالي، فإن الحرف **نهاية_سطر.** هو الذي يتسبب في الانتقال الى السطر التالي، ولو ذكر الحرف **نهاية_سطر** اكثر من مرة في النص، لشهدنا التنقل الى السطر. التالي مع كل حرف **نهاية_سطر،** ونتيجة لما تحدثه تعليمة السطر. 7، فان كتابة اجابة المستعمل ردا على تعليمة السطر 8، تكون في سطر غير السطر الذي كتبته تعليمة السطر 7 **(**خطأ: لم يُعثر على مصدر المرجع–(ص)).

يبين خطأ: لم يُعثر على مصدر المرجع–(ق) ما تحدثه تعليمة السطر 9، ونلاحظ ان المؤشر بقي في آخر السطر بعد اخر حرف من النص الذي كتب، ثم تنفذ تعليمة السطر 10، ونلاحظ من خلال خطأ: لم يُعثر على مصدر المرجع–(ك) ان نص التعليمة **اكتب_سطر.** قد كتب مباشرة بعد النص السابق ثم انتقل المؤشر. الى اول السطر التالي.

يبين خطأ: لم يُعثر على مصدر المرجع–(ل) ما تحدثه تعليمة السطر 11، ونلاحظ في هذا السطر استعمال التعليمة اكتب وكذلك ظهور حرف الرجوع الى السطر (القيمة **نهاية_سطر)** في وسط النص الذي ترسله التعليمة اكتب الى الشاشة، ومعنى هذا ان النص يكتب على سطرين، ويبق المؤشر. في آخر السطر. الثاني.

1 – 2 خوارزم عرض محتوى جدول ذو بعد واحد على الشاشة

ليتمكن الخوارزم من عرض محتوى جدول ذو بعد واحد، عليه زيارة كل خانات الجدول، الواحدة تلو الأخرى، وكلما وصلت الزيارة الى خانة ما، يكتب الخوارزم على الشاشة قيمة الخانة اذا كانت الخانة من نمط بدائي او سمة من سمات الخانة ان كانت الخانة من نمط غير بدائي.

لكي يتمكن الخوارزم من زيارة كل الخانات دون نسيان اي خانة، على الخوارزم اتباع طريقة فعالة تمكنه من الزيارة الفعلية لكل الخانات، وفيما يخص الجداول ذو بعد واحد، هناك طريقتين:

- في الطريقة الأولى تبدأ الزيارة من الخانة رقم 0، اي الخانة الأولى، ثم تنتقل الى الخانة رقم 1، ثم الخانة رقم 2 الى ان نصل الى آخر خانة، ورقمها هو **حجم الجدول – 1** وفي هذه الطريقة تتكرر التعليمتين التاليتين:

- تعليمة عرض محتوى الخانة.
- تعليمة الانتقال الى الخانة التالية.

- في الطريقة الثانية تبدأ الزيارة من الخانة الأخيرة ورقمها هو **حجم الجدول – 1** ، ثم تنتقل الى الخانة السابقة، اي الخانة التي رقمها هو حجم الجدول – 2 ، ثم الخانة رقم حجم الجدول – 3 الى ان نصل الى اول خانة، ورقمها هو 0 وفي هذه الطريقة تتكرر التعليمتين التاليتين:
  - تعليمة عرض محتوى الخانة
  - تعليمة الانتقال الى الخانة السابقة

لعرض جدول ذو بعد واحد، نجد طريقتين:

- العرض الأفقي: وفيه تعرض كل الخانات على سطر واحد من اسطر الشاشة
- العرض العمودي: وفيه تعرض كل خانة وحيدة في سطر من اسطر الشاشة وفي مكان محدد من السطر، مثلا في اول السطر او في منتصفه.

فيما يلي، وحتي نتمكن من كتابة نصوص يمكن اعادة استعمالها فيما بعد، نجعل خوارزم عرض الجدول في اجراء مرن غير مرتبط بحجم الجدول الذي يعرض.

1 – 2 – 1 العرض الأفقي للجدول

يمثل النص 64 وبالأخص الإجراء **عرض_افقي** ، خوارزما لعرض جدول طبيعي افقيا والطريقة المستعملة في الزيارة هي الطريقة الأولى، اي من الخانة رقم 0 الى الخانة الأخيرة، ولكي تتمكن الوظيفة من الانتقال من خانة الى اخرى، تلجأ الى استعمال متغيرة لاحتواء رقم الخانة، وكلما ارادت الوظيفة ان تنتقل من خانة الى اخرى، تقوم بتغيير قيمة هذه المتغيرة.

	نص خوارزم تعمير جدول المعدلات
	خوارزم_2_الفصل_10_عرض_جدول_على_الشاشة {
1	اجراء عرض_افقي (طبيعي[] جد_ط1){
2	طبيعي رقم = 0 ؛
3	سلسلة سطر_التزيين = "----------"
4	حرف حد_الخانة = '|' ؛
5	اكتب_سطر (سطر_التزيين)
6	مادام(رقم < جد_ط1.حجم){
7	اكتب (حد_الخانة)
8	اكتب (جد_ط1[رقم]
9	رقم++ ؛
10	}
11	اكتب_سطر (حد_الخانة) // الحد الأخير
12	اكتب_سطر (سطر_التزيين)
13	}
14	اجراء اساسي(){
15	طبيعي[ ] جدول1 = {12، 24، 48، 96، 10، 20، 40، 80} ؛
	اكتب_سطر ("اهلا و سهلا، اختبار اجراء عرض جدول افقيا"
16	عرض_افقي (جدول1) ؛
	اكتب_سطر ("نأمل ان يكون العرض الأفقي للجدول قد نال اعجابك....."
17	} // نهاية الإجراء اساسي
	} //نهاية الخوارزم_2_الفصل_10_عرض_جدول_على_الشاشة

النص 64: اجراء عرض افقي و مزين لجدول طبيعي ذو بعد واحد

```
اهلا و سهلا، اختبار اجراء عرض جدول افقيا
------------------------------------------------------------------------
| 12 | 24 | 48 | 96 | 10 | 20 | 49 | 49 |
------------------------------------------------------------------------
نأمل ان يكون العرض الأفقي للجدول قد نال اعجابك.....
_
```

الشكل 2: عرض افقيو مزين للجدول (النص 24)

في نص الإجراء **عرض_أفقي** تستعمل مباشرة تعليمة التكرار، وترى ان عبارة التحكم في التكرار تتكفل بمراقبة قيمة رقم الخانة، فعندما يتجاوز رقم الخانة الرقم الاقصى، وهو **حجم الجدول – 1**، تنته تعليمة التكرار وينتقل التنفيذ الى التعليمة التي تلي تعليمة التكرار.

كما يظهر في خطأ: لم يُعثر على مصدر المرجع، يقوم الإجراء **عرض_افقي من** النص 64، بتزيين العرض، فيُخرِج اولا عدد من الحروف لتمثيل سطر، ويمثل هذا السطر الحد العلوي للخانات ثم في سطر. ثان تكتب قيم الخانات، و بين قيمة وقيمة اخرى تضع حرفا يمثل حدا بين الخانات، ثم في سطر ثالث ترسم الوظيفة سطرا ثان يمثل الحد الأدنى للخانات، ونلاحظ ان قيمة السطر الممثل للحدين العلوي والسفلي وقيمة حرف الحد قد وضعتا في متغيرتين (السطرين 3 و 4)، ومثل هذه الكتابة سوف تعطي نوعا من المرونة في حالة ما اذا اردنا ان نغير الحروف الممثلة لحدود الخانات، فالتغيير يكون فقط على مستوى هذين السطرين ودون اي تأثير على باق النص، وسوف نرى فيما بعد طريقة فيها مرونة اكبر.

لرقم	نص خوارزم تعمير جدول المعدلات
خوارزم_3_الفصل_10_عرض_جدول_على_الشاشة {	
1	اجراء عرض_عمودي (طبيعي[] جد_ط1){
2	طبيعي رقم = 0 ؛
3	سلسلة سطر_التزيين = "-----"
4	حرف حد_الخانة = '|' ؛
5	اكتب_سطر (سطر_التزيين)
6	مادام(رقم < جد_ط1.حجم){
7	اكتب_سطر (حد_الخانة + جد_ط1[رقم] + حد_الخانة)
8	اكتب_سطر (سطر_التزيين)
9	رقم++ ؛
10	}
11	}
12	اجراء اساسي(){
13	طبيعي[ ] جدول1 = { 48، 96، 40، 80} ؛
14	اكتب_سطر ("اهلا و سهلا، اختبار اجراء عرض جدول عموديا"
15	عرض_عمودي(جدول1) ؛
16	اكتب_سطر ("نأمل ان يكون العرض العمودي للجدول قد نال اعجابك....."
17	} // نهاية الإجراء اساسي

```
} //نهاية الخوارزم_3_الفصل_10_عرض_جدول_على_الشاشة
```

النص 65: اجراء عرض عمودي و مزين لجدول طبيعي ذو بعد واحد

```
اهلا و سهلا، اختبار اجراء عرض جدول افقيا
----------
|  48  |
----------
|  96  |
----------
|  40  |
----------
|  80  |
----------
.....نأمل ان يكون العرض الأفقي للجدول قد نال اعجابك

_
```

الشكل 3: عرض عمودي و مزين للجدول (النص 3)

1 – 2 – 2 العرض العمودي للجدول:

في العرض العمودي (خطأ: لم يُعثر على مصدر المرجع) الذي يتكفل به الإجراء **عرض_عمودي** (النص 65) نرى ان سطر التزيين قد تراجع حجمه، واصبح هذا الأخير يكف لتمثيل الحد العلوي والسفلي لكل خانة، ونلاحظ في السطر. 7 من النص 65 تعليمة **اكتب_سطر** التي تكتب قيمة الخانة بين حديها الأيمن و الإيسر.

1 – 2 – 3 تحسين الإجرائين المكلفين بعرض جدول طبيعي ذو بعد واحد

في نص الإجرائين (النص 64 والنص 65) نرى ان الأحرف التي استعملت لتمثيل حدود الخانة قد ثبتت في النص، ونريد ان يتمكن مستعمل الإجراء من اختيار الأحرف التي تمثل حدود الخانة، والطريقة التي اصبحنا نعرفها. هي تزويد الإجراء بمداخل تمكنه من اخذ هذه المعلومات من خارجة، فمثلا، بدل التصريح في جسم الإجراء بالمتغيرتين **سطر_التزيين** و **حد_الخانة**، نحول هاتين المتغيرتين الى مداخل كما نراه في النص 66 والنص 67، ونرى ايضا من خلال النصين ان الإجراء **أساسي** هو الذي يحدد قيمة المدخلين، فيمكن له ان يصف اي أحرف اخرى لتمثيل حدود الخانات، كما يظهر من خلال السطرين 15 و 17 من النص 66 ، والسطرين 13 و 15 من النص 67.

لرقم	نص خوارزم تعمير جدول المعدلات
خوارزم_4_الفصل_10_عرض_جدول_على_الشاشة	
1	اجراء عرض_افقي (طبيعي[] جد_ط1؛ سلسلة سطر_التزيين؛ حرف حد_الخانة){
2	طبيعي رقم = 0 ؛
3	اكتب_سطر (سطر_التزيين)
4	مادام(رقم > جد_ط1.حجم){
5	اكتب (حد_الخانة)
6	اكتب (جد_ط1[رقم]
7	رقم++ ؛
8	}

```
9     اكتب_سطر (حد_الخانة)  // الحد الأخير
10    اكتب_سطر (سطر_التزيين)
11  }
12  اجراء اساسي(){
13    طبيعي[ ] جدول1 = {12، 24، 48، 96، 10، 20، 40، 80} ؛
14    اكتب_سطر ("اهلا و سهلا، اختبار اول  للنسخة الثانية من اجراء عرض جدول افقيا"
15    عرض_افقي (جدول1، "----------"، '|') ؛
16    اكتب_سطر ("اهلا و سهلا، اختبار ثان للنسخة الثانية من اجراء عرض جدول افقيا"
17    عرض_افقي (جدول1، "**********"، '*') ؛
18    اكتب_سطر ("تأمل ان يكون العرض الأفقي للجدول قد نال اعجابك....."
19  } // نهاية الإجراء اساسي
} //نهاية الخوارزم_4_الفصل_10_عرض_جدول_على_الشاشة
```

النص 66: النسخة الثانية للإجراء عرض افقي لجدول طبيعي ذو بعد واحد

1 – 2 – 3 محدودية تصميم الإجراء عرض_افقي

هذا الإجراء في نسخته الحالية، عندما يُطلب تنفيذه، يجب ان يوضع في المدخل **سطر_التزيين** عددا معينا من الاحرف ويجب على هذا العدد ان يتوافق مع حجم الجدول الذي يذكر في المدخل **جد_ط1**، فمثلا اذا كان حجم الجدول هو 5 خانات، واذا كانت كل خانة تستهلك 6 احرف، فلا بد من كتابة 30 حرفا لتمثيل السطر العلوي والسفلي لعرض الجدول، واذا كان الجدول مكونا من 10 خانات، وجب كتابة 60 حرف، وهكذا، فعلى كاتب الخوارزم ان يكتب العدد الكافي من الاحرف كلما كتب تعليمة تنفيذ الإجراء **عرض_افقي_جدول_ط1.**

لرقم	نص خوارزم تعمير جدول المعدلات
	خوارزم_5_الفصل_10_عرض_جدول_على_الشاشة {
1	اجراء عرض_عمودي (طبيعي[] جد_ط1، سلسلة سطر_التزيين؛ حرف حد_الخانة){
2	طبيعي رقم = 0 ؛
3	اكتب_سطر (سطر_التزيين)
4	مادام(رقم > جد_ط1.حجم){
5	اكتب_سطر (حد_الخانة) + جد_ط1[رقم] + (حد_الخانة)
6	اكتب_سطر (سطر_التزيين)
7	رقم++ ؛
8	}
9	}
10	اجراء اساسي(){
11	طبيعي[ ] جدول1 = { 48، 96، 40، 80} ؛
12	اكتب_سطر ("اهلا و سهلا، اختبار اول للنسخة الثانية من اجراء عرض جدول عموديا"
13	عرض_عمودي (جدول1، "------" ، '|') ؛
14	اكتب_سطر ("اهلا و سهلا، اختبار ثان للنسخة الثانية من اجراء عرض جدول عموديا"
15	عرض_عمودي (جدول1، "=====" ، '*') ؛
16	اكتب_سطر ("تأمل ان يكون العرض العمودي للجدول قد نال اعجابك....."
17	} // نهاية الإجراء اساسي
	} //نهاية الخوارزم_5_الفصل_10_عرض_جدول_على_الشاشة

النص 67: النسخة الثانية للإجراء عرض عمودي و مزين لجدول طبيعي ذو بعد واحد

في حقيقة الأمر، كتابة العدد الكافي من الأحرف في المدخل **سطر_التزيين** فيه نوع من الصعوبة، لكنه ليس بالمشكل العويص، فحله سهلا جدا، اذ على كاتب الخوارزم ان يصرح بمتغيرة فيها سطر التزيين، ثم يصف هذه المتغيرة في المدخل **سطر_التزيين،** لكن هناك حالات تكون فيها كتابة العدد الملائم من الأحرف امرا في غاية الصعوبة، وهذه الصعوبة نراها من خلال الخوارزم

اجمع_خانتين_متتاليتين (النص 68) الذي يقوم بجمع خانتين متتاليتين لجدول طبيعي، ثم يضع النتيجة في جدول آخر، وحجم هذا الجدول الأخير يكون في العموم نصف حجم الجدول الأول خطأ: لم يُعثر على مصدر المرجع)، والسلوك العام لهذا الخوارزم كالتالي:

يقوم الخوارزم **اجمع_خانتين_متتاليتين** (النص 68) بإنشاء جدول، ثم يعمره بالتفاعل مع المستعمل، وبعدها ينشئ جدولا آخر يكون حجمه هو نصف حجم الجدول الأول، ثم يقوم بالتجوال في الجدولين ابتداء من الخانة الأولى، ففي كل فترة من فترات تنقله في الجدولين، يتعامل النص مع خانتين متتاليتين من الجدول الأول ومع خانة من الجدول الثاني، فيأخذ قيمتي خانتين متتاليتين من الجدول الأول فيجمعها ويضع النتيجة في خانة الجدول ثان، ثم ينتقل الى الخانتين التاليتين من الجدول الأول والخانة التالية من الجدول الثاني، ويبين خطأ: لم يُعثر على مصدر المرجع مثالا لجدولين بعد انتهاء عمل الخوارزم **اجمع_خانتين_متتاليتين** ونفترض ان عدد خانات الجدول الأول زوجي.



الشكل 4: كل خانة من الجدول الثاني تحتوي على قيمة جمع خانتين متتاليتين من الجدول الأول

يحتوي النص 68 على اربع خوارزميات: الإجراء **اساسي** (السطر 40)، الوظيفة **عمر_جدول** (السطر 1) ، الوظيفة **اجمع_خانتان_متتاليتان** (السطر 16) والإجراء **عرض_افقي** (السطر 28).

يقوم الإجراء **اساسي** بإنشاء اسمين لجدولين (السطر 41): **جدول1** و **جدول2**، وهذين الاسمين في حقيقتهما صلتين. الجدول **جدول1** موجه لاحتواء المعطيات التي سوف يدخلها المستعمل، الجدول **جدول2** موجه لاحتواء نتائج جمع محتوى خانتان متتاليتان من الجدول **جدول1**، ثم يطلب الإجراء اساسي تنفيذ الوظيفة **عمر_جدول** (السطر 42)، وهذه الأخيرة تتفاعل مع المستعمل لمعرفة عدد المعلومات التي ينوي المستعمل ادخالها، ثم تقوم بإنشاء جدول يعمر بالتفاعل مع المستعمل ، وبعد تعميره تنته الوظيفة **عمر_جدول** وترجع صلة الجدول الذي انشأته والتي يلتقطها الإجراء اساسي في الصلة **جدول1**.

بعد حصوله على الجدول الأول، يطلب الإجراء اساسي تنفيذ الوظيفة **اجمع_خانتان_متتاليتان** التي ينطلق عملها من الجدول **جدول1** الذي يوضع في مدخلها (السطر 43)، ثم تنتج الوظيفة جدولا يربط بالصلة **جدول2**، واول ما تقوم به الوظيفة **اجمع_خانتان_متتاليتان** هو انشاء الجدول الثاني (اي **جدول2**) انطلاقا من حجم الجدول الأول(اي **جدول1**)، وبعدها تقوم هذه الوظيفة بالتجوال في الجدولين، في الجدول الأول تكون اولا مع الخانة 0 ، وفي نفس الوقت تكون مع الخانة 0 من الجول الثاني ، فتجمع محتوى الخانة 0 و التي تليها، اي الخانة 1 من الجدول الأول وتضع النتيجة في

الخانة 0 من الجدول الثاني، ثم تنتقل الوظيفة الى الخانة 2 من الجدول الأول والى الخانة 1 من الجدول الثاني ، وتعاود نفس ما جرى سابقا لكن هذه المرة تجمع الخانتين 2 والتي تليها من الجدول الأول، اي 3، وتضع النتيجة في الخانة 1 من الجدول الثاني، ونلاحظ هنا ان الوظيفة عند انتقالها من مرحلة الى اخرى في اطار زيارتها للجدولين، تقفز قفزتين مع الجدول الأول وتقفز قفزة واحدة مع الجدول الأول, وبعد الانتهاء من عملها تُرجع الوظيفة **اجمع_خانتان_متتاليتان** صلة الجدول الثاني الى الإجراء **اساسي** الذي يلتقطها في الصلة **جدول2** التي تمثل اسم الجدول الثاني.

بعد تحصله على إنتاج الوظيفة **اجمع_خانتان_متتاليتان** ، يطلب الإجراء **اساسي** من الإجراء **عرض_افقي** اخراج محتوى الجدولين على الشاشة، وهنا يواجه كاتب الخوارزم معضلة يجب عليه حلها، فعند كتابة النص لم يكن يعلم الحجم الحقيقي للجدول الأول (والثاني طبعا)، فلا يمكن له ان يتخيل الحجم الحقيقي للجدول ليتسنى له الكتابة الصحيحة لسطر التزيين، كما هو الشأن بالنسبة للتعلمتين في السطرين 15 و 17 من النص 66 ، ففي السطرين 44 و45 (الإجراء **اساسي** النص 68) وصف مؤلف النص سطرا مزينا حجمه اربعين حرفا، وهذا السطر ملائم لعرض جدول حجمه اربع خانات، وهذا السطر سوف لن يتناسق مع اي جدول ذو حجم مختلف عن اربع، فاذا كان الحجم اقل من اربعة يكون سطر التزيين اطول، فتذهب مهمته في التزيين، واذا كان الحجم اكبر من اربعة يكون سطر التزيين قصير، فتذهب ايضا وظيفته في التزيين، فهنا تظهر اذا محدودية التصميم الحالي للإجراء **عرض_افقي**.

لرقم	الوظيفة عمر_جدول
خوارزم 6_الفصل_10_اجمع_خانتان_متتاليتان {	
1	طبيعي[ ] عمر_جدول(){
2	طبيعي[ ] جد_ط
3	طبيعي حجم ، رقم ؛
4	اكتب " فضلا، وفني بعدد القيم التي تريد ادخالها "
5	اقرأ (حجم)
6	جد_ط = شيد طبيعي[حجم]؛
7	رقم =0؛
8	مادام (رقم < حجم){
9	اكتب "أدخل قيمة : " ؛
10	اقرأ (جد_ط[رقم]) ؛
11	رقم ++ ؛
12	}
13	ارجع جد_ط ؛
14	}
15	
16	طبيعي[ ] اجمع_خانتان_متتاليتان (طبيعي[ ] جدول1){
17	طبيعي حجم_جد2 = جدول1.حجم / 2
18	طبيعي[ ] جدول2 = شيد طبيعي[حجم_جد2]؛
19	طبيعي رقم1=0 ، رقم 2 =0 ؛
20	مادام (رقم2 < حجم_جد2){
21	جدول2[رقم2] = جدول1[رقم1] + جدول1[رقم1 + 1]
22	رقم2 ++ ؛
23	رقم1 = رقم1 + 2 ؛
24	}
25	ارجع جدول2؛
26	}

```
اجراء عرض_افقي (طبيعي[ ] جد_ط1؛ سلسلة سطر_التزيين؛ حرف حد_الخانة){
    طبيعي رقم = 0 ؛
    اكتب_سطر (سطر_التزيين)
    مادام(رقم > جد_ط1.حجم){
        اكتب (حد_الخانة)
        اكتب (جد_ط1[رقم]
        رقم++ ؛
    }
    اكتب_سطر (حد_الخانة)   // الحد الأخير
    اكتب_سطر (سطر_التزيين)
}

اجراء اساسي(){
    طبيعي[ ] جدول1 ، جدول2 ؛
    جدول1 = عمر_جدول()
    جدول2 = اجمع_خانتان_متتاليتان (جدول1)
    عرض_افقي (جدول1، "---------------------------------------"، '|') ؛
    عرض_افقي (جدول2، "--------------------"، '|') ؛
}  // نهاية الإجراء اساسي
} // نهاية الخوارزم_6_الفصل_10_اجمع_خانتان_متتاليتان
```

النص 68: خوارزم ينتج جدولا بجمع الخانات المتتالية، خانتين خانتين، من جدول آخر

1 – 2 – 3 تحسين إضافي للإجراء عرض_افقي :

لتدارك النقص في التصميم، يجب على الإجراء **عرض_افقي** ان يتكيف مع اي جدول، وبصورة ادق يجب على حجم سطر. التزيين ان يتكيف مع حجم الجدول، فعلى الإجراء **عرض_افقي** ان يستنتج بنفسه الحجم الصحيح لسطر التزيين، ويمكن للكاتب وصف فقط الحرف المستعمل في سطر. التزيين.

ملاحظة: في اطار اعادة تصميم الإجراء **عرض_افقي** نفترض ان كتابة اي عدد طبيعي تستهلك نفس المساحة على الشاشة، ونحددها هنا بعشرة احرف، وهذا الافتراض غير حقيقي مع ما علمناه من سلوك تعليمة **اكتب**، فمثلا تنفيذ تعليمات النص 69 تخرج على الشاشة ما نراه في خطأ: لم يُعثر على مصدر. المرجع وهذا ما يحدث في الحقيقة، فاذا اعتبرنا ان التعليمة **اكتب** تحترم ما افترضناه فان تنفيذ تعليمات النص 69 تخرج على الشاشة ما نراه في خطأ: لم يُعثر على مصدر المرجع، وسوف نرى فيما بعد كيفية جعل التعليمة اكتب تحترم ما يفترضه كاتب الخوارزم في شأن اخراج القيم على الشاشة.

لرقم | سلوك تعليمات الكتابة على الشاشة

```
خوارزم_7_الفصل_10_سلوك_تعليمات_الكتابة_على_الشاشة
اجراء اساسي(){
    طبيعي ط1 = 12، ط2 = 2456 ؛
    اكتب (ط1)
    اكتب (ط2) ؛
    اكتب (14567)
    اكتب_سطر ()
    اكتب_سطر (ط1)
    اكتب_سطر (ط2)
```

9	اكتب_سطر (14567)
10	} // نهاية الإجراء اساسي
} // نهاية الخوارزم_7_الفصل_10_سلوك_تعليمات_الكتابة_على_الشاشة	

النص 69: سلوك تعليمات الكتابة على الشاشة

```
14567245612
12
2456
14567
```

الشكل 4: السطر الأول يبين النتيجة الحقيقية لتنفيذ تعليمات الأسطر 3، 4 و 5 من النص 7

```
12      2456    14567

2
2456
14567
```

الشكل 5: السطر الأول لنتيجة تنفيذ تعليمات النص 7 مع فرضية ان كل قيمة طبيعية تستهلك مساحة 10 احرف

في النسخة الجديدة للإجراء **عرض_افقي** (النص 70) نرى ان المدخل الذي كان يستعمل لوصف سطر التزيين قد استبدل بحرف يستعمل في انشاء سطر التزيين (المدخل **حرفر_التزيين**) ، ولإنشاء سطر التزيين يستعين الإجراء **عرض_افقي** بإجراء آخر اسمه **أخرج_خط** (السطرين 4 و 11، الإجراء **عرض_افقي**، النص 70)، همه الوحيد هو اخراج خط التزيين على الشاشة، وهذا الإجراء يستعمل المدخلين **حرف التزيين و حجم**، المدخل **حرف التزيين** يجلب للإجراء الحرف الذي يبنى عليه سطر التزيين، والمدخل **حجم** يحدد عدد الأحرف المكونة لسطر التزيين، اي حجم سطر التزيين.

لرقم	الإجراء أخرج_خط
خوارزم_8_الفصل_10_اجمع_خانتان_متتاليتان {	
	.....
	.....
1	اجراء أخرج_خط (حرف حرفر_التزيين؛ طبيعي حج){
2	طبيعي عداد = 0 ؛
3	مادام(عداد < حجم){ ؛
4	اكتب (حرف)؛
5	عداد ++ ؛
6	}
7	اكتب_سطر()
8	}
	.....
	.....
1	اجراء عرض_افقي (طبيعي[ ] جد_ط1؛ حرف حرفر_التزيين؛ حرف حد_الخانة){
2	طبيعي رقم = 0 ؛
3	طبيعي حجم_خط_التزيين = جد_ط1.حجم *11 +1 ؛
4	أخرج_خط(حرف_التزيين، حجم_خط_التزيين)
5	مادام(رقم < جد_ط1.حجم){
6	اكتب (حد_الخانة)
7	اكتب (جد_ط1[رقم]
8	رقم++ ؛
9	}

```
10      اكتب_سطر (حد_الخانة)  // الحد الأخير
11      أخرج_خط (حرف_التزيين، حجم_خط_التزيين)
12  }
.....
.....
} // نهاية الخوارزم_8_الفصل_10_اجمع_خانتان_متتاليتان
```

النص 70: النسخة الثالثة للإجراء عرض_افقي

قبل ان يطلب خدمة الإجراء **أخرج_خط**، يقوم الإجراء **عرض_افقي** بحساب حجم خط التزيين انطلاقا من حجم الجدول ومن حجم المساحة التي تستهلكها كتابة كل قيمة طبيعية، وكذلك الأحرف التي تفصل الخانات عن بعضها في الشاشة، فكل خانة تستهلك 11 حرفا ما عدا الخانة الأخيرة فإنها تستهلك 12 حرفا، وبذلك تكون عبارة حساب حجم خط التزيين على الشكل : حجم الجدول * 11 + 1، وهذا ما نراه في السطر 3 من الإجراء **عرض_افقي** (النص 70)، ومع هذه النسخة الجديدة للإجراء **عرض_افقي**، تصبح كتابة الإجراء **اساسي** اسهل كما يبينها السطرين 5 و 6 من النص 71 .

```
لرقم | الإجراء اساسي
خوارزم_9_الفصل_10_اجمع_خانتان_متتاليتان {
. . . . . .
. . . . . .
1   اجراء اساسي(){
2   | طبيعي[ ] جدول1 ، جدول2 ؛
3   | جدول1 = عمر_جدول()
4   | جدول2 = اجمع_خانتان_متتاليتان (جدول1)
5   | عرض_افقي (جدول1، "-"، '|') ؛
6   | عرض_افقي (جدول2، "-"، '|') ؛
7   } // نهاية الإجراء اساسي
. . . . . .
. . . . . .
} // نهاية الخوارزم_9_الفصل_10_اجمع_خانتان_متتاليتان
```

النص 71: نص الإجراء اساسي مستغلا النسخة الثالثة للإجراء عرض_افقي

1 – 2 – 4 رؤية ثانية لكتابة خط التزيين:

يمكن ان نستعمل وظيفة همها انشاء سلسلة من نفس الحرف تمثل خط التزيين (النص 72)، وترجع الوظيفة هذه السلسلة، وفي هذه الحالة طلب كتابة السلسلة على الشاشة لن يكون في صدر الوظيفة بل خارجها، أي عند من يطلب تنفيذها، اي الإجراء **عرض_افقي** فيما يخصنا، وهذا ما نراه في السطرين 5 و12 من الإجراء **عرض_افقي** (النص 72)، ونرى في السطر 4 من الإجراء **عرض_افقي** كيفية طلب خدمة الوظيفة **اصنع_خط** المكلفة بإنشاء سطر. التزيين.

تنطلق الوظيفة **اصنع_خط** من سلسلة فارغة تنشؤها في السطر 2 (الوظيفة **اصنع_خط**، النص 72)، ثم تدخل في دوران (السطر 4 الى 7 من الوظيفة) ويتم في كل دورة اضافة حرف الى السلسلة، وعدد الدورات هو عدد الأحرف المكونة لسطر التزيين، وهذا العدد والحرف المكون لسطر التزيين يُأخذان من مداخل الوظيفة.

لرقم	الإجراء اصنع_خط
خوارزم_10_الفصل_10_اجمع_خانتان_متتاليتان {	
	. . . . .
	. . . . .
1	سلسلة اصنع_خط (حرف حرفر_التزيين؛ طبيعي حج){
2	سلسلة خط_التزيين = "" ؛ // انشاء ساسلة حرزف فارعة
3	طبيعي عداد = 0 ؛
4	مادام(عداد < حجم){ ؛
5	خط_التزيين = خط_التزيين + حرف ؛ // اضافة حرف للسلسلة
6	عداد ++ ؛
7	}
8	ارجع خط_التزيين ،
9	}
	. . . . .
	. . . . .
1	اجراء عرض_افقي (طبيعي[ ] جد_ط1؛ حرف حرفر_التزيين؛ حرف حد_الخانة){
2	طبيعي رقم = 0 ؛
3	طبيعي حجم_خط_التزيين = جد_ط1.حجم *11 +1 ؛
4	سلسلة خط = اصنع_خط (حرف_التزيين، حجم_خط_التزيين) ؛
5	اكتب_سطر (خط) ؛
6	مادام(رقم > جد_ط1.حجم){
7	اكتب (حد_الخانة)
8	اكتب (جد_ط1[رقم]
9	رقم++ ؛
10	}
11	اكتب_سطر (حد_الخانة) // الحد الأخير
12	اكتب_سطر (خط)
13	}
	. . . . .
	. . . . .
} // نهاية الخوارزم_10_الفصل_10_اجمع_خانتان_متتاليتان	

النص 72: استعمال وظيفة لإنشاء خط التزيين

الفصل الحادي عشر

الأوعية: تمثيل ولخراج جداول الولقع المعاش

1 – مقدمة:

الواقع المعاش الذي تتناول الخوارزميات جانبا منه لحل بعض مشاكله او تحسين الأداء فيه يحتوي على جداول معقدة في الشكل والمضمون، فمثلا يمكن ان نرى جدولا فيه عدد من الأعمدة ، وفي كل عمود معلومة من نمط مختلف عن نمط عمود محاذ له، والشكلين (خطأ: لم يُعثر على مصدر المرجع و خطأ: لم يُعثر على مصدر المرجع) مثالين بسيطين لجداول الواقع المعاش، فالجدول الأول (خطأ: لم يُعثر على مصدر المرجع) يحتوي على مختلف المعلومات المتعلقة بالمستفيدين من عملية بيع مساكن بالتقسيط، فالعمود الأول (الاسم واللقب) يحتوي معلومات من النمط **سلسلة** (سلسلة حروف)، والعمود الثاني (رقم الشقة) يحتوي على معلومات من النمط **طبيعي**، والعمود الرابع (القيمة الإجمالية) يحتوي على معلومات من النمط **حقيقي** والعمود الثامن يحتوي على معلومات من نمط مهيكل يمثل تاريخا ما، فإذا اردنا ان نمثل كل معلومة بالنمط المناسب لها، فمثل هكذا جدول لا يمكن تمثيله مباشرة في الخوارزم بجدول واحد، ففي عالم الخوارزميات، معلومات الجدول يجب ان تكون من نفس النمط، والإشكال الذي سوف يطرح هنا هو كيفية تمثيل مثل هكذا جدول.

المؤسسة الأهلية لبناء العمارات و المساكن الفردية **المنطقة الصناعية، شارع المقاولين، رقم 124** **مدينة الاعمال و التجارة**
جدول المستفيدين من المشروع الف باء تاء **عمارة رقم 7**

الشكل 1: نموذج لجدول يستخدم لمتابعة بيع البيوت بالتقسيط

مدرسة الإمام عبد الحميد بين باديس **حي العلماء المسلمين** **مدينة العلوم**
وضعية المستويات والأقسام والتلاميذ

الشكل 2: نموذج لجدول يبين عدد التلاميذ في اقسام مؤسسة تعليمية

الجدول الثاني (خطأ: لم يُعثر على مصدر المرجع) فيه بعض التعقيد، فبالرغم من ان المحتوى هو من نفس النمط، فالأسطر. والخانات وما تحتويه من معلومات لها دلالات مختلفة، فالسطر. بحد ذاته يمثل مستوى تعليمي، والخانة تمثل قسما ان كان محتواها مختلف عن القيمة 0، والقيمة تمثل مرة عدد التلاميذ في القسم ومرة تمثل عدد التلاميذ في المستوى ومرة ثالثة تمثل عدد التلاميذ في المؤسسة.

اذا تأملنا خطأ: لم يُعثر على مصدر المرجع و خطأ: لم يُعثر على مصدر المرجع نرى ان الجداول تحتوي على معلومات اخرى في غاية الأهمية يجب تمثيلها في الخوارزم وهذه المعلومات هي اسماء الأعمدة واسماء الأسطر، ففي خطأ: لم يُعثر على مصدر المرجع نرى فقط اسماء الاعمدة، اما في الشكل 2 فهناك اسماء الأعمدة، واسماء الأسطر واسم لخلية (المجموع الكلي).

مع مثل هذه الجداول، يجب على كاتب الخوارزم ان يفكر مليا في طريقة تمثيلها وتنظيمها. واخراجها. على الشاشة ، وهذا ما سوف نراه عبر الأمثلة التالية.

2 – آليات التحكم الدقيق في اخراج النصوص

لإخراج الجداول على صورتها. الحقيقية في الشاشة (او اي جهاز اخراج كالطابعات)، لا تكف التعليميتين **اكتب** و**اكتب_سطر**، فللوصول الى اخراج جدول كجدول خطأ: لم يُعثر على مصدر المرجع على مساحة الشاشة، يكون العمل جد صعب اذا استعملنا التعليميتين **اكتب** و **اكتب_سطر**، خاصة عندما يكون الشكل الحقيقي مرتبط بمتغيرات مثل عدد الاقسام. فالجدول. لا يمكن ان يكون عرض اعمدته ثابتا، فهذا العرض يتكيف مع عدد الأقسام في مستوى ما، فاذا كان العدد كبير كان عرض كل العمود صغيرا. واذا كان عدد الأقسام ضئيل يكون عرض الأعمدة اكبر.

حتى نتمكن من الكتابة السهلة والفعالة لخوارزميات الإخراج على الشاشة، سوف نستعمل صيغة ثانية للتعليميتين **اكتب** و **اكتب_سطر**، وفي هذه الصيغة تُمكن التعليمتين كاتب الخوارزم من حجز مساحة من الشاشة خاصة لاستقبال إخراج قيمة ما وكذلك وصف شكل اخراج قيمة ما، مثلا اضافة الإشارة + قبل قيمة طبيعية او حقيقية، او تحديد عدد الارقام بعد الفاصلة في سياق اخراج عدد حقيقي.

3 – التعليمة اكتب (او اكتب_سطر) و كيفية التحكم في اخراج النصوص على الشاشة:

في الصيغة الأولى للتعليمة **اكتب (او اكتب_سطر)**، نستعمل مدخلا واحدا، وهو من النمط **سلسلة**، اما في الصيغة الثانية للتعليمة **اكتب** (او **اكتب_سطر.)** فيمكن اضافة مداخل اخرى غير محددة العدد، وفي كل مدخل من هذه المداخل الإضافية توضع اي عبارة يمكن تقييمها، وفي الغالب تكون العبارة بسيطة جدا، تتمثل في اسم متغيرة او قيمة ثابتة.

في هذه الصيغة الثانية تحتوي السلسلة التي توضع في المدخل الأول على **رموز خاصة** هدفها التحكم في كيفية اخراج القيم التي وضعت في المداخل الأخرى، وكونها تحتوي على **رموز التحكم في الإخراج**، غالبا ما نسمي المدخل الأول **بالسلسلة المتحكمة في إخراج النصوص**.

الشكل العام لكتابة التعليمة **اكتب (او اكتب سطر)** في صيغتها الثانية يكون كما يلي:

اكتب ("سلسلة التحكم في إخراج النصوص" ، عبارة1 ، عبارة2 ..... عبارة س)

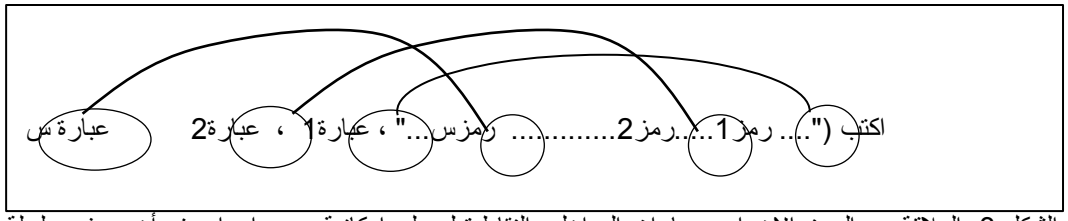
او

اكتب_سطر ("سلسلة التحكم في إخراج النصوص". ، عبارة1 ، عبارة2 ..... عبارة س)

العبارات **عبارة1،عبارة2** الخ... يمكن ان تكون نتيجة تقييمها من اي نمط، فبعد تقييم عبارة ما (مثلا العبارة 2)، تحول قيمة العبارة الى سلسلة من الحروف حسب ما ورد في رمز الإخراج المخصص لها، ثم توضع هذه السلسلة في مكان الرمز المناسب (رمز2 المناسب للعبارة 2) ثم ترسل الى الشاشة.

تذكير: الفرق بين التعليمة **اكتب** و التعليمة **اكتب_سطر** يكمن في ان التعليمة **اكتب_سطر** تكتب حرفا اضافيا هو الحرف الذي يمكن من الانتقال الى اول السطر التالي في الشاشة (او اي طابعة).

كما اشرنا اليه سابقا، فالتحكم في الإخراج يتم باستعمال **رموز خاصة** داخل سلسلة التحكم في الإخراج، وكل رمز يبين كيفية اخراج القيمة التي وضعت في المدخل المناسب له، فمثلا اذا وجدت في سلسلة التحكم في الإخراج ثلاثة رموز، وجب استعمال ثلاث مداخل اخرى زيادة على المدخل المخصص لسلسلة التحكم في الإخراج ، فالرمز الأول يناسب المدخل الأول الذي يلي مدخل سلسلة التحكم في الإخراج، و الرمز الثاني يناسب المدخل الثاني الذي يلي مدخل سلسلة التحكم في الإخراج، و هكذا، كما يظهر في خطأ: لم يُعثر على مصدر المرجع.



الشكل 3: العلاقة بين الرمز الإخراج وعبارات المداخل والنقاط تدل على امكانية وجود اي احرف أخرى في سلسلة التحم في الإخراج

تبدأ كتابة الرمز المتحكم في الإخراج بالحرف %، تليه إشارات تبين حجم المساحة التي يجب ان تحجز للإخراج وكيفية الإخراج، وتنته كتابة الرمز بحرف يشير الى نمط القيم المستهدفة، فالحرف **ط** يشير الى النمط **طبيعي**، والحرف **ح** يشير الى النمط **حقيقي**، والحرف **م** يشير الى النمط **منطقي**، والحرف **ف** يشير الى النمط **حرف**، والحرف **س** يشير الى النمط **سلسلة**.

من بين الإشارات التي تُوجه كيفية الإخراج نجد على سبيل المثال:

- ضرورة ملئ الفراغ باحرف الصفر قبل القيمة الطبيعية او الحقيقية، فمثلا اذا كانت القيمة التي تخرج هي القيمة الطبيعية 25 واذا كانت مساحة اخراج القيمة هي 6 احرف، فالإخراج بهذه الأشارة ينتج السلسلة "000025".
- وضع اشارة القيمة (+ او –) قبل القيمة الطبيعية او الحقيقية.
- الدقة التي تحديد عدد الأرقام بعد الفاصلة فيما يخص القيم الحقيقية.

3 – 1 رموز التحكم في ابسط صورها:

تتكون رموز التحكم في الإخراج في ابسط صورها من الحرف % متبوعا مباشرة بالحرف الدال على نمط القيمة المستهدفة، وهي كالتالي:

%ط : المدخل المستهدف فيه قيمة من النمط طبيعي,

%ح : المدخل المستهدف فيه قيمة من النمط حقيقي.

%س : المدخل المستهدف فيه قيمة من النمط سلسلة حروف

%م : المدخل المستهدف فيه قيمة من النمط منطقي

%ف : المدخل المستهدف فيه قيمة من النمط حرف

رموز التحكم في القيم الطبيعية و الحقيقية:

لإخراج القيم الطبيعية او الحقيقية يمكننا استعمال الرمز في شكله البسيط، اي %ط او %ح، او في شكل يحتوي على اشارات او توجيهات الإخراج مثل:

- حجم المساحة التي تحتجز
- استعمال حرف الصفر لملء الفراغات التي لم تستعمل عند كتابة رموز القيمة الطبيعية او الحقيقية
- اظهار الإشارة + عند اخراج القيم الموجبة
- تحديد عدد الأرقام بعد الفاصلة، وهذا ما نسميه بالدقة، عند اخراج قيمة حقيقية

لتبيان ما اوردناه، نسرد الأمثلة التالية:

- ما هو الرمز لكتابة القيمة الطبيعية 278 في مساحة قدرها 25 حرفا ؟ الجواب: يكون رمز الإخراج على الشكل **%25ط.**
- ما هو الرمز لكتابة القيمة الطبيعية 278 في مساحة قدرها 25 حرفا، و نريد ملء الفراغ الذي يسبق كتابة القيمة بالحرف 0؟ الجواب: يكون رمز الإخراج على الشكل **%025ط.**
- ما هو الرمز لكتابة القيمة الطبيعية 278 في مساحة قدرها 25 حرفا، و نريد اظهار اشارة + ؟ الجواب: يكون رمز الإخراج على الشكل **%+25ط.**

- ما هو الرمز لكتابة القيمة الحقيقية 134,87654 في مساحة قدرها 20 حرفا، وبدقة ثلاثة اعداد (اي ثلاثة اعداد بعد الفاصلة)؟ الجواب: رمز الإخراج يكون على الشكل **20.3%ح**، وتحتسب الفاصلة في الإخراج.
- ما هو الرمز لكتابة القيمة الحقيقية 134,87654 في مساحة قدرها 20 حرفا، بدقة بأربعة اعداد، و نريد ان يكتب الرقم 0 بدل الفراغات؟ الجواب: يكون رمز الإخراج على الشكل **%020.4ح**.

3- 2 سلوك التعليمة اكتب:

تقوم التعليمة **اكتب** بتصفح مدخلها الأول المحتوي على سلسلة التحكم في الإخراج، وتحتوي هذه السلسلة على حروف ترسل كما هي الى الشاشة وعلى رموز التحكم في الإخراج ، ويجب انجاز ما تحتويه هذه الرموز من اشارات وتوجيهات.

تقوم تعليمة **اكتب** (او **اكتب_سطر**) بالتجوال في السلسلة المتحكمة في الإخراج انطلاقا من الحرف الأول، فترسل كل حرف تصادفه الى الشاشة، الا في حالة واحدة، وهي عندما تصادف التعليمة رمزا من رموز التحكم في الإخراج، ففي هذه الحالة لا ترسل التعليمة الى الشاشة الحروف المكوِنة لرمز الإخراج، بل تنظر التعليمة في المدخل المناسب للرمز، فتأخذ القيمة الموجودة في المدخل المناسب وتخرجه حسب ما يحتويه الرمز من توجيهات، وانطلاقا من توجيهات الرمز في كيفية الإخراج، تحول التعليمة اكتب قيمة المدخل المناسب للرمز الى سلسلة من الحروف ثم ترسلها الى الشاشة.

للشرح العملي لسلوك التعليمة اكتب، نستعين بالنصين، النص 73 والنص 74، المحتويان على بعض الامثلة المبينة لكيفية استعمال التعليمة **اكتب** في صيغتها الثانية ويظهر الشكلان، الشكل 2 و خطأ: لم يُعثر على مصدر المرجع، نتيجة تنفيذ تعليمات النصين على الشاشة، ومن اجل ان يكون شرح سلوك التعليمة **اكتب** في صيغتها الثانية واضحا وعمليا، اضفنا الى الشكلين شبكة تبين مواقع الحروف في الشاشة، وهذه المواقع موجودة حقيقة في تصاميم اخراج النصوص على الشاشة.

3 – 3 استعمال رموز الإخراج في ابسط صيغها (النص 73 و الشكل 2).

في السطر 7 من النص 73 نرى ان عدد مداخل التعليمة اكتب هو اربع مداخل: اولا السلسلة المتحكمة في الإخراج وبعدها ثلاث متغيرات، ونلاحظ ان السلسلة المتحكمة في الإخراج هي في ابسط صورها، ومع مثل هذه الرموز المذكورة في السلسلة المتحكمة في الإخراج، تُخصص لكل متغيرة المساحة الكافية لها على الشاشة، لا تزيد ولا تنقص، فمثلا:

- المتغيرة **الف** تحتوي على القيمة 1234 ولإخراجها على الشاشة تحجز لها 4 مواقع لاستضافة الرموز 1، 2،3 و4.

- المتغيرة **حق** المحتوية على القيمة 45.36، تستهلك 5 مواقع، ونرى ان النقطة الفاصلة تستهلك موقعا واحدا.
- المتغيرة **منط** تستهلك 4 مواقع، لكونها تحتوي على القيمةالمنطقية **صحيح.**

```
لرقم | الوظيفة حصل_على_معلومات_المؤسسة
خوارزم المثال_1_النص_11_امثلة_في_الإخراج {
1 |  اجراء اساسي() {
2 |  |  طبيعي الف = 1234
3 |  |  حقيقي حق = 45.36
4 |  |  منطقي منط = صحيح
5 |  |  حرف  حر  =  '،'
6 |  |  سلسلة  فارق = " ، "
7 |  |  اكتب_سطر ( "%ط%ح%م"، الف، حق، منط )  ؛  // سلسلة التحكم في الإخراج مكونة فقط رموز  الإخراج للمتغيرات
8 |  |  اكتب_سطر ( "%ط%ف%ح%ف%م"، الف، حر، حق، حر، منط )  ؛
9 |  |  اكتب_سطر ( "%ط%س%ح%س%م"، الف، فارق، حق، فارق، منط ) ؛
10 |  |  اكتب_سطر ( "الف=%ط، حق=%ح، منط%م"، الف، حق، منط )
11 |  |  اكتب_سطر ( "%ط "، الف )
12 |  |  اكتب_سطر ( "%ح "، حق )
13 |  |  اكتب_سطر ( "%م"، منط )
14 |  } //نهاية اجراء اساسي
} // نهاية الخوارزم المثال_1_النص_11_امثلة_في_الإخراج
```

النص 73: استعمال رموز الإخراج دون حجز مساحة للإخراج

نرى من خلال السطر الأول للشاشة (الشكل 2) نتيجة تنفيذ تعليمة السطر 7 **من** النص 73 ونلاحظ ان قيم المتغيرات الثلاث قد اخرجت وهي ملتصقة بعضها البعض، وهذا راجع الى الصيغة التي استعملناها في سلسلة التحكم في الإخراج، وللتفريق بين هذه القيم عند اخراجها على الشاشة في سطر واحد، يمكن ان نستعمل عدة طرق، وسوف نستعرض بعضا منها.



الشكل 2: نتيجة تنفيذ تعليمات النص 73

من بين ابسط هذه الطرق، ادخال متغيرة من النمط حرف بين قيمتين، كما يظهر من خلال تعليمة السطر 8، ففيها وضعنا المتغيرة **حر** بين المتغيرتين **الف** و **حق** و بين المتغيرتين **حق** و **منط،**

وزدنا في سلسلة التحكم في الإخراج رمز الإخراج الخاص بالحرف، و يبين السطر الثاني من الشاشة نتيجة تنفيذ تعليمة السطر 8.

بدل استعمال متغيرة من النمط حرف، يمكننا استعمال متغيرة من النمط سلسلة، وهذا ما نراه من خلال تعليمة السطر 9، فاستبدلنا في المداخل المتغيرة **حر** بالمتغيرة **فارق**. وفي سلسلة التحكم في الإخراج استبدلنا الرمز الخاص بإخراج الحروف، بالرمز الخاص باستخراج سلاسل الحروف، وتظهر نتيجة التنفيذ في السطر الثالث من الشاشة.

كما اشرنا اليه عند التعريف بسلوك التعليمة اكتب، يمكننا الاستغناء عن المتغيرات التي تلعب دور الفارق اذا كان الفارق ثابتا، وفي هذه الحالة نضع الأحرف الممثلة للفارق (او اي امر آخر) مباشرة في سلسلة التحكم كما يظهر ذلك من خلال تعليمة السطر 10، اين زج بالقيمة الثابتة " ، " مباشرة في سلسلة التحكم في الإخراج، وتظهر نتيجة تنفيذ تعليمة السطر 10 في السطر الرابع من الشاشة.

3-4 استعمال رموز الإخراج مع تحديد حجم مساحة الإخراج (النص 74 و خطأ: لم يُعثر على مصدر المرجع).

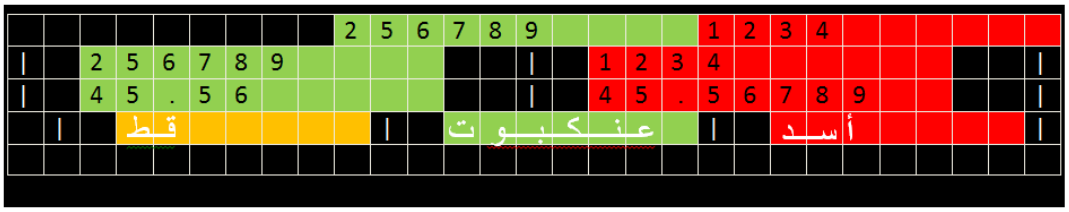
في امثلة النص 73 لم نستعمل بعد الخاصية الهامة لرموز الإخراج المتعلقة بتحديد مساحة إخراج قيمة ما، فالرموز التي استعملناها في النص 73 لا تحتوي على هذه الخاصية، ولذلك فقد حدد من المساحة ما يكفي فقط لكتابة قيمة ما.

تُذكر حجم مساحة الإخراج مباشرة قبل الحرف الدال على نمط القيمة التي يستهدفها رمز الإخراج، فمثلا الكتابة **%10ط** تعني ان اخراج القيمة الطبيعية الموجودة في المدخل المناسب يكون في مساحة تَسَعُ لعشرة احرف (او عشرة مواقع)، فاذا كانت **س** هي المتغيرة التي وضعت في المدخل المناسب، وكانت قيمة س هي 278، فكتابة المتغيرة **س** على الشاشة تكون كالتالي: اولا تكتب سبعة فراغات (7 احرف فراغ) ثم تكتب الرموز 2،7 و 8 و اذا كانت قيمة **س** هي 78654 فتكتب اولا 5 فراغات ثم تكتب القيمة 78654.

```
لرقم
خوارزم المثال_2_النص_11_امثلة_في_الإخراج {
1   اجراء اساسي() {
2       طبيعي الف = 1234، باء = 256789
3       حقيقي حق = 45.56789
4       سلسلة[ ] اسماء = {"اسد" ، "عنكبوت" ، "قط"} ؛
5       اكتب_سطر ( "%10ط%10ط"، الف، باء ) ؛
6       اكتب_سطر ( "% |10ط% | 10ط| "، الف، باء ) ؛
7       اكتب_سطر ( "% |10ح% | 10.2ح| "، حق، حق ) ؛
8       طبيعي عداد = 0 ؛
9       مادام (عداد < اسماء.حجم){
10          اكتب ( "%|7س "، اسماء[عداد] ) ؛
11          عدادا++
```

```
12          }
13          اكتب_سطر ( " |"
14      } //نهاية الإجراء اساسي
} // نهاية الخوارزم المثال_2_النص_11_امثلة_في_الإخراج
```

النص 74: استعمال رموز الإخراج مع حجز مساحة للإخراج



الشكل 5: نتيجة تنفيذ تعليمات النص 2

في السطر رقم 5 من النص 74 تقوم تعليمة **اكتب** بكتابة متتالية لقيمتين طبيعتين كل واحدة منها في مساحة قدرها 10 احرف، ولكون القيمتين الموجودتين في المتغيرتين **الف** و **باء** يمكن كتابتهما بأقل من عشرة رموز. (الرموز من 0 الى 9)، فإن العددين لا يظهران ملتصقين، كما نراه في السطر الأول من الشاشة **(خطأ: لم يُعثر على مصدر المرجع)**، وفي هذا الشكل تظهر بلون واحد المساحة المحتجزة لكل قيمة، ولو لم تحدد مساحة في رموز الإخراج لظهرت القيمتين متلاصقتين، بل اكثر من هذا، فلو كانت المساحة غير كافية لظهرت القيمتين متلاصقتين.

في السطر 6 من النص 74، أُضيف لسلسلة التحكم في الإخراج قيم ثابتة تتمثل في حروف تفصل مساحة **الف** و مساحة **باء** ، ونرى في السطر الثاني على الشاشة **(خطأ: لم يُعثر على مصدر المرجع)** نتيجة تنفيذ التعليمة رقم 6، وتظهر النتيجة وكأنها تمثل سطرا لجدول ما والحروف الفاصلة تبين اعمدة الجدول.

السطر 7 من النص 74 يبين كيفية اخراج قيمة حقيقية، فعندما لا نحدد الدقة يكتب كل ما في القيمة في المساحة المخصصة، وعندما تحدد الدقة لن تكتب الا الحروف التي تشملها دقة القيمة.

تظهر في السطر الرابع من الشاشة نتيجة تنفيذ تعليمة التكرار (الأسطر من 8 الى 12) التي تتجول في الجدول **اسماء**، فكلما زار الخوارزم خانة من خانات الجدول، يقوم بكتابة محتوى الخانة في مساحة قدرها 7 احرف (التعليمة رقم 10).

ملاحظة: ما لا يمكن وصفه في رموز الإخراج في اللغة الشرمزية: لا يمكن ان نحدد من خلال رموز الإخراج وضعية القيم في المساحة، فوضعيتها تكون دائما في الجهة اليمنى، فلا يمكن ان نجعلها في وسط المساحة المحجوزة او في يسارها. للوصول الى مثل هذه الأهداف، اي تحديد موقع القيمة في المساحة، علينا بإنجاز خوارزم خاص بهذا الأمر.

3 – 5 استعمال المتغيرات مكان القيم الثابتة في رموز الإخراج.

الى هذا الحد رأينا ان القيمة التي توصف في رمز الإخراج والدالة على مساحة الإخراج، هي قيمة ثابتة، ومعنى هذا ان مساحة الإخراج ثابتة تتعلق بمحيط ثابت، ومثل هذا المحيط نجده في جدول خطأ: لم يُعثر على مصدر المرجع، فعدد اعمدة الجدول ثابت وكذلك عرضها، والسؤال الذي يطرح هنا: هل استعمال القيم الثابتة في رموز الإخراج يكون مجديا اذا كان المحيط متغيرا، ومثل هذا المحيط نجده في خطأ: لم يُعثر على مصدر المرجع، فعدد الأعمدة، الذي يمثل عدد الاقسام في مستوى ما، يتغير من مؤسسة الى اخرى، واذا كان عدد الأقسام مختلف فان عدد الأعمدة يختلف، وبما ان الإخراج يتم في مساحة حجم عرضها محدود، كالشاشة او الطابعة، فان حجم عرض الأعمدة يصغر كلما كبر عدد الأقسام ويكبر كلما صغر عدد الأقسام(الشكل 3) مع وجود حد اقصى وحد ادنى للحجم لا يمكن تجاوزهما.

إكمالية ابن خلدون
حي المعلمين
مدينة العلوم

وضعية المستويات و الأقسام والتلاميذ

	قسم 1	قسم 2	قسم 3	قسم 4	قسم 5	قسم 6	قسم 7	المجموع
السنة الأولى	25	27	31	26	22	29	28	188
السنة الثانية	29	25	28	24	27	28	27	188
السنة الثالثة	30	29	31	30	31	30	×××	181
السنة الرابعة	28	28	29	28	29	×××	×××	142
السنة الخامسة	30	30	31	40	×××	×××	×××	131
المجموع الكلي								830

مدرسة الإمام عبد الحميد بين باديس
حي العلماء المسلمين
مدينة العلوم

وضعية المستويات و الأقسام والتلاميذ

	القسم ألف	القسم باء	القسم تاء	القسم جيم	المجموع
السنة الأولى	25	27	22	29	103
السنة الثانية	29	25	27	×××	81
السنة الثالثة	30	29	×××	×××	59
المجموع الكلي					243

الشكل 3: عرض الاعمدة مرتبط بالعدد الأكبر من الأقسام في مستوى ما

للتمكن من التكيف مع محيط متغير، وجب استعمال متغيرة، او اي عبارة مبنية على الأقل على متغيرة واحدة، في رمز الإخراج، والسؤال الذي علينا الإجابة عليه هو كيف نستعمل متغيرة او عبارة في رمز الإخراج؟

قلنا فيما سبق ان المدخل الأول للتعليمة **اكتب** في صيغتها الثانية هو من النمط "**سلسلة من الحروف**"، وعندما نقول ان المدخل يجب ان نضع فيه سلسلة من الحروف، فهذا قول موجز لقول ادق مفاده انه يجب ان نضع في المدخل الأول اي عبارة ينته تقييمها الى سلسلة من الحروف، واذا استعملنا هذا التعريف الأدق فيمكننا مثلا ان نكتب الكتابات المتساوية التي تظهر في الأسطر من 5 الى 9 من النص 75 .

```
خوارزم المثال_3_النص_11_امثلة_في_الإخراج
إجراء اساسي
    طبيعي ط = 123
    حقيقي ح = 24.76
    طبيعي م_ط = 20 ، م_ح = 40 ؛ // م_ط: مساحة طبيعي  م_ح: مساحة حقيقي
    اكتب("مساحة طبيعي 20 حرفا:%20ط و مساحة حقيقي 40 حرفا:%40ح"، ط، ح) ؛
    اكتب("مساحة طبيعي 20 حرفا: " + "%20ط" + " و مساحة حقيقي 40 حرفا:" + "%40ح" ، ط، ح) ؛
    اكتب("مساحة طبيعي 20 حرفا:%" + "20" + "ط" + " و مساحة حقيقي 40 حرفا" + "%" + "40" + "ح" ، ط، ح) ؛
    اكتب("مساحة طبيعي 20 حرفا:%" + 20 + "ط" + " و مساحة حقيقي 40 حرفا" + "%" + 40 + "ح" ، ط، ح) ؛
    اكتب("مساحة طبيعي 20 حرفا:%" + م_ط + "ط" + " و مساحة حقيقي 40 حرفا" + "%" +مح + "ح"، ط، ح) ؛
} //نهاية الإجراء اساسي
```

```
} // نهاية الخوارزم المثال_3_النص_11_امثلة_في_الإخراج
```

النص 75: في الأسطر من 4 الى 7 كتابات متساوية مرتكزة على عبارات يفضي تقييمها الى سلسلة من الحروف

تقييم عبارات المدخل الأول للتعليمة اكتب في الأسطر من 5 الى 9 ينتج العبارة الموجودة في المدخل الأول لتعليمة السطر 4 الممثلة لسلسلة التحكم في الإخراج، ونلاحظ ان كل ما ورد في المدخل الأول للتعليمة اكتب في السطرين 5 و 6 قيمة ثابتة من النمط سلسلة، ونلاحظ في السطر 7 ان المساحة وضعت لوحدها في سلسلة حروف ("20" و "40")، و عند تقييم العبارة تلصق السلسلتين الممثلتين للمساحة بالحرف % الموجود في السلسلتين التي سبقتهما في العبارة.

في السطر 8 استعملت قيم ثابتة من النمط طبيعي وهي 20 و 40 بدل السلسلتين "20" و " 40"، ونذكر ان القيمتين 20 و "20" مختلفتين، فالأولى من النمط طبيعي و الثانية من النمط سلسلة، وعند تقييم العملية + التي تربط بين سلسلة من الحروف و قيمة طبيعية، فان القيمة الطبيعية تحول الى سلسلة من الحروف ثم تربط بالسلسلة، وهذا السلوك للعملية + قد رأيناه من قبل واستعملناه بكثرة في التعليمات "**اكتب**".

في السطر 9، عوض ذكر القيمتين الطبيعيتين 20 و 40، ذكرنا متغيرتين من النمط طبيعي، وهما **م_ط** و **م_ح**، وبما ان احد معاملي العملية + هو من النمط طبيعي، فان تقييم العبارة يكون اولا بالتحصل على قيمة المتغيرة من النمط طبيعي، ثم تحويلها الى سلسلة من الحروف ثم الصاقها بالسلسلة الممثلة للمعامل الآخر للعملية +.

وهكذا، من خلال السطر 9، نرى انه بإمكاننا كتابة عبارة فيها متغيرات تستعمل لبناء رموز الإخراج، وجعل سلسلة التحكم في الإخراج تتأقلم مع متغيرات المحيط التي تستعمل فيه.

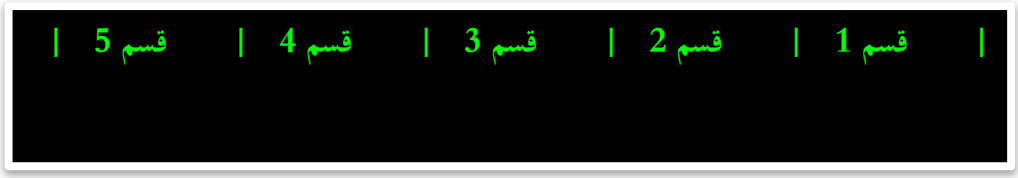
3 – 6 مثال توضيحي لكيفية بناء رموز الإخراج تتوافق مع تغير المحيط

نريد انجاز اجراء، اسمه **اخرج_عناوين،** هدفه كتابة سطر على الشاشة فيه عدد من الأسماء (او العناوين)، يشبه الشكل 4، ويستقي هذا الإجراء الاسماء التي يكتبها على السطر من مدخله الوحيد، وتوفر الأسماء في المدخل على شكل جدول من الأسماء، اي ان رأس الخوارزم يكون على الشكل التالي:

اجراء اخرج_عناوين(سلسلة[ ] جد_اسماء)

نفترض في هذا الخوارزم ان السطر في الشاشة لا يمكنه استيعاب اكثر من 80 حرفا، ويجب على الخوارزم توزيع سطر الشاشة بالسوية على كل الأسماء التي تكتب في السطر، فمثلا اذا وفرت للإجراء 4 اسماء، فعلى الخوارزم ان يقسم السطر الى 4 مناطق، بحجم 20 حرفا لكل منطقة، وفي كل منطقة يظهر اسم من الاسماء، وإذا كان عدد الاسماء 8، فعلى الخوارزم ان يقسم السطر الى 8 مناطق، بحجم 10 حروف لكل منطقة.

تبدأ كل منطقة بكتابة حرف يبين حدود كل منطقة وهو الحرف | (حرف **العمود** او حرف **العصا**)، فيكتب هذا الحرف في بداية كل منطقة ، ثم يكتب الاسم مباشرة بعده وتنته كل منطقة بثلاثة فراغات، وينته السطر بالحرف |، ويبين الشكل 4 سطرا فيه 5 مناطق، وحجم كل منطقة 16 حرفا.



الشكل 4: سطر فيه مناطق سعة كل منطقة 16 حرفا

في الشكل 4 نجد في كل منطقة ما يلي:

- موقعا واحد مخصص لحرف الحد (|)
- 12 موقعا لاستضافة الاسم (او العنوان)، ولتحديد هذه المساحة نستعمل الرمز **12%س**
- 3 فراغات بعد الاسم وقبل حد المنطقة التالية
- المنطقة الأخيرة من السطر تحتوي في آخرها على فراغين والحرف الممثل للحد.

فكل منطقة، يظهر فيها اسم، سعتها 16 حرفا.

انطلاقا من هذا المثال، نسوق التمرين التالي: ما هي القيمة التي نذكرها في رمز الإخراج اذا كان عدد الأسماء هو 8؟ والجواب يكون كالتالي: اذا كان عدد الاسماء 8 فهذا يعني ان سعة كل منطقة هي 10 احرف (نذكر ان حجم السطر في الشاشة هو 80 حرفا) ، واذا استثنينا حاجز بداية المنطقة والفراغات الثلاث في آخر كل منطقة، فانه لن يتبق للاسم الا 6 احرف، وهكذا يكون رمز الإخراج على الشكل 6%س، وهنا يجب طبعا ان نستعمل اسماء لا يتجاوز حجمها 6 احرف، ومن هذا التمرين نستخلص العلاقة التي تبين لن كيفية ايجاد قيمة مساحة رمز الإخراج، و هي كالتالي:

حجم_المنطقة = حجم_سطر_الشاشة / عدد_العناوين

مساحة_رمز_الإخراج = **حجم_المنطقة – الفراغات – مساحة_حرف_الحد،**

و فيما يخص هذا التمرين: قيمة الفراغات هي 3 ومساحة حرف الحد هي 1،

وبهذا تكون **مساحة_رمز_الإخراج** على الشكل التالي:

مساحة_رمز_الإخراج = حجم_سطر_الشاشة / عدد_العناوين – 4

اي **(80 / جد_اسماء.حجم) – 4** ، فعدد العناوين هو حجم الجدول **جد_اسماء**

```
خوارزم المثال_4_النص_11_إخراج_عناوين {
// ---------------------------------------- الإجراء اخرج_عناوين
اجراء اخرج_عناوين( سلسلة[ ] جد_اسماء ){
    طبيعي مساحة = (80 / جد_اسماء.حجم) - 4 ؛
    طبيعي عداد = 0،
    مادام (عداد < جد_اسماء.حجم - 1) {
        اكتب ("%"|" + مساحة +"س "، جد_اسماء[عداد] ) ؛ // كتابة كل المناطق ما عدا المنطقة الأخيرة
        عداد ++
    }
    اكتب_سطر ("%"|" + مساحة +"س |"، جد_اسماء[عداد] ) ؛ // كتابة المنطقة الأخيرة
}
```

```
1  // ---------- الإجراء أخرج_خط
2  اجراء أخرج_خط (حرف حرف_الخط؛ طبيعي حجم){
3      طبيعي عداد = 0 ؛
4      مادام(عداد < حجم){ ؛
5          اكتب (حرف_الخط )؛
6          عداد ++ ؛
7      }
8      اكتب_سطر()
9  }
10 // ---------- الإجراء اساسي
11 اجراء اساسي(){
12     سلسلة [ ] عناوين_01 = { "قسم اول"، "قسم ثاني"، "قسم ثالث" ، "قسم رابع" }؛
13     سلسلة [ ] عناوين_01 = { "ق 01"، "ق 02"، "ق 03"، "ق 04"، "ق 05"، "ق 06"، "ق 07"، "ق 08" }؛
14     أخرج_خط('-' ، 80)،
15     اخرج_عناوين(عناوين_01) ؛
16     أخرج_خط('-' ، 80)،
17     اخرج_عناوين(عناوين_02) ؛
18     أخرج_خط('-' ، 80)،
19 }
20 } // نهاية الخوارزم المثال_4_النص_11_إخراج_عناوين
```

النص 76: خوارزم اخراج عناوين مختلفة الحجم في سطر واحد

3 – 6 – 1 سلوك الإجراء:

ينطلق عمل الإجراء **اخرج_عناوين** (السطر 1 من النص 76) مما يجده في الجدول الذي يوضع في مدخله المسمى **جد_اسماء** ، فمن هذا الجدول يتحصل الإجراء على عدد الاسماء (السطر 2)، وبعد التحصل على عدد الأسماء، يقوم الخوارزم بتحديد حجم المناطق، ثم على القيمة التي يجب ان تعبر عن حجم المساحة في رمز الإخراج، ومعنى هذا ان الذي يذكر في رمز الإخراج انما هي متغيرة تحمل القيمة التي تحدد حجم المساحة في رمز الإخراج.

لجلب قيمة المتغيرة الى رمز الإخراج نستعمل طريقة تعليمة السطر 9 من النص 75، اي كتابة عبارة يفضي تقييمها الى وضع قيمة المتغيرة في الرمز، وهذا ما نراه في العبارة: **"%|"+ مساحة +"** " في السطر 5 من النص 76.

```
--------------------------------------------------------------------------------
|  قسم اول  |       قسم اول  |       قسم اول  |       قسم اول  |
--------------------------------------------------------------------------------
| ق 01  |ق 01  |ق 01  |ق 01  |ق 01  |ق 01  |ق 01  |ق 01 |
--------------------------------------------------------------------------------
```

الشكل 8: نتيجة تنفيذ خوارزم النص 4

3 – 6 – 2 النسخة الثانية للإجرء اخرج_عناوين:

نلاحظ من خلال النص 76 ان الإجراء **اخرج_عناوين** مرتبط حيويا بقيمتين ثابتتين تجعل استعماله ممكنا في حالات خاصة، وهي الحالات التي يكون عرض مساحة الإخراج هي 80 حرفا ويكون حرف الحد هو الحرف |، فمثلا اذا اردنا ان نستعمل الإجراء مع شاشة عرضها 70 حرف،

ونريد ان تكون النجمة (الحرف *) هي حرف الحد، فلا يمكن للإجراء **اخرج_عناوين** في نسخته الحالية من ايصالنا الى مبتغانا.

```
خوارزم المثال_5_النص_11_إخراج_عناوين {
الإجراء اخرج_عناوين
1   اجراء اخرج_عناوين( سلسلة[ ] جد_اسماء ؛ حرف حد ؛ طبيعي حجم )}
2       طبيعي مساحة = (حجم / جد_اسماء.حجم) - 4 ؛
3       طبيعي  عداد = 0،
4       مادام (عداد < جد_اسماء.حجم - 1) {
5           اكتب (جد+"%" + مساحة +"  "، جد_اسماء[عداد] ) ؛  // كتابة كل المناطق ما عدا المنطقة الأخيرة
6           عداد ++
7       }
8       اكتب_سطر (جد+"%" + مساحة +"  " +جد ، جد_اسماء[عداد] ) ؛  // كتابة المنطقة الأخيرة
9   {
.....   . . . . . .
....    . . . . . .
} // نهاية الخوارزم المثال_5_النص_11_إخراج_عناوين
```

النص 77: النسخة الثانية من الإجراء اخرج_عناوين

لجعل الإجراء **اخرج_عناوين** اكثر مُرونة، علينا ان نُخَلصه من القيمتين الثابتين 80 وحرف العمود(الحرف |). لبلوغ هذا الهدف نضيف مدخلين الى النسخة الثانية من الإجراء **اخرج_عناوين(** النص 77)، احدهما لجلب حجم مساحة العرض (المدخل **حجم**)، والثاني لجلب الحرف الذي يستعمل كحرف حد بين مناطق الإخراج (المدخل **حد**).

4 – تمثيل و اخراج جداول الواقع المعاش:

رأينا من قبل ان جداول الواقع المعاش ليست محدودة كجداول الخوارزميات، فهي، عكس جداول الخوارزميات، تحتمل انواع مختلفة من المعلومات سواء على مستوى النمط او المعنى، وهذا التعقيد نابع من حقيقة المتعامل مع مثل هذه الجداول، فالذي يتعامل مع مثل هذه الجداول المعقدة انما هو الإنسان وليس الآلة، ومثل هذه الجداول لا يمكن تمثيلها مباشرة في الخوارزم بجدول واحد، ففي عالم الخوارزميات، يجب على معلومات الجدول ان تكون من نفس النمط، والإشكال الذي سوف يطرح هنا هو كيفية تمثيل مثل هذه الجداول المعقدة وكيفية اخراجها للإنسان على الصورة التي يريدها الإنسان.

من اجل تبيان كيفية تمثيل هذه الجداول المعقدة في عالم الخوارزميات نقوم بدراسة خوارزم معاينة وضعية المستويات والأقسام والتلاميذ، والذي يعتمد في الواقع المعاش على جدول مشابه لجدول خطأ: لم يُعثر على مصدر المرجع.

4 – 1 خوارزم معاينة وضعية المستويات و الأقسام والتلاميذ

في هذا المثال نريد انجاز خوارزم يتعاطى مع جدول خطأ: لم يُعثر على مصدر المرجع، فيُمكن هذا الخوارزم المستعمل من تعمير هذا الجدول وعرضه على الشاشة.

جدول خطأ: لم يُعثر على مصدر المرجع هو جدول ذو بعدين يحمل المفاهيم التالية :

- المستويات التعليمية في مؤسسة تعليمية.
- الأقسام في كل مستوى تعليمي.
- عدد التلاميذ في كل قسم.
- عدد التلاميذ في كل مستوى.
- عدد التلاميذ في المؤسسة.

اضافة الى تعاطيه مع جدول خطأ: لم يُعثر على مصدر المرجع، يجب على هذا الخوارزم ان يتأقلم مع اي مؤسسة تستعمل جدولا مشابها لجدول خطأ: لم يُعثر على مصدر المرجع، وتختلف جداول المؤسسات التعليمية في عدد المستويات التعليمية وعدد الأقسام في كل مستوى، ونلاحظ من خلال خطأ: لم يُعثر على مصدر المرجع، أن كل سطر من الجدول يمثل مستوى تعليمي ما، وتمثل كل خانة من كل سطر قسما من اقسام المستوى التعليمي الذي يمثله السطر.

4 – 1 – 1 تنظيم معلومات الخوارزم:

قبل ان نتطرق الى السلوك العام للخوارزم، يُستحب اولا ان نتعرف على المعلومات التي يتعاطى معها الخوارزم وايجاد افضل السبل لتمثيلها، وفي حالنا علينا ان نجد كيفية تمثيل جدول خطأ: لم يُعثر على مصدر المرجع.

ملاحظة: نلفت النظر الى امكانية تمثيل جدول ما كما هو في الواقع اذا قمنا بعدم التفريق بين نوعية المعلومات التي يحتويها الجدول والنظر اليها كلها وكأنها معطيات من نمط واحد هو النمط سلسلة، لكن مثل هذا التوجه في تمثيل الجدول يكون سببا في تعقيد محتوى الخوارزم خاصة عندما نريد تنفيذ عمليات حسابية على محتوى الحدول، فمثلا اذا كان محتوى خانة اولى هي السلسلة "25" ومحتوى خانة ثانية هي السلسلة "75"، واذا اردنا ان نجمع القيمتين على انهما قيمتان طبيعياتان ثم نضع النتيجة في خانة ثالثة، فعلينا اولا ان نحول السلسلتين "25" و "75" الى القيم الطبيعية 25 و 75، ثم في مرحلة ثانية نقوم بجمعهما فنتحصل على القيمة الطبيعية 100، وفي مرحلة ثالثة نحول القيمة الطبيعية 100 الى السلسلة "100" كي يصبح من الممكن وضعها في الخانة الثالثة.

علمنا مما سبق ان :

- جداول الواقع المعاش يمكن ان تكون مكونة من اعمدة محتوية على معلومات من انماط مختلفة.
- جداول الخوارزميات لا يمكنها تحمل اكثر من نمط واحد من المعلومات.

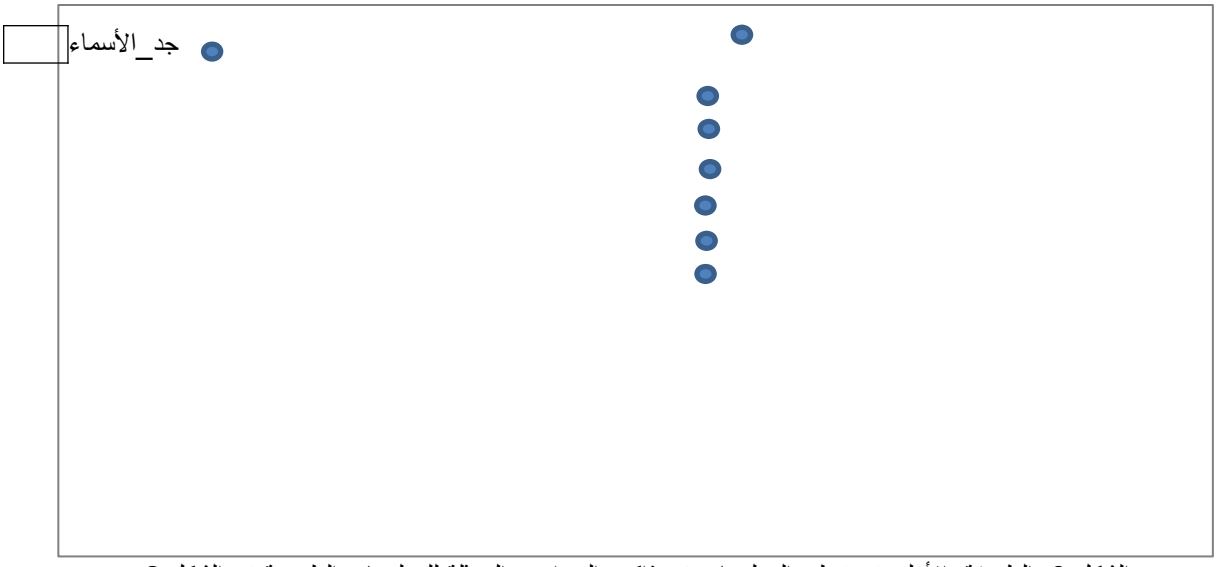
لنتمكن من تمثيل جدول الواقع المعاش وجب علينا ان نفكر مليا عن كيفية تمثيله في الخوارزم، وهذه العملية التفكيرية غالبا ما نسميها عملية تنظيم المعلومات.

في جدول خطأ: لم يُعثر على مصدر المرجع نرى اولا وجود نمطين من المعلومات :

- النمط **طبيعي :** المعلومات من هذا النمط هي على التوالي:
  - عدد التلاميذ في القسم
  - عدد التلاميذ في مستوى ما
  - عدد تلاميذ المؤسسة التعليمية.
- النمط **سلسلة:** المعلومات من هذا النمط هي على التوالي:
  - اسماء المستويات
  - اسماء الأقسام

4 – 1 – 2 استعمال جداول من الأنماط الأساسية لتمثيل جدول خطأ: لم يُعثر على مصدر المرجع

انطلاقا من الملاحظتين السابقتين، فان جدول خطأ: لم يُعثر على مصدر المرجع لا يمكن تمثيله بجدول واحد اذا اردنا ان نمثل كل قيمة بنمطها الطبيعي، فمن خصائص الجدول في الخوارزميات ان كل معطياته من نفس النمط، ولتمثيله يجب على الأقل استعمال جدولين: في الجدول الأول نضع فيه المعلومات من النمط سلسلة ، وفي الجدول الثاني نضع المعطيات من النمط طبيعي كما يظهر ذلك في خطأ: لم يُعثر على مصدر المرجع.



الشكل 9: الطريقة الأولى في تنظيم المعلومات في ذاكرة الخوارزم الممثلة للمعلومات الطبيعية في الشكل 2

اذا تعمقنا اكثر في التفكير من اجل تنظيم المعلومات، سوف نكتشف انه حتى المعلومات التي تنتمي الى نفس النمط، فليس من المستحسن وضعها كلها في نفس الجدول، فيمكن ان يكون لكل معلومة من نمط ما مفهوما خاصا بها، بعيد كل البعد عن مفاهيم المعطيات الأخرى من نفس النمط، وهنا يستحسن تفريقها.

	قسم 1	قسم 2	قسم 3	قسم 4	قسم 5	قسم 6	المجموع
السنة الأولى	25	27	31	26	22	29	160
السنة الثانية	29	25	28	24	27	×××	133
السنة الثالثة	30	29	31	30	×××	×××	120
السنة الرابعة	28	28	29	28	×××	×××	113
السنة الخامسة	30	30	31	×××	×××	×××	91
					المجموع الكلي		617

جدول 14: تنظيم اولي لمعلومات خطأ: لم يُعثر على مصدر المرجع

اذا تفحصنا الجدول 14: المستخرج من خطأ: لم يُعثر على مصدر المرجع و ركزنا على المنطقة التي تحتوي فقط على المعلومات من النمط طبيعي،نرى وجود المفاهيم التالية :

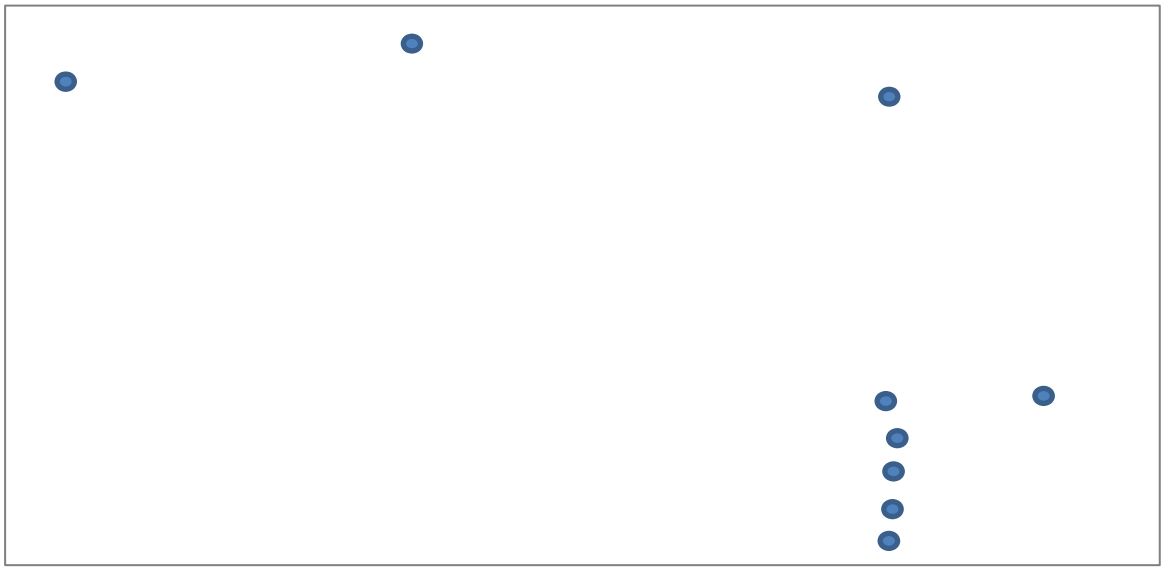
- مفهوم عدد الطلبة في قسم ما: وهي كثيرة و تظهر على شكل جدول ذو بعدين في خطأ: لم يُعثر على مصدر المرجع.
- مفهوم العدد الكلي للتلاميذ في مستوى ما، وهي ايضا منتظمة على شكل جدول ذو بعد واحد.
- مفهوم العدد الكلي للتلاميذ في المؤسسة، وهي معلومة وحيدة، يمكن ان تحتويها متغيرة من النمط طبيعي.

مع امكانية وضع كل هذه المعلومات من النمط طبيعي في جدول واحد، فمن الأفضل تفريقها لكونها تمثل مفاهيم مختلفة، فمثلا، يمكننا ان ننظم المعلومات من النمط طبيعي على النجو التالي (خطأ: لم يُعثر على مصدر المرجع):

- اولا، نخصص جدولا ذو بعدين، نسميه **جد_اقسام**، و يمثل كل سطر منه مستوى ما من التعليم، فالسطر الأول يمثل **السنة الأولى**، والثاني **السنة الثانية الخ**، ويحتوي كل سطر على عدد من الخانات، كل خانة تمثل قسما ما، فالخانة الأولى تمثل **القسم رقم 1**، والخانة الثانية تمثل **القسم رقم 2** الخ، وتحتوي كل خانة من خانات السطر على قيمة طبيعية تمثل عدد التلاميذ في القسم، ففي الخانة الأولى نجد عدد التلاميذ في القسم رقم 1 وفي الخانة الثانية نجد عدد التلاميذ في القسم رقم 2 الخ. اذا كانت قيمة الخانة 0 فهذا يعني ان القسم غير موجود في المستوى المتعلق بالخانة، وهكذا اذا تأملنا في خطأ: لم يُعثر على مصدر المرجع، نرى ان المؤسسة تحتوي على 5 مستويات، و يحتوي المستوى الأول على 4 اقسام و المستوى الثالث على 6 اقسام.
- ثانيا نخصص جدولا ذو بعد واحد، نسميه **جد_المجموع** ، ونضع فيه عدد التلاميذ في مستوى ما، و حجم هذا الجدول هو عدد اسطر الجدول ذو بعدين **جد_اقسام** ، وكل خانة من الجدول **جد_المجموع** سوف تحتوي على مجموع القيم الموجودة في السطر المقابل لها في الجدول

جد_اقسام ، فالخلية الأولى (او الخانة الأولى) من الجدول **جد_المجموع** تقابل السطر الأول من الجدول **جد_اقسام** ، والخانة الثانية تقابل السطر الثاني وهكذا، كما يظهر ذلك جليا في (خطأ: لم يُعثر على مصدر المرجع) .

- ثالثا، بما ان العدد الكلي للتلاميذ في المؤسسة يمثل بقيمة واحدة فقط، فلهذه القيمة نخصص متغيرة، وقيمة هذه المتغيرة هي مجوع القيم الموجودة في الجدول ذو بعد واحد **جد_المجموع**، ونسمي **المجموع_الكلي** هذه المتغيرة.



الشكل 10: الطريقة الثانية في تنظيم المعلومات الممثلة للشكل (الشكل 2)

نفس الطريقة في تنظيم المعلومات من النمط طبيعي يمكننا ان نتبعها مع المعلومات من النمط سلسلة، ففي مجمل المعلومات من النمط سلسلة نرى ثلاثة اصناف:

- اسماء مستويات التعليم
- اسماء الأقسام
- اسم يدل على مجموع التلاميذ في كل الاقسام التابعة لمستوى ما.
- اسم يدل على مجموع التلاميذ في المؤسسة التعليمية.

فمن هذه الملاحظة نرى انه من الأفضل تخصيص(خطأ: لم يُعثر على مصدر المرجع):

- جدولا لأسماء المستويات، نسميه **جد_أسماء_المستويات**، والخانة الأولى من هذا الجدول تحمل اسم المستوى الأول، وتقابل كل خانة من خانات الجدول **جد_أسماء_المستويات** سطرا في الجدول **جد_اقسام** وخانة في الجدول **جد_المجموع**، فمثلا اذا اردنا ان نعرف اسم المستوى الثالث نتصفح الخانة الثالثة في الجدول **جد_أسماء_المستويات** والتي تحمل الرقم 2، واذا اردنا ان نعرف عدد الأقسام وعدد التلاميذ في كل قسم في المستوى الثالث، علينا ان نتصفح السطر الثالث من الجدول ذو بعدين **جد_اقسام** والذي يحمل الرقم 2، واذا اردنا ان

نعرف مجموع الطلبة في المستوى الثالث علينا ان نتصفح الخانة الثالثة من الجدول **جد_المجموع** والتي تحمل الرقم 2.

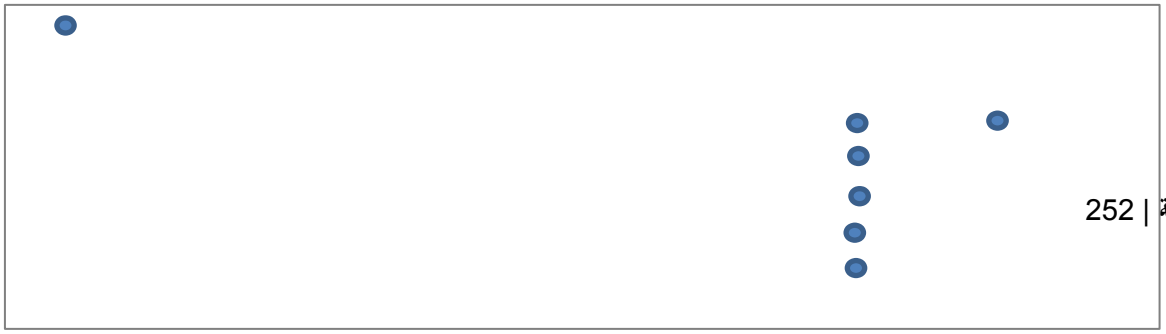
- جدولا لأسماء الأقسام، و نسميه **جد_أسماء_الأقسام**، وكل خانة منه تقابل عمودا في الجدول **جد_اقسام**،
- اما الاسم الذي يدل على المجموع فنراه وحيدا، ولذا تمثيله بمتغيرة من النمط سلسلة كاف.

ملاحظة هامة: الاسم **المجموع_الكلي** هو اسم لمتغيرة من النمط طبيعي تحتوي على عدد التلاميذ في المؤسسة، وهذه المتغيرة ادخلت لتمثل الخانة الأخيرة من السطر الأخير في الجدول **جد_اقسام** الظاهر في خطأ: لم يُعثر على مصدر المرجع. أما المتغيرة **علامة_المجموع_الكلي** فهي متغيرة من النوع سلسلة تحتوي على العبارة (او العلامة) **المجموع الكلي**، و نلاحظ ان الكتابة **المجموع_الكلي** مختلفة عن الكتابة **"المجموع الكلي"**، فالأولى اسم لمتغيرة لا يمكنه احتواء فراغات والثاني علامة تدل على الاسم الموجود في الخانتين ما قبل الأخيرتين في السطر الأخير من جدول خطأ: لم يُعثر على مصدر المرجع، وهذه العلامة مخزنة في متغيرة يمكن ان تحتوي على مختلف الحروف ومن بينها الفراغ و يمكن تبديلها ان شاء كاتب الخوارزم وهذه العلامة هي التي تظهر على الشاشة عندما يخرج الخوارزم جدول خطأ: لم يُعثر على مصدر المرجع.

4 – 1 – 3 كيفية انجاز جدول الأقسام:

هناك طريقتين لإنجاز الجدول **جد_اقسام**:

- الجدول له عدد ثابت من الأعمدة، وعدد الأعمدة هو اكبر عدد من الاقسام في مستوى ما، فمثلا في خطأ: لم يُعثر على مصدر المرجع، العدد الأكبر من الأقسام موجود في المستوى الثالث، اي 6 اقسام ، وفي هذه الحالة اذا كان عدد المستويات هو 5 فالجدول يكون بخمسة اسطر وستة اعمدة اي 5 × 6، وفي هذه الحالة على الخوارزم ان يطلب هذه المعلومة، اي اقصى عدد في الأقسام من المستعمل ثم ينشأ الجدول **جد_اقسام** مرة واحدة.
- كل سطر له عدده الخاص من الأعمدة (خطأ: لم يُعثر على مصدر المرجع)، وفي هذه الحالة لا حاجة للخوارزم ان يعلم مسبقا العدد الأكبر من الأقسام، فهو يحتاج فقط الى عدد الأقسام في مستوى ما، وعند علمه به ينشأ السطر، وفي هذه الحالة ينشأ الجدول تدريجيا، فينشأ اولا الجدول الممثل للمستويات، ثم لكل مستوى ينشأ جدول الأقسام حسب عدد الأقسام، وهذه هي الطريقة التي نتبعها في النص 78 من اجل بناء الجدول **جد_اقسام**.



الشكل 11: الطريقة الثانية لتمثيل الجدول جد_اقسام

4 – 1 – 4 دراسة نص الخوارزم (النص 78):

في بداية الخوارزم، نرى اولا التعريف بالنمط **معلومات_مؤسسة،** وهدفه انشاء متغيرات تحمل كل المعلومات التي تُعرف مؤسسة ما، وفيما يخصنا اقتصرنا فقط على الاسم والعنوان، ولاستيعاب العناوين الكبيرة يحتوي النمط على سمتين مخصصتين لعنوان المؤسسة وهما: **العنوان_1** و **العنوان_2.**

من خلال التدبر في نص الإجراء اساسي، الذي يمثل قلب الخوارزم، نرى وجود 4 مراحل في هذا الإجراء:

المرحلة الأولى : و فيها تهيأ المتغيرات اللازمة لعمل الإجراء(السطر 2 الى السطر 7 من النص 78)

في هذه المرحلة تُنشأ المتغيرات التالية:

- **جد_اقسام** (السطر 2) وهي صلة لجدول ذو بعدين من النمط طبيعي، ومجازا نقول الجدول **جد_اقسام،** وفيه تمثل الأسطر مختلف المستويات والأقسام وعدد التلاميذ في كل قسم، وهذا الجدول سوف يُنشأ على مراحل، في المرحلة الأولى التي نحن فيها يتم فقط التصريح باسم الجدول، وفي المرحلة الثانية فسوف ينشأ البعد الأول (صلات الأسطر)، وفي المرحلة التي تلي سوف يُنشأ البعد الثاني (الأسطر الممثلة للأقسام).
- **جد_المجموع** (السطر 3) و هو جدول حجمه يكون كحجم البعد الأول من الجدول **جد_اقسام،** و تحتوي كل خانة منه على مجموع الطلبة في مستوى ما.

لرقم	النمط معلومات_مؤسسة
1	عرف_نمط معلومات_مؤسسة {
2	| سلسلة الاسم_الكامل ، العنوان_1 ، العنوان_2 ؛
3	}
خوارزم_المثال_6_الفصل_11_معاينة_مستويات_أقسام_تلاميذ {	
	معلومات_مؤسسة حصل_على_معلومات_المؤسسة(){ ..... المحتوى الكامل لهذه الوظيفة في النص 79
	ساسلة[ ] حصل_اسماء_المستويات(طبيعي حجم) { .....المحتوى الكامل لهذه الوظيفة في النص 80
	ساسلة[ ] حصل_اسماء_الأقسام() { ..... المحتوى الكامل لهذه الوظيفة في النص 81
	اجراء اخرج_جدول_الأقسام(معلومات_مؤسسة م ؛ طبيعي[][] جد_قس؛ طبيعي[] جد_مج؛ سلسلة[] اسم_مس، اسم_قس؛ طبيعي كل) { .... المحتوى الكامل لهذه الوظيفة في النص 82
	اجراء اخرج_سطر_عناوين(سلسلة[ ] جد_اسماء ؛ حرف حد ، فراغ؛ طبيعي عرض_شاشة، حجم_فراغ){ ..... المحتوى في النص 83
	اجراء اخرج_مستوى (طبيعي[] اقسام؛ سلسلة اسم ؛ طبيعي مساحة_اسم، عدد_خانات، مساحة_خانة ؛ حرف حد ، فراغ){ .... المحتوى الكامل لهذه الوظيفة في النص 84
	اخرج_سطر_المجموع(كل ، حجم_الفراغ_الأول ، مساحة_الخانة، عدد_الأقسام، الحد ، الفراغ){ ..... المحتوى في النص 85
// ---------- الإجراء اساسي	
1	اجراء اساسي(){
2	| طبيعي[ ][ ] جد_اقسام؛ // جدول يحتوي على الاقسام و عدد التلاميذ في كل قسم
3	| طبيعي[ ] جد_المجموع؛ // جدول يحتوي على عدد التلاميذ في كل مستوى

```
4   ساسلة[ ] جد_أسماء_المستويات // جدول لاحتواء اسماء المستويات
5   ساسلة[ ] جد_أسماء_الأقسام  // جدول لاحتواء اسماء الأقسام
6   طبيعي المجموع_الكلي = 0 ؛
7   طبيعي عدد_مس، عدد_قس ، عداد_المستويات ، عداد_الاقسام ، عدد_التلاميذ_في _القسم، عدد_تلاميذ_المستوى؛
8   اكتب " أهلا و سهلا بكم في برنامج تسيير التلاميذ في المؤسسة"
9   معلومات_مؤسسة  مؤسسة = حصل_على_معلومات_المؤسسة() ؛
10  اكتب "فضلا ما هو عدد مستويات مؤسستكم"
11  اقرأ (عدد_مس)  // الآن عدد المستويات موجود في المتغيرة عدد_مس
12  // انشاء البعد الأول للجدول للجدول جد_اقسام. حجم البعد الأول (اي عدد الأسطر) تحدده قيمة المتغيرة عدد_مس
13  جد_اقسام = شيد طبيعي[عدد_مس][ ] ،
14  // انشاء جدول اسماء المستويات عن طريق الوظيفة حصل اسماء_المستويات و حجم الجدول هو محتوى المتغيرة  عدد_مس
15  جد_أسماء_المستويات = حصل_اسماء_المستويات(عدد_مس)
16  // انشاء الجدول ذو بعد واحد جد_المجموع. حجم الجدول تحدده قيمة المتغيرة عدد_مس
17  جد_المجموع = شيد طبيعي[عدد_مس] ؛
18  // التحصل على اسماء الأقسام في المؤسسة
19  جد_أسماء_الأقسام = حصل_اسماء_الأقسام() ؛
20
21  // يبدأ لآن التجوال في الجدول جد_اقسام لإنشاء اسطره و كذا التجوال في خانات السطر المنشئ لتعمير عدد تلاميذ الأقسام
22  عداد_المستويات = 0،
23  مادام (عداد_المستويات <  عدد_مس){
24      اكتب "فضلا ادخل عدد أقسام المستوى " + جد_أسماء_المستويات[عداد_المستويات] + " : "؛
25      اقرأ عدد_قس  // الآن عدد أقسام  المستوى الحالي موجود في المتغيرة عدد_قس
26      // انشاء سطر الجدول جد_اقسام الممثل للمستوى الحالي ، وحجمه تحدده قيمة المتغيرة عدد_قس
27      جد_اقسام[عداد_المستويات]  = شيد طبيعي[عدد_قس] ؛
28      // تعمير السطرالممثل للمستوى  الحالي بعدد تلاميذ كل قسم.
29      عداد_الاقسام = 0 ؛
30      عدد_تلاميذ_المستوى  = 0،
31      مادام (عداد_الاقسام <  عدد_قس){
32          اكتب "فضلا ادخل عدد التلاميذ في القسم " + جد_أسماء_الأقسام [عداد_الاقسام] + ":" ؛
33          اقرأ عدد_التلاميذ_في_القسم ؛
34          جد_اقسام[عداد_المستويات] [عداد_الاقسام] = عدد_التلاميذ_في_القسم ،
35          عدد_تلاميذ_المستوى  = عدد_تلاميذ_المستوى + عدد_التلاميذ_في_القسم ؛
36          اكتب "ننتقل الآن الى القسم التالي "
37          عداد_الاقسام++ ؛
38      }
39      جد_المجموع[عداد_المستويات] = عدد_تلاميذ_المستوى  ؛
40      المجموع_الكلي = المجموع_الكلي + عدد_تلاميذ_المستوى  ؛
41      اكتب "ننتقل الآن الى المستوى التالي "
42      عداد_المستويات++
43  } // نهاية التكرار
44  اخرج_جدول_الأقسام(مؤسسة ، جد_أقسام، جد_المجموع، جد_أسماء_المستويات، جد_أسماء_الأقسام، المجموع_الكلي) ؛
45 } // نهاية الإجراء اساسي
} // نهاية الخوارزم_المثال_6_الفصل_11_معاينة_مستويات_أقسام_تلاميذ
```

النص 78: خطوات انشاء جدول ذو بعدين

- **جد_أسماء_المستويات** (السطر 4) وهو جدول من النمط سلسلة، و حجمه هو كحجم البعد الأول من الجدول **جد_اقسام،** و تحتوي كل خانة منه على اسم المستوى.
- **جد_أسماء_الأقسام** (السطر 5)، وهو جدول من النمط سلسلة، وتحتوي كل خانة منه على اسم قسم من اقسام المستويات، و بما ان عدد الأقسام في المستويات المختلفة يمكن ان يكون مختلفا، ومن اجل احتوائه على اسماء كل الإقسام، فحجمه هو حجم اطول سطر من الجدول **جد_اقسام،** ونعني بأطول سطر، السطر الأكبر حجما.
- المتغيرة **المجموع_الكلي**(السطر 6)، وتُعمرُ هذه المتغيرة بجمع محتوى كل خانا ت الجدول **جد_المجموع** .

- المتغيرة **عدد_مس** (السطر 7) التي تمثل عدد مستويات المؤسسة، وتُستعمل هذه المتغيرة في عملية انشاء البعد الأول للجدول **جد_اقسام،** وهذا البعد الأول يمثل جدول المستويات ويحتوي على صلات الجداول الممثلة للمستويات (اي لأسطر الجدول **جد_اقسام)**.
- المتغيرة **عدد_قس** (السطر 7) التي تمثل عدد الأقسام في مستوى ما، وتُستعمل هذه المتغيرة في عملية انشاء اسطر الجدول **جد_اقسام،** اي اقسام مستوى ما.

لإنجاز عمله، يستعين الإجراء اساسي بمتغيرات اخرى مثل:

- المتغيرتين **عداد_المستويات** و **عداد_الاقسام** (السطر 7) اللتان تلعبان دور العداد في تعليمات التكرار
- المتغيرتين **عدد_التلاميذ_في_القسم** و **عدد_تلاميذ_المستوى** (السطر 7) ، فالأولى تستعمل عند التفاعل مع المستعمل للحصول على عدد التلاميذ في قسم ما، والثانية تستعمل لحساب عدد التلاميذ في مستوى ما.

لرقم	الوظيفة حصل_على_معلومات_المؤسسة
1	معلومات_مؤسسة حصل_على_معلومات_المؤسسة(){
2	معلومات_مؤسسة مع_مو = شيد (معلومات_مؤسسة()) ؛
3	اكتب_سطر(" تعريف المؤسسة التعليمية")
4	اكتب (" فضلا، ادخل الاسم الكامل لمؤسستكم: ")
5	اقرأ (مع_مو.الاسم_الكامل) ؛
6	اكتب (" فضلا، ادخل الفقرة الأولى من العنوان (الشارع، الحي، الأرقام الخ ")
7	اقرأ (مع_مو. العنوان_1) ؛
8	اكتب (" فضلا، ادخل الفقرة الثانية من العنوان (المدينة و الولاية ")
9	اقرأ (مع_مو. العنوان_2) ؛
10	ارجع مع_مو ؛
11	} //نهاية الوظيفة حصل_على_معلومات_المؤسسة

النص 79: الوظيفة حصل_على_معلومات_المؤسسة

المرحلة الثانية: الحصول على المعلومات العامة حول المؤسسة:

من خلال السطر 9 للإجراء **اساسي (**النص 78**)**، نرى ان مهمة الحصول على المعلومات العامة حول المؤسسة، هي من اختصاص الوظيفة **حصل_على_معلومات_المؤسسة** (النص 79)، ففي هذا النص نرى ان الوظيفة ترجع متغيرة مهيكلة من النمط **معلومات_مؤسسة،** ففي السطر 2 من النص 79 تُنشأ المتغيرة **مع_مو** من النمط **معلومات_مؤسسة،** ثم في الأسطر من 3 الى 9 من النص 79 تتفاعل الوظيفة **حصل_على_معلومات_المؤسسة** مع المستعمل لتعمير سمات المتغيرة المهيكلة **مع_مو،** وعند انتهائها في السطر 10، ترجع الوظيفة المتغيرة **مع_مو،** وبكلام ادق ترجع الصلة **مع_مو،** لأننا اصبحنا نعلم ان المتغيرات المهيكلة انما هي في حقيقتها متغيرات طارئة تُستغل باستعمال الصلات، ومجازا نقول ان الصلة هي اسم المتغيرة المهيكلة الطارئة.

اذا رجعنا الى السطر 9 من النص 78، نرى ان الإجراء اساسي يلتقط في الصلة "**مؤسسة"** قيمة الصلة التي ترجعها الوظيفة **حصل_على_معلومات_المؤسسة.**

المرحلة الثالثة: انشاء البعد الأول من الجدول "جد_اقسام". و إنشاء الجداول الأخرى

في السطر 10 من الإجراء اساسي (النص 78) يبدأ التفاعل مع المستعمل، واول ما يطلبه الإجراء اساسي هو عدد المستويات الموجودة في المؤسسة (السطر 10)، ويتلق الرد في المتغيرة **عدد_مس** (السطر 11)، ومباشرة بعد التعرف على قيمة هذه المعلومة التي خزنت في المتغيرة **عدد_مس**، يقوم الإجراء اساسي بإنشاء البعد الأول من الجدول **جد_اقسام** مرتكزا على محتوى المتغيرة **عدد_مس**(السطر. 13)، وهذا البعد الأول يبين عدد الأسطر. في الجدول، وهذه هي الخطوة الأولى في سياق انشاء الجدول **جد_اقسام**، ونذكر ان كل سطر يمثل مستوى تعليمي في المؤسسة.

بعد إنشائه للبعد الأول من الجدول **جد_اقسام** والذي يبين عدد مستويات المؤسسة، يقوم الإجراء اساسي ببناء جدول توضع في اسماء كل المستويات، وتنفيذ هذا البناء يسنده الإجراء اساسي الى الوظيفة **حصل_اسماء_المستويات** (السطر 15).

لرقم	الوظيفة حصل_اسماء_المستويات
1	ساسلة[ ] حصل_اسماء_المستويات(طبيعي حجم) {
2	ساسلة[ ] جد_اسماء = شيد سلسلة[حجم] ؛
3	اكتب_سطر(" عرف اسماء المستويات في مؤسستكم، هناك " + حجم + " مستويات")
4	طبيعي عداد = 0 ؛
5	مادام (عداد < حجم) {
6	اكتب (" فضلا، ادخل ا سم المستوى رقم "+ (عداد + 1) + " : ")
7	اقرأ (جد_اسماء[عداد]) ؛
8	عداد++ ؛
9	}
10	ارجع جد_اسماء ؛
11	} //نهاية الوظيفة حصل_اسماء_المستويات

النص 80: الوظيفة حصل_اسماء_المستويات

من خلال النص 80، السطر الأول، نرى ان الوظيفة **حصل_اسماء_المستويات** هي من النمط **جدول ذو بعد واحد من سلاسل الحروف**، اي ان الوظيفة ترجع عند انتهاء تنفيذها جدولا فيه عدد من سلاسل الحروف، وكل سلسلة تمثل اسما لمستوى ما من مستويات المؤسسة.

من اجل اتمام عملها، ترتكز. الوظيفة **حصل_اسماء_المستويات** على عدد الأسماء التي تتكفل بجلبها من المستعمل وتضعها في جدول سلاسل الحروف، وهذا العدد تتحصل عليه عبر منفذها المسمى **حجم**، وأول ما تقوم به الوظيفة هو استعمال قيمة المدخل **حجم** لإنشاء جدول الأسماء **جد_اسماء** (السطر. 2 من النص 80)، وبعدها يبدأ التفاعل مع المستعمل من اجل تعمير الجدول **جد_اسماء** (الأسطر. من 4 حتى 9)، وعند الانتهاء من تعمير الجدول **جد_اسماء** ترجع الوظيفة قيمة **الصلة** الى من طلب تنفيذها، اي الإجراء اساسي فيما يخصنا، وفي السطر 15 من النص 78، يلتقط الإجراء اساسي رد الوظيفة **حصل_اسماء_المستويات** في الصلة **جد_أسماء_المستويات** وهي مجازا إسم جدول اسماء المستويات، وسوف تستعمل هذه الاسماء للتدليل على اسماء المستويات عندما يُخرج الخوارزم على الشاشة جدول خطأ: لم يُعثر على مصدر المرجع.

دائما في هذه المرحلة الثالثة، في السطر 17، ينشأ الإجراء اساسي الجدول **جد_المجموع**، وحجم هذا الجدول يساوي حجم البعد الأول من الجدول **جد_اقسام**، ويمثل هذا الحجم عدد المستويات في المؤسسة (انظر خطأ: لم يُعثر على مصدر المرجع و خطأ: لم يُعثر على مصدر المرجع).

تنته المرحلة الثالثة بإنشاء الجدول **جد_أسماء_الأقسام (السطر 19 من** النص 78**)** وهو جدول ذو بعد واحد يحتوي على اسماء كل الأقسام في المؤسسة، وهذه الأسماء سوف تستعمل عندما يريد الخوارزم اخراج عناوين اعمدة جدول خطأ: لم يُعثر على مصدر المرجع، وهنا ايضا يكلف الإجراء اساسي الوظيفة **حصل_اسماء_الأقسام** (النص 81) بإنجاز. هذه المهمة، اي بُنيان جدول يحتوي على اسماء كل الأقسام في المؤسسة.

```
لرقم | الوظيفة حصل_اسماء_الأقسام
1  | سلسلة[ ] حصل_اسماء_الأقسام() {
2  |     سلسلة[ ] جد_اسماء ؛
3  |     طبيعي اقصى_عدد_من_الاقسام ، عداد = 0 ؛
4  |     اكتب (" فضلا، افدنا بأكبر عدد من الإقسام يمكن ان يحتويه مستوى ما في مؤسستكم " ) ؛
5  |     اقرأ (اقصى_عدد_من_الاقسام) ؛
6  |     جد_اسماء = شيد سلسلة[حجم اقصى_عدد_من_الاقسام] ؛
7  |     اكتب_سطر(" سنبدأ الآن تعريف اسماء الأقسم في مؤسستكم ") ؛
8  |     مادام (عداد < اقصى_عدد_من_الاقسام) {
9  |         اكتب (" فضلا، ادخل اسم القسم رقم "+ (عداد + 1) + " : ")
10 |         اقرأ (جد_اسماء[عداد] ) ؛
11 |         عداد++ ؛
12 |     }
13 |     ارجع جد_اسماء ؛
14 | } //نهاية الوظيفة حصل_اسماء_المستويات
```

النص 81: الوظيفة حصل_اسماء_الأقسام

من خلال النص 81، يتبين لنا ان الوظيفة **حصل_اسماء_الأقسام** ترجع جدولا ذو بعد واحد من النمط **سلسلة** (السطر 1)، وهذا الجدول يجب ان يتسع لكل اسماء الأقسام في المؤسسة، فمثلا، ان كان ثلاث هو عدد المستويات ، وان كان في المستوى الأول 5 اقسام وفي المستوى الثاني 3 اقسام وفي المستوى الثالث 7 اقسام، يكون حجم الجدول هو 7.

يتم انشاء الجدول الذي ترجعه الوظيفة **حصل_اسماء_الأقسام** في السطر. 6 بعد ان تتحصل الوظيفة في السطر 5 من قبل المستعمل على اكبر عدد ممكن من الأقسام في مستوى ما، وبعد الانتهاء من انشاء الجدول **جد_اسماء**، تقوم الوظيفة في الأسطر من 8 الى 12 بتعمير جدول اسماء الأقسام، وفي السطر 13، و بعد الانتهاء من تعميره ترجع الوظيفة صلة الجدول الى من طلب تنفيذها، اي الإجراء اساسي (النص 78)، وهذا الأخير يلتقط رد الوظيفة في الصلة **جد_أسماء_الأقسام** (السطر 19 من النص 78).

المرحلة الرابعة: اتمام انجاز الجدول جد_اقسام

في هذه المرحلة يتم اتمام انشاء الجدول **جد_اقسام** وتعميره و كذا تعمير الجدول **جد_المجموع** وتحديد قيمة المتغيرة **المجموع_الكلي.** للوصول الى هذه الأهداف يمكن ان ننجز هذه المرحلة بعدة طرق، من ابرزها الطريقتين التاليين:

الطريقة الأولى : في هذه الطريقة يقوم الخوارزم بما يلي:

- اولا باستكمال انشاء الجدول **جد_اقسام،** ويتم هذا الأمر بإنشاء كل الأسطر الممثلة للمستويات والأقسام، ولبلوغ هذا الهدف، يتجول الخوارزم في الجدول الممثل للبعد الأول وكلما وصل الى خانة منه، قام بإنشاء السطر مرتكزا على عدد اقسام المستوى الذي يتحصل عليه بالتفاعل مع المستعمل.
- ثانيا بتعمير الجدول **جد_اقسام** بعدد تلاميذ كل قسم، ولبلوغ هذا الهدف، يتجول الخوارزم في البعد الأول من الجدول **جد_اقسام،** وكلما كان في خانة من خانات البعد الأول، يقوم بزيارة خانات البعد الثاني المتمثل في السطر المرتبط بخانة البعد الأول، والممثل لمستوى ما، وكلما وصل الى خانة في سطر ما، وهي خانة تمثل قسما ما، يتفاعل الإجراء مع المستعمل للحصول على عدد تلاميذ القسم.
- ثالثا بتعمير الجدول **جد_المجموع،** وهنا يجب على الخوارزم ان يتجول في الجدول **جد_اقسام** للحصول على عدد التلاميذ في كل قسم ليجمعها ويضعها في الخانة المناسبة من الجدول **جد_المجموع،** اي انه في نفس الوقت يتجول في الجدول **جد_المجموع.**
- رابعا تحديد قيمة المتغيرة **المجموع_الكلي،** و لبلوغ هذا الهدف وجب على الخوارزم زيارة كل خانات الجدول **جد_المجموع** و تجميع ما فيها من القيم.

الطريقة الثانية:

من ابرز ما تتسم به الطريقة الأولى، كثرة التجوال في جدول ما، فكلما دعت الضرورة، اعيد التجوال في جدول ما، فمثلا التجوال في الجدول **جد_اقسام** يعاد في المراحل الثلاثة الأولى لبلوغ الأهداف المسطرة في هذه المراحل، والتجوال في الجدول **جد_المجموع** يعاد في المرحلتين الثالثة والرابعة، وهذه الإعادة في التجوال تجعل وقت تنفيذ الخوارزم طويلا نوعا ما، وفي هذه الطريقة الثانية نحاول تفادي ما امكن اعادة التجوال في جدول ما، وهذه هي الطريقة المعتمدة في النص 78.

في هذه الطريقة الثانية، عندما ننشأ سطرا ما في الجدول **جد_اقسام** ونقوم بتعمير خاناته الممثلة للأقسام، نقوم في نفس الوقت بحساب عدد التلاميذ في المستوى الممثل بسطر الجدول **جد_اقسام،** وهذا ما يحدث في التكرار الذي يُهيأ في السطر 22 ويبدأ من السطر 23 وينته في السطر 43، ففي السطر 27 يتم انشاء السطر الممثل للمستوى الذي تحتوي المتغيرة **عداد_المستويات** على رقمه، وتتم هذه العملية بالارتكاز على عدد الأقسام في المستوى الذي تحصل عليه الإجراء من خلال التفاعل مع المستعمل في السطرين 24 و 25.

بعد ان ينشأ السطر الممثل لمستوى ما، تبدأ فورا عملية تعمير الخانات الممثلة للأقسام بعدد تلاميذ كل القسم، وتتم عملية التعمير بالتجوال في خانات السطر، ويتم هذا التجوال في عملية التكرار الذي تهيأ في السطرين 30 و 31، و تبدأ في السطر 32، وهنا لتفادي اعادة التجوال في الأقسام من اجل حساب عدد تلاميذ المستوى، يقوم الخوارزم اولا بوضع القيمة التي يُصرح بها المستعمل في

الخانة الممثلة للقسم، ومباشرة بعد هذه العملية يقوم الخوارزم بإضافة القيمة الى المتغيرة **عدد_تلاميذ_المستوى** التي هيأت في السطر 31 (قبل الدخول في تعليمة التكرار. شحنت المتغيرة **عدد_تلاميذ_المستوى** بالقيمة 0)، وهكذا عند الانتهاء من زيارة كل خانات السطر الممثل لمستوى ما، تنته تعليمة التكرار وتكون حصيلة المتغيرة **عدد_تلاميذ_المستوى** هو عدد التلاميذ في المستوى، فتوضع هذه القيمة مباشرة في الخانة المقابلة للسطر على مستوى الجدول **جد_المجموع (السطر 39)،** و بهذه الطريقة سوف لن يتجول الإجراء مرة أخرى لحساب عدد تلاميذ كل المستويات.

نفس الفكرة نتبعها في حساب المجموع الكلي للتلاميذ ، ففي السطر. 6، انشأت المتغيرة **المجموع_الكلي** بقيمة اولية هي 0، و في السطر 40 ، وفي اطار التكرار. الأول (المحتوي على التكرار الثاني)، كلما تحصلنا على عدد تلاميذ مستوى ما اضفناه الى المتغيرة **المجموع_الكلي،** وهكذا عندما ينته التكرار. الأول نكون قد بلغنا اهدافنا(تعمير. عدد تلاميذ كل قسم، حساب عدد تلاميذ كل المستويات، حساب المجموع الكلي للتلاميذ في المؤسسة) بالتجوال مرة واحدة فقط في الجدولين **جد_اقسام** و**جد_المجموع.**

المرحلة الخامسة: اخراج جدول خطأ: لم يُعثر على مصدر المرجع على الشاشة

هذه المرحلة قد اسندت الى الإجراء **اخرج_جدول_الأقسام،** والهم الوحيد لهذا الإجراء هو استعمال مختلف الجداول لإخراج جدول مشابه لجدول خطأ: لم يُعثر على مصدر المرجع على الشاشة.

السلوك العام للإجراء أخرج_جدول_الأقسام

لإنجاز عمله، يرتكز هذا الإجراء على معلومات عدة تُوفر. في مداخله(السطر 1 من النص 82) و هي على التوالي:

- تعريف المؤسسة، اي **اسم المؤسسة** و**عناوينها،** وهذه المعلومات توفر. عبر المدخل **م** من النمط **معلومات_مؤسسة.**
- **عدد المستويات** و**اسمائهم،** وهاتين المعلومتين يتم توفيرهما عبر جدول من النمط سلسلة مخصص لأسماء المستويات. يوضع في المدخل **اسم_مس،** وحجم هذا الجدول هو نفسه عدد المستويات.

لرقم	النص
1	اجراء اخرج_جدول_الأقسام(معلومات_مؤسسة م ؛ طبيعي[][] جد_قس؛ طبيعي[] جد_مج؛ سلسلة[] اسم_مس، اسم_قس؛ طبيعي كل) {
2	ثابت طبيعي حجم_الخط_الأفقي = 80
3	ثابت حرف الفراغ = ' '؛
4	ثابت حرف الحد ='|' ؛
5	ثابت طبيعي حجم_الفراغ_ألأول = 15
6	طبيعي العدد_الأكبر_من_الأقسام = اسماء_الأقسام.حجم
7	طبيعي مساحة_الخانة = (حجم_الخط_الأفقي - الفراغ_ألأول) / (العدد_الأكبر_من_الأقسام + 1)
8	
9	// المرحلة الأولى
10	اخرج_خط ("-"، حجم_الخط_الأفقي)؛ // رسم الخط االأفقي لأول
11	أكتب_سطر("%|20ف%40س%19ف|"، الفراغ ، م.الأسم_الكامل، الفراغ) ؛
12	أكتب_سطر("%|20ف%40س%19ف|"، الفراغ ، م.العنوان_1 ، الفراغ) ؛

```
13  أكتب_سطر("%|20ف%40س%19ف|"، الفراغ ، م.العنوان_2  ، الفراغ ) ؛
14  اخرج_خط("-"، حجم_الخط_الأفقي)؛  // رسم الخط الأفقي الثاني
15  أكتب_سطر("%|20ف%40س%19ف|"، الفراغ ، "وضعية المستويات و الأقسام والتلاميذ  " ، الفراغ ) ؛
16  اكتب_خط("-"، حجم_الخط_الأفقي)؛  // رسم الخط الأفقي الثالث ما قبل جدول الأقسام
17
18  // المرحلة الثانية : كتابة عناوين الأعمدة  على طريقة خطأ: لم يُعثر على مصدر المرجع
19  اكتب("%"+حجم_الفراغ_ألأول ، الفراغ) ؛
20  اكتب_خط("-"، حجم_الخط_الأفقي - حجم_الفراغ_ألأول )؛  // رسم الخط الأفقي الذي يبدأ مع نهاية الفراغ الأول و هو فوق العناوين
21  اخرج_سطر_عناوين (اسم_قس، الحد، فراغ، حجم_الخط_الأفقي، حجم_فراغ_الأول )؛
22  اكتب_خط("-"، حجم_الخط_الأفقي)؛  // رسم الخط الأفقي الثالث ما قبل جدول الأقسام
23
24  // المرحلة الثالثة : كتابة محتوى الأقسام في مختلف المستويات التعليمية حسب جدول  خطأ: لم يُعثر على مصدر المرجع
25
26  طبيعي عداد = 0؛
27  مادام (عداد < جد_قس.حجم ) { // جد_قس.حجم هو عدد اسطر الجدول ذو بعدين جد_قس
28      اخرج_مستوى (جد_قس[عداد] ، اسم_مس[عداد] حجم_الفراغ_ألأول، العدد_الأكبر_من_الأقسام ، مساحة_الخانة ؛ الحد ، الفراغ ) ؛
29      عداد++
30  }
31  // المرحلة الرابعة: اخراج السطر الذي يبين العدد الكلي لتلاميذ المؤسسة
32  اخرج_المجموع(كل ، حجم_الفراغ_ألأول ، مساحة_الخانة، العدد_الأكبر_من_الأقسام + 1، الحد ، الفراغ )
33
34  // المرحلة الخامسة: اخراج آخر سطر في الجدول
35  طبيعي فراغ_اخر_السطر = حجم_الفراغ_ألأول + مساحة_الخانة *( العدد_الأكبر_من_الأقسام – 2)
36  اكتب("%فراغ_اخر_السط"، الفراغ) ؛
37  اكتب_خط("-"، حجم_الخط_الأفقي - فراغ_اخر_السط)؛  // رسم الخط الأفقي الذي يبدأ مع نهاية الفراغ الأول
38  } // نهاية الإجراء اخرج_جدول_الأقسام
```

النص 82: الإجراء اخرج_جدول_الأقسام

- **اسماء الأقسام،** وهذه الأسماء توفر عبر جدول من النمط سلسلة مخصص لأسماء الأقسام، يوضع في المدخل **اسم_قس،** وحجم هذا الجدول يمثل ايضا **العدد الأكبر من الأقسام** الذي يمكن ان يحتويه مستوى ما.
- **عدد الأقسام في كل مستوى** و **عدد التلاميذ في كل قسم،** وهذه المعلومات توفر عبر جدول طبيعي ذو بعدين، عدد اسطره هو عدد المستويات، وكل سطر يحتوي على الأقسام في مستوى من المستويات وكل خانة في السطر تمثل قسما ومحتواها هو عدد التلاميذ في القسم، ويوضع مثل هذا الجدول في المدخل **جد_قس.**
- العدد الاجمالي للتلاميذ في كل مستوى، ويتوفر هذا عبر جدول من النمط الطبيعي، كل خانة منه تمثل عدد التلاميذ في مستوى ما، ويوضع مثل هذا الجدول في المدخل **جد_مج.**
- العدد الكلي للتلاميذ في المؤسسة، وتوضع هذه المعلومة في المدخل المسمى **كل.**

ملاحظة: يمكن للإجراء **اخرج_جدول_الأقسام** ان يستغني عن المداخل التي تجلب العدد الكلي للتلاميذ (المدخل **كل)** وعدد التلاميذ في كل مستوى(المدخل **جد_مج**) ، ويقوم هو بنفسه بحساب هذه المعلومات بالتجوال في الجدول الذي يوضع في المدخل **جد_قس** ، ومعنى هذا انه علينا كتابة التعليمات التي تمكن من الوصول الى هذين الهدفين، وهذا ما سوف نتجنبه في هذه النسخة من الإجراء **اخرج_جدول_الأقسام.**

في بداية الإجراء **اخرج_جدول_الأقسام**، من السطر 2 الى السطر 5، نرى التصريح ببعض المتغيرات ذات القيمة الثابتة، والتي تصف المحيط الذي يتم فيه الإخراج، كحجم عرض الشاشة، وحجم الفراغ الذي يبدأ به السطر الذي يحتوي على عناوين الجدول، والاحرف المستعملة للتفريق بين خانات الجدول (حرف الحد)، والحرف الذي يلعب دور الفراغ.

بعد التصريح بالمتغيرات الثابتة، نرى التصريح بمتغيرتين بقيم اولية: المتغيرة المسماة **العدد_الأكبر_من_الأقسام** والمتغيرة المسماة **مساحة_الخانة.** المتغيرة **العدد_الأكبر_من_الأقسام** تستخلص قيمتها الأولية من حجم الجدول **اسماء_الأقسام** الذي يحتوي على اسماء كل الأقسام في المؤسسة (السطر 6)، وادخلت هذه المتغيرة فقط لجعل النص اوضع، فكان من الممكن الاستغناء عنها واستعمال السمة **حجم** التابعة للجدول **اسماء_الأقسام** عبر العبارة **اسماء_الأقسام.حجم.**

المتغيرة **مساحة_الخانة** تستعمل لاحتواء قيمة مساحة الخانات التي فيها تكتب عناوين الأقسام او عدد التلاميذ او مجموع التلاميذ في مستوى ما، وقيمة هذه المساحة تستنتج من عبارة السطر 7، وفي هذه العبارة نرى ان المساحة المخصصة لكل العناوين هي مساحة الشاشة (المعبر عنها بالثابتة **حجم_الخط_الأفقي)** منقوصة من قيمة المتغيرة **حجم_الفراغ_ألأول** (وهي قيمة مساحة الفراغ في اول سطر العناوين وكذلك هي قيمة المساحة المخصصة لأسماء المستويات) والنتيجة تقسم بين عدد الخانات المخصصة للأقسام (المتغيرة **العدد_الأكبر_من_الأقسام**) من جهة والخانة التي تحتوي على مجموع التلاميذ في مستوى ما من جهة اخرى، وهي آخر خانة في السطر.

لإنجاز جدول مشابه لجدول خطأ: لم يُعثر على مصدر المرجع، يمر الخوارزم **اخرج_جدول_الأقسام** (النص 82) بخمس مراحل:

المرحلة الأولى: في هذه المرحلة يقوم الإجراء **اخرج_جدول_الأقسام** بكتابة مختلف العناوين التي تسبق الجدول الذي يحتوي على مختلف المعلومات حول الأقسام والمستويات التعليمية، ولإنجاز هذه المرحلة الأولى يقوم الخوارزم بكتابة خط افقي متبوعا بثلاثة اسطر فيها اسم المؤسسة وعنوانها الأول وعنوانها الثاني، ثم خط افقي ثان وتحته الجملة الثابتة " **وضعية المستويات والأقسام والتلاميذ**" متبوعة بخط افقي ثالث، ولرسم الخطوط الأفقية نستعمل الإجراء **أخرج_خط** (النص 76)، وهذه المرحلة الأولى تتكفل بها تعليمات الأسطر من 10 الى16.

في مثل جداول خطأ: لم يُعثر على مصدر المرجع، يكون دائما حجم الخط الأفقي ثابت، و يَتبع عرض المكان الذي يرسم فيه (مثلا: ورقة مكتوبة يدويا، ورقة اخرجت من طابعة، الشاشة الخ..)، وفي حال النص 82 قدر هذا الحجم بحجم العرض التقليدي للشاشة وهو ثمانين حرفا، وتمثله في النص 82 المتغيرة الثابتة **حجم_الخط_الأفقي** (السطر 2).

المرحلة الثانية : في هذه المرحلة يقوم الإجراء **اخرج_جدول_الأقسام** برسم عناوين الجدول، ولبلوغ هذا الهدف يستعين الإجراء **اخرج_جدول_الأقسام** (السطر. 21) بالإجراء **اخرج_سطر_عناوين** (النص 83).

نلاحظ ان رسم عناوين جدول خطأ: لم يُعثر على مصدر المرجع تتشابه بشكل كبير مع رسم العناوين التي يقوم بها الإجراء **اخرج_عناوين** في نسخته الأولى والثانية (النص 76 و النص 77) مع الفروق التالية:

- يسبق المنطقة الأولى من عناوين الجدول، فراغ في السطر. (خطأ: لم يُعثر على مصدر المرجع)، ثم تأتي اول منطقة لأول عنوان،
- كل العناوين ليس لها نفس المفهوم، فالعناوين الأولى هي اسماء اقسام، اما العنوان الأخير فيتمثل في كلمة "**المجموع**"

وهكذا، لإنجاز الإجراء **اخرج_سطر_عناوين،** يمكننا ان ننطلق من نص الإجراء **اخرج_عناوين،** ثم ندخل عليه تغييرات طفيفة تمكنه من ادخال المنطقة الفارغة قبل اول منطقة من مناطق العناوين، واخراج الكلمة "**المجموع**" في أخر منطقة في سطر العناوين، وهذا ما نراه من خلال النص 83.

لرقم	الإجراء اخرج_عناوين
1	اجراء اخرج_سطر_عناوين(سلسلة[ ] جد_اسماء ؛ حرف حد ، فراغ؛ طبيعي عرض_شاشة، حجم_فراغ){
2	طبيعي مساحة = ((عرض_شاشة - حجم_فراغ)/ (جد_اسماء.حجم +1)) - 4 ؛// عدد أحرف الفراغ في منطقة+حرف الحد=4
3	
4	اكتب ("%"+حجم_فراغ، فراغ) ؛ // كتابة الفراغ في اول السطر
5	طبيعي عداد = 0،
6	مادام (عداد < جد_اسماء.حجم) {
7	اكتب (حد+"%" + مساحة +"س "، جد_اسماء[عداد]) ؛ // كتابة كل المناطق ما عدا المنطقة الأخيرة
8	عداد ++
9	}
10	اكتب_سطر (جد+"%" + مساحة +"س " +جد ، "المجموع") ؛ // كتابة اسم العمود الأخير
11	{

النص 83: النسخة الثالثة من الإجراء اخرج_عناوين

في النص 83، نرى ان المساحة المخصصة للعناوين، والتي تأتي بعد الفراغ الأول، تُقَسم بالتساوي بين كل اسماء العناوين، ونذكر. ان اسماء العناوين هي اسماء كل الأقسام متبوعة بالكلمة "**المجموع**" التي تمثل عنوان العمود الذي تحتوي خاناته على مجموع التلاميذ في مستوى ما، فالعدد الكلي للعناوين هو عدد اسماء الأقسام زائد العنوان الخاص بالعمود الأخير **(جد_اسماء.حجم +1)** (السطر 2)، و يُأخذ عدد اسماء الأقسام مباشرة من حجم الجدول **جد_اسماء** (السطر. 2)، وتمثل المتغيرة **مساحة** المساحة التي تُحجز لكل منطقة تستحوذ عليها خانة من خانات العناوين، وتستنتج قيمتها من خلال عبارة السطر 2.

من خلال السطر 4، نرى ان اول ما يكتب في السطر هو الفراغ الأول، وحجم منطقة الفراغ يحدده المدخل **حجم_فراغ**، وبعد اخراج المنطقة الفارغة في اول السطر، يقوم الإجراء

الجزء الثاني: الفصل الحادي عشر: الأوعية: تمثيل واستغلال جداول الواقع المعاش في الخوارزميات

اخرج_سطر_عناوين بإخراج ا سماء الأقسام **(الأسطر من 6 الى 9)** ثم اسم العمود الأخير في السطر **(السطر 10)**.

المرحلة الثالثة: في هذه المرحلة تبدأ كتابة اسطر الجدول المحتوية على المعلومات حول مختلف المستويات، وهنا لا بد ان نشير الى امر هام، هو ان اسطر الجدول **جد_اقسام** (اي الجدول الذي يستغله الإجراء) لا تشبه تماما جدول خطأ: لم يُعثر على مصدر المرجع، فأسطر الجدول **جد_اقسام** ليس لها نفس العدد من الخانات لكون عدد الأقسام في مستوى ما يمكن ان يكون مختلفا عن سواه من المستويات، اما في جدول خطأ: لم يُعثر على مصدر المرجع فنلاحظ ان كل الأسطر لها نفس العدد من الخانات، و نستنتج العدد الحقيقي للأقسام في كل مستوى بعدد الخانات المحتوية على قيمة طبيعية، اما الخانات التي تحتوي على الإشارة ×× فهي تمثل اقسام غير موجودة في المستوى الدراسي الذي يمثله سطر الجدول، ومعنى هذا ان رسم جدول خطأ: لم يُعثر على مصدر المرجع يعتمد على حجم المستوى الذي يحتوي على اكبر عدد من الأقسام.

لرسم سطر مشابه لسطر جدول خطأ: لم يُعثر على مصدر المرجع، يستعين الإجراء **اخرج_جدول_الأقسام** بالإجراء **اخرج_مستوى** (السطر 28 من النص 82)، واول عمل يقوم به الإجراء **اخرج_مستوى** (النص 84)، هو كتابة حرف الحد متبوعا باسم المستوى التعليمي، ثم يكتب عدد التلاميذ في كل قسم من اقسام المستوى، وتنته كتابة السطر بكتابة مجموع تلاميذ المستوى التعليمي، وتحدد لإسم المستوى التعليمي نفس المساحة التي حددت للفراغ الأول في سطر عناوين الأقسام، اي قيمة المتغيرة **حجم_الفراغ_ألأول** (السطر 7 من النص 82) وتحدد للقيم الطبيعية الممثلة لعدد التلاميذ في القسم ولعدد التلاميذ في مستوى ما نفس المساحة التي حددت للعناوين في سطر العناوين، اي قيمة المتغيرة **مساحة_الخانة** (السطر 7 من النص 82) وهذه المعلومات توفر عبر مداخل الإجراء **اخرج_مستوى.**

كتابة السطر الممثل للمستوى تشبه كثيرا كتابة العناوين في الإجراء **اخرج_سطر_عناوين (**النص 83) ، والفرق بينهما يكمن في الأمرين التاليين:

- **في** الإجراء **اخرج_مستوى** يكتب حرف الحد متبوعا باسم المرحلة في اول السطر بينما يكتب فراغا في الإجراء **اخرج_سطر_عناوين.**
- في الإجراء **اخرج_مستوى**، و في الخانات التي تلي الخانة الأولى المخصصة لاسم المرحلة، تكتب القيم الممثلة لعدد التلاميذ ومجموعهم والتي تأخذ من جدول من النمط طبيعي، اما في الإجراء **اخرج_سطر_عناوين**، فتكتب اسماء الأقسام التي تأخذ من جدول من النمط سلسلة.

اضافة الى هذين الأمرين، فان الإجراء **اخرج_مستوى** يقوم بالمهمتين التاليتين:

- حساب مجموع الطلبة في المستوى، وهذه المهمة يقوم بها اثر تجوله في السطر الممثل للأقسام، ثم يضع هذا المجموع في العمود الأخير المعنون "**المجموع**".
- اخراج القيم الطبيعية في وسط المنطقة التي تظهر فيها هذه القيم، ونذكر ان هذه القيم تمثل عدد التلاميذ في الأقسام ومجموعهم في مستوى تعليمي ما.

```
لرقم | الإجراء اخرج_عناوين
1  | اجراء اخرج_مستوى (طبيعي[] اقسام؛ سلسلة اسم ؛ طبيعي مساحة_اسم، عدد_خانات، مساحة_خانة ؛ حرف حد ، فراغ ){
2  |     طبيعي مساحة_عت = 2 ؛   //المساحة الكافية لكتابة عدد التلاميذ في القسم
3  |     طبيعي مساحة_مجموع= 3 ؛  //المساحة الكافية لكتابة عدد التلاميذ في المستوى
4  |     طبيعي فراغ_قبل_عت = (مساحة_خانة - مساحة_عت)/2 ؛
5  |     طبيعي فراغ_قبل_مجموع = (مساحة_خانة - مساحة_ مجموع)/2  ؛
6  |     طبيعي فراغ_بعد_عت،
7  |     طبيعي فراغ_بعد_مجموع ؛
8  |
9  |     اذاكان ((مساحة_خانة - مساحة_عت)%2 == 0)
10 |         فراغ_بعد_عت = (مساحة_خانة - مساحة_عت)/2
11 |     والا
12 |         فراغ_بعد_عت = (مساحة_خانة - مساحة_ عت)/2 + 1
13 |
14 |     اذاكان ((مساحة_خانة - مساحة_ مجموع)%2 == 0)
15 |         فراغ_بعد_مجموع = (مساحة_خانة - مساحة_ مجموع)/2
16 |     والا
17 |         فراغ_بعد_مجموع = (مساحة_خانة - مساحة_ مجموع)/2 + 1
18 |
19 |
20 |     اكتب (حد+ "%"+ مساحة_اسم +"س" ، اسم ) ؛  // كتابة اسم المرحلة
21 |     طبيعي مجموع = 0،
22 |     طبيعي  عداد = 0،
23 |     مادام (عداد < اقسام.حجم) {
24 |         اكتب (حد+"%" + فراغ_قبل_عت+ "ف%"+ مساحة_عت +"ط%" + فراغ_بعد_عت+"ف" ، فراغ، اقسام[عداد]، فراغ  ) ؛
25 |         مجموع = مجموع + اقسام[عداد]  ؛
26 |         عداد ++
27 |     }
28 |     اكتب_سطر(حد+"%" + فراغ_قبل_مجموع+"ف%"+ مساحة_مجموع +"ط%"+ فراغ_بعد_مجموع+"ف"، فراغ، مجموع ، فراغ ) ؛
29 | }
```

النص 84: الإجراء اخرج_مستوى، و يشبه كثيرا الإجراء اخرج_عناوين

في النص 84 يبين السطر الأول مداخل الإجراء **اخرج_مستوى،** وهي على التوالي:

- المدخل **اقسام** وفيه يوضع جدول من النمط طبيعي يحتوي على خانات تمثل الأقسام وفي كل خانة يوجد عدد التلاميذ في القسم، ونشير هنا ان حجم الجدول، اي العدد الحقيقي للأقسام في مستوى ما، لا يمكن ان يكون اكبر من قيمة المدخل **عدد_الخانات،** فقيمة هذا المدخل تبين العدد الحقيقي للخانات التي يجب رسمها، وعدد الأقسام في مستوى ما يمكن ان يكون اقل، فمثلا تظهر في السطر خانات فيها قيم طبيعية، وهي خانات تمثل الأقسام، و تظهر اخرى فارغة لا تحتوي على قيم طبيعية، وهذه الخانات لا تمثل اقساما.
- المدخل **اسم** و فيه يوضع اسم المستوى
- المدخل **مساحة_اسم** وهي المساحة التي يستهلكها اسم المستوى، وهي تساوي الفراغ في اول سطر للسطر الذي يحتوي على عناوين الأقسام (او عناوين الأعمدة)

- المدخل **عدد_الخانات** الذي من خلاله يتحصل الخوارزم على عدد الخانات التي يجب رسمها، و كما سبق و ان اشرنا اليه، يمكن ان يكو عدد الأقسام اقل من عدد الخانات التي ترسم.
- المدخل **مساحة_خانة** الذي يبين مساحة الخانة التي يوضع فيها عدد تلاميذ القسم او المجموع
- المدخل **حد** و فيه يوضع الحرف الذي يستعمل كحد بين مناطق السطر عند اخراج سطر من الجدول على الشاشة.
- المدخل **فراغ** وفيه يوضع الحرف الذي يستعمل كفراغ في منطقة كتابة قيمة ما، وهذا الفراغ يكون ما بين القيمة وحدي الخانة، فهناك فراغ يسبق العدد وهناك فراغ يأتي بعد العدد، وهنا نفترض ما يلي:
  - عدد التلاميذ يكون دائما اكبر من 10 و اقل من 99، اي ان عدد التلاميذ يستهلك مساحة حرفين على الشاشة.
  - مجموع التلاميذ في مستوى ما يكون دائما اقل من 999، اي ان القيمة الممثلة لمجموع التلاميذ في مستوى ما تستهلك مساحة ثلاثة احرف على الشاشة.

في الأسطر من 2 الى 7 صُرح بمتغيرات تستعمل في الكتابة المرنة للسلاسل المتحكمة في الإخراج، ومن بينها المتغيرات التي تستعمل في جعل القيم تظهر في وسط مناطقها، فالمتغيرة **فراغ_قبل_عت** تتكفل باحتواء عدد الفراغات قبل قيمة تمثل عدد التلاميذ، اما المتغيرة **فراغ_بعد_عت** فتتكفل باحتواء عدد الفراغات بعد قيمة تمثل عدد التلاميذ، وقيمة هاتين المتغيرتين متساوتان او تختلفان بواحد فقط حسب حجم مساحة المنطقة التي فيها يظهر عدد التلاميذ، والمتغيرتين **فراغ_قبل_مجموع** و**فراغ_بعد_مجموع** تلعبان نفس الدور لكن مع القيمة التي تمثل مجموع التلاميذ في مستوى ما.

في الأسطر من 9 الى 17 من النص 84، كتبت تعليمات هدفها تحديد عدد الفراغات قبل وبعد كتابة عدد التلاميذ وكتابة المجموع، اي قيمة المتغيرات السالفة الذكر، وفي هذا الصدد نفترض ان عدد التلاميذ لا يمكنه تجاوز العدد 99 والعدد الكلي في اي مستوى لا يمكننه ان يتجاوز 999، اي ان عدد التلاميذ ستهلك مساحة حرفين على الأكثر والمجموع في المستوى يستهلك 3 حروف على الأكثر، فاذا كانت قيمة المساحة المخصصة للمنطقة قيمة زوجية فان الفراغان ما قبل عدد التلاميذ وما بعد عدد التلاميذ متساويان، و يكونان مختلفان فيما يخص المجموع، والعكس يحدث اذا كانت قيمة المساحة المخصصة للمنطقة عددا فرديا، فمثلا اذا كانت مساحة المنطقة هي 13 حرفا، وبما ان عدد التلاميذ يستهلك دائما حرفين والمجموع يستهلك ثلاث احرف، فان عدد الفراغات قبل عدد التلاميذ هو 5 فراغات وكذلك 5 فراغات قبل المجموع، اما الفراغات بعد عدد التلاميذ فهي 6 وبعد المجموع فهي 5، اما اذا كان حجم مساحة المنطقة 16 حرفا، فان الفراغات التي تكتب قبل عدد التلاميذ وبعده هي 7 احرف، اما الفراغات قبل المجموع فهي 6 و بعد المجموع فهي 7.

مدرسة الإمام عبد الحميد بين باديس **حي العلماء المسلمين** **مدينة العلوم**
وضعية المستويات و الأقسام والتلاميذ

الشكل 12: نموذج لجدول يبين عدد التلاميذ في اقسام مؤسسة تعليمية

المرحلة الرابعة: كتابة السطر الذي يحتوي على المجموع الكلي للتلاميذ

و هذه المهمة يقوم بها الإجراء **اخرج_سطر_المجموع** (السطر 31 من النص 82)، والمعلومات التي يحتاجها هذا الإجراء (النص 85) هي المساحة التي يستعملها لكتابة السلسلة "**المجموع الكلي**"، وهي مساحة خانتين (خطأ: لم يُعثر على مصدر المرجع)، ثم مساحة الخانة التي تحتوي على القيمة الممثلة لمجموع الطلبة في المؤسسة التعليمية، ولتوفير هاتين المعلومتين هناك طرق عدة، فمثلا يمكن ان نوفر مساحة عنوان المستوى، ثم عدد الأقسام ثم مساحة كل الخانة التي توضع فيها القيمة الممثلة لمجموع الطلبة في المؤسسة التعليمية، فانطلاقا من هذه المعلومات يمكن ان نستنتج القيم التي توصف في سلسلة التحكم في الإخراج، فالحد الأول الذي يسبق السلسلة "**المجموع الكلي**" يكتب في مساحة هي مساحة عنوان المرحلة زائد مساحة خانات عدد الأقسام ناقص مساحة خانتين(خطأ: لم يُعثر على مصدر المرجع)، مباشرة بعد الحد تكتب السلسلة "**المجموع الكلي**" في مساحة قدرها مساحة خانتين(خطأ: لم يُعثر على مصدر المرجع)، ثم يكتب المجموع في متوسط الخانة الأخيرة، و نفترض ان القيمة الممثلة لمجموع الطلبة في المؤسسة لا يمكنها ان تتجاوز 9999، اي انها تستهلك من الخانة 4 احرف، ولكتابتها في الوسط نستعين بما توصلنا اليه في الإجراء اخرج سطر.

في بداية الإجراء **اخرج_سطر_المجموع** (النص 82)، من السطر 3 الى السطر 5 صُرِح بمتغيرات تستعمل في توسيط قيمة المدخل المسمى **كل** في وسط آخر خانة من السطر، ومن السطر 7 الى السطر 10 كتبت تعليمات يُستَنتج منها الفراغ ما قبل قيمة المدخل **كل** وما بعدها عندما تخرج في الخانة الأخيرة من السطر، وقد شرحنا في اطار الإجراء **اخرج_مستوى** (النص 84) هدف مثل هذه التعليمات.

يحتوي السطر 13على تعليمة اخراج المنطقة الفارغة التي تسبق المنطقة التي تحتوي على السلسلة "**المجموع الكلي**"، وتنته هذه المنطقة الفارغة بحرف الحد وهو اول حرف للمنطقة التي تحتوي على السلسلة "**المجموع الكلي**"، وهكذا يكون هدف تعليمة السطر 13 هو كتابة حرف **الحد** في منطقة حجم مساحتها يُستنتج من العبارة **حجم_الفراغ_ألأول + (عدد_الأقسام – 2)* (مساحة_الخانة+ 1) + 1**، فعدد الخانات بعد الفراغ الأول (الفراغ الموجود مباشرة تحت عناوين المستويات) هي **عدد_الأقسام+1** (1 يمثل العمود الأخير المحتوي على مجموع عدد تلاميذ مستوى ما)، ويقع حد المنطقة الفارغة في بداية الخانة الثالثة ما قبل آخر خانة، والمسافة من اول السطر الى موقع الحد هو **(عدد_الأقسام+1– 3)*(مساحة_الخانة+1)**، واضيف 1 الى **مساحة_الخانة**، كون مساحة الخانة لم يحتسب معها موقع الحد الأول، وفي كل ما سبق لم يحتسب موقع الحد الذي تنته به المنطقة الفارغة، وهو الحد الأيمن للمنطقة التي تحتوي على السلسلة "**المجموع الكلي**"، و لذا اضيف 1 في آخر العبارة.

تواصل تعليمة السطر 16 ما بدأه السطر 13 في اطار اخراج السطر الذي يحتوي على المجموع، فتقوم هذه التعليمة يكتابة السلسلة "**المجموع الكلي**" متبوعة بحرف **الحد** في مساحة خانتين مباشرة بعد الكتابة السابقة و في نفس السطر.

وتقوم تعليمة السطر 18 بإنهاء اخراج السطر الذي يحتوي على المجموع، فتكتب قيمة المجموع الكلي المتوفر في المدخل **كل**، في وسط الخانة وفي نفس السطر مباشرة بعد الكتابة السابقة متبوعا بحرف الحد والرجوع الى السطر التالي.

	الإجراء اخرج_سطر_المجموع
1	اخرج_سطر_المجموع(كل ، حجم_الفراغ_ألأول ، مساحة_الخانة، عدد_الأقسام، الحد ، الفراغ){
2	
3	طبيعي مساحة_الكل= 4 ؛ //المساحة الكافية لكتابة عدد التلاميذ في المؤسسة
4	طبيعي فراغ_قبل_الكل = (مساحة_خانة - مساحة_الكل)/2 ؛
5	طبيعي فراغ_بعد_ الكل ،
6	
7	اذاكان ((مساحة_خانة - مساحة_ الكل)%2 == 0)
8	فراغ_بعد_الكل = (مساحة_خانة - مساحة_الكل)/2
9	والا
10	فراغ_بعد_الكل = (مساحة_خانة - مساحة_الكل)/2 + 1
11	
12	// كتابة الحد الأول في مساحة تستنتج من العبارة : حجم_الفراغ_ألأول + (عدد_الأقسام – 2)* (مساحة_الخانة+1)
13	اكتب ("%ح"+حجم_الفراغ_ألأول + (عدد_الأقسام – 2)* (مساحة_الخانة+1) +1 ، الحد) ؛
14	
15	// كتابة السلسلة "المجموع الكلي" متبوعة بحرف الحد في مساحة خانتين مباشرة بعد الكتابة السابقة و في نفس السطر
16	اكتب ("%س"+ (مساحة_الخانة +1)*2 ، "المجموع الكلي"+الحد) ؛
17	
18	// كتابة المجموع الكلي في وسط الخانة و نفس السطر مباشرة بعد الكتابة السابقة متبوعا بحرف الحد و الرجوع الى السطر التالي
19	اكتب_سطر ("%ف" + فراغ_قبل_الكل+ "%ط"+ مساحة_الكل +"%ف" + فراغ _بعد_الكل + الحد ، فراغ، كل، فراغ) ؛
20	}

النص 85: الإجراء اخرج_سطر_المجموع

المرحلة الخامسة: رسم آخر سطر في جدول خطأ: لم يُعثر على مصدر المرجع :

ينته خوارزم النص 82 في هذه المرحلة الخامسة بكتابة آخر سطر في جدول خطأ: لم يُعثر على مصدر المرجع، ويحتوي هذا السطر الأخير على سلسلة حروف مكونة من الحرف –، مكتوبة مباشرة تحت السطر الذي يبين مجموع تلاميذ المؤسسة، اي ان حجم السلسلة المكونة من الحرف– هو حجم ثلاث خانات + 3 حدود، و يبدأ هذا السطر بمنطقة فارغة حجمها هو نفس حجم المنطقة الفارغة في السطر السابق (الذي يبين مجموع تلاميذ المؤسسة)، و هذا الحجم نستخلصه من عبارة السطر 35، ونلاحظ هنا كتابة ليست بجديدة، وهي التصريح في وسط النص بالمتغيرة **فراغ_اخر_السطر** (السطر 35)، وقد ادخلت هذه المتغيرة لتسهيل الكتابة في الأسطر التالية، ففي السطر 36 تكتب الفراغات التي تسبق السلسلة المكونة من الحرف –، و بعد كتابة هذا الفراغ، يستعين خوارزم النص 82 مرة اخرى في السطر 37 بالإجراء **اكتب_خط** لإتمام رسم السطر.

ملاحظة: بعد اتمام انجاز الإجراء **اخرج_جدول_الأقسام** (النص 82)، نلاحظ ان الجدول الذي يوضع في المدخل **جد_مج** لم يستعمل، ويعني هذا ان المدخل زائد لا جدوى منه، مع ان المعلومات التي تجلب من هذا المدخل يجب ان تخرج في الجدول الذي يرسم على الشاشة، وفي حقيقة الأمر هذه المعلومات تم الوصول اليها واستغلالها دون استعمال المدخل **جد_مج**، فقد تم حسابها في الإجراء **اخرج_مستوى**، عندما كان هذا الإجراء يتجول في كل سطر من اسطر جدول الأقسام. والملاحظ في الإجراء **اخرج_مستوى**، ان عملية حساب عدد التلاميذ في كل مستوى لم تكن مكلفة، بل كان الأمر سهلا، ومن هذه الملاحظة نرى انه في بعض الأحيان، عندما تكون سهلة وغير مكلفة عملية استنتاج بعض المعطيات انطلاقا من معطيات المداخل، يستحسن تقليص المداخل والتعويل على اضافة بعض التعليمات لحساب المعطيات الغائبة في المداخل، وانطلاقا من هذه الملاحضة نستخلص انه من الأفضل ان يكون رأس الإجراء **اخرج_جدول_الأقسام** على الشكل التالي:

```
اجراء اخرج_جدول_الأقسام(معلومات_مؤسسة م ؛ طبيعي[][] جد_قس؛ سلسلة[] اسم_مس، اسم_قس؛ طبيعي كل)
```

الفصل الثاني عشر

الأوعية: سلاسل الحروف

1- مقدمة:

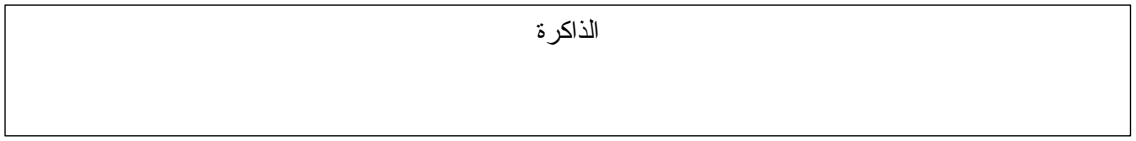
يعتبر النمط ***سلسلة_حروف*** أو باختصار ***سلسلة***، من اكثر الأنماط استعمالا، وهذا ما لحضناه من خلال الخوارزميات التي كتبناها حتى الآن، فهي التي من خلالها يتمكن الخوارزم من اخراج مختلف الرسائل على الشاشة، فأي معلومة نريد كتابتها على الشاشة، او ارسالها عبر شبكة ما من شبكات الحاسوب، يجب ان تحول اولا الى سلسلة من الحروف ثم ترسل الى الشاشة او غيرها من منافذ الحاسوب، فمثلا، اذا كانت **س** متغيرة من النمط **حقيقي** و قيمتها هي 123456.897، فإنها تكون ممثلة في ذاكرة الحاسوب على الشكل الذي يستعمل لتمثيل الأعداد الحقيقية والذي، في الغالب، يستهلك من الذاكرة 4 وحدات (خطأ: لم يُعثر على مصدر المرجع)، وعند ارسالها الى الشاشة باستخدام التعليمة اكتب **(اكتب (س))**، فان التعليمة اكتب تقوم اولا بتحويل القيمة الحقيقة الى **سلسلة من الحروف** ثم ترسلها بعد ذلك الى الشاشة، و في الشكل الممثل للنمط **سلسلة** يستهلك كل حرف وحدة واحدة من الذاكرة، اي ان تمثيل القيمة الحقيقية 123456.897 باستعمال النمط **سلسلة** سوف يستهلك 10 وحدات من الذاكرة.

الشكل 1 : تمثيل نفس القيمة في الذاكرة باستعمال النمط حقيقي والنمط سلسلة

النمط **سلسلة** نمط غير بدائي، بل هو نمط مهيكل، واقرب نمط له هو النمط **جدول حروف**، اذ يمكن ان نستعمل **جدول حروف** لتمثيل سلسلة من حروف، ويظهر وكان النمطين هما اسمين لنمط واحد، لكن في واقع لغات البرمجة ولغات الخوارزميات، فان هناك فرق يمكن ان يكون شاسعا بين النمط **جدول حروف** و النمط **سلسلة**.

في لغة **سي** و **سي++**، الفرق بينهما طفيف، فالمتغيرة من النمط سلسلة تمثل في الذاكرة بجدول حروف، الا ان هناك فرق طفيف لكن هام جدا، ان الجدول الممثل لسلسلة الحروف يجب ان تكون آخر خانة منه محتوية على الحرف "لا شيء" (قيمة الحرف "لا شيء" هي 0 و رمزه هو \0) ، فالجدول تعرف حدوده بموقع الخانة الأولى و بحجمه، اما السلسلة فيعرف حدها بموقع الحرف الأول وبوجود الحرف "لا شيء"، وفي خطأ: لم يُعثر على مصدر المرجع نرى تمثيل لمتغيرتين من النمط سلسلة، فالمتغيرة الأولى تبدأ بموقع الحرف 'ب' و تنته بالحرف '\0'، والمتغيرة الثانية تبدأ بالحرف 'ا' و تنته بالحرف '\0'.

اما في لغة **جافا**، فالفرق شاسع بين النمط **سلسلة** والنمط **جدول حروف**، فسلسلة الحروف نمط مهيكل يختلف تماما عن النمط جدول الحروف، فلا يعلم كاتب البرامج حقيقة تمثيل سلاسل الحروف في الذاكرة، فهو يكتف فقط بمعرفة كيفية استعمال عدد من الوظائف المنتسبة الى النمط **سلسلة** لإجراء العمليات المختلفة على المتغيرات من النمط سلسلة، وهذه هي الطريقة التي سوف نتبعها في هذا الكتاب.



++الشكل 2 : تمثيل الذاكرة لسلاسل الحروف في لغتي سي و سي

2 – التعريف:

النمط ***سلسلة حروف***، او باختصار ***سلسلة***، نمط مهيكل يحتوي على عدد من الحروف المتسلسلة، لا يهم كيف يكون تمثيلها الحقيقي في الذاكرة ، المهم في كل هذا ان الأحرف منتظمة في تسلسل، فهناك الحرف الأول يتبعه الحرف الثاني ثم الثالث الى آخره، وعدد الأحرف هو حجم سلسلة الأحرف، ويوفر هذا النمط عددا كبيرا من الوظائف المنتسبة تمكن من استغلال محتوى المتغيرات من النمط **سلسلة**، و يبين النص 86 بعض رؤوس اهم الوظائف التي تنتمي الى النمط **سلسلة**، ومن بين رؤوس هذه الوظائف نلاحظ في الأسطر. من 3 الى 6 رؤوس المشيدات الأكثر استعمالا.

```
1  عرف نمط  سلسلة {
2      // المشيدات
3      سلسلة (سلسلة س) ؛  // تشييد سلسلة انطلاقا من سلسلة اخرى تذكر في المدخل
4      سلسلة (طبيعي ط) ؛  // تشييد سلسلة تمثل القيمة الطبيعية المذكورة  في المدخل
5      سلسلة (حقيقي ح) ؛
6      سلسلة (منطقي م) ؛
7      // الوظائف المنتسبة الأكثر استعمالا
8      طبيعي  حجم() ؛ // تنبأ عن حجم السلسلة، اي عدد احرفها
9      حرف حرف_في_موقع(طبيعي موقع) // ترجع الحرف الموجود في الموقع الذي ذكر في المدخل
10     طبيعي  موقع_حرف(حرف ح) ؛ // ترجع اول موقع للحرف في السلسلة للحرف الموجود في المدخل ح
11     طبيعي  موقع_حرف(حرف ح، طبيعي موقع_بداية_البحث) ؛ // اول موقع للحرف ح بعد موقع بداية
البحث
12     منطقي تساوي (سلسلة س) // المقارنة بالتساوي بين سلسلتين
13     طبيعي قارن(سلسلة س)  // مقارنة سلسلتين
14 }
```

النص 86: تعريف جزئي للنمط ال مهيكل سلسلة

2 – 1 هام جدا:

عند تنفيذ الوظائف المنتمية الى النمط سلسلة، فان هذه الوظائف لا تدخل اي تغيير على محتوى السلسلة التي تنفذ عليها، فأي متغيرة من النمط سلسلة لا يمكن تغيير محتواها بعد انشائها،

وكأن المتغيرة من النمط سلسلة متغيرة ثابتة، اي انها تأخذ قيمتها مرة واحدة لا يمكن تغييرها. بعد ذلك، وكل ما تنتجه الوظائف المنتمية الى النمط **سلسلة** هي في حقيقة الأمر متغيرات اخرى تحتوي على التغييرات التي طلبت، فمثلا اذا كان هدف وظيفة ما هو استبدال كل حرف 'ج' بالحرف 'ع'، فان هذه الوظيفة تقوم بإنتاج سلسلة حروف اخرى اين نجد فيها النتيجة، اي كل حرف 'ج' قد حول الى حرف 'ع'، ولا تمس السلسلة الأصلية.

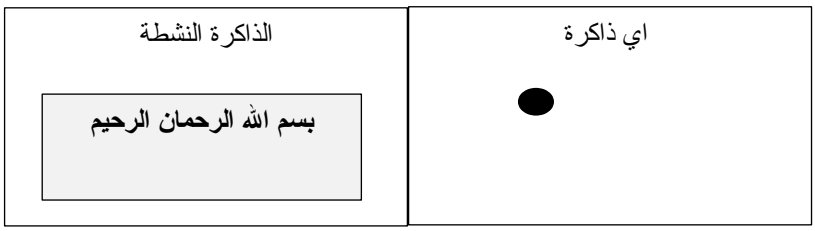
2 – 2 السلسلة الثابتة:

يطلق هذا الاسم على كل كتابة التي من خلالها نرى محتوى السلسلة، وهي كل كتابة تكتب بين حدين ممثلين بحرف الاقتباس(")، كالكتابة التالية (وقد مارسنا كثيرا هذه الكتابة).

"بسم الله الرحمان الرحيم"

3 – كيفية الوصول الى متغيرة من النمط سلسلة:

بما ان النمط **سلسلة** نمط مهيكل، وانطلاقا من المبادئ التي حددناها. من قبل بخصوص طرق الوصول الى محتوى المتغيرات المهيكلة، فان المتغيرات من النمط **سلسلة** يوصل اليها باستعمال **الصلات**، فالمتغيرة من النمط سلسلة ليست متغيرة بدائية، فالكتابة التالية **سلسلة بسملة = "بسم الله الرحمان الرحيم"**. مفادها اننا صرحنا بصلة من النمط سلسلة واسمها **بسملة** وربطنا هذه الصلة بالمتغيرة الطارئة والثابتة **"بسم الله الرحمان الرحيم". (خطأ: لم يُعثر على مصدر المرجع)**



"الشكل 3 : التمثيل في الذاكرة للتصريح: سلسلة بسملة = "بسم الله الرحمان الرحيم

التعامل مع محتوى متغيرة من النمط سلسلة:

بما ان المتغيرة من النمط **سلسلة** ليست بجدول، وبما ان كاتب الخوارزم يجهل تماما كيفية التمثيل في الذاكرة لمتغيرة من النمط سلسلة، فإن التعامل مع محتوى المتغيرة من النمط سلسة لن يكون الا عبر الوظائف المنتسبة للنمط سلسلة.

4 – كيفية انشاء المتغيرات من النمط سلسلة:

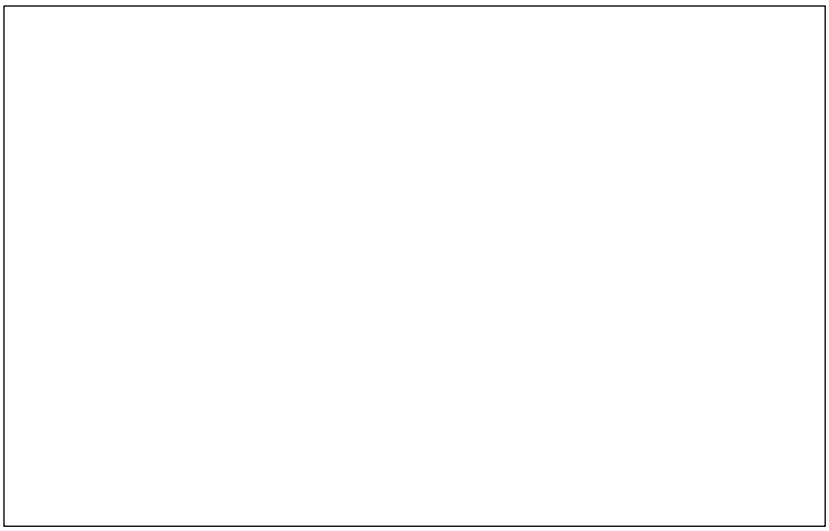
تنشأ المتغيرات من النمط **سلسلة** كغيرها من الأنماط المهيكلة، اي باستعمال المشيدات، فإضافة الى الوظائف المنتسبة كما يظهر ذلك في خطأ: لم يُعثر على مصدر المرجع، يوفر النمط سلسلة مشيدات عدة، و من ابرزها :

- سلسلة(سلسلة س) ويقوم هذا المشيد بإنشاء سلسلة انطلاقا. من سلسلة اخرى.

- سلسلة(طبيعي ط) ويقوم هذا المشيد بإنشاء سلسلة انطلاقا من قيمة طبيعية.
- سلسلة(حقيقي ح) ويقوم هذا المشيد بإنشاء سلسلة انطلاقا من قيمة حقيقية.
- سلسلة(منطقي م) ويقوم هذا المشيد بإنشاء سلسلة انطلاقا من قيمة منطقية، وتكون السلسلة المنتجة هي السلسلة "**صحيح**" او السلسلة "**خطأ**".

الكتابة المختصرة للمشيد سلسلة(سلسلة س)

في واقع الأمر، قليلا ما تُستعمل مشيدات النمط سلسلة على حالتهم التي ذكرناها سالفا، خاصة المشيد سلسلة(سلسلة س) و هذا راجع اساسا الى ان اي سلسلة مهما كانت طريقة انشائها، فهي ثابتة لا يمكن تغييرها، فبدل استعمال كتابة السطر 5 من خطأ: لم يُعثر على مصدر المرجع نكتب مباشرة ما في السطر 8، فالكتابة **"سلسلة ما"** هي في حقيقة الأمر تعادل (بل كتابة اخرى) للكتابة **شيد سلسة(سل1)** او للكتابة **شيد سلسة("سلسلة ما").**



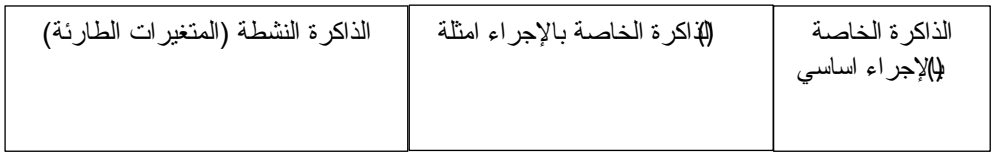
النص 2: امثلة تبين كيفية استعمال مشيدات النمط سلسلة

الكتابتين **"سلسلة ما"** و**شيد سلسة("سلسلة ما")** عبارتين متساويتين هدف كل واحدة منهما انشاء متغيرة ثابتة من النمط سلسلة، ونتيجة تقييمهما بعد انشاء المتغيرة، هي صلة ظرفية مرتبطة بالمتغيرة ويجب الاحتفاظ بها في صلة معروفة حتى نتمكن فيما بعد بالتعامل مع محتوى المتغيرة، ولذا تكتب الكتابتين عموما على يسار عملية الشحن (=)، كما يظهر في السطرين 5 و 8 من خطأ: لم يُعثر على مصدر المرجع.

5 – أمثلة توضيحية:

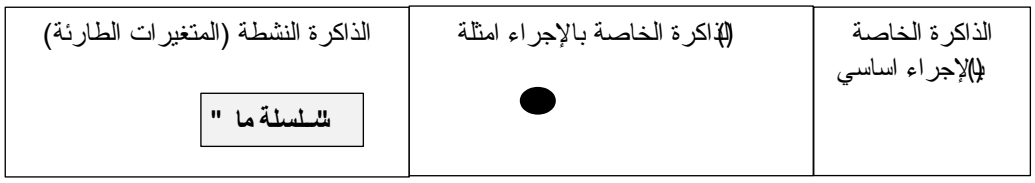
حتى نتمكن من الفهم العملي لما يحدث في الذاكرة عند انشاء المتغيرات من النمط **سلسلة** وما يحدث عندما نستعمل عملية الشحن (=)، نستعين فيما يلي ببعض الأشكال التي تعبر عن حالة الذاكرة اثر تنفيذ تعليمات خطأ: لم يُعثر على مصدر المرجع، ونلفت النظر الى اننا لن نظهر كيفية تمثيل سلاسل الحروف في الذاكرة باستعمال وحدات الذاكرة بل نستعمل فقط مستطيل تُكتب فيه قيمة السلسلة.

- المرحلة 1: بداية تنفيذ الإجراء اساسي(السطر 14)، ثم يطلب الإجراء اساسي تنفيذ الأجراء **امثلة()** ثم يتم الدخول الى الأجراء **امثلة()** لتنفيذ تعليماته. في هذه المرحلة تكون الذاكرة على خطأ: لم يُعثر على مصدر المرجع.



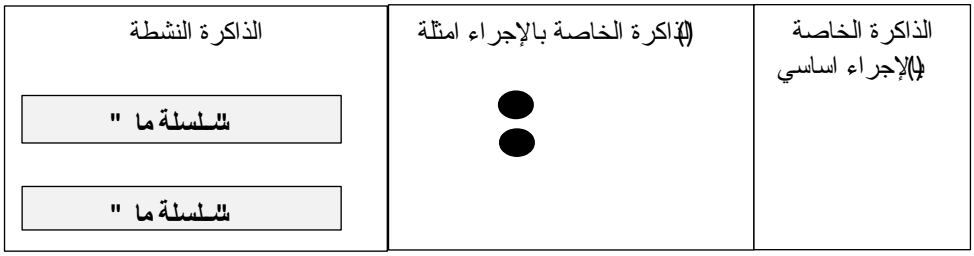
الشكل 4: حالة الذاكرة عند الشروع في تنفيذ الإجراء امثلة() من النص 1

- **المرحلة 2:** على مستوى الإجراء **امثلة()** يتم اولا انجاز التصريح بقيمه اوليه الموجود في السطر 3 من خطأ: لم يُعثر على مصدر المرجع، أي: **سلسلة سل1 = "سلسلة ما"؛** فتصبح الذاكرة على خطأ: لم يُعثر على مصدر المرجع، فالصلة **سل1** تنشأ في الذاكرة المحلية (او الخاصة) للإجراء **امثلة()** والمتغيرة الحاملة للسلسلة **"سلسلة ما"** تنشأ في الذاكرة النشطة، ونذكر ان المتغيرات من النمط **سلسلة** متغيرات مهيكلة تنشأ في الذاكرة النشطة.

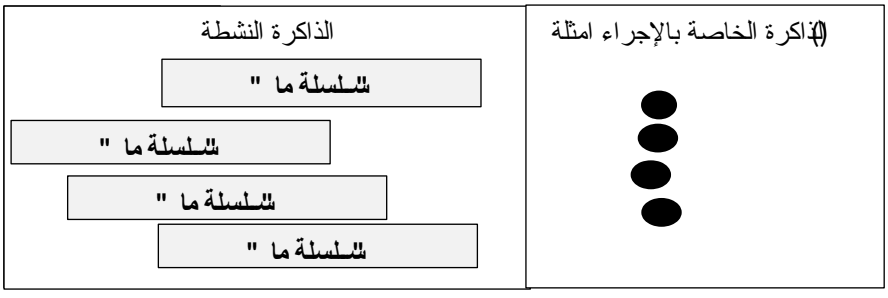


الشكل 5: حالة الذاكرة بعد تنفيذ تعليمة السطر 3 من الإجراء امثلة() من النص 1

- المرحلة 3: ينجز التصريح: **سلسلة سل2 = شيد سلسة(سل1)؛** (السطر 5 من خطأ: لم يُعثر على مصدر المرجع)، وتصبح الذاكرة على خطأ: لم يُعثر على مصدر المرجع، ونلاحظ هنا ان المتغيرتين مستقلتين بعضهما البعض مع كونهما يحتويان على نفس القيمة، فقد تم انشاء المتغيرة الثانية (المشار اليها بالصلة سل2) انطلاقا مما تحتويه المتغيرة الأولى (المشار اليها بالصلة سل1).

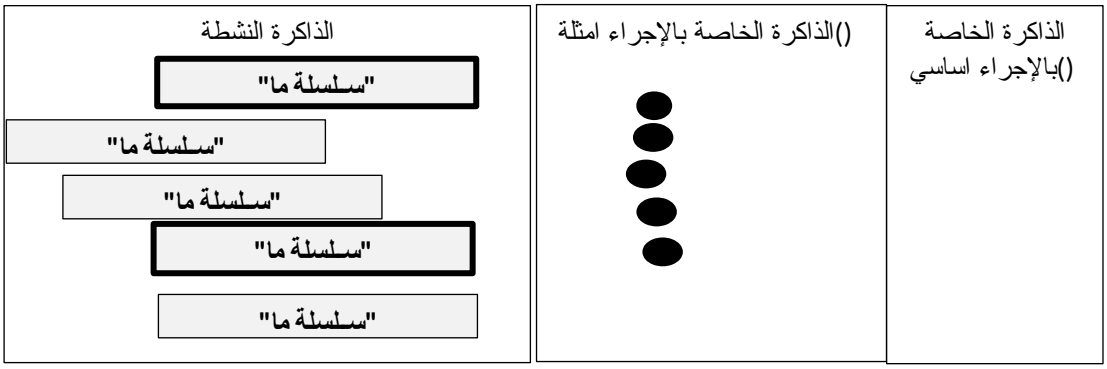


الشكل 6: حالة الذاكرة بعد تنفيذ تعليمة السطر 5 من الإجراء امثلة() من النص 1



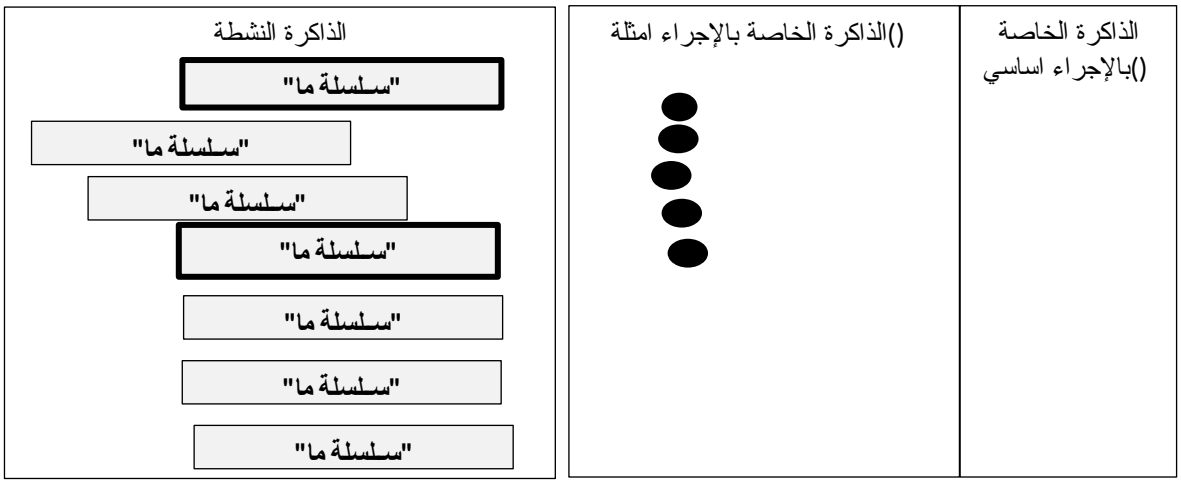
الشكل 7: حالة الذاكرة بعد تنفيذ تعليمتي السطر 7 و8 من الإجراء امثلة() من النص 1

- المرحلة 4: تنفيذ التعليمتين **سلسلة سل3="سلسلة ما"** ثم **سلسلة سل4 = "سلسلة ما"**، (السطرين 7و8 من خطأ: لم يُعثر على مصدر المرجع)، وتصبح الذاكرة على خطأ: لم يُعثر على مصدر المرجع.
- المرحلة 4: تنفيذ التعليمة **سلسلة سل5 = سل 4 = سل 1 = "سلسلة ما"** (السطر 9 من خطأ: لم يُعثر على مصدر المرجع): بعد تنفيذ التعليمة تصبح الذاكرة على خطأ: لم يُعثر على مصدر المرجع، ونلاحظ من خلال هذا الشكل، ان المتغيرتين اللتين كانتا مشار اليهما بالصلتين سل1 و سل4، اصبحتا طليقتين لا يمكن الوصول اليهما لعدم وجود اي صلة مرتبطة بهما.



الشكل 8 : حالة الذاكرة بعد تنفيذ تعليمة السطر 9 من الإجراء امثلة() من النص 1

- المرحلة 5: بعد تنفيذ التعليمتين **سل4 = "سلسلة ما"** ثم **سل5 = "سلسلة ما"** (السطرين 10و 11 من خطأ: لم يُعثر على مصدر المرجع) تصبح الذاكرة على خطأ: لم يُعثر على مصدر. المرجع.



الشكل 9 : حالة الذاكرة بعد تنفيذ تعليمتي السطرين 10 و 11 من الإجراء امثلة() من النص 1

5- 1 خلاصة ما ورد في الأشكال:

في حقيقة الأمر هذه الخلاصة انما هي تذكير لما ورد من قبل، فمن خلال هذه الأشكال نرى ان كتابة السلسلة الثابتة (سلسلة من الأحرف يحدهما حرف الاقتباس (") هو في الحقيقة طلب انشاء متغيرة ثابتة من النمط سلسلة وتعادل الكتابة التي يظهر فيها المشيد **سلسلة(سلسلة سل).**

5 – 2 السلسلة الفارغة:

هي كتابة مكونة فقط من الحدين الممثلين بحرف الاقتباس(")، فالكتابة "" هي السلسلة الفارغة، اما الكتابة التي نجد فيها حرف الفراغ كالكتابة " "، فهي سلسلة غير فارغة، مكونة بأحرف الفراغ، ولو كان حرفا فارغا واحدا.

تنبيه و تذكير: الكتابة 'ق' لا علاقة لها بالكتابة "ق"، فالأولى قيمة ثابتة من النمط حرف والثانية سلسلة ثابتة، ففي الأولى نجد رمز لحرف واحد فقط، فلا يمكن ان يضع رمزين او اكثر بين حرف الاقتباس الفردي(')، فمثلا الكتابة 'عمر' خاطئة، وكما علما من قبل لكل حرف رمز خاص به.

6 – عملية إلصاق سلاسل الحروف ورمزها الحرف +:

عندما يكون احد معاملي العملية + من النمط سلسلة، تقوم هذه العملية باعتبار المعامل الآخر على انه ايضا من النمط سلسلة:

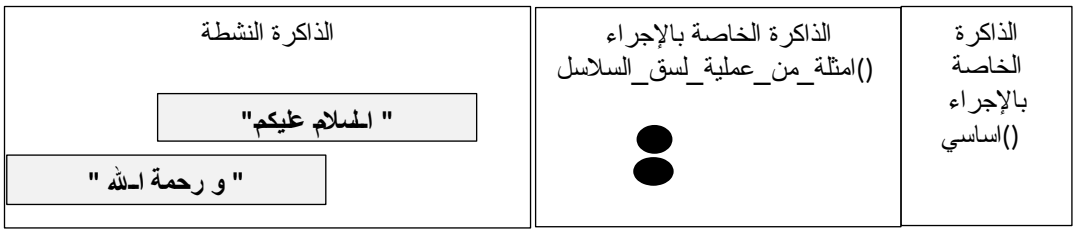
- فان كان المعامل الآخر فعلا من النمط سلسلة تقوم هذه العملية بإنتاج سلسلة تكون فيها السلسلتان متلاصقتين كما يظهر ذلك في الأشكال، من خطأ: لم يُعثر على مصدر المرجع الى خطأ: لم يُعثر على مصدر المرجع، التي تبين نتيجة تنفيذ التعليمات من السطر 3 الى السطر 7 من خطأ: لم يُعثر على مصدر المرجع.
- اما اذا كان المعامل الثاني من نمط غير النمط سلسلة، فإن العملية + تقوم اولا بتحويل قيمة المعامل الثاني الى قيمة من النمط سلسلة، ثم تقوم بإلصاق السلسلتين كما تبينه الأشكال من خطأ: لم يُعثر على مصدر المرجع الى خطأ: لم يُعثر على مصدر المرجع التي تبين نتيجة تنفيذ التعليمات من السطر. 11 الى السطر. 16 من خطأ: لم يُعثر على مصدر. المرجع.

الجزء

النص 3: امثلة تبين كيفية استعمال عملية الصاق السلاسل

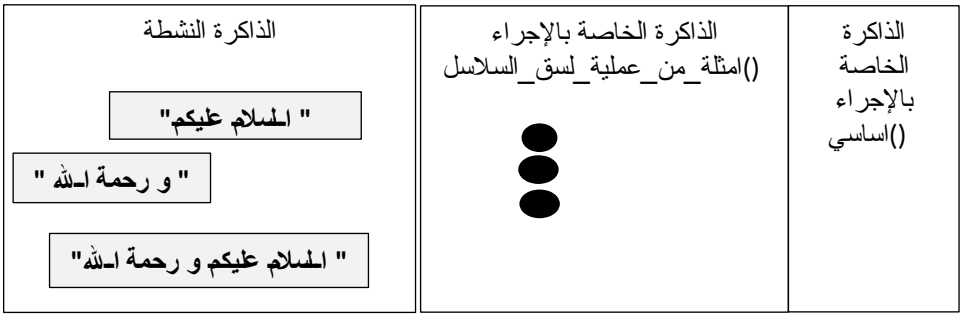
7 دراسة نص مع أشكال تبين حالة الذاكرة بعد تنفيذ تعليمات خطأ: لم يُعثر على مصدر المرجع:

بعد الدخول في الإجراء **اساسي** وطلب تنفيذ الإجراء **امثلة_من_عملية_لسق_السلاسل()**، وبعد الشروع في تنفيذ الإجراء **امثلة_من_عملية_لسق_السلاسل()** واثر تنفيذ تعليمتي السطرين 3 و 4 تصبع الذاكرة على الصفة التي يبينها خطأ: لم يُعثر على مصدر المرجع، وفي هاذين السطرين يتم التصريح بمتغيرتين من النمط سلسلة بقيمة اولية مشار اليهما بصلتين هما: **سلام** و **رحمة**، وبما ان الصلات في حد ذاتها معلومات بدائية، فإن موقعها يكون في الذاكرة الخاصة بالإجراء **امثلة_من_عملية_لسق_السلاسل()**، اما المتغيرات الحاملة للقيم **"السلام عليكم"** و **"ورحمة الله"**، فمحلها الذاكرة النشطة لكونها معلومات مهيكلة.

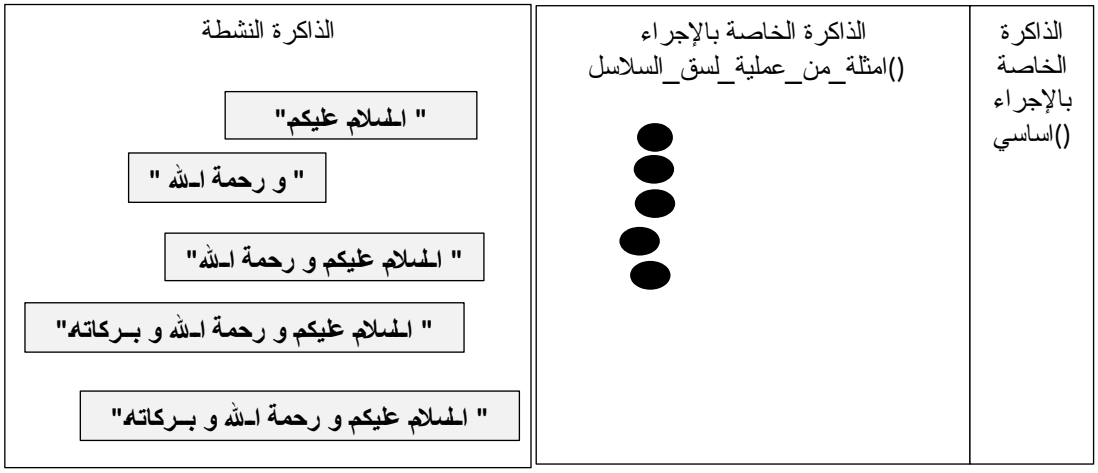


الشكل 10 : حالة الذاكرة بعد تنفيذ تعليمات النص 3 من السطر 3 الى السطر 4

يبين خطأ: لم يُعثر على مصدر المرجع حالة الذاكرة اثر تنفيذ تعليمة السطر 5، إذ تُنشأ متغيرة من النمط **سلسلة** مشار اليها بالصلة **سلام_و_رحمة** وتحتوي هذه المتغيرة على نتيجة الصاق القيمة **" و رحمة الله "** (التي نسخت من السلسلة المشار اليها بالصلة **رحمة)** بالقيمة **"السلام عليكم "** (التي نسخت من السلسلة المشار اليها بالصلة **سلام)**.



الشكل 11 : حالة الذاكرة بعد تنفيذ تعليمة السطر 5 من النص 3



الشكل 12 : حالة الذاكرة بعد تنفيذ تعليمتي السطرين 6 و 7 من النص 2

تنفيذ السطرين 6 و 7 يحولان الذاكرة الى ما يبينه خطأ: لم يُعثر على مصدر المرجع، فنرى ظهور متغيرتين تحتويان على نفس القيمة، اي ان تعليمة السطر 6 تؤدي الى نفس نتيجة تعليمة السطر 7، ففي السطر 6 استعملنا النتيجة التي تحصلنا عليها سابقا من خلال تعليمة السطر 5، اما في السطر 7 استعملنا المتغيرتين **سلام** و **رحمة (السطرين 3 و 4)** دون استعمال ما تحصلنا عليه في السطر 5.

تنفيذ تعليمات الاسطر من 9 الى 11 (خطأ: لم يُعثر على مصدر المرجع)، ففي السطر 11 نرى عبارة تحتوي عمليات تنتمي الى ثلاثة اصناف: عملية شحن (=) واحدة، ثلاثة عمليات جمع (+) وعملية ضرب(*) واحدة، وكما علمناه من قبل من سلم أولويات العمليات(الفصل 12، الجزء الأول)، فان الأولوية في بداية تقييم العبارة تكون لعملية الضرب (*)، فأول عبارة جزئية تقيم هي: **سعر_الوحدة * الكمية**، ونتيجة تقييم هذه العبارة هي القيمة الحقيقية 52958384,70 التي وضعت في متغيرة ظرفية من النمط حقيقي نسميها **مظ1**، وبعد هذا التقييم الأول للعبارة الجزئية **سعر_الوحدة * الكمية** تصبح عبارة السطر 11 على الصيغة التالية:

السعر_الكلي = "السعر الإجمالي للكمية "+ الكمية + " هو: " +	**مظ1**

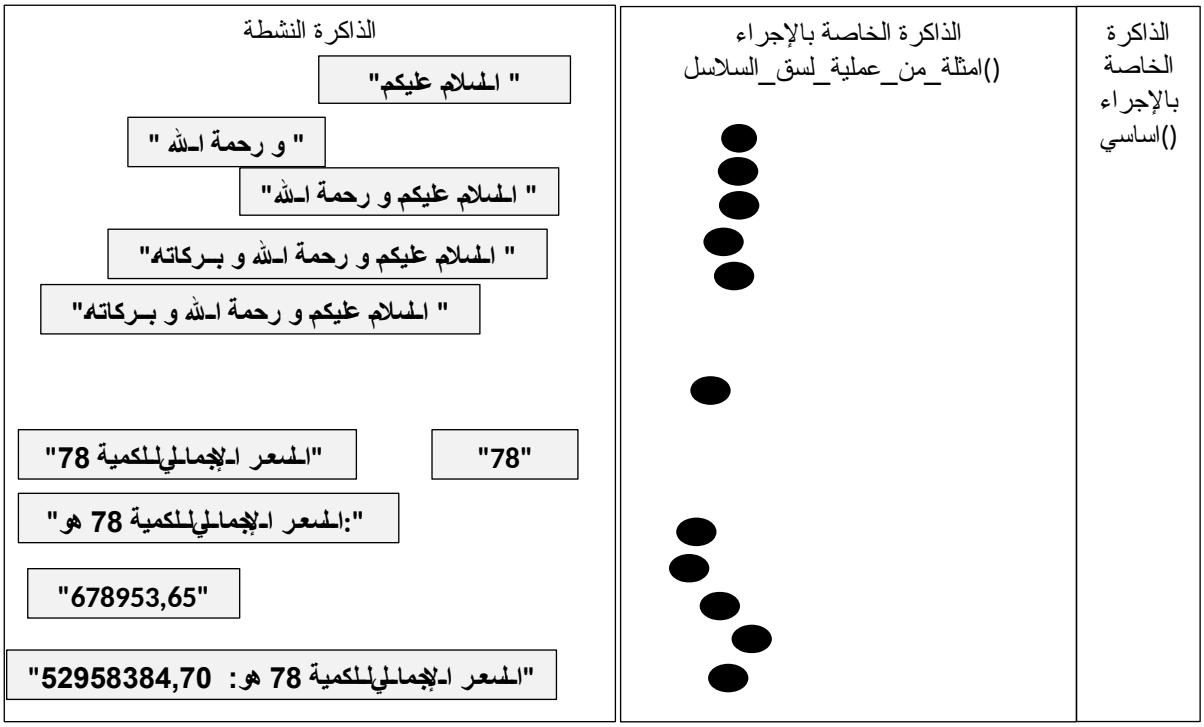
في هذه الصيغة الجديدة لعبارة السطر 11، نجد عملية شحن (=) واحدة وثلاثة عمليات جمع (+)، وعلمنا ان في مثل هذه الحالة تكون عملية الجمع (+) اولى بالتنفيذ، واذا رجعنا الى سلم الأولويات(الفصل 12، الجزء الأول)، نرى ان من بين العمليات الثلاث للجمع، فان العملية التي تنتخب للتنفيذ هي تلك الموجودة في اقصى اليمين، اي ان العبارة الجزئية الأولى بالتقييم هي العبارة: **"السعر الإجمالي للكمية " + الكمية.**

في العبارة الجزئية **"السعر الإجمالي للكمية " + الكمية** نرى ان المعامل الأيمن لعملية الجمع ينتمي الى النمط سلسلة وان المعامل الأيسر من النمط طبيعي، وكما علمناه من قبل فعملية الجمع (+) تصبح هنا عملية الصاق سلسلة بأخرى، وحتى تتم العملية، تقوم اولا عملية جمع السلاسل (او إلصاق السلاسل) بتحويل القيمة الموجودة في المتغيرة **الكمية**، وهي 78، الى سلسلة مكونة من الحرفين '8' و

'7'، اي السلسلة "78"، وبعد التحويل تكون السلسلة الناتجة، اي "78" مرتبطة بصلة ظرفية، لا يعلمها الا منفذ الخوارزم، واشرنا الى هذه الصلة الظرفية في خطأ: لم يُعثر على مصدر المرجع بالاسم **صظ 1،** وبعد الحصول على السلسلة **صظ1** تصبح العبارة الجزئية **"السعر الإجمالي للكمية " + الكمية** على الشكل **"السعر الإجمالي للكمية "+ صظ1،** وتقييم هذه العبارة الأخيرة ينتج السلسلة **"السعر الإجمالي للكمية 78"** مرتبطة بصلة ظرفية سميناها **صظ2** في خطأ: لم يُعثر على مصدر المرجع، وبعد الانتهاء من تقييم العبارة الجزئية، تصبح العبارة الأصلية (السطر 11) على الشكل التالي:

السعر_الكلي =	صظ2	+ " هو: " +	مظ1

وفي هذه الصيغة الحديثة للعبارة الأصلية (السطر 11) نجد عملية شحن(=) واحدة وعمليتي جمع (+) ، واستنادا الى سلم اولويات العمليات، فان العبارة الجزئية المرشحة للتنفيذ هي العبارة: **صظ2 + " هو:** "، و في هذه العبارة نرى ان كلا المعاملين من النمط سلسلة، فهنا تنفذ مباشرة عملية لصق السلسلتين، وتكون نتيجة تنفيذ عملية الجمع السلسلة **"السعر الإجمالي للكمية 78 هو:"، و هذه القيمة** مرتبطة بصلة ظرفية سميناها. **صظ3** في خطأ: لم يُعثر على مصدر المرجع.



الشكل 13 : حالة الذاكرة اثر تنفيذ تعليمات الاسطر من 9 الى 11 من النص 3

بعد تقييم هذه العبارة الجزئية الأخيرة، تصبح العبارة الأصلية على الشكل التالي:

السعر_الكلي =	صظ3	+ مظ1

وفي هذه العبارة نجد عملية شحن واحدة وعملية جمع واحدة، واستنادا الى سلم اولويات العمليات، فان العبارة الجزئية **صظ3 + مظ1** هي العبارة المرشحة للتنفيذ، ونرى ان المعامل الأيمن

من النمط **سلسلة** والمعامل الأيسر من النمط **حقيقي**، وهنا تبدأ عملية الجمع بتحويل القيمة الحقيقية الموجودة في المتغيرة **مظ1**، اي **52958384,70** ، الى السلسلة "**52958384,70**"، وهذه السلسلة، كما نراه من خلال خطأ: لم يُعثر على مصدر المرجع، مرتبطة بالصلة الظرفية **صظ4**. وبعد تحويل محتوى المتغيرة **مظ1** الى السلسلة **صظ4**، تصبح العبارة الجزئية الأخيرة على الشكل:

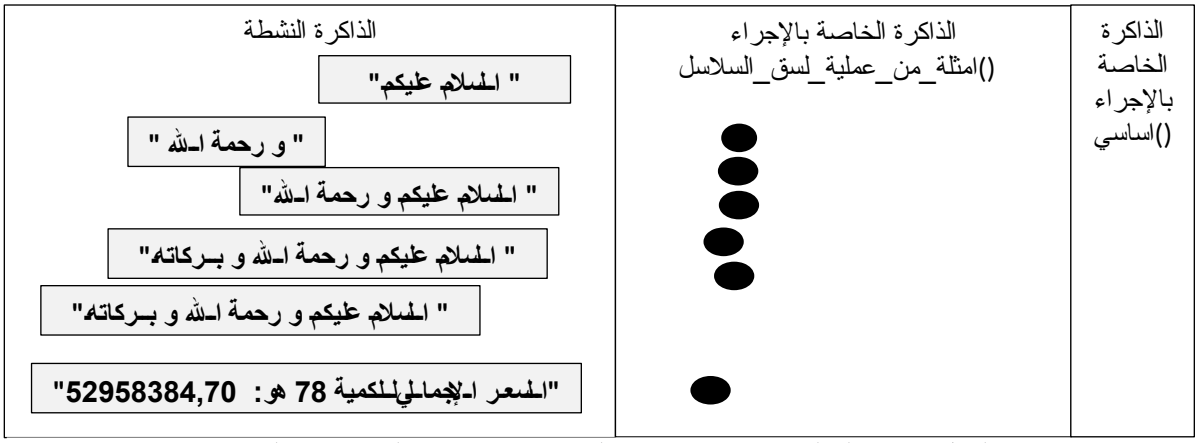
السعر_الكلي =	صظ3	+ صظ4

في هذه العبارة الأخيرة كلا المعاملين لعملية الجمع من النمط سلسلة، فيتم اذا الصاق السلسلتين وانتاج سلسلة مرتبطة بالصلة **صظ5** وقيمتها: "**السعر الإجمالي للكمية 78 هو: 52958384,70**" ، كما نراه في خطأ: لم يُعثر على مصدر المرجع، تصبح العبارة الأصلية (السطر 11) على الشكل التالي:

السعر_الكلي =	صظ5

بتقييم هذه الصيغة الأخيرة للعبارة الأصلية (صيغة فيها عملية واحدة فقط) ينته تقييم العبارة الأصلية، وفي هذا التقييم الأخير تقوم عملية الشحن بشحن الصلة **السعر_الكلي** بالمعلومات الموجودة في الصلة **صظ5**، اي ان الصلتين تصبحان مرتبطتان بنفس المتغيرة ، اي تلك التي تحتوي على السلسلة "السعر الإجمالي للكمية 78 هو: 52958384,70".

تنبيه هام جدا: نذكر انه بعد اتمام تقييم العبارة الأصلية تُمح من الذاكرة كل المتغيرات والصلات الظرفية، و كذلك كل المتغيرات الطارئة التي ليس لها ارتباط مع اي صلة، وهكذا تصبح الذاكرة على حالة خطأ: لم يُعثر على مصدر المرجع.



الشكل 14 : حالة الذاكرة بعد اتمام تنفيذ تعليمات الاسطر من 9 الى 11 من النص 3

- تنفيذ تعليمات الاسطر من 14 الى 17(خطا: لم يَعثر على مصدر المرجع): من خلال هذه التعليمات سوف نرى
  - كيفية الحصول على قيمة من النمط سلسلة انطلاقا من قيمة ليست من النمط سلسلة (اي اننا نريد ان نرى قيمة ما على الصورة سلسلة).

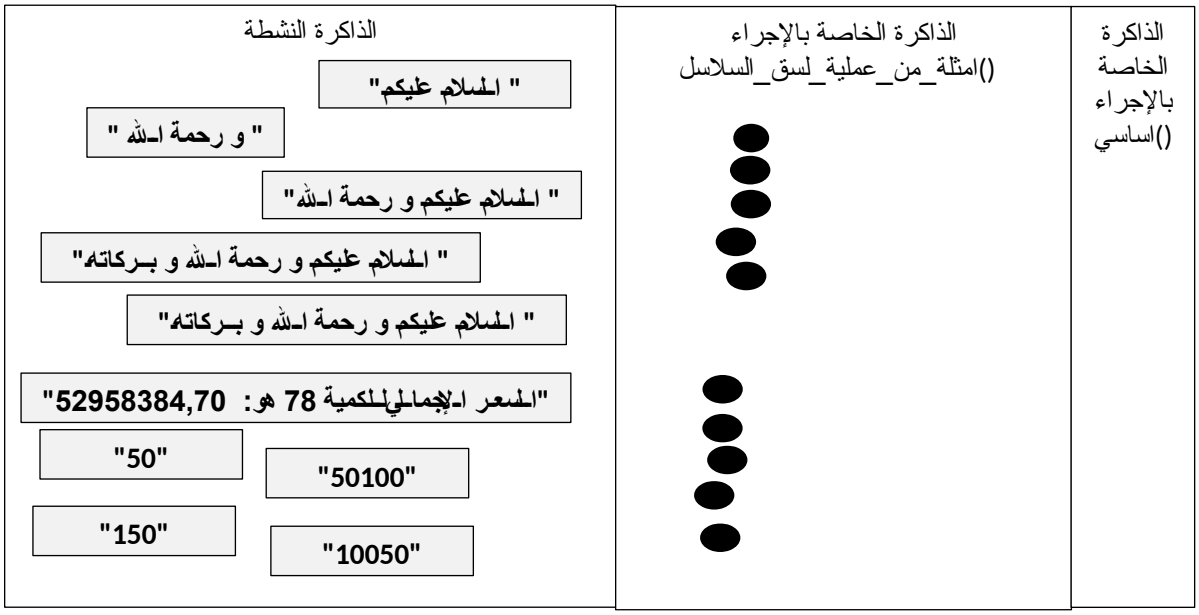
الجزء الثاني: الفصل الثاني عشر: الأوعية: سلاسل الحروف

- أثر اولويات العمليات على نتيجة تقييم عبارات فيها معامل من النمط سلسلة,

للحصول على الصفة **سلسلة** لقيمة ما، هناك على الأقل طريقتان: طريقة مباشرة واخرى غير مباشرة. في الطريقة المباشرة، اذا اردنا ان نتحصل على الصيغة سلسلة لقيمة ما، نستعمل مشيدا من المشيدات التي يوفرها النمط **سلسلة**، فمثلا اذا اردنا الصيغة سلسلة لقيمة حقيقية نستعمل المشيد **سلسلة(حقيقي ح)**، واذا اردنا الصيغة سلسلة لقيمة منطقية نستعمل المشيد **سلسلة(منطقي م)**، واذا اردنا الصيغة سلسلة لقيمة طبيعية نستعمل المشيد **سلسلة (طبيعي ط)** لإنتاج الصورة من النمط سلسلة للقيمة الطبيعية **ط**.

في الطريقة الغير مباشرة، فبدل من استعمال المشيدات المتاحة، نلجأ الى خصائص عملية الجمع التي رأيناها وهذه الخاصية هي قدرة عملية الجمع على انتاج الصورة سلسلة لأي نمط مختلف عن النمط سلسلة، والسطر 14 يمثل هذه الطريقة الغير مباشرة، ففي العبارة الجزئية **""+ 50** لعبارة السطر 14، نرى ان احد معاملي عملية الجمع هي السلسلة الفارغة، وهي من النمط سلسلة، وفي هذه الحالة فان عملية الجمع تصبح عملية لصق سلسلة بأخرى، فتقوم اولا بتحويل القيمة الطبيعية 50 الى السلسلة "50" ثم تلصقها بالسلسلة الفارغة التي لا تحتوي على اي حرف، وهكذا لن يتم اضافة اي حرف الى السلسلة "50"، وبهذه الطريقة يتم انتاج الصورة سلسلة للقيمة 50.

في السطر 15 نرى من خلال اسم الصلة **سلسلة مئة_و_خمسون_1** وكأن كاتب العبارة اراد الحصول على السلسلة "150"، لكن في واقع الأمر، عملية الجمع لا تقوم بعملية جمع القيمة الطبيعية 100 بالقيمة الطبيعية 50، وانما تقوم بإلصاق السلسلة "50" بالسلسلة "100" التي تنتجها عملية تحويل القيمة 100، وهذا قبل بداية عملية اللصق، والنتيجة الحقيقية هي السلسلة "50100" كما يظهر ذلك في خطأ: لم يُعثر على مصدر المرجع.



الشكل 15 : حالة الذاكرة بعد تنفيذ تعليمات الاسطر من 14 الى 16 من النص 3

في السطرين 16 و 17 نرى اثر أولوية تنفيذ العمليات على نتائج تقييم العبارات، ففي السطرين نرى من خلال اسم الصلتين **سلسلة مئة_و_خمسون_2** و**سلسلة مئة_و_خمسون_3** وكأن كاتب العبارة اراد الحصول على السلسلة "150"، في عبارة السطر 16 يتم هذا الأمر، ففي تقييم عبارة السطر 16 يتم انتخاب العبارة 100 + 50 للتقييم، وفي هذه العبارة عملية الجمع هي عملية جمع قيمتين طبيعيتين، فالنتيجة تكون 150، وبعد هذا التقييم تصبع العبارة الأصلية على الشكل **مئة_و_خمسون_2 = 150 + ""**، و لتقييم هذه الأخيرة تنتخب العبارة الجزئية **150 + ""** وكما رأيناه من قبل، تكون السلسلة "150" هي نتيجة تقييم عبارة السطر 16.

اما في السطر 17، فالنتيجة مختلفة، فأول عبارة جزئية يتم انتخابها للتقييم هي العبارة الجزئية "" + 50، وهنا عملية الجمع تنتج السلسلة "50"، فتصبح العبارة الأصلية على الشكل **مئة_و_خمسون_3 = "50" + 100** ، ثم تنتخب للتقييم العبارة الجزئية "50" + 100، و تكون نتيجتها السلسلة "10050".

8 – ادراج الحروف الخاصة في سلاسل الحروف:

في بعض الأحيان نريد ادراج بعض الحروف الخاصة في سلسلة ما، وفي اطار النمط سلسلة، نعني بالحروف الخاصة:

- تلك التي ليس لها رسم خاص بها نراه على الشاشة عند الضغط على بعض ازرار لوحة الحروف، بل نرى مفعولها، كحرف الرجوع الى السطر، او حرف الانتقال الى الصفحة التالية او حرف الحذف الى آخره من الحروف الغير المرئية (راجع الفصل 10،الجزء 1).
- الأحرف التي لها رسم ولها ايضا دورا خاصا في كتابة سلاسل الحروف، وفي هذا الشأن هناك حرف خاص واحد هو حرف الاقتباس المزدوج (") الذي يستعمل لتحديد سلسلة من الحروف.

لأدراج الحروف الخاصة التي ليس لها رسم، نستعمل رموزها المركبة، وهي الرموز التي تبدأ بالحرف / (راجع الفصل 10،الجزء 1)، فمثلا، رمز حرف الرجوع الى اول السطر هو **/ع**، ورمز حرف الانتقال الى السطر التالي هو **/س** ورمز حرف الحذف هو **/م**.

لأدراج الحروف الخاصة التي لها رسم ولها دور خاص في كتابة السلاسل، نستعمل رسم الحرف مسبوقا بالحرف '/'، فنكتب **/"** بدل من ".

فمثلا، يقوم خوارزم خطأ: لم يُعثر على مصدر المرجع بإخراج على الشاشة ما نراه في خطأ: لم يُعثر على مصدر المرجع، ففي هذا الخوارزم لم نستعمل الا سلسلة واحدة أُدرج فيها مرتين الحرف /" و ثلاث مرات الحرفين '/ع' و '/س'.

النص 4: ادراج الأحرف الخاصة في سلاسل الحروف

```
! "يا "احمد
السلام
! عليكم
```

الشكل 16 : نتيجة تنفيذ خوارزم النص 3

الفصل الثالث عشر

الأوعية: اهم وظائف النمط سلسلة

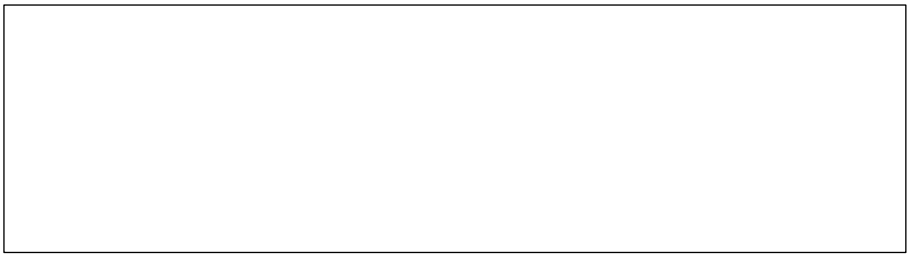
1 – مقدمة:

يمتاز النمط سلسلة باحتوائه على عدد كبير من الوظائف المنتسبة هدفها اجراء مختلف العمليات على السلسلة واخراج النتائج المرجوة دون المساس بالسلسلة في حد ذاتها، وتستعمل هذه الوظائف في خوارزميات مختلفة، واكثرها استعمالا الخوارزميات التي تقارن بين سلسلتين من الحروف والخوارزميات الموجهة لمعالجة النصوص، وفيما يلي ولأهميتها البالغة نبدأ بعمليات مقارنة سلاسل الحروف.

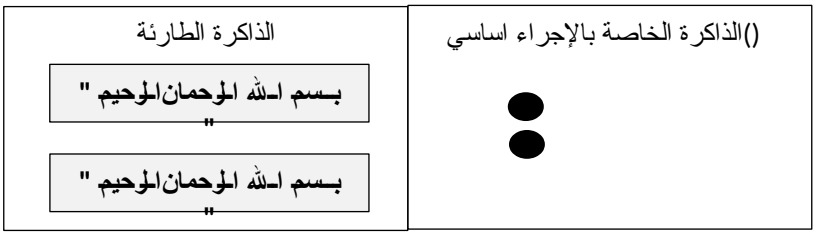
المقارنة بالتساوي بين سلسلتين من الحروف: الوظيفة: منطقي تساوي (سلسلة س)

في الفصل السابق، علمنا ان المتغيرات من النمط سلسلة ليست بالمتغيرات البدائية وانما هي متغيرات مهيكلة بشكل منتظم يشبه الجداول ذو بعد واحد، وارتكازا على هذا فإنه لا يمكن لنا استعمال عملية المقارنة على التساوي التي تستعمل مع المتغيرات البدائية ورمزها ==، ويجب ان نحذر عند استعمال هذه العملية مع المتغيرات المهيكلة، فالنتيجة يمكن ان تكون عكس المنتظر، فعلى سبيل المثال، يظهر في السطر 2 من خطأ: لم يُعثر على مصدر المرجع التصريح بمتغيرتين من النمط **سلسلة**، ويظهر خطأ: لم يُعثر على مصدر المرجع ما يحدثه التصريحين في الذاكرة، ونرى جليا ان الصلتين **بسملة1** و **بسملة2** مرتبطتين بمتغيرتين مختلفتين تحتويان على السلسلة **"بسم الله الرحمان الرحيم"**، وبما ان الصلة تحتوي على معلومة تواجد المتغيرة المرتبطة بها في الذاكرة، فكلتا المتغيرتين الطارئتين المحتويتين على السلسلة **"بسم الله الرحمان الرحيم"** موجودتان في مكانين مختلفين، اي ان قيمة الصلة **بسملة1** مختلفة عن قيمة الصلة **بسملة2** ، وفي مثل هذه الحالة تكون القيمة **خطأ** هي نتيجة تقييم العبارة **بسملة1 == بسملة 2** (السطر 3 من خطأ: لم يُعثر على مصدر المرجع و السطر 1 من خطأ: لم يُعثر على مصدر المرجع)، فالمقارنة باستعمال العملية == تأخذ بعين الاعتبار محتوى الصلتين وليس محتوى السلسلتين المشار اليهما بالصلتين.

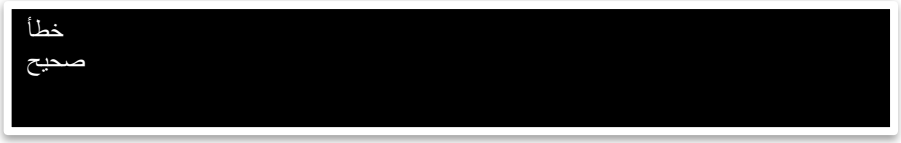
اما نتيجة العبارة **بسملة1 == بسملة 2** (السطر 4 من خطأ: لم يُعثر على مصدر المرجع و السطر 2 من خطأ: لم يُعثر على مصدر المرجع) فهي **صحيح**، و هذه النتيجة جاءت بعد تنفيذ تعليمة السطر 3 التي تجعل الذاكرة على الحالة التي يظهرها خطأ: لم يُعثر على مصدر المرجع، وفي هذا الشكل الصلتين مرتبطتين بنفس **السلسلة**، اي في حقيقة الأمر لا توحي الكتابة بتاتا المقارنة بالتساوي بين متغيرتين مختلفتين بل بمتغيرة مع نفسها.



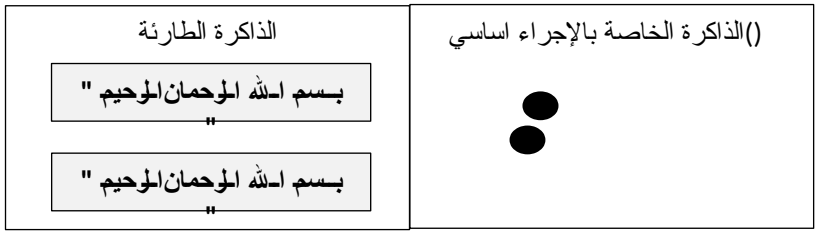
== النص 1: مقارنة سلاسل الحروف باستعمال عملية التساوي



الشكل 1 : حالة الذاكرة بعد تجسيد تصريح السطر 1 من الإجراء اساسي() من النص 1



الشكل 2 : نتيجة تنفيذ خوارزم النص 1



الشكل 3 : حالة الذاكرة بعد تنفيذ تعليمة السطر 4 من الإجراء اساسي() من النص 1

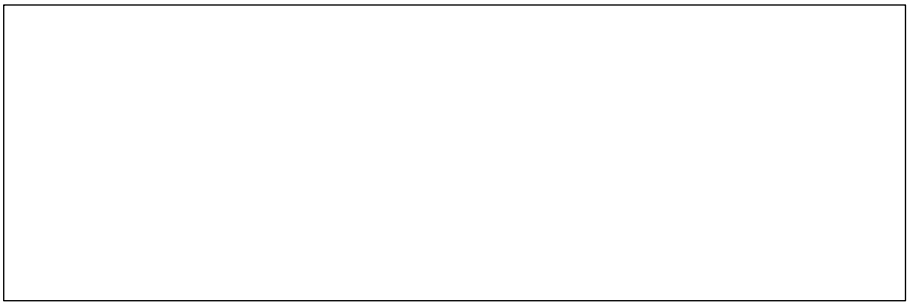
بما ان عملية المقارنة بالتساوي (و رمزها ==) لا تصلح لمقارنة سلسلتين من الحروف، فما هو السبيل المتاح للمقارنة بالتساوي بين سلسلتين.

في اطار المتغيرات المهيكلة ومنها المتغيرات من النمط سلسلة، تتم عملية المقارنة باستعمال وظيفة منتسبة تحمل عموما الاسم **تساوي** وهي من النمط منطقي، فمثلا بالنسبة للمتغيرات من النمط سلسلة يكون رأس هذه الوظيفة على الشكل التالي:

منطقي تساوي (سلسلة س)

و تقوم هذه الوظيفة بمقارنة السلسلة التي تنفذ من خلالها الوظيفة **تساوي** (مثلا السلسلة **بسملة1** في السطر 3 من خطأ: لم يُعثر على مصدر المرجع) مع السلسلة التي توضع في مدخلها (مثلا السلسلة **بسملة2** في السطر 3 من خطأ: لم يُعثر على مصدر المرجع) ، وفي خطأ: لم يُعثر على مصدر. المرجع نجد بعض الأمثلة التي تبين كيفية استعمال هذه الوظيفة المنتسبة وفي خطأ: لم يُعثر على مصدر المرجع نرى نتيجة تنفيذ خوارزم خطأ: لم يُعثر على مصدر المرجع.

نلاحظ في السطر الأخير من خطأ: لم يُعثر على مصدر المرجع ان النتيجة هي خطأ، وهذا راجع الى كون السلسلة التي ذكرت في مدخل الوظيفة **تساوي** تختلف عن الأخرى كونها تحتوي على الأحرف الممثلة للحركات كحركة الكسر في اول حرف والفتحة فوق آخر راء، فحرف بحركة ما مختلف تماما عن حرف بلا حركة او حركة مغايرة، وحرف بحركة مكون في حقيقته من حرفين: الحرف والحركة.



(..)النص 2: مقارنة سلاسل الحروف باستعمال الوظيفة المنتسبة تساوي



الشكل 4 : نتيجة تنفيذ خوارزم النص 2

2- المقارنة بين سلسلتين من الحروف: الوظيفة المنتسبة: طبيعي قارن(سلسلة س)

تهدف المقارنة بين سلسلتين من الحروف الى معرفة ترتيب سلسلة اولى بالنسبة لسلسلة ثانية على اساس ترتيب الحروف، هل الأولى اسبق (وكثيرا ما نستعمل الكلمة اصغر) من الثانية، هل الأولى تساوي الثانية، هل الأولى تتأخر عن الثانية (وكثيرا ما نستعمل الكلمة اكبر) ، فمثلا السلسلة "احمد" تُرَتب قبل السلسلة "محمد"، اي ان السلسة "احمد" اصغر من السلسلة "محمد"، لأن الحرف الأول 'ا' اصغر من الحرف 'م' في الترتيب الأبجدي وكذلك من ناحية القيمة الممثلة للحرف في الذاكرة، فكل حرف في حقيقته ممثل بقيمة ما في الذاكرة.

عندما تنفذ الوظيفة على سلسلة ما وهي التي سميناه بالسلسلة الأولى، وتوضع سلسلة ثانية في مدخلها، نتحصل على احدى النتائج التالية

- القيمة الطبيعية 1 وتعني ان السلسلة الأولى اكبر من السلسلة الثانية
- القيمة الطبيعية -1 و تعني ان السلسلة الأولى اصغر من السلسلة الثانية

- القيمة الطبيعية 0، وتعني ان السلسلة الأولى تساوي السلسلة الثانية

و يبين خطأ: لم يُعثر على مصدر المرجع كيفية استعمال الوظيفة **قارن** في كتابة الخوارزميات ويظهر خطأ: لم يُعثر على مصدر المرجع نتيجة تنفيذ خوارزم خطأ: لم يُعثر على مصدر. المرجع، وهدف خوارزم خطأ: لم يُعثر على مصدر المرجع هو المقارنة بين سلسلتين من الحروف يُدخلهما المستعمل وفيما يلي السلوك العام لخوارزم خطأ: لم يُعثر على مصدر المرجع.

(..)النص 3: مقارنة سلاسل الحروف باستعمال الوظيفة المنتسبة قارن

- بعد التصريح بصلتين من النمط سلسلة في السطر 2، يطلب الخوارزم (الإجراء اساسي) من المستعمل ادخال جملة اولى (اي سلسلة من الحروف)، أو ادخال الكلمة "**نهاية**" ان اراد المستعمل انهاء عمل الخوارزم (السطر. الأول من خطأ: لم يُعثر على مصدر المرجع)
- بعد الحصول على الجملة الأولى (السطر. الثاني من خطأ: لم يُعثر على مصدر المرجع)، ينتقل الخوارزم الى تعليمة التكرار. التي يبدأ رأسها في السطر 5 من خطأ: لم يُعثر على مصدر. المرجع، و يحتوي رأس تعليمة التكرار على عبارة هدفها التحقق من محتوى السلسلة **جملة1** التي تمثل الجملة الأولى، فان كانت السلسلة **جملة1** تساوي السلسلة الثابتة

"نهاية"، تنته تعليمة التكرار وينتقل التنفيذ الى تعليمة السطر 23، ولمعرفة هل محتوى السلسلة **جملة1** يساوي السلسلة **"نهاية"**، نستعمل الوظيفة المنتسبة **تساوي** ، واستعمالنا لعملية النفي الممثلة بحرف التعجب ! يجعل معنى العبارة المنطقية في السطر 5 كالتالي: **ما دام جملة1 لا تساوي السلسلة "نهاية"**، واذا تحقق هذا الشرط يلج التنفيذ جسم تعليمة التكرار.

- في اول عملية في جسم تعليمة التكرار يطلب الخوارزم ادخال الجملة الثانية (السطر 6 من خطأ: لم يُعثر على مصدر المرجع و السطر الثالث من خطأ: لم يُعثر على مصدر المرجع)، وبعد ادخالها (السطر 7 من خطأ: لم يُعثر على مصدر المرجع والسطر الرابع من خطأ: لم يُعثر على مصدر المرجع) يقوم بمقارنتها مع محتوى الجملة الأولى، ولبلوغ هذا الهدف نستعمل الوظيفة المنتسبة **قارن** وننفذها على السلسلة **جملة1** ونزود مدخلها بالسلسلة **جملة 2**، كما يظهر ذلك في السطر 8 من خطأ: لم يُعثر على مصدر المرجع، وبعد تنفيذ الوظيفة نتحصل على ردها في المتغيرة **نتيجة_المقارنة**، والنتيجة تكون حتما احدي القيم التالية: 0، او 1 او -1، و نلاحظ ان المتغيرة **نتيجة_المقارنة** قد صرح بها في اول استعمال لها، وهذا امر مقبول كما رأيناه من قبل.

```
فضلا ادخل جملة اولى  او كلمة نهاية ان اردت الخروج
ذهب عمر الى  المدرسة قبل الظهر
فضلا ادخل جملة ثانية
ذهب علي الى المدرسة على الساعة العاشرة صباحا
الجملة: ذهب عمر الى  المدرسة قبل الظهر -- اكبر من الجملة -- ذهب علي الى المدرسة على الساعة العاشرة صباحا
----------------------------------
فضلا ادخل جملة اولى  او كلمة نهاية ان اردت الخروج
ذاك الشبل من ذلكم الاسد
فضلا ادخل جملة ثانية
هذا الشبل من ذاك الاسد
الجملة: ذاك الشبل من ذلكم الاسد -- اصغر من الجملة -- هذا الشبل من ذاك الاسد
----------------------------------
فضلا ادخل جملة اولى  او كلمة نهاية ان اردت الخروج
نهاية
----------------------------------
الى اللقاء...
----------------------------------
```

الشكل 5 : نتيجة تنفيذ خوارزم النص 3

- في المرحلة الثانية في جسم عملية التكرار (الأسطر من 9 الى 18) يحاول الخوارزم معرفة محتوي المتغيرة **نتيجة_المقارنة** ، ولكل نتيجة فعل مناسب:
  - فإذا كانت قيمة المتغيرة **نتيجة_المقارنة** هي 0 (الأسطر من 9 الى 11 من خطأ: لم يُعثر على مصدر المرجع) ، يتم ارسال النص **الجملتين متساويتين..** الى الشاشة ثم ينتقل التنفيذ الى السطر 19.
  - اما اذا كانت القيمة مختلفة عن الصفر يحاول الخوارزم ان يرى هل القيمة الموجودة في المتغيرة **نتيجة_المقارنة** هي 1 (الأسطر من 12 الى 14 من خطأ: لم يُعثر على مصدر المرجع)، فاذا كان الأمر كذلك، يستنتج الخوارزم ان السلسلة **جملة1** اكبر من السلسلة **جملة2**، ويرسل الى الشاشة نصا يفيد بهذه النتيجة (وهذا ما يحصل في تنفيذ النص الذي يعبر عنه خطأ: لم يُعثر على مصدر المرجع في سطره الخامس)، و بعد ارسال النص ينتقل التنفيذ الى السطر 19.
  - اذا كان محتوى المتغيرة **نتيجة_المقارنة** مختلفا عن 0 او 1، فمعناه ان المتغيرة تحتوي حتما على القيمة -1، وفي هذه الحالة لا حاجة للمقارنة بين محتوى المتغيرة والقيمة -1، وفي هذه الحالة تكون السلسلة **جملة1** اصغر من السلسلة **جملة2.**
- تبدأ المرحلة الثالثة لجسم تعليمة التكرار مع تعليمة السطر 19، فيطلب الخوارزم ادخال جملة اولى تقارن فيما بعد مع جملة ثانية اذا كانت مختلفة عن السلسلة **"نهاية".**

3 – معرفة حجم السلسة (عدد احرف السلسلة)، الوظيفة: طبيعي حجم().

تُخبر الوظيفة **حجم()** عن **عدد حروف** السلسلة، او ما نسميه ايضا **حجم** السلسلة او **طول** السلسلة، ونلاحظ هنا اختلاف النمط سلسلة من النمط جدول، ففي النمط جدول نتحصل على حجم الجدول عبر السمة **حجم**، اما في النمط سلسلة فنتحصل على الحجم باستعمال الوظيفة **حجم()**، ويبين خطأ: لم يُعثر على مصدر المرجع و خطأ: لم يُعثر على مصدر المرجع كيفية استعمال هذه الوظيفة لمعرفة طول السلاسل.

الجزء الثاني: الفصل الثالث عشر: الأوعية: اهم وظائف النمط سلسلة

ففي السطرين 4 و5 نتحصل في المتغيرتين الطبيعيتين **طول_سلام** و **طول_رحمة** على طول السلسلتين التي عرفتا في السطرين 2 و 3، وهذا باستعمال الوظيفة **حجم()**، ففي السطرين 4 و5 استعملنا صلات السلسلتين للوصول الى الوظيفة **حجم()**، وفي الأسطر من 6 الى 8 اخرجنا على الشاشة نتائج السطرين 4 و 5 ، ونلاحظ من خلال الشاشة (خطأ: لم يُعثر على مصدر المرجع) ان نتيجة تنفيذ السطر 7 ليست بالنتيجة المرجوة وهذا راجع الى اولية تنفيذ عمليات العبارة.

"طول سلام و الرحمة =" + طول_سلام + طول_رحمة

فأول عبارة جزئية تقيم هي العبارة الجزئية: **"طول سلام و الرحمة =" + طول_سلام**، وهنا يكون سلوك عملية الجمع (+) هو سلوك لصق سلسلتين، فأولا تحول القيمة الموجودة في المتغيرة الطبيعية **طول_سلام** (و هي 12 حسب خطأ: لم يُعثر على مصدر المرجع و خطأ: لم يُعثر على مصدر المرجع) الى سلسلة من الحروف (و هي "12")، ثم تلصق بالسلسلة **"طول سلام و الرحمة =**"، لتصبح النتيجة السلسلة **"طول سلام و الرحمة = 12"**، وبعد الحصول على هذه النتيجة ينته تقييم العبارة الجزئية وتصبح العبارة الأصلية على الشكل التالي:

"طول سلام و الرحمة = 12" + طول_رحمة

و في هذا الشكل الأخير للعبارة الأصلية يكون سلوك عملية الجمع (+) هو سلوك لصق سلسلتين، فتُحَول اولا قيمة المتغيرة **طول_رحمة** (اي 11 حسب خطأ: لم يُعثر على مصدر المرجع و خطأ: لم يُعثر على مصدر المرجع) الى شكل سلسلة، (اي "11") ثم تلصق بالسلسلة **"طول سلام والرحمة = 12"** لتصبح النتيجة النهائية السلسلة **"طول سلام و الرحمة = 1112"** كما يظهر ذلك في خطأ: لم يُعثر على مصدر المرجع.

في السطر 8، تكون النتيجة موافقة لما ينتظر، ففي عبارة هذا السطر، وحسب ما ورد في سلم الأوليات الخاص بالعمليات، فأول عملية تنفذ هي عملية التقويس، وفي اطار عملية التقويس عبارة مبنية على عملية الجمع(+)، فتنفذ اولا هذه العملية، ولكون معاملي هذه العملية من النمط طبيعي، فسلوكها هو جمع عددين طبيعيين، ونتيجتها تكون القيمة 23، وبعد اتمام عملية الجمع وعملية التقويس تصبح العبارة الأصلية على الشكل التالي:

"طول سلام و الرحمة = " + 23

وفي هذه العبارة يكون سلوك عملية الجمع هو لصق سلسلتين لأن احد المعاملين من النمط سلسلة، فتحول اولا القيمة الطبيعية 23 الى السلسلة "23" ، وتكون النتيجة النهائية السلسلة **"طول سلام و الرحمة = 23"** التي ترسل الى الشاشة (خطأ: لم يُعثر على مصدر المرجع)

في السطر 9 نرى استعمالا جديدا لكيفية كتابة طلب تنفيذ وظيفة منتسبة، ففي هذا السطر كتبنا مباشرة المتغيرة من النمط سلسلة و**هي "السلام عليكم و رحمة الله و بركاته"** متبوعة مباشرة بعملية الولوج في المتغيرة (العملية الممثلة بالنقطة)، فما عهدناه هو كتابة صلة متبوعة بعملية الولوج في المتغيرة المشار اليها بالصلة، كما هو الحال في السطرين 4 و 5، وفي حقيقة الأمر فالكتابة **"السلام عليكم و رحمة الله و بركاته"** يجب ان ترى على انها صلة لكن هذه الصلة غير متوفرة لكاتب الخوارزم لعدم احتياجه لها بعد هذه الكتابة، فمثلا اذا كان اسم الصلة هو **كظ1**، فالكتابة **"السلام عليكم و رحمة الله و بركاته".حجم()** يجب ان ترى انها في الحقيقة على الشكل **كظ1.حجم()**.

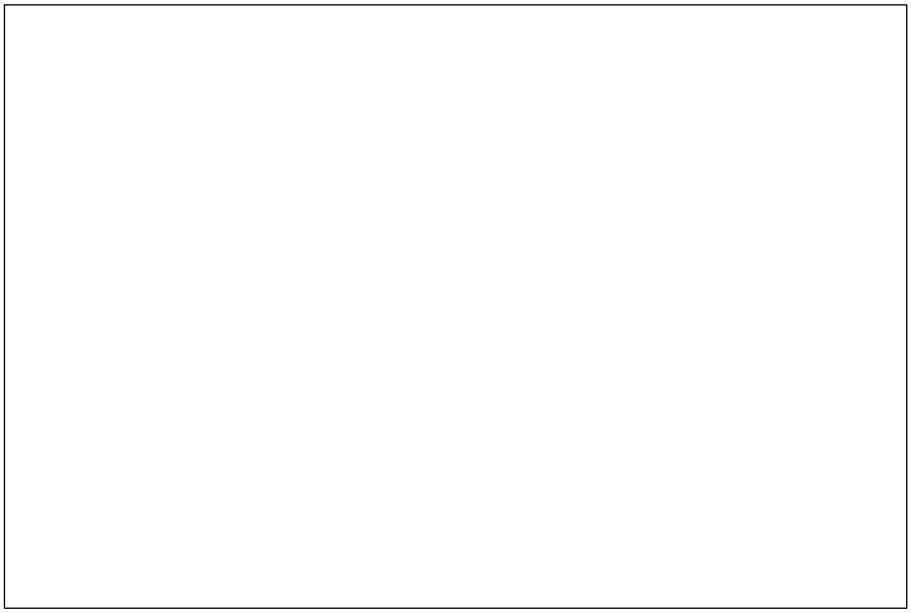
بما ان الكتابة **"السلام عليكم و رحمة الله و بركاته"** هي في حقيقة الأمر كتابة اخرى لعملية تشييد متغيرة طارئة من النمط سلسلة، أي **شيد سلسلة("السلام عليكم و رحمة الله و بركاته")**، فيمكن ايضا كتابة العبارة **"السلام عليكم و رحمة الله و بركاته".حجم()** على الشكل **شيد سلسلة("السلام عليكم و رحمة الله و بركاته").حجم()**

```
طول سلام = 12
طول سلام و الرحمة = 1112
طول سلام و الرحمة الحقيقي = 23
طول السلام كاملا = 33
طول سلام مرة ثانية = 12
طول سلام و الرحمة مرة ثانية = 23
طول السلسلة الفارغة هو : 0
فضلا ادخل جملة مفيدة: هذا درس يتناول النمط سلاسل الحروف.
الجملة التي ادخلت مكونة من 35 حرفا
```

الشكل 6 : نتيجة تنفيذ خوارزم النص 4

4 – معرفة الحرف الموجود في موقع ما من سلسلة ما، الوظيفة: حرف حرف_في (طبيعي موقع):

تقوم الوظيفة **حرف_في** بإرجاع الحرف الموجود في الموقع المذكور. من السلسلة، وأول حرف في السلسلة له الموقع 0، وموقع آخر حرف يساوي حجم السلسلة ناقص 1، والسلسلة الفارغة لا موقع لها، فلا يمكن استعمال مثل هذه الوظيفة مع سلسلة فارغة، ومن الخطأ ذكر موقع اقل من 0 او اكبر او يساوي حجم السلسلة، و يبين خطأ: لم يُعثر على مصدر المرجع وخطأ: لم يُعثر على مصدر. المرجع بعض الاستعمالات لهذه الوظيفة.



النص 5: كيفية استغلال الوظيفة حرف_في المنتسبة للنمط سلسلة

في بداية الأمر يقوم الخوارزم بقراءة جملة يكتبها المستعمل (الأسطر. من 2 الى 4 من خطأ: لم يُعثر على مصدر المرجع، والسطر الأول من خطأ: لم يُعثر على مصدر المرجع) ومباشرة بعد قراءة الجملة يقوم خوارزم خطأ: لم يُعثر على مصدر المرجع بإبلاغ المستعمل عن حجم الجملة التي كتبها (السطرين 5 و 6 من خطأ: لم يُعثر على مصدر المرجع، والسطر الثاني في خطأ: لم يُعثر على مصدر المرجع).

في الأسطر من 7 الى 9 من خطأ: لم يُعثر على مصدر المرجع (الأسطر من 3 الى 5 في خطأ: لم يُعثر على مصدر المرجع) يبين الخوارزم كيفية استغلاله، ثم في الأسطر من 10 الى 12 (السطر 6 في خطأ: لم يُعثر على مصدر المرجع) يطلب الخوارزم **رقم_موقع**، فيرد المستعمل بالرقم 1، وبعد قراءة القيمة الممثلة للموقع يحتفظ بها الخوارزم في المتغيرة **رقم_الموقع** التي صرح بها في السطر. 11.

بعد الحصول على اول قيمة تمثل موقعا في السلسلة التي ادخلت من قبل المستعمل، يدخل الخوارزم في تكرار (السطر. 13)، فيتحقق اولا من ان القيمة التي اعطيت له والموجودة في المتغيرة **رقم_الموقع** هي قيمة لموقع حقيقي، فإن كان رقم الموقع حقيقي (اي انه في حالة خطأ: لم يُعثر على مصدر. المرجع اصغر من 27 او اكبر او يساوي 0) يدخل الخوارزم جسد عملية التكرار(الأسطر. من 14 الى 16)، وفي جسد التكرار يقوم الخوارزم اولا بإيجاد الحرف الموجود في الموقع الذي اعطي (اي 1 في حالة خطأ: لم يُعثر على مصدر المرجع) من خلال الوظيفة **حرف_في** (السطر 14)، و نرى جليا في السطر 14 ان الوظيفة **حرف_في** يُطلب تنفيذها على المتغيرة جملة (**جملة.حرف_في (رقم_الموقع)**)، وبعد الحصول على الحرف يكتب الخوارزم رده على المستعمل (السطر 6 في خطأ: لم يُعثر على مصدر المرجع)، وبعد هذا الرد يطلب الخوارزم موقعا. جديدا (السطرين 15 و 16،

والسطر. 8 في خطأ: لم يُعثر على مصدر المرجع)، وفور. الحصول على قيمة الموقع يعاود الخوارزم من خلال عملية التكرار التأكد من صحة قيمة الموقع، وحسب خطأ: لم يُعثر على مصدر المرجع يدخل المستعمل القيمة 0 فيخرج له الحرف الموجود في الموقع 0، اي الحرف 'ا'، ثم الموقع 18 الى ان يدخل المستعمل الموقع 30، وهنا يتحقق الخوارزم من خلال تعليمة التكرار ان الموقع غير صحيح فينته التكرار وينتقل التنفيذ الى التعليمة التي تلي تعليمة التكرار اي تلك الموجودة في السطر. 18، وبتنفيذها. (السطر 19 في خطأ: لم يُعثر على مصدر المرجع) ينته تنفيذ الخوارزم.

```
فضلا ادخل جملة مفيدة: السيف أصدق إنباءا من الكتب.
الجملة التي ادخلت مكونة من 27 حرفا.
اعطني موقعا حقيقيا في الجملة التي ادخلت اعطك حرفه.
انتبه... الموقع يجب ان يكون من 0 و لا يتجاوز 26
انتبه...في حالة اعطائك لموقع غير حقيقي، ينته الخوارزم...
الموقع : 1
الحرف الموجود في الموقع 1 هو: ل
الموقع : 0
الحرف الموجود في الموقع 0 هو: ا
الموقع : 18
الحرف الموجود في الموقع 18 هو:م
الموقع : 22
الحرف الموجود في الموقع 22 هو:ل
الموقع : 25
الحرف الموجود في الموقع 1 هو:ب
الموقع : 30
الموقع 30 غير حقيقي... الى اللقاء في المرة القادمة.
```

الشكل 7 : نتيجة تنفيذ خوارزم النص 5

5 – البحث عن حرف ما في السلسة وارجاع موقعه فيها، الوظيفة: طبيعي موقع (حرف ح)

تقوم الوظيفة **موقع** بالبحث عن اول ظهور لحرف ما في السلسلة ، ويبدأ البحث انطلاقا من موقع الحرف الأول في السلسلة (الموقع رقم 0)، فاذا تمكنت الوظيفة من ايجاد الحرف، تنته وترجع موقع الحرف، وفي حالة ما اذا كان الحرف غير موجود في السلسلة، ترجع الوظيفة القيمة –1، وهذه القيمة تخبر ان الحرف غير موجود في السلسلة، ومن خلال خطأ: لم يُعثر على مصدر المرجع وخطأ: لم يُعثر على مصدر المرجع نرى كيفية الاستغلال.

6 – البحث عن موقع حرف ما في السلسلة انطلاقا من البداية، الوظيفة طبيعي موقع (حرف ح).

بعد قراءة جملة ادخلها المستعمل بواسطة لوحة الحروف (الأسطر من 2 الى 4 في خطأ: لم يُعثر على مصدر المرجع، والسطر الأول في خطأ: لم يُعثر على مصدر المرجع) ثم إخبار المستعمل بطول الجملة (اي عدد حروفها) التي ادخلها (السطرين5 و 6 في خطأ: لم يُعثر على مصدر. المرجع والسطر. الثاني في خطأ: لم يُعثر على مصدر المرجع)، يبين الخوارزم كيفية استعماله

(السطرين7 و 8 في خطأ: لم يُعثر على مصدر المرجع والسطرين 3 و 4 في خطأ: لم يُعثر على مصدر. المرجع).

النص 6: كيفية استغلال الوظيفة حرف_في المنتسبة للنمط سلسلة

```
فضلا ادخل جملة مفيدة: ولقد أبيت على الطوى وأظله حتى انال به كريم المأكل.
الجملة التي ادخلت مكونة من 50 حرفا.
اعطني حرفا ما ابلغك بأول ظهور له في الجملة التي ادخلت.
ادخل الحرف 0 ان اردت توقيف تنفيذ الخوارزم
ادخل الحرف الذي تريد معرفة موقعه في الجملة : و
موقع الحرف و هو: 0
ادخل الحرف الذي تريد معرفة موقعه في الجملة : س
الحرف س غير موجود في الجملة التي ادخلت.
ادخل الحرف الذي تريد معرفة موقعه في الجملة : ب
موقع الحرف ب هو: 7
ادخل الحرف الذي تريد معرفة موقعه في الجملة : 0
الى اللقاء في المرة القادمة.
```

الشكل 8 : نتيجة تنفيذ خوارزم النص 6

في سياق استعماله، يطلب الخوارزم من المستعمل اعطائه حرفا. (السطر 10 في خطأ: لم يُعثر على مصدر المرجع، والسطر 5 في خطأ: لم يُعثر على مصدر المرجع) ثم يقوم الخوارزم بقراءة الحرف ووضعه في المتغيرة المسماة **الحرف**(السطر 11 في خطأ: لم يُعثر على مصدر المرجع)، وقد تم التصريح بالمتغيرة **الحرف** في السطر 9 من خطأ: لم يُعثر على مصدر المرجع.

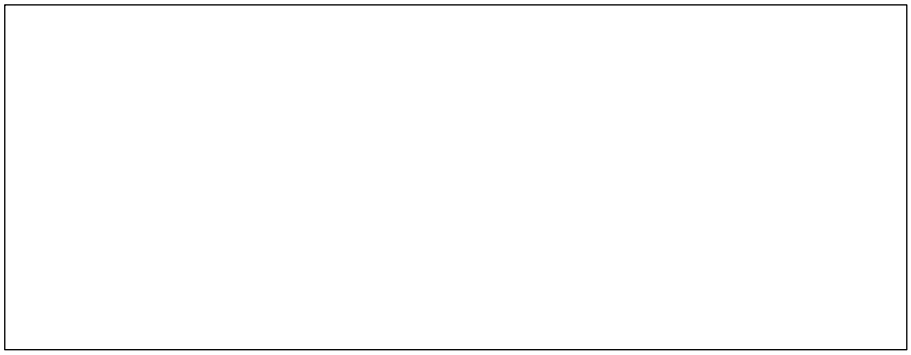
بعد قراءة اول حرف يدخل الخوارزم في عملية تكرا ر لا ينته الا اذا كان الحرف الذي قُرِأ هو الحرف '0' (السطر. 12 من خطأ: لم يُعثر على مصدر المرجع)، فكلما كان الحرف الذي قُرِأ مختلف عن الحرف '0'، يقوم الخوارزم، وهو في جسد تعليمة التكرار بما يلي:

- استعمال الوظيفة **موقع** لإيجاد موقع الحرف الذي قُرِأ (السطر 13 من خطأ: لم يُعثر على مصدر. المرجع)، ونرى من خلال السطر 13 ان الوظيفة تنفذ على السلسلة المسماة جملة **(جملة. موقع(الحرف))**، وترجع الوظيفة نتيجتها. في المتغيرة **موقع_الحرف** التي صرح بها في نفس السطر، اي السطر 13.
- التحقق من القيمة التي ارجعتها. الوظيفة **موقع()** في المتغيرة **موقع_الحرف**، هل هي حقا موقعا ام لا (السطر 14) ، فاذا كانت القيمة التي تحتويها المتغيرة **موقع_الحرف** قيمة تساوي او تكبر القيمة 0 فهي موقعا، وفي هذه الحالة يخبر الخوارزم عن موقع الحرف في الجملة (السطر 17 من خطأ: لم يُعثر على مصدر المرجع والسطرين 6، 10 في خطأ: لم يُعثر على مصدر المرجع)، اما اذا كانت القيمة تساوي −1 فالقيمة لا تمثل موقعا بل تمثل جوابا يفيد بان الحرف الذي ادخل من قبل المستعمل غير موجود في السلسلة، وفي هذه الحالة يخبر الخوارزم المستعمل بعدم وجود الحرف في الجملة (السطر 15 من خطأ: لم يُعثر على مصدر المرجع والسطر. 8 في خطأ: لم يُعثر على مصدر المرجع).

عند الانتهاء من عملية ايجاد موقع الحرف في السلسلة، ان كان الحرف موجودا. حقا، ينتقل الخوارزم، وهو دائما في جسد تعليمة التكرار، الى طلب ادخال حرف ثم الى قراءة الحرف ووضعه في المتغيرة **الحرف**(السطرين 18 و 19 من خطأ: لم يُعثر على مصدر المرجع)، ثم يعود الى اول تعليمة في عملية التكرار (السطر 12 من خطأ: لم يُعثر على مصدر المرجع) وهي النظر في قيمة الحرف الذي ادخله المستعمل، فاذا ادخل حرفا مختلفا عن الحرف '0' تكرر تعليمات جسد التكرار، والا تنته تعليمة التكرار وينتقل التنفيذ خارج تعليمة التكرار وبالتدقيق. الى السطر 21، وفي هذا السطر اخر تعليمة تنفذ وينته تنفيذ الخوارزم.

7 – البحث عن موقع حرف انطلاقا. من موقع ما، الوظيفة: طبيعي موقع (حرف ح، طبيعي البداية)

سلوك هذه الوظيفة هو نفس سلوك الوظيفة السابقة **(طبيعي موقع (حرف ح))**، والفرق بينهما ان الوظيفة السابقة تبدأ البحث من الموقع الأول (الموقع 0)، بينما هذه الوظيفة تبدأ من موقع يحدد من خلال المدخل المسمى **البداية**، و يبين خطأ: لم يُعثر على مصدر المرجع وخطأ: لم يُعثر على مصدر المرجع كيفية استغلال الوظيفة "**طبيعي موقع (حرف ح، طبيعي البداية)**".



النص 7: كيفية استغلال الوظيفة طبيعي موقع (حرف ح، طبيعي البداية) المنتسبة للنمط سلسلة

```
موقع الحرف س انطلاقا من بداية الجملة  هو: 4
موقع الحرف س انطلاقا من بداية الجملة  هو: 4
موقع الحرف س انطلاقا من الموقع رقم 5 هو: 11
موقع الحرف س انطلاقا من الموقع رقم 12 هو: 30
موقع الحرف س انطلاقا من الموقع رقم 31 هو: 46
موقع الحرف س انطلاقا من الموقع رقم 47 هو: -1
```

الشكل 9 : نتيجة تنفيذ خوارزم النص 7

يحتوي السطر 2 من خطأ: لم يُعثر على مصدر المرجع على تصريح بقيمة اولية لمتغيرة من النوع **سلسلة،** وفي السطر 3 نرى استعمال الوظيفة السابقة، اي "**طبيعي موقع (حرف ح)**"، وهذه الوظيفة تبحث عن الحرف **'س'** انطلاقا من الموقع 0، فتجد في الموقع 4 اول 'س' بعد الموقع 0، وفي السطر 4 نستعمل الوظيفة "**طبيعي موقع (حرف ح، طبيعي البداية)**"، وفي هذا السطر تقوم الوظيفة بالبحث عن اول حرف 'س' يظهر بعد الموقع المشار اليه في مدخله، و هو الموقع 0، و نرى ان النتيجة هي نفس النتيجة التي توصلت اليها تعليمة السطر 2، ونرى من خلال التعليمات التي تلي ان البحث عن الحرف **'س'** الذي يلي يبدأ دائما بعد الموقع المحتوي على الحرف **'س'** الذي وجد من قبل، وفي التعليمة الأخيرة، وبعد آخر س وجد في الموقع 46، نرى ان الوظيفة ترجع القيمة –1 منبهة انه بعد الموقع 47 لا وجود لحرف 'س'.

7 – 1 النسخة الثانية لخوارزم خطأ: لم يُعثر على مصدر المرجع

في هذه النسخة (خطأ: لم يُعثر على مصدر المرجع)، نستبدل القيم الثابتة الدالة على موقع بداية البحث بمتغيرات آملين ان نعطي بعض المرونة للخوارزم، فبدلا من ذكر قيمة ثابتة في المدخل **البداية،** نذكر متغيرة قد هيئت بالقيمة المناسبة، فالقيمة الأولى هي 0، وهذه القيمة الأولية نضعها في المتغيرة **بداية_البحث** وهذا ما يحدث في السطر 3، اذ فيه يتم التصريح بالمتغيرة **بداية_البحث** وشحنها بقيمة اولية هي الصفر.

في السطر 4 نرى الطلب الأول لتنفيذ الوظيفة **موقع()، ونلاحظ** استعمال المتغيرة **بداية_البحث** في المدخل **البداية،** ونرى نتيجة التنفيذ في السطر الأول من خطأ: لم يُعثر على مصدر المرجع.

حتى نتمكن في الأسطر التالية من الخوارزم من تهيئة المتغيرة **بداية_البحث** بقيمة مناسبة، وجب علينا الاحتفاظ بالقيمة التي ترجعها الوظيفة **موقع** في السطر 4، فالبحث التالي (السطر 6) يبدأ دائما من الموقع الذي يلي الموقع الذي وجد فيه الحرف المبحوث عنه، اي الحرف **'س'** فيما يخص خوارزم خطأ: لم يُعثر على مصدر المرجع وخطأ: لم يُعثر على مصدر المرجع، ونرى انه في السطر 4 من خطأ: لم يُعثر على مصدر المرجع تم الاحتفاظ برد الوظيفة **موقع** في المتغيرة **الموقع_الحالي** التي صرح بها في السطر 3، وقبل الانتقال الى البحث التالي (السطر 7) نهيأ المتغيرة **بداية_البحث** بقيمة هي قيمة المتغيرة **بداية_البحث** زائد 1 (السطر 6).

النص 8: نسخة 2 لخوارزم النص 7

```
موقع الحرف س انطلاقا من بداية الجملة هو: 4
موقع الحرف س انطلاقا من الموقع رقم 5 هو: 11
موقع الحرف س انطلاقا من الموقع رقم 12 هو: 30
موقع الحرف س انطلاقا من الموقع رقم 31 هو: 46
موقع الحرف س انطلاقا من الموقع رقم 47 هو: -1
```

الشكل 10 : نتيجة تنفيذ خوارزم النص 8

من خلال نص الخوارزم، نرى تكرار السلوك التالي: – طلب ايجاد الحرف من بداية ما – الاحتفاظ بموقع وجوده – ثم تهيئة البداية الجديدة انطلاقا من الموقع الذي وجد فيه الحرف، وهذه الملاحظة تجعلنا نفكر في انجاز اجراء آخر نستغل فيه هذه الملاحظة.

أخيرا، ينته الخوارزم في السطر 17 بإخراج رسالة السطر 5 من خطأ: لم يُعثر على مصدر المرجع، و تفيد هذه الرسالة بعدم وجود الحرف س بعد البداية التي اعطيت للوظيفة موقع في السطر 16.

7 – 2 حساب عدد المرات التي يظهر فيها حرف ما في سلسلة ما

برغم الاستعانة ببعض المتغيرات في خوارزم خطأ: لم يُعثر على مصدر المرجع الا ان هذا الخوارزم يبق خاص بالجملة التي يبحث فيها وكذلك بالحرف 'س' وبعدد المرات (اي 4) التي يظهر فيها الحرف 'س' في الجملة، وحتى المتغيرات التي ادخلت لم تُضف الا بعض الوضوح في كتابة الخوارزم، وهذا الوضوح الأخير هو الذي نستغله في فهم ادق للخوارزم وفي عملية استخراج نص خوارزم جديد هدفه ايجاد عدد المرات التي يظهر فيها حرف ما في جملة ما، اي ان الخوارزم عند كتابته لا يعرف لا الجملة ولا الحرف.

الخوارزم الذي نحن بصدد انجازه، والذي نسميه **عدد_الأحرف**، هدفة ما يلي: ايجاد **عدد المرات** التي يظهر فيها **حرف ما** في **سلسلة ما**، ومن هذا الوصف للخوارزم نستخرج المفاهيم التالية:

- الخوارزم يرجع عددا هو عدد المرات التي يظهر فيها **حرف ما** في **جملة ما**، اي ان الخوارزم يجب ان يكون على شكل وظيفة، ولابد ان يكون **طبيعي** هو نمط هذه الوظيفة.
- الوظيفة تبحث عن **حرف ما**، ومعنى هذا ان الحرف غير معروف في نص الوظيفة، وتعريفه يكون عند طلب تنفيذ الوظيفة، ومعنى هذا ان الوظيفة لا بد لها من مدخل تتحصل من خلاله على الحرف، وهنا لا بد ان يكون **حرف** هو نمط المدخل.
- البحث عن الحرف الذي وصف في المدخل يتم في اطار **جملة ما**، اي ان الجملة غير معروفة في نص الخوارزم، وكما كان الشأن بالنسبة للحرف الذي يُبحث عنه، فلا بد للوظيفة ان يكون لها مدخل تتحصل من خلاله على الجملة، وهنا لا بد ان تكون **سلسلة** هي نمط المدخل.

و هكذا نستنتج ان شكل الوظيفة يكون على النحو التالي:

طبيعي عدد_الأحرف(حرف ح ، سلسلة جملة).

و قبل ان نبدأ في كتابة نص الخوارزم **عدد_الأحرف** نتأمل قليلا في خوارزم خطأ: لم يُعثر على مصدر المرجع.

من خلال خطأ: لم يُعثر على مصدر المرجع نرى جليا ان بعض التعليمات يتم اعادة كتابتها كما هي 4 مرات ، وهي:

4	**الموقع_الحالي = جملة.موقع('س'، بداية_البحث)) ؛**
5	**اكتب_سطر("موقع الحرف س انطلاقا من الموقع "+ البداية + "في الجملة هو: "+ الموقع_الحالي) ؛**
6	**بداية_البحث = الموقع الحالي + 1 ؛**

و تم التكرار 4 مرات لأن كاتب الخوارزم علم ان الحرف 'س' يظهر 4 مرات في الجملة، ولو ظهر الحرف 'س' مئة مرة في الجملة التي عرفها كاتب الخوارزم في نص الخوارزم ، لكرر كتابة تعليمات الأسطر 4، 5 و 6 مئة مرة.

من خلال خطأ: لم يُعثر على مصدر المرجع نرى ايضا ان الخوارزم ينته عندما ينته ايجاد كل الأحرف وفي هذه الحالة ترد الوظيفة **موقع** القيمة –1 ، اي ان تكرار عملية البحث عن موقع الحرف بعد هذه الحالة لا يكون مجديا.

من هذه الملاحظات نستنتج انه لا بد ان تكون الوظيفة **عدد_الأحرف** مبنية على تعليمة التكرار (خطأ: لم يُعثر على مصدر المرجع)، وينته التكرار. عندما ترد الوظيفة **موقع** القيمة –1.

يحتوي خطأ: لم يُعثر على مصدر المرجع على خوارزم كامل يحتوي عل وظيفة واجراء. الوظيفة هي **عدد_الأحرف** والإجراء هو **اساسي،** اي ان خطأ: لم يُعثر على مصدر المرجع قابل للتنفيذ على الحاسوب لو حُوِل الى لغة برمجة.

النص 9: الوظيفة عدد_الاحرف(..) مبنية على تعليمة التكرار

7 – 3 سلوك الوظيفة عدد_الأحرف:

تحتوي الوظيفة **عدد_الأحرف** على مدخلين (السطر 1):

- المدخل الأول من النمط **حرف** يحمل الاسم **ح** وهو المدخل الذي تتحصل منه الوظيفة على الحرف الذي يُبحث عنه في جملة ما ويُحسب عدد المرات التي يظهر فيها في الجملة.
- المدخل الثاني من النمط **سلسلة** ويحمل الاسم **جملة،** ومنه تتحصل الوظيفة على الجملة التي يتم البحث فيها عن الحرف.

في بداية الوظيفة يتم التصريح بمتغيرات تحتاجها الوظيفة في عملها:

2	**طبيعي الموقع_الحالي، بداية_البحث = 0 ؛**
3	**طبيعي عدد = 0؛**

- المتغيرة **الموقع_الحالي** تستعمل للاحتفاظ برد الوظيفة **موقع** (السطرين 4 و 8).
- المتغيرة **بداية_البحث** تُستعمل لإرشاد الوظيفة **موقع** عن الموقع في الجملة الذي منه ينطلق البحث عن الحرف كما يظهر في السطرين 4 و 8، و بما ان اول موقع ينطلق منه البحث هو الموقع الأول، فقد صرح بالمتغيرة **بداية_البحث** بقيمة اولية هي 0، وفي المرات التالية للبحث عن موقع الحرف يتم تسديد قيمة المتغيرة **بداية_البحث** قبل اعطائها للوظيفة **موقع.**
- المتغيرة **عدد،** وفيها يتم تخزين القيمة التي تُخبر عن عدد المرات التي يظهر فيها الحرف في الجملة، في بداية الخوارزم تكون القيمة المخزنة في المتغيرة **عدد** هي 0، وكلما وُجد موقعا آخرا للحرف في الجملة يتم اضافة القيمة 1 الى قيمة المتغيرة **عدد،** وعند انتهاء الوظيفة **عدد_الأحرف** من عملها تكون المتغيرة **عدد** تحتوي على النتيجة النهائية، وهي التي ترجعها الوظيفة.

في السطر 4 من خطأ: لم يُعثر على مصدر المرجع تنطلق اول عملية بحث عن الحرف الموجود في المتغيرة **ح** (الممثلة للمدخل الأول) في الجملة المخزنة في المتغيرة **جملة** (الممثلة للمدخل الثاني)، فان وجد الحرف في الجملة نتحصل على موقعه في المتغيرة **الموقع_الحالي،** وتكون القيمة تساوي او تكبر القيمة 0، وان لم يوجد الحرف في الجملة نتحصل على القيمة −1 في المتغيرة **الموقع_الحالي.**

في السطر 5 تبدأ عملية التكرار، ونرى جليا ان دخول جسد عملية التكرار لن يتم الا اذا كانت قيمة المتغيرة **الموقع_الحالي** مختلفة عن القيمة −1، وهكذا ان تم الحصول في تعليمة السطر 4 على القيمة −1، فان دخول جسد تعليمة التكرار لن يتم، بل ينته تنفيذ تعليمة التكرار والانتقال الى التعليمة التي تلي تعليمة التكرار، اي تعليمة السطر 10 التي بها ينته تنفيذ الوظيفة وتكون النتيجة التي ترجع هي 0.

اما اذا كانت نتيجة السطر 4 (اي محتوى المتغيرة **الموقع_الحالي)** مختلفة عم القيمة −1، فان الدخول الأول لجسد تعليمة التكرار يصبح ممكنا، وفي اول تعليمة لجسد التكرار (السطر 6) تُوثق عملية ايجاد الحرف في الجملة بزيادة القيمة 1 الى محتوى المتغيرة **عدد،** وفي اول دخول لجسد تعليمة التكرار تصبح قيمة المتغيرة **عدد** هي القيمة 1 وهي توثيق لنتيجة تعليمة السطر 4.

بعد توثيق العملية الأولى للبحث عن الحرف، ننتقل من خلال تعليمتي السطرين 7 و 8 الى البحث مجددا عن موقع آخر للحرف الموجود في المتغيرة **ح،** وحتى نتفادى ان يشمل البحث الموقع الذي وجد فيه الحرف من قبل، نبدأ البحث انطلاقا من الموقع الذي يلي مباشرة آخر موقع وجد فيه الحرف المبحوث عنه، ولهذا يتم في السطر 7 تسديد قيمة المتغيرة **بداية_البحث،** فيتم البحث انطلاقا من الموقع الذي يلي مباشرة الموقع الحالي : **بداية_البحث = الموقع_الحالي + 1.**

بعد تنفيذ تعليمة السطر 8، والحصول على نتيجة في المتغيرة **الموقع_الحالي**، يخرج التنفيذ من جسم التكرار وينتقل الى رأس تعليمة التكرار، ويحدث هنا ما حدث بعد تنفيذ تعليمة السطر 4، اي ان رأس التكرار يتحقق من قيمة المتغيرة **الموقع_الحالي**، فان كانت قيمتها تساوي –1، ينته التكرار وتُرجِع الوظيفة **عدد_الاحرف** ما في المتغيرة عدد، اما ان كانت قيمة المتغيرة **الموقع_الحالي** لا تساوي –1، فان التنفيذ يدخل جسم تعليمة التكرار اين تُوثق النتيجة الجديدة للوظيفة **عدد_الاحرف** ثم تنفذ تعليمات المتعلقة بالبحث التالي عن موقع الحرف الموجود في المتغيرة **ح**.

7 – 4 سلوك الإجراء اساسي (خطأ: لم يُعثر على مصدر المرجع)

يقوم الإجراء اساسي بقراءة جملة يدخلها المستعمل، ثم يطلب حرفا ليتم البحث عن عدد المرات التي ذكر فيها في الجملة التي ادخلت، ويبق الإجراء مع نفس الجملة واحرف متغيرة الى ان يتم ادخال الحرف '0'، وهنا ينته الخوارزم، وكلما ادخل المستعمل حرفا، يستعين الإجراء اساسي بالوظيفة **عدد_الاحرف** (السطر 22).

8 – البحث عن سلسلة داخل سلسلة اخرى، الوظيفة: طبيعي موقع (سلسلة س)

تقوم هذه الوظيفة بالإخبار عن موقع جزء من سلسلة من الحروف موجودة في السلسلة التي تنفذ عليها هذه الوظيفة، فاذا وُجِد الجزء ترد الوظيفة المنتسبة موقع اول حرف تابع للجزء، وفي حالة عدم وجود الجزء تُرجع الوظيفة القيمة –1، ويبين خطأ: لم يُعثر على مصدر المرجع حالات ثلاث طُلب فيها تنفيذ خوارزم خطأ: لم يُعثر على مصدر المرجع، و نلاحظ في الطلب الثاني للتنفيذ ان الجزء الذي تحصلنا على موقعه، اي الكلمة "**فرس**"، هو الجزء الذي يظهر اولا في السلسلة.

النص 10: كيفية استغلال الوظيفة المنتسبة موقع(سلسلة س)

```
ادخل نصا صغيرا:السيف اصدق انباءا من الكتب
ادخل جزءا من النص الذي ادخلت سالفا لنخبرك عن موقعه في النص:صدق
السلسلة - صدق - موجودة في الموقع رقم 7
```

```
ادخل نصا صغيرا: أثرى الأموال فرس يتبعها فرس في بطنها فرس
ادخل جزءا من النص الذي ادخلت سالفا لنخبرك عن موقعه في النص:فرس
السلسلة - فرس - موجودة في الموقع رقم 13
```

```
ادخل نصا صغيرا: ابتسم فأن في الابتسامة راحة وصحة
ادخل جزءا من النص الذي ادخلت سالفا لنخبرك عن موقعه في النص:الراحة والصحة
```

الشكل 11 : نتيجة تنفيذ خوارزم النص 10

9 – البحث عن سلسلة انطلاقا من موقع ما، الوظيفة: طبيعي موقع (سلسلة س، طبيعي البداية)

سلوك هذه الوظيفة هو السلوك ذاته للوظيفة السابقة (**طبيعي موقع (سلسلة س)**)، والفرق بينهما انه مع هذه الوظيفة نحدد من اي موقع في السلسلة يبدأ البحث عن الجزء، فمثلا، اذا كانت الصلة **جملة 1** مرتبطة بالسلسلة "**أثرى الأموال فرس يتبعها فرس في بطنها فرس**"، فان نتيجة تنفيذ التعليمة **جملة 1.موقع("فرس"، 15)** هي القيمة 24 وهي موقع المرة الثانية التي تظهر فيها كلمة "**فرس**"، ونتيجة تنفيذ التعليمة **جملة1.موقع("فرس"، 25)** هي القيمة 37 وهي موقع المرة الثالثة التي تظهر فيها كلمة "**فرس**"، اما نتيجة تنفيذ التعليمة **جملة1.موقع("فرس"، 38)** فهي القيمة –1، ومعنى هذا ان الكلمة "**فرس**" غير موجودة بعد الموقع 37، وبفصل هذه الوظيفة يمكننا ان نكتب خوارزما يخبرنا عن عدد المرات التي يتردد فيها جزء من سلسلة في سلسلة ما.

الفصل الرابع عشر

الأوعية: العمليات المعتادة على الجداول

1 – مقدمة:

تُعد الجداول من الأوعية (او الحاويات) الشائعة الاستعمال في كتابة الخوارزميات، وفي اطارها صممت خوارزميات كثيرة الاستعمال، ومن ابرزها:

- خوارزميات تعمير الجداول.
- خوارزميات نسخ محتوى جدول في جدول آخر.
- خوارزميات البحث عن قيمة او معلومة ما في الجدول.
- خوارزميات البحث عن اصغر (او اكبر) قيمة في الجدول.
- خوارزميات ترتيب محتوى الجدول.

و تعتمد كل هذه الخوارزميات على طريقة ما من طرق التجوال في الجدول: هل يبدأ التجوال من اول خانة في اتجاه آخر خانة؟ هل يبدأ التجوال من آخر خانة في اتجاه اول خانة؟ الخ.. ففي اطار طريقة تجوال ما، وعندما يصل التجوال الى خانة ما، يقوم الخوارزم بعمليات مختلفة على الخانة، كملئها بقيمة ما، أو مقارنة محتواها بقيمة ما، او تغيير محتواها بعملية ما الخ...

في كل هذه الطرق يرتكز التجوال على متغيرة تحتوي على رقم الخانة التي يشرع في الانتقال اليها، وغالبا ما نسمي هذه المتغيرة **بمتغيرة التجوال** وهي بالضرورة من النمط طبيعي، فمثلا اذا كان التجوال من اول خانة الى آخر خانة، تكون قيمة متغيرة التجوال هي الصفر (0) في اول الأمر، اي موقع الخانة الأولى، ثم تصبح القيمة 1 بزيادة 1 الى قيمة متغيرة التجوال، فيكون التجوال قد وصل الى الخانة الثانية، وهكذا حتى تصبح قيمة متغيرة التجوال هي حجم الجدول ناقص 1، اي آخر خانة، وعندما تصل (او تتجاوز) قيمة متغيرة التجوال القيمة الممثلة لحجم الجدول، ينته التجوال ويكون الخوارزم قد زار كل الخانات، وننبه الى ان ترقيم خانات الجدول يبدأ من الصفر وينته عند القيمة الممثلة لحجم الجدول ناقص 1.

يتم التجوال حسب الطرق التي اشرنا اليها باستعمال تعليمات التكرار (مثلا تعليمة **مادام** وتعليمة "**منحتى**" التي سوف نعيد تقديمها فيما بعد)، وتبدأ هذه التعليمات بمرحلة اولى، تُحضر فيها متغيرة التجوال، فمثلا اذا كانت الزيارة تبدأ من اول خانة، توضع في متغيرة التجوال القيمة 0 اما ان كانت الزيارة تبدأ من الخانة الأخيرة في اتجاه الخانة الأولى فإننا نضع في متغيرة التجوال القيمة الممثلة لحجم الجدول ناقص واحد، وبعد المرحلة الأولى تأتي مرحلة ثانية، وهي البداية الحقيقية لتعليمة التكرار، فيها يتم تقييم عبارة منطقية مبنية اساسا على قيمة متغيرة التجوال التي تحتوي على رقم الخانة التي يشرع في زيارتها، ولن يتم الولوج في جسد تعليمة التكرار الا اذا كانت القيمة **صحيح** هي قيمة تقييم العبارة المنطقية، اما اذا كانت القيمة **خطأ** هي قيمة تقييم العبارة المنطقية، تنته تعليمة التكرار و ينته معها التجوال في الجدول.

فمثلا اذا كانت الزيارة من بداية الجدول في اتجاه آخر خانة، تكون العبارة كالتالي: **هل قيمة متغيرة التجوال اصغر من حجم الجدول**، فما دامت القيمة اقل من حجم الجدول فمعنى هذا ان محتوى متغيرة التجوال هو حقا رقم خانة وعلى الخوارزم زيارتها، اما اذا كانت القيمة تساوي او تكبر حجم الجدول، فمعنى هذا ان متغيرة التجوال لا تحتوي على رقم خانة من خانات الجدول، وان كل الخانات قد تم زيارتها ولا يمكن للتجوال ان يستمر، اما اذا كانت الزيارة من نهاية الجدول في اتجاه اول خانة، تكون اول قيمة لمتغيرة التجوال هي حجم الجدول – 1، والقيم التالية لمتغيرة التجوال تتناقص بواحد، وتكون العبارة المنطقية المتحكمة في التكرار على الشكل التالي: **هل قيمة متغيرة التجوال اكبر او تساوي 0**، فما دامت القيمة لم تنزلق الى اقل من 0 فمعنى هذا ان محتوى متغيرة التجوال هو حقا رقم خانة وعلى الخوارزم زيارتها، اما اذا كانت القيمة اقل من 0، فمعنى هذا ان متغيرة التجوال لا تحتوي على رقم خانة من خانات الجدول وان كل الخانات قد تم زيارتها ولا يمكن للتجوال ان يستمر.

في الطرق السابقة للتجوال نرى جليا ان كاتب الخوارزم هو الذي يحدد مسار التجوال في الجدول باستعمال متغيرة التجوال، وهذه الطرق ترتكز على تعليمة التكرار المحتوية في رأسها على عبارة منطقية مبنية على متغيرة التجوال ، اي انه على كاتب الخوارزم ان ينتبه جدا الى القيم التي توضع في متغيرة التجوال ولا يسمح ان تستعمل اذا كان محتواها لا يتناسب مع مواقع الخانات.

في بعض الاحيان، ما يهم كاتب الخوارزم هو زيارة كل الخانات ولا يهتم بأي خانة يبدأ منها التجوال ولا الاتجاه الذي يسير عليه التجوال، اي انه لا يريد ان ينشغل بالتحكم في قيمة متغيرة التجوال ولا يريد استعمالها، ففي مثل هذه الحالة، توجد تعليمة تكرار لا تستعمل متغيرة التجوال، وهذه التعليمة تضمن زيارة كل خانات ولا تظهر لكاتب الخوارزم طريقة تجوالها، فكاتب الخوارزم لا يهمه هذا الأمر، وهذه التعليمة الخاصة تسمى تعليمة **لكل**، وسوف ندرسها فيما يلي.

2 – تعليمات التكرار :

لحد الآن تعلمنا كيفية استعمال تعليمة **مادام** وفيما يلي نقدم تعليمتين للتكرار اشرنا اليهما فيما سبق، وهما تعليمة "**منحتى**". و تعليمة "**لكل**".

2 – 1 التعليمة "منحتى"

هي فقط كتابة مختصرة لتعليمة **مادام**، ونذكر ان تعليمة **مادام** تحتوي على اربعة مراحل:

- **المرحلة الأولى**، وتكتب تعليماتها قبل رأس تعليمة التكرار، والمرحلة الأولى تتكون من التعليمات (في العموم تعليمة واحدة) التي تهيأ فيها المتغيرات المتحكمة في التكرار (في العموم متغيرة واحدة) والتي تُبنى عليها العبارة المنطقية الموجودة في رأس تعليمة التكرار، وتنفذ المرحلة الأولى مرة واحدة فقط قبل الوصول الى المرحلة الثانية.

- **المرحلة الثانية**، وهي التي نسميها رأس تعليمة التكرار وتبدأ بكلمة **مادام** وتحتوي على عبارة منطقية اذا صحت تنفذ تعليمات جسد التكرار والا تنته تعليمة التكرار.
- **المرحلة الثالثة** وتحتوي عل تعليمات جسد التكرار التي تهدف الى انجاز السلوك الذي من اجله كتبت تعليمة التكرار، ويعاد تنفيذ تعليمات الجسد كلما اعيد تقييم العبارة المنطقية وأفضى تقييمها الى القيمة **صحيح**.
- **المرحلة الرابعة** و تحتوي على التعليمات (في العموم تعليمة واحدة) التي تهيأ فيها من جديد المتغيرات (عموما متغيرة واحدة) المتحكمة في التكرار والتي بنيت عليها العبارة المنطقية في رأس تعليمة **مادام**، وفي العموم تكون هذه التعليمات هي التعليمات الموجودة في جسد تعليمة التكرار، وتنفذ هذه التعليمات مباشرة قبل الرجوع الى رأس تعليمة التكرار.

```
} خوارزم المثال_1_الفصل_14
"النسخة الثانية للإجراء الذي يضاعف محتوى الخانات بالضعف : استعمال التعليمة "منحتى /// 
} اجراء ضاعف_ن2(طبيعي[ ] جط ؛ طبيعي ض)
منحتى( طبيعي ج = 0 ؛ ج > جط.حجم؛ ج = ج + 1) { // المراحل : الأولى، الثانية و الرابعة
جط[ج] = جط[ج]*ض ؛ // المرحلة الثالثة: التعليمات الخاصة بالهدف الأصلي للخوارزم،اي ضرب
القيمة الحالية بالمضاعف ض
}
إجراء عرض الجدول افقيا على الشاشة ///
}اجراء عرض_افقي (طبيعي[] جد_ط1)
"----------------------------------------------------------------------" = سلسلة سطر_التزيين
حرف حد_الخانة = '|' ؛
اكتب_سطر (سطر_التزيين)
}منحتى(طبيعي رقم = 0 ؛ رقم > جد_ط1.حجم ؛ رقم++)
اكتب (حد_الخانة)
اكتب (جد_ط1[رقم]
}
اكتب_سطر (حد_الخانة) // الحد الأخير
اكتب_سطر (سطر_التزيين)
}
الإجراء اساسي ///
} () اجراء اساسي
طبيعي[] جدول_طب1 = {6 ، 12، 24، 48، 5، 10، 25، 25 } ؛
ضاعف_ن2 (جدول_طب1، 3) ؛
عرض_افقي (جدول_طب1) ؛
}
نهاية الخوارزم // {
```

النص 1 : استعمال التعليمة مادام في التجوال

بما انه في غالب الأحيان تتكون المرحلة الأولى من تعليمة واحدة فقط والمرحلة الرابعة من تعليمة واحدة، ادخلت كتابة مختصرة لتعليمة **مادام** وتسمي تعليمة **منحتى.**

تعليمة **منحتى** تحتوي على رأس وجسد، ويحتوي الجسد على المرحلة الثالثة فقط بينما يحتوي رأس تعليمة **منحتى** على المرحلة الأولى، والثانية، والرابعة، وكل هذه المراحل توضع بين قوسين وتكون كتابة التعليمة **منحتى** على الشكل العام التالي:

```
منحتى (المرحلة الأولى؛ العبارة المنطقية المتحكمة في التكرار، المرحلة الرابعة){
        تعليمات جسد التكرار
}
```

النسختين الموجودتين في خطأ: لم يُعثر على مصدر المرجع للإجراء الذي يضاعف محتوى الخانات بالضعف **ض (ضاعف_ن1** و **ضاعف_ن2)** متساويتان تماما في المعنى والهدف، فكلا الإجراءين هدفهما التجوال في جدول من النمط طبعي وتحويل قيمة كل خانة منه الى اضعافها حسب ما يحمله المدخل **ض**، ففي الإجراء **ضاعف_ن1**، استعملنا تعليمة **مادام**، ونرى في هذا النص كل مرحلة منفصلة في الكتابة عن الأخرى، اما في الإجراء **ضاعف_ن1**، استعملنا تعليمة **منحتى**، ونرى المراحل الأولى والثانية والرابعة كلها كتبت في رأس التعليمة.

في خطأ: لم يُعثر على مصدر المرجع، أُعيد كتابة الإجراء **عرض_افقي** الذي رأيناه في الفصل 10 من هذا الجزء الثاني، وفي هذه النسخة استعملنا التعليمة **منحتى** في مكان التعليمة **مادام**، وفي الإجراء **اساسي** نبين كيفية استعمال الوظيفتين **ضاعف_ن2** و**عرض_افقي**، وتنفيذ الخوارزم ينتج على الشاشة ما يبينه خطأ: لم يُعثر على مصدر المرجع.

```
-------------------------------------------------------------------------
| 12 | 24 | 48 | 96 | 10 | 20 | 50 | 50 |
-------------------------------------------------------------------------
```

الشكل 1: نتيجة تنفيذ خوارزم النص 1

تنبيه : الفرق الكبير الذي يجب الانتباه اليه بين التعليمتين "مادام" و "منحتى" :الفرق الكبير بين التعليمتين موجود على مستوى متغيرة التجوال، ففي ما يخص تعليمة "**مادام**"، فمتغيرة التجوال يمكن استعمالها داخل جسد التكرار وخارجه، فقد صرح بها خارج التعليمة، فهي حاضرة في المجال الذي صرحت فيه وكل المجالات الداخلية، اما متغيرة التجوال التابعة لتعليمة "**منحتى**" فهي لا ترى الا في اطار تعليمة **منحتى** ولا يمكن استعمالها خارج التعليمة **منحتى**.

2 – 2 التعليمة "لكل"

في هذه التعليمة لا يهتم كاتب الخوارزم برقم الخانة التي تُزار، ولا اتجاه الزيارة، فما يهمه هو زيارة كل الخانات، ولذا نجد ان كتابة التعليمة "**لكل**" لا تذكر متغيرة التجوال، وتكون على الشكل العام التالي:

```
لكل (نمط_متغيرة اسم_المتغيرة : اسم_الوعاء) {
// تعليمات يمكن ان تستعمل المتغيرة اسم_المتغيرة ؛
{
```

و بالنسبة للجدول تكون على الشكل الخاص التالي:

```
لكل (نمط_متغيرة اسم_المتغيرة : اسم_الجدول){
// تعليمات يمكن ان تستعمل المتغيرة اسم_المتغيرة ؛
{
```

وبمثل هذه التعليمة يمكن التجوال في الجدول، دون استعمال متغيرة التجوال، ففي كل مرحلة من مراحل التجوال، توفر. التعليمة "**لكل**" قيمة الخانة قيد الزيارة في المتغيرة المذكورة في رأس تعليمة "لكل" (مثلا، المتغيرة **ق** في خطأ: لم يُعثر على مصدر المرجع)، فمثلا في خطأ: لم يُعثر على مصدر المرجع، اذا كانت الخانة رقم 6 (اي السابعة) من الجدول "**جط**" هي التي تُزار، ففي هذه المرحلة بالذات تقوم التعليمة "**لكل**" بنقل محتوى الخانة **جط[6]** الى المتغيرة **ق،** وتتم نفس هذه العملية حتى تُزار كل الخانات وتنته بذلك التعليمة "**لكل**".

```
} خوارزم المثال_2_الفصل_14
```

"النص 2 : استعمال التعليمة "لكل

هام جدا: مع مثل هذه التعليمة، ولأن كاتب الخوارزم لا يعلم رقم الخانة التي تزار، فانه لن يكون له بالإمكان تغيير الخانة التي وصلت اليها الزيارة، وهذا يقودنا الى النتيجة التالية وهي ان تعليمة التكرار "**لكل**" لا تستعمل في الحالات التي يريد المستعمل تغيير محتوى الخانات.

3 – خوارزميات تعمير الجداول:

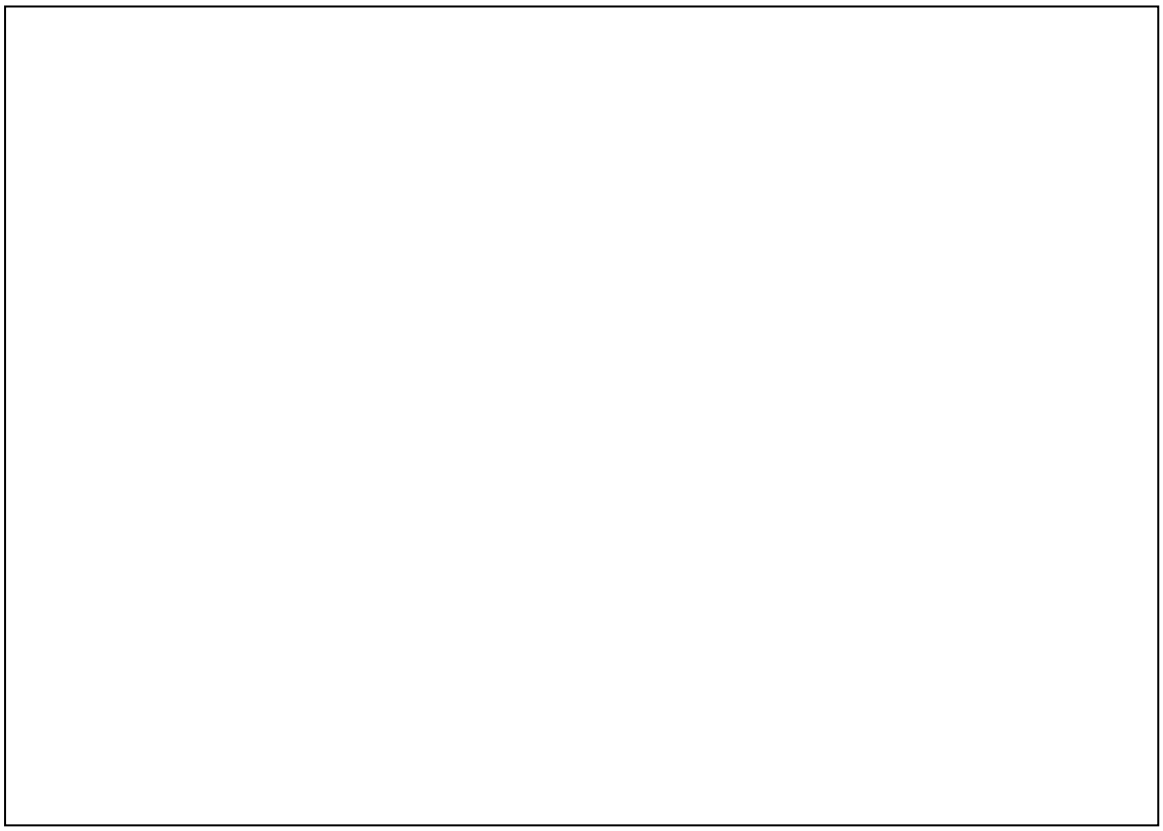
هدف هذه الخوارزميات هي تغذية خانات الجدول بمعطيات تُجلب من مصدر ما، وهناك مصادر عديدة للمعطيات، وعلى سبيل المثال يمكن ان يكون المصدر:

- لوحة المفاتيح.

- خوارزما آخر ينتج المعلومات.
- جدولا آخر.

تعمير الجدول يمكن ان يستهدف كل خانات الجدول او خانات بعينها.

بما ان ابسط عملية في التعمير هي وضع قيمة ما في خانة محددة، فان استعمال رقم الخانة المستهدفة يصبح ضروريا، فالوصول لأي خانة لتغييرها لا يمكن ان يحدث الا باستعمال رقم الخانة، ومعنى هذا انه في مثل هذه الحالة، اي تعمير جدول، سوف لن يكون بإمكاننا استعمال التعليمة "**لكل**"، بل سوف نستعمل اما التعليمة "**مادام**" او التعليمة "**منحتى**".



"النص 3 : تعمير جدول ذو بعد واحد باستعمال تعليمة التكرار "مادام

3 – 1 تعمير جدول انطلاقا من لوحة الحروف:

يقوم خوارزم خطأ: لم يُعثر على مصدر المرجع بتعمير جدول ذو بعد واحد (وهو ما نعتاد بتسميته جدولا دون ذكر البعد) بقيم طبيعية انطلاقا مما يدخله المستعمل بواسطة لوحة الحروف، فهذا الخوارزم، وهو اجراء، يقوم اولا بالتحقق من ان الجدول الذي توضع صلته في المدخل **جط** هو جدول حقيقي، اي انه موجود، وفي مثل هذه الحالة تكون قيمة الصلة **جط** مختلفة عن القيمة **صلة_حرة**، وفي حالة وجود الجدول يتحقق الإجراء من وجود خانات في الجدول، فكل جدول يحتوي على خانات تكون قيمة سمته **حجم** مختلفة عن القيمة 0، فالجدول الذي لا يحتوي على اي خانة تكون فيه قيمة السِمة **حجم** هي 0.

بعد التحق من وجود الجدول ومن وجود خانات في الجدول، تبدأ عملية التعمير انطلاقا من الخانة الأولى في اطار تعليمة التكرار. **مادام،** ونفس الخوارزم أعيد كتابته في خطأ: لم يُعثر على مصدر. المرجع باستعمال التعليمة "**منحتى**".

```
}خوارزم المثال_4_الفصل_14
نـهاية الـخوارزم الـمثالـ4_الـفصل_14 // {
```

"النص 4 : اعادة كتابة النص 3 باستعمال التعليمة "منحتى

النص 5: وظيفة تقوم بإنتاج و تعمير جدول

3 – 2 انتاج جدول معمر من لوحة الحروف

الإجراء السابق يقوم بتعمير جدول يُعطى له من خلال المدخل **جط،** اي لا بد للجدول ان يكون موجودا قبل طلب تنفيذ الإجراء **عمر_جدول_من_لوحة_المفاتيح،** وهذا ما نجده على مستوى الإجراء **اساسي** في خطأ: لم يُعثر على مصدر المرجع، حيث يقوم الإجراء اساسي اولا بإنشاء الجدول ثم يطلب تعميره، وهذه المرة، الجدول الذي يُعمر غير موجود عند طلب تنفيذ الخوارزم **عمر_جدول_من_لوحة_المفاتيح** ، وهذا ما نراه من خلال الإجراء اساسي (السطر 21 من خطأ: لم يُعثر على مصدر المرجع)، ففي خطأ: لم يُعثر على مصدر المرجع تقوم الوظيفة **عمر_جدول_من_لوحة_المفاتيح** بإنتاج الجدول (السطر 11) ثم تعميره (من السطر 14 الى السطر.

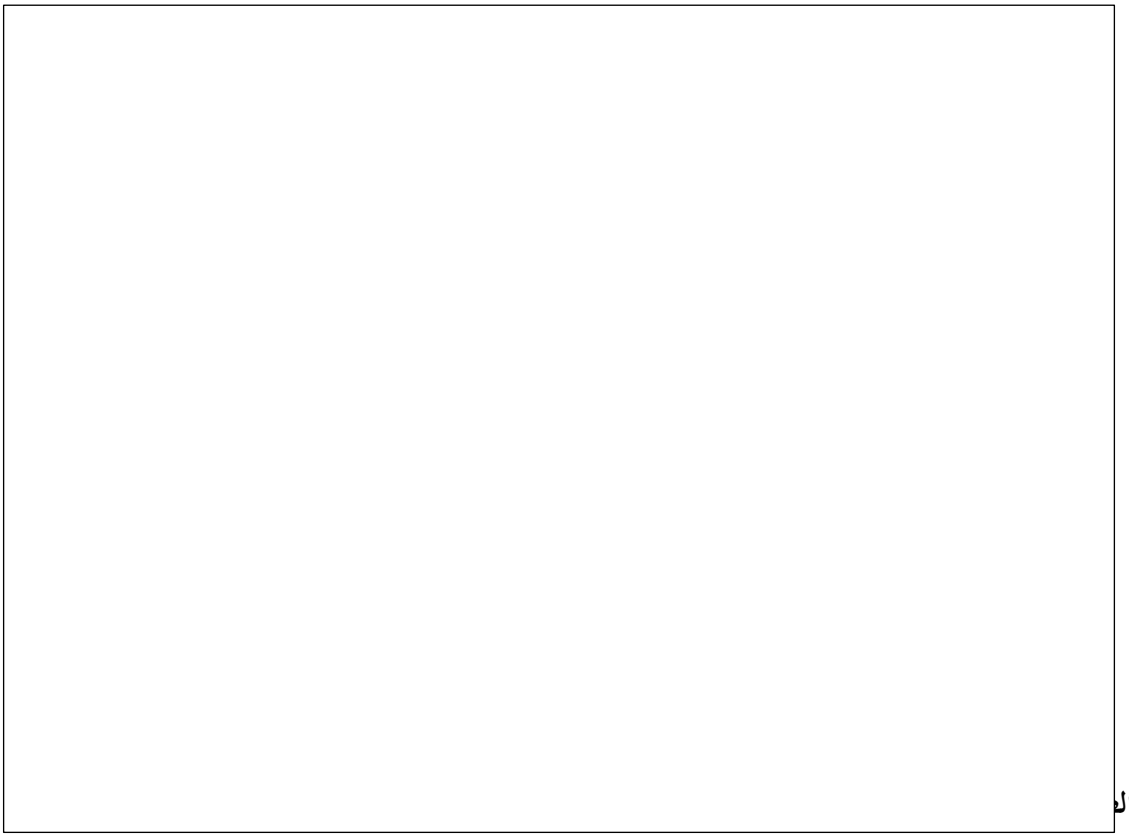
17) وفي الأخير تُرجع الوظيفة الجدول الذي انتجته وعمرته الى من طلب تنفيذه (السطر 18)، ولإنتاج الجدول تحتاج الوظيفة الى معلومة كي تمارس عملها، وهذه المعلومة هي حجم الجدول الذي يجب انتاجه وتعميره (السطر 1، المدخل "ح").

3 – 3 تعمير جدول انطلاقا من محتوى جدول آخر

تعمير جدول، ونسميه الهدف، انطلاقا من جدول آخر، نسميه المنبع، يمكن ان يكون على اشكال عدة، وفي كل هذه الأشكال يجب ان يكون المنبع موجودا قبل طلب تنفيذ الخوارزم الذي يقوم بالتعمير، واهم هذه الأشكال:

الشكل الأول: الهدف موجود، اي انه قد تم انشاؤه قبل طلب تنفيذ خوارزم التعمير، وعند طلب تنفيذ خوارزم التعمير تُوفر له في مداخله صلة المنبع وصلة الهدف (السطر 1 من خطأ: لم يُعثر على مصدر المرجع)، فيقوم الخوارزم بنقل محتوى المنبع الى الهدف، وفي مثل هذا الشكل يظهر إشكال متعلق بحجم الهدف، فان كان حجم الهدف اصغر من حجم المنبع، سوف لن يتم نقل كل القيم الموجودة في خانات المنبع، ومعنى هذا ان الخوارزم الذي يعمل على هذا الشكل لن يكون فعالا في بعض الأحيان.

يمثل هذا الشكل الأول الإجراء **عمر_جدول_من_جدول** (خطأ: لم يُعثر على مصدر المرجع) ورأس هذا الإجراء يظهر في السطر 1 من خطأ: لم يُعثر على مصدر المرجع، ففي بدايته، يقوم الإجراء من تصفح محتوى الصلتين اللتين استعملتا في المدخلين "**المنبع**" و "**الهدف**" (من السطر 3 الى السطر 13)، وفي حالة عدم وجود المنبع او الهدف ينته الخوارزم.



النص 6: اجراء تعمير جدول انطلاقا من جدول آخر

بعد التأكد من وجود المنبع والهدف تبدأ عملية تعمير الهدف انطلاقا من المنبع، وفي كل الحالات، الا واحدة، يكون عدد الخانات التي تعمر هي بحجم عدد خانات المنبع، والحالة الوحيدة التي يتم فيها تعمير عدد مختلف عن حجم المنبع هي الحالة التي يكون فيها حجم الهدف اقل من حجم المنبع، وفي هذه الحالة عدد الخانات التي يتم تعميرها هي بحجم عدد خانات الهدف، والفصل في عدد الخانات التي يجب تعميرها تتكفل به تعليمات الأسطر من 16 الى 17.

ملاحظة هامة: نلاحظ في السطر 17 والسطر 19 من خطأ: لم يُعثر على مصدر المرجع، ان جسم التعليمية الشرطية (السطر 17) يحتوي تعليمة واحدة، ففي هذه الحالة يمكن ان لا نستعمل حاضنتي بداية ونهاية جسم التعليمة الشرطية، ولبساطة تعليمة الجسم، كتبناها في نفس السطر التي كتب فيه رأس التعليمة الشرطية، ونفس الطريقة استعملت مع تعليمة التكرار. **"منحتى" (السطر. 19).**

الشكل الثاني : المنبع والهدف موجودان ويظهران في المداخل (السطر 1 من خطأ: لم يُعثر على مصدر المرجع)، وفي حالة ما اذا كان حجم الهدف اصغر من حجم المنبع، يستبدل الهدف بجدول آخر ينشأ في الخوارزم ويكون مرتبطا بالصلة **"الهدف"**، وحتى يتمكن طالب تنفيذ الخوارزم من معرفة صلة الهدف الذي أنشأ، لا بد من ارجاعه في نهاية تنفيذ الخوارزم (السطر 13 من خطأ: لم يُعثر على مصدر المرجع) فالصلة التي ترجع تدل على الهدف الحقيقي الذي تم تعميره، فيمكن ان تكون القيمة هي قيمة الصلة الأصلية للهدف او قيمة الصلة الجديدة للهدف الجديد الذي انشأ لكي يستوعب قيم كافة خانات المنبع، ومعنى كل هذا ان خوارزم التعمير يجب ان يكون على شكل وظيفة كما يظهر في السطر 1 من خطأ: لم يُعثر على مصدر المرجع.

النص 7: وظيفة تعمير جدول انطلاقا من جدول آخر: المنبع و الهدف في المداخل

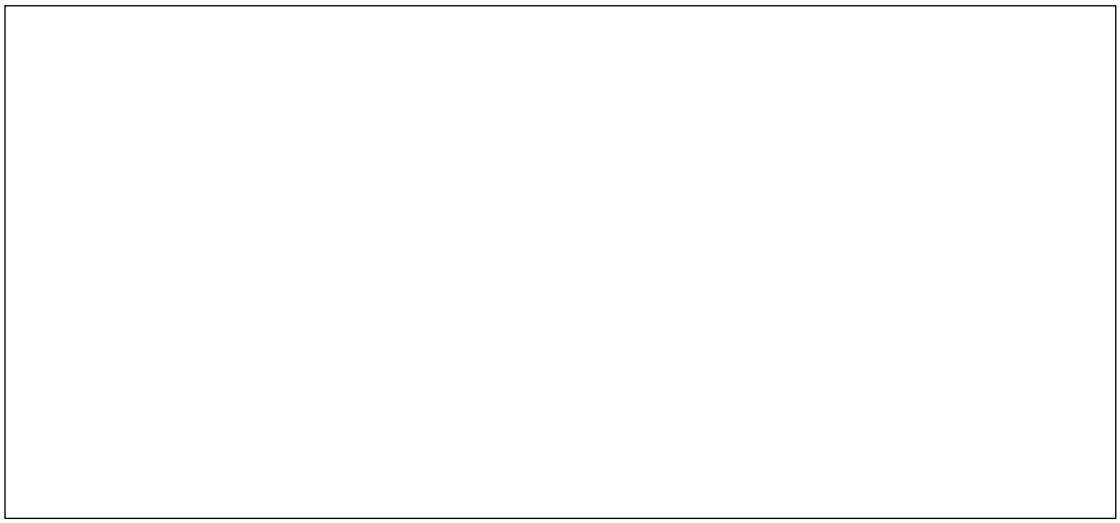
في البداية تتأكد الوظيفة **عمر_جدول_من_جدول** (خطأ: لم يُعثر على مصدر المرجع) من وجود المنبع، فان غاب المنبع لا يمكن اتمام العملية، وفي هذه الحالة تُرجع الوظيفة القيمة **صلة_حرة** (السطر 3) وهذه القيمة تبين ان الوظيفة لا يمكنها تعمير جدول انطلاقا من جدول غير موجود، ونلاحظ في كتابة التعليمة الشرطية (السطر 3)، اننا لم نستعمل حرفي الحد للتدليل على بداية جسم التعليمة الشرطية كما رأينا سابقا.

بعد التأكد من وجود المنبع، تقوم الوظيفة **عمر_جدول_من_جدول** بتصفح صلة الهدف، فان كان الهدف غير موجود، اي ان قيمة الصلة هي **صلة_حرة**، وعكس خوارزم خطأ: لم يُعثر على مصدر المرجع، يقوم خوارزم خطأ: لم يُعثر على مصدر المرجع بإنشاء الهدف (السطر 5).

بعد ما اصبع المنبع والهدف موجودان، يتأكد الخوارزم من سعة الهدف (السطر 8)، فان كان حجم الهدف اقل من المنبع، يقوم الخوارزم بإنشاء هدف آخر قادر على احتواء كافة قيم المنبع.

ابتداءا من السطر 10 تكون كل الشروط قد هُيأت لإنجاح عملية تعمير الهدف، وبعد اتمام التعمير يُرجع الخوارزم قيمة الصلة المرتبطة بالهدف.

الشكل الثالث: الهدف غير موجود عند طلب تنفيذ وظيفة التعمير ولذا يحذف المدخل الخاص به (السطر 1 من خطأ: لم يُعثر على مصدر المرجع)، فيجب انشاؤه داخل الخوارزم (السطر 5) ثم تبدأ عملية التعمير(السطر 8)، ومع مثل هذا الشكل الثالث لن يكون هناك اي اشكال فيما يخص خاصية حجم الهدف، فالهدف سوف ينشأ بحجم المنبع.



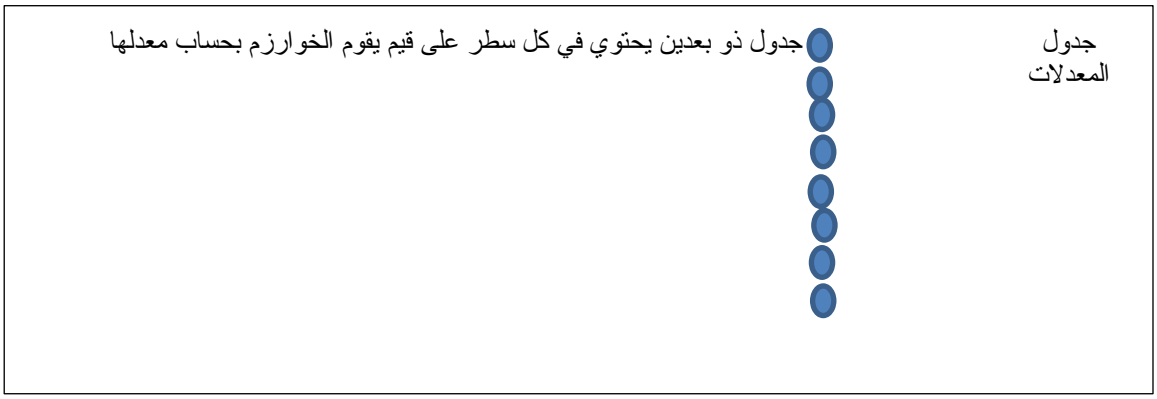
النص 8: وظيفة تعمير جدول انطلاقا من جدول آخر: فقط المنبع في المدخل

3 – 4 تعمير جدول بمعطيات تُنتج إثر عمليات حسابية

يمكن ان تكون العملية الحسابية بسيطة جدا او معقدة جدا، فعلى سبيل المثال، تعمير جدول بقيم هي ضعف القيم التي يحتويها تعتبر عملية بسيطة جدا (خطأ: لم يُعثر على مصدر المرجع)، اما تعمير جدول بمعدلات قيم موجودة في عدة جداول اخرى، يمكن ان تكون معقدة نسبيا، كتعمير جدول معدلات انطلاقا من جدول ذو بعدين يحتوي على القيم التي يجب حساب معدلاتها (خطأ: لم يُعثر على مصدر المرجع)، وفيما يلي ندرس خوارزم تعمير جدول المعدلات في حالته البسيطة.



النص 9: اجراء لمضاعفة قيم خانات جدول طبيعي



الشكل 2: تحتوي كل خانة من جدول المعدلات على معدل القيم الموجودة في السطر المقابل في الجدول ذو بعدين

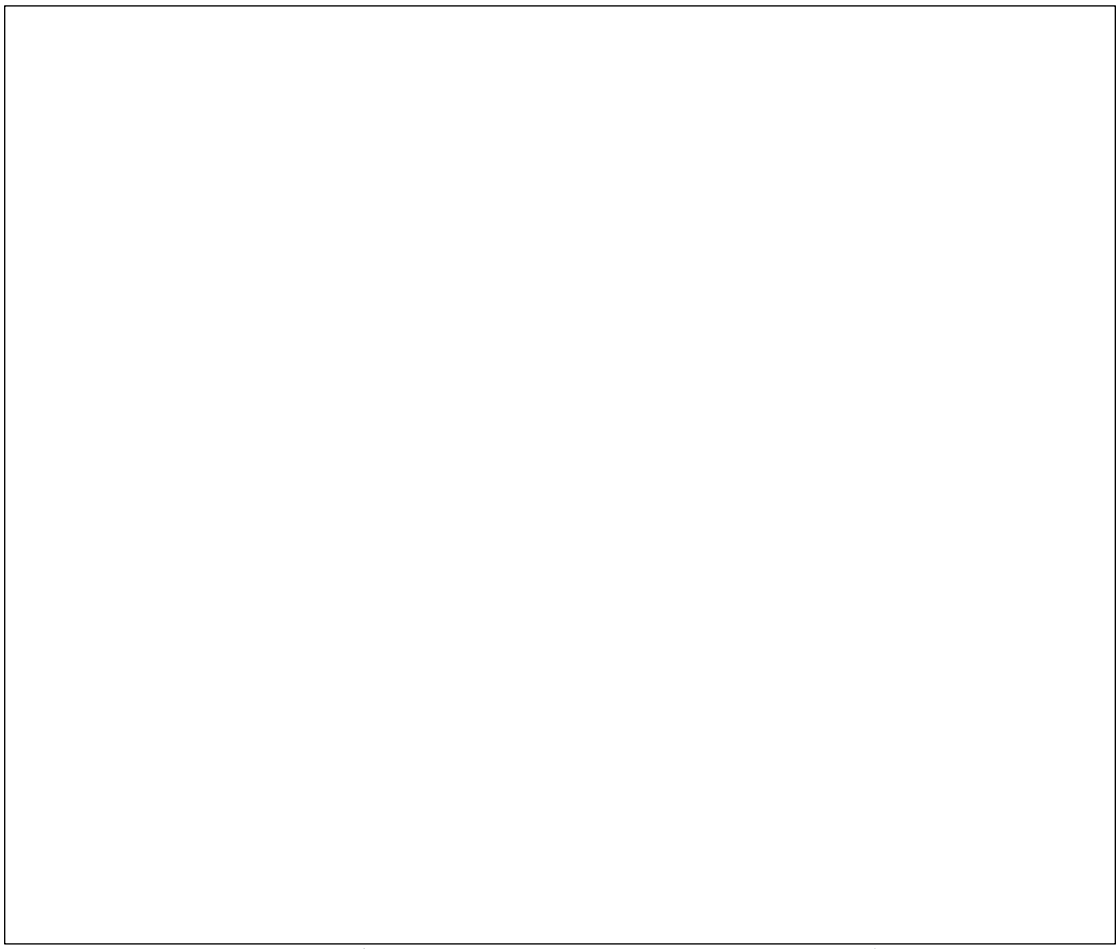
3 – 5 تعمير جدول المعدلات انطلاقا من جدول النقاط

يقوم خوارزم خطأ: لم يُعثر على مصدر المرجع بحساب معدلات قيم موجودة في جدول ذو بعدين، فيمكن ان تكون القيم الموجودة في الجدول ذو بعدين هي قيم تمثل النقاط التي تحصل عليها طالب ما في مواد مختلفة، فمثلا يمكن لكل سطر من اسطر الجدول ذو بعدين ان يمثل النقاط التي تحصل عليها الطالب في مادة ما، وقد راينا فيما سبق مثل هذا الجدول.

يحتوي خوارزم خطأ: لم يُعثر على مصدر المرجع على وظيفتين:

- الوظيفة "**معدل**" و تقوم بحساب معدل قيم موجودة في جدول ذو بعد واحد.
- الوظيفة " **احسب_المعدلات**" وتقوم بإنتاج جدول ذو بعد واحد يحتوي على معدلات قيم موجودة في جدول ذو بعدين، فكل خانة من الجدول المنتج، وهو ذو بعد واحد، تحتوي على معدل القيم الموجودة في السطر المقابل لها في الجدول ذو بعدين، ولبلوغ هدفها

تستغل هذه الوظيفة (اي " **احسب_المعدلات"**) الوظيفة "**معدل"،** فتعطيها تباعا اسطر الجدول ذو بعدين وهذا في اطار التكرار بالتعليمة "**منحتى**" (السطر 19، خطأ: لم يُعثر على مصدر. المرجع).



النص 10: وظيفة تعمير بمعدلات قيم موجودة في جدول ذو بعدين

في كلتا الوظيفتين نرى في اول الأمر تعليمات هدفها التأكد من حقيقة ما وضع في المداخل، اي هل ما وضع في المدخل صلة جدول ام صلة حرة.

في الوظيفة معدل، تقوم تعليمات السطر 5 بالتجوال في الجدول وجمع كل القيم الموجودة في خانات الجدول، ونتحصل على النتيجة في المتغيرة "**م**" التي انشأت في السطر 4 بقيمة اولية هي 0، وبعد الحصول على مجموع قيم الخانات، يقسم المجموع بعدد الخانات في عبارة السطر 6، وتنته الوظيفة بإرجاع نتيجة العبارة في نفس السطر.

تقوم الوظيفة **احسب_المعدلات** في السطر 15 بإنشاء الجدول الذي سوف يحتوي على المعدلات، وحجم هذا الجدول هو حجم اسطر الجدول ذو بعدين الموجود في السِمة "**المنبع.حجم**" وفي السطر 19، في اطار تعليمة التكرار يتم التجوال في خانات جدول المعدلات وفي اسطر الجدول ذو بعدين، كلما انتقلنا الى خانة من خانات جدول المعدلات ننتقل الى السطر المقابل في الجدول ذو بعدين

و في اطار هذا التجوال تنفذ التعليمة **"الهدف[رقمخ] = معدل(المنبع[رقمخ])"** التي تُستغل فيها الوظيفة **"معدل".**

الفصل الخامس عشر

الأوعية: العمليات المعتادة على الجداول: عمليات البحث

1 – مقدمة:

تعد عمليات البحث عن قيمة ما في الجداول من العمليات الشائعة في عالم الخوارزميات، فهناك انواع عدة من هذه الخوارزميات و اهمها:

- البحث عن قيمة معينة ما في الجدول
- البحث عن عدد المرات التي تظهر فيها قيمة معينة في جدول ما
- البحث عن اصغر (او اكبر) قيمة في الجدول

1 – البحث عن قيمة معينة ما في الجدول

يقوم مثل هذا الخوارزم بالبحث عن قيمة ما في جدول ما، فاذا كانت القيمة موجودة في خانة او اكثر، يرجع الخوارزم قيمة طبيعية تمثل موقع اول خانة في الجدول تحتوي على القيمة المبحوث عنها، وان لم توجد القيمة يرجع الخوارزم قيمة لا يمكن ان تكون موقعا لأي خانة، كالقيمة –1، ورأس مثل هذا الخوارزم له شكل مرتبط بنمط القيمة التي يبحث عنها، فمثلا :

النص 1: وظيفة البحث عن قيمة طبيعية في جدول طبيعي ذو بعد واحد

- اذا كان البحث عن قيمة طبيعية في جدول من النمط طبيعي، يكون رأس مثل هذا الخوارزم في الغالب على الشكل التالي: **طبيعي ابحث(طبيعي ط، طبيعي[ ] جط)**
- اذا كان البحث عن قيمة حقيقية في جدول من النمط حقيقي، يكون رأس مثل هذا الخوارزم في الغالب على الشكل التالي: **طبيعي ابحث(حقيقي ح، حقيقي[ ] جح)**

- اذا كان البحث عن اسم او اي سلسلة من الحروف، في جدول يحتوي على سلاسل من الحروف، يكون رأس مثل هذا الخوارزم في الغالب على الشكل التالي: **طبيعي ابحث(سلسلة س، سلسلة[ ] جس)**

يحتوي خوارزم خطأ: لم يُعثر على مصدر المرجع على الوظيفة "**ابحث**" و الإجراء "**اساسي**"، وتقوم الوظيفة "**ابحث**" بالبحث عن قيمة طبيعية (تُعطى لها عبر المدخل "**ق**") في جدول طبيعي (يوصف لها عبر المدخل "**جدط**") ، ويبين الإجراء "**اساسي**" كيفية استغلال الوظيفة "**ابحث**"، ويبين خطأ: لم يُعثر على مصدر المرجع نتائج تنفيذ خوارزم خطأ: لم يُعثر على مصدر المرجع.

```
رسالة 1: موقع القيمة 81 في الجدول هو: 8
رسالة 2: موقع القيمة 56 في الجدول هو: 2
رسالة 3: موقع القيمة 54 في الجدول هو: 5
رسالة 4: موقع القيمة 55 في الجدول هو: -1
```

الشكل 1 : نتيجة تنفيذ خوارزم النص 1 على الشاشة

النص 2: البحث عن عدد المرات التي تظهر فيها قيمة طبيعية في جدول طبيعي ذو بعد واحد

2 – البحث عن عدد المرات التي تظهر فيها قيمة معينة في جدول ما.

هذه المهمة تقوم بها وظيفة ترجع قيمة طبيعية، وهذه القيمة تكون دائما اكبر او تساوي الصفر، فلا يمكن ان يكون هذا العدد سالبا، وفي حالة ما اذا واجهت الوظيفة اشكالا يمنع عملها، كإعطائها

جدولا غير موجود، فإنها ترجع القيمة –1، وتعني هذه القيمة ان مُستغل الوظيفة لم يضع في مدخلها جدولا.

يحتوي خوارزم خطأ: لم يُعثر على مصدر المرجع على الوظيفة "**عدد_مرات**" و الإجراء "**اساسي**"، وتقوم الوظيفة "**عدد_مرات**" بالبحث عن قيمة طبيعية (تعطى لها عبر المدخل "**ق**") في جدول طبيعي (يوصف لها عبر المدخل "**جدط**")، و في سياق التجوال في الجدول، كلما وُجدت خانة تحتوي على القيمة التي وصفت في المدخل **ق**، يضاف واحد الى المتغيرة **عداد** التي صرح بها بقيمة اولية تساوي 0، ويبين الإجراء "**اساسي**" كيفية استغلال الوظيفة "**عدد_مرات**"، ويبين خطأ: لم يُعثر على مصدر المرجع نتائج تنفيذ خوارزم خطأ: لم يُعثر على مصدر المرجع.

```
رسالة 1: عدد المرات التي تظهر فيها القيمة 81 في الجدول هو: 1
رسالة 2: عدد المرات التي تظهر فيها القيمة 13 في الجدول هو: 0
رسالة 3: عدد المرات التي تظهر فيها القيمة 78 في الجدول هو: 3
رسالة 4: عدد المرات التي تظهر فيها القيمة 79 في الجدول هو: -1
```

الشكل 2 : نتيجة تنفيذ خوارزم النص 2 على الشاشة

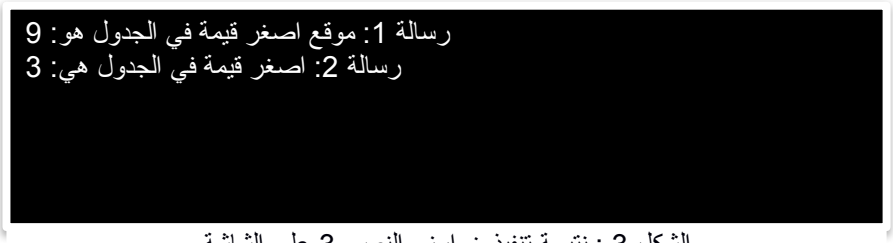
3 – البحث عن اصغر (او اكبر) قيمة في الجدول

هذه المهمة تقوم بها وظيفة ترجع **موقع الخانة** المحتوية على القيمة الأصغر (او الأكبر) في الجدول، والموقع في الجدول قيمة طبيعية اكبر او تساوي الصفر، فلا يمكن ان يكون رقم الموقع سالبا، وفي حالة ما اذا واجهت الوظيفة اشكالا يمنع عملها، كإعطائها جدولا غير موجود، فإنها ترجع القيمة السالبة –1، وتعني هذه القيمة ان مستغل الوظيفة لم يضع في مدخلها جدولا.

ملاحظة: اذا ترددت القيم الأصغر (او الأكبر) في الجدول، فالموقع الذي ترجعه هذه الوظيفة هو موقع اول ظهور للقيمة الأصغر (او الأكبر) في الجدول.

الصفح

النص 3: البحث عن موقع اصغر قيمة في جدول طبيعي ذو بعد واحد



الشكل 3 : نتيجة تنفيذ خوارزم النص 3 على الشاشة

3 – 1 سلوك الوظيفة الأصغر: عند انطلاقها، تفترض الوظيفة ان اول خانة تحتوي على اصغر قيمة (السطرين 7 و 8 من خطأ: لم يُعثر على مصدر المرجع)، وانطلاقا من هذا الافتراض تقوم الوظيفة بوضع القيمة 0 في المتغيرة **موقع_الأصغر،** ثم ينطلق التجوال في الجدول من الخانة الثانية و رقمها 1 (السطر 11) في اطار تعليمة التكرار **"منحتى".** وكلما وصل التجوال الى خانة ما، يقوم الخوارزم بمقارنة قيمة الخانة قيد الزيارة بالخانة التي تحتوي على اصغر قيمة(السطر 12)، فإذا كانت قيمة الخانة المُزارة اكبر او تساوي الخانة التي تحتوي على اصغر قيمة (المشار اليها بمحتوى المتغيرة **موقع_الأصغر)**، تنتقل الوظيفة الى الخانة التالية، فتعاد المقارنة، اما اذا تبين ان قيمة الخانة المزارة اصغر من الخانة المشار اليها بالمتغيرة **موقع_الأصغر.** تقوم الوظيفة هنا بتحديث محتوى المتغيرة **موقع_الأصغر.** لتحتوي رقم الخانة المزارة التي اصبحت تحتوي على اصغر قيمة من بين قيم الخانات التي تمت زيارتها، ثم تنتقل الى الخانة التالية فتعاد المقارنة.

عند الانتهاء من زيارة كل الخانات، تنته عملية التكرار التي تتكفل بترتيب زيارة الخانات، وعند انتهائها تكون المتغيرة **موقع_الأصغر** حاملة لرقم الخانة التي تحتوي على اصغر قيمة، فترجع الوظيفة محتوى المتغيرة **موقع_الأصغر**.

4 – البحث في الجداول المرتبطة بانماط مهيكلة:

الوظائف التي قمنا بإنجازها لحد الآن تتعامل مع الأعداد الطبيعية، واذا اردناها ان تتعامل مع الأعداد الحقيقية، علينا بإنجاز هذه الوظائف ولا تختلف هذه الوظائف من سابقاتها الا بنمط الجدول ونمط القيمة التي يبحث عنها.

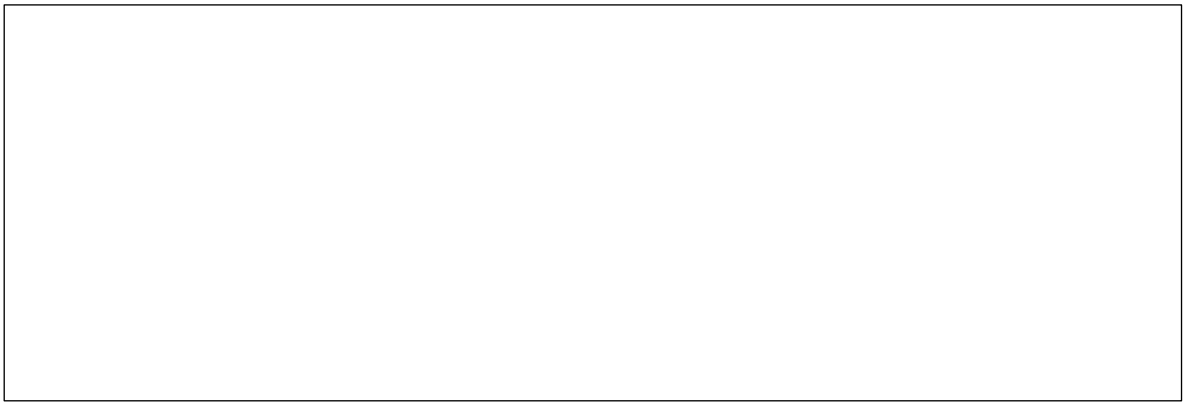
اما ان كان الجدول يحتوي على قيم من انماط مهيكلة، ونسميها قيم مهيكلة او معقدة، فان خوارزميات البحث في مثل هذه الجداول مختلفة عن خوارزميات البحث في الجداول المحتوية على معطيات من انماط قاعدية (او اساسية، او بدائية)، والاختلاف مع ما مضى من خوارزميات البحث يكون في الملاحظات التالية:

- عند البحث عن معلومة مهيكلة، يجب مقارنة ما في الجدول مع ما نبحث عنه، والمقارنة بين متغيرتين مهيكلتين لا يمكن ان تنجر باستعمال عملية المقارنة (==) التي تُستعمل مع الأنماط البدائية، وقد اوضحنا هذا الأمر في الفصل الرابع من هذا الجزء الثاني.
- انطلاقا من الملاحظة السابقة، يجب علينا التعريف الدقيق لعملية المقارنة بين قيمتين مهيكلتين، ثم انجاز هذه العملية على شكل وظيفة منتسبة او غير منتسبة، فمثلا اذا كان النمط المهيكل يصف مستطيلا فما معنى المقارنة بين مستطيلين، هل هي مقارنة بين موقعي المستطيلين في الرسم، ام هي مقارنة حجم المستطيلين ام هي مقارنة مساحة المستطيلين، ونفس التساؤلات تطرح مع قيمة مهيكلة تصف شخصين او طاولتين او سيارتين او عددين مركبين او عددين كسرين.
- عند البحث عن معلومة مهيكلة في جدول ما، وعكس ما هو عليه الحال مع جداول الأنماط البداية، يجب علينا قبل اي مقارنة ان نتحقق من الوجود الفعلي للمعلومة المهيكلة في الجدول، ووجود معلومة مهيكلة في خانة من خانات الجدول يكون بوجود قيمة مختلفة عن القيمة **صلة_حرة** في خانة الجدول، فاذا كانت قيمة الخانة هي القيمة المتعارف عليها والمسماة بالقيمة **صلة_حرة** ، فهذا معناه ان الخانة لا تحتوي على اي قيمة مهيكلة، وبكلام ادق، فان الخانة لا تحتوي على قيمة تمكن من الوصول الى متغيرة مهيكلة مرتبطة بخانة الجدول، اما اذا كانت قيمة الخانة مختلفة عن القيمة **صلة_حرة** فهذا يعني ان الخانة مرتبطة بمعلومة مهيكلة.

4 – 1 المثال الأول: البحث عن عدد المرات التي يظهر فيها عدد مركب في جدول للأعداد المركبة:

هذه المهمة تقوم بها وظيفة ترجع قيمة طبيعية، وهذه القيمة تكون دائما اكبر او تساوي الصفر، فلا يمكن ان يكون هذا العدد سالبا، وفي حالة ما اذا واجهت الوظيفة اشكالا يمنع عملها، كإعطائها جدولا غير موجود او قيمة مهيكلة غير موجودة، ففي حالة الخطأ ترجع الوظيفة اما القيمة –1 او القيمة –2، فالقيمة –1 تبين ان الجدول الذي اعطي لها غير موجود، وتعني هذه القيمة ان مستغل الوظيفة لم يضع في مدخلها جدولا، ونفس المنطق مع القيمة –2 التي تبين ان مستغل الوظيفة لم يضع

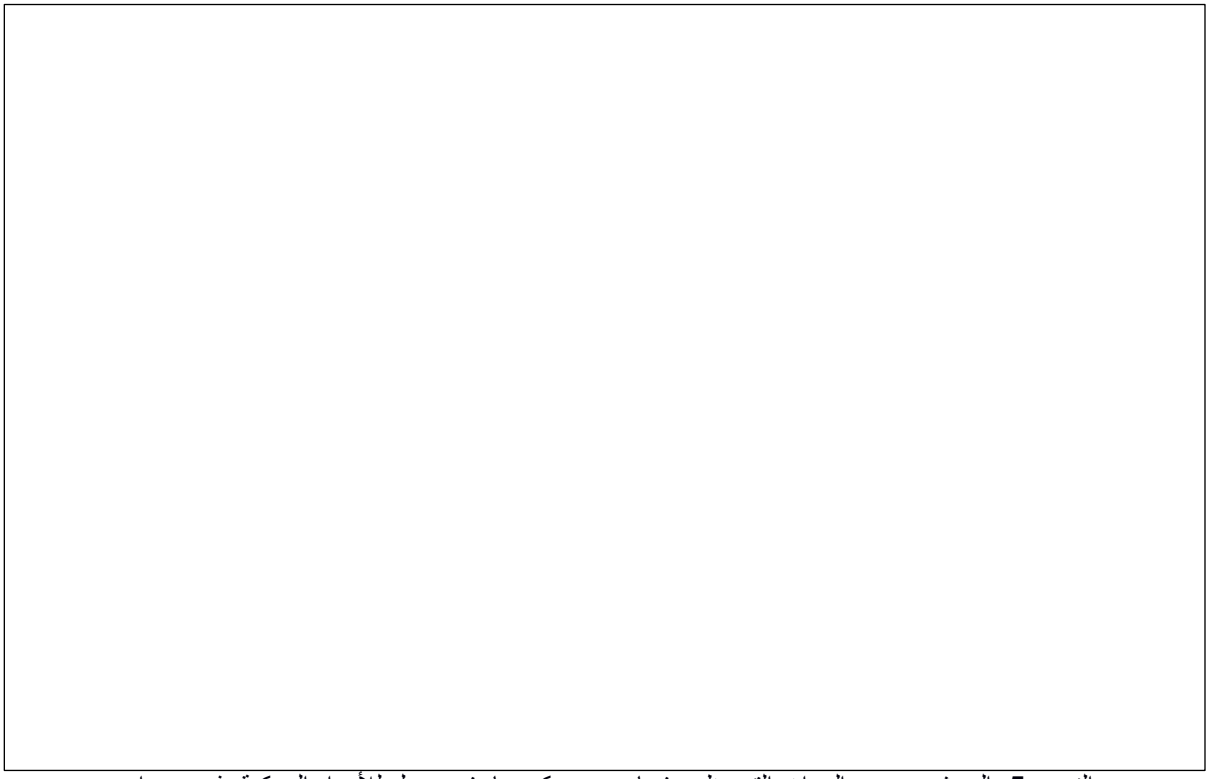
في مدخلها الأول قيمة مركبة، بل وضع صلة حرة.



النص 4: التعريف بنمط يمثل الأعداد المركبة

4 – 1 – 1 تعريف النمط الممثل للأعداد المركبة

بما ان الجدول الذي تتم فيع عملية تعداد الأعداد المركبة هو جدول لنمط مهيكل يمثل الأعداد المركبة، علينا اولا ان نُعرف بالنمط الممثل للأعداد المركبة، وهذا ما نراه في خطأ: لم يُعثر على مصدر. المرجع، وفيه نلاحظ ان النمط **عدد_مركب** يحتوي على مشيد واحد وعلى الوظيفة المنتسبة "**يساوي**" التي تمكن من مقارنة عددين مركبين: العدد الذي من خلاله تنفذ الوظيفة **يساوي** والعدد الذي يذكر في مدخل الوظيفة **يساوي**.



النص 5: البحث عن عدد المرات التي يظهر فيها عدد مركب ما في جدول للأعداد المركبة ذو بعد واحد

4 – 1 – 2: خوارزم البحث عن عدد المرات التي يظهر فيها عدد مركب في جدول للأعداد المركبة

يحتوي خوارزم خطأ: لم يُعثر على مصدر المرجع على الوظيفة **عدد_مرات** و الإجراء **اساسي**، وتقوم الوظيفة **عدد_مرات** بالبحث عن عدد مركب (يعطى لها عبر المدخل **عم**) في جدول للأعداد المركبة (يوصف لها عبر المدخل **جعم**)، و في سياق التجوال في الجدول، كلما وُجدت خانة تحتوي على عدد مركب يساوي العدد المركب الذي وصف في المدخل **عم**، يضاف واحد الى المتغيرة **عداد** التي صرح بها بقيمة اولية تساوي 0 في السطر. 8، ويبين الإجراء "**اساسي**" كيفية استغلال الوظيفة "**عدد_مرات**"، ويبين خطأ: لم يُعثر على مصدر المرجع نتائج تنفيذ خوارزم خطأ: لم يُعثر على مصدر المرجع.

```
رسالة 1: عدد المرات التي تظهر فيها القيمة المركبة (8.5، 16.0) في الجدول هو: 2
رسالة 2: عدد المرات التي تظهر فيها القيمة المركبة (96.5، 76.87) في الجدول هو: 1
```

الشكل 4 : نتيجة تنفيذ خوارزم النص 5 على الشاشة

نلاحظ على مستوى الوظيفة **عدد_مرات** وفي السطر. 12، استعمال الوظيفة المنتسبة **يساوي** بدل عملية المقارنة بالتساوي == التي لا تصلح الا مع المتغيرات والقيم البدائية، فمنطق المساواة بين عددين مركبين تحمله الوظيفة المنتسبة **يساوي**.

على مستوى الإجراء اساسي، يحتوي السطران 19 و 20 (السطر 20 هو تتمة لما في السطر 19) على تصريح بقيم اولية للجدول **جدول_عم**، وهو جدول للأعداد المركبة، وقد رأينا في الفصل السابع من هذا الجزء الثاني كيفية التصريح بقيم أولية لجداول الأنماط ال مهيكلة.

في السطر 21، وبعد انشاء المتغيرة **ع**، يقوم الإجراء **اساسي** بطلب تنفيذ الوظيفة **عدد_مرات،** ويضع في المدخل الأول العدد المركب (8.5، 16.0) الذي يتم انشاؤه في اطار عملية طلب تنفيذ الوظيفة **عدد_مرات،** اما في المدخل الثاني فيضع الجدول **جدول_عم** الذي أُنشأ في السطرين 19 و 20.

في السطر 24 يُطلب ثانية تنفيذ الوظيفة **عدد_مرات،** وهذه المرة يضع الإجراء **اساسي** المتغيرة المهيكلة **عم** في المدخل الأول للوظيفة **عدد_مرات،** وقد أُنشأت المتغيرة المهيكلة **عم** في السطر 23 على الشكل الذي حدده النمط **عدد_مركب**.

4 – 1 – 3 تحسين الإجراء اساسي في خوارزم خطأ: لم يُعثر على مصدر المرجع

نريد احداث تغيير طفيف في السطر 23 من خطأ: لم يُعثر على مصدر المرجع، وهذا التغيير هو تغيير القيم التي وصفت للمشيد، اي نحول الكتابة "**شيد عدد_مركب(96.5، 76.87)؛**" الى "**شيد عدد_مركب(10,0، 20,0)؛**"، ثم نطلب اعادة تنفيذ الخوارزم، وهنا نتفاجأ بالنتيجة التي تظهر في خطأ: لم يُعثر على مصدر المرجع، فنحن نريد معرفة عدد المرات التي يظهر فيها العدد المركب (

10.0، 20.0)، لكن ما ظهر في خطأ: لم يُعثر على مصدر المرجع غير الذي اردناه، زيادة على ذلك فان القيمة المركبة (96.5، 76.87) موجودة حقا في الجدول لكن خطأ: لم يُعثر على مصدر المرجع يتحدث بغير ذلك، فاين هو الخلل؟

```
رسالة 1: عدد المرات التي تظهر فيها القيمة المركبة (8.5، 16.0) في الجدول هو: 2
رسالة 2: عدد المرات التي تظهر فيها القيمة المركبة (96.5، 76.87) في الجدول هو: 0
```

الشكل 5 : نتيجة تنفيذ خوارزم النص 5 على الشاشة بعد التغيير الطفيف في قيم المتغيرة عم1

الطريقة التي اتبعت في كتابة خطأ: لم يُعثر على مصدر المرجع هي السبب وراء هذا الخلل، فعندما احدثنا التغيير في السطر 23، كان علينا ان نغير كل الأسطر المرتبطة بشكل مباشر بالقيم الثابتة التي ذكرت صراحة في السطر 23، ففي خطأ: لم يُعثر على مصدر المرجع يوجد سطر واحد، وهو السطر 24، فكان علينا ان نحدث تغييرا في السطر 24 بعد تغيير السطر 23، وهكذا، واستنادا الى الطريقة التي اتُبعت في كتابة خطأ: لم يُعثر على مصدر المرجع ، كلما اردنا احداث تغيير في السطر 23، علينا ان نسدد السطر 24.

فيما يخص خطأ: لم يُعثر على مصدر المرجع ، حل هذا الإشكال غير مكلف وغير مقلق جدا، لكن لو كان في الخوارزم اكثر من سطر مرتبط بالقيمة الثابتة للسطر 23، فان الإشكال يصبح مكلف ومقلق ومصدر هام للأخطاء.

السؤال الذي نطرحه الآن: هل هناك طريقة تمكننا من تفادي المشقة المكلفة عندما نريد احداث تغيير في السطر 23 دون اللجوء الى احداث تغييرات في الأسطر الأخرى؟ الجواب قد اشرنا اليه اكثر من مرة في الفصول السابقة: علينا ان نتجنب ،ما امكن، استعمال القيم الثابتة في كتابة نصوص الخوارزميات، فمن الأفضل استعمال متغيرات او متغيرات ثابتة، فمثلا في خطأ: لم يُعثر على مصدر المرجع كان علينا في السطر 24 استعمال المتغيرة المهيكلة **عم1** التي انشأت في السطر 23,

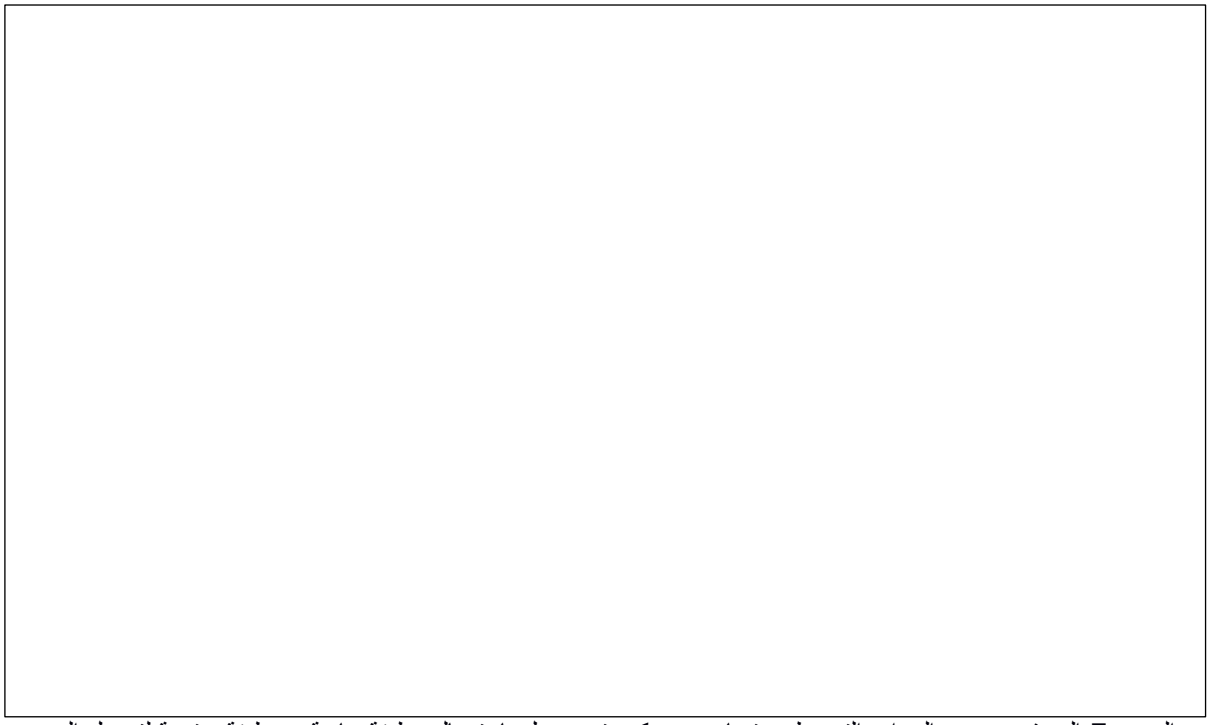
المشكل المطروح في استعمال المتغيرة **عم1** في السطر 23 يكمن في الكيفية، فالمتغيرة **عم1** ليست من المتغيرات البدائية التي يعرف الخوارزم كيف يحولها الى سلسلة حروف عندما تذكر للتعليمة اكتب او عندما تظهر في عبارة تكون نتيجتها سلسلة من الحروف.

```
الص
```

النص 6: اثراء النمط عدد_مركب بوظيفة منتسبة تظهر عددا مركبا على شكل سلسلة حروف

الإجابة على المشكل المطروح يمكن ان تكون على الأقل على احد الشكلين التاليين:

- انشاء وظيفة تقوم بإخراج العدد المركب على الصورة التي عهدناها في خطأ: لم يُعثر على مصدر. المرجع وفي خطأ: لم يُعثر على مصدر المرجع، اي اظهار الجزء الحقيقي و الجزء التخيلي بين قوسين، ويكون لهذه الوظيفة مدخلا واحدا لإدخال القيمة المركبة ومخرجا. واحدا هو المخرج الأساسي، منه تخرج سلسلة الحروف الممثلة للقيمة المركبة، و تظهر هذه الوظيفة **(حول_الى_سلسلة)** في خطأ: لم يُعثر على مصدر المرجع من السطر 18 الى السطر. 20، و يظهر استعمالها في السطر 30 من خطأ: لم يُعثر على مصدر المرجع.
- انشاء وظيفة منتسبة للنمط **عدد_مركب**، وسميت هذه الوظيفة **الى_ساسلة**(السطر. 12 من خطأ: لم يُعثر على مصدر المرجع)، والفرق بين الوظيفة المنتسبة **الى_ساسلة** وبين الوظيفة الغير منتسبة **حول_الى_سلسلة** يكمن فقط في كيفية طلب التنفيذ وعدد المداخل.
  - فمع الوظيفة المنتسبة يُطلب التنفيذ من خلال المتغيرة المهيكلة التي نريد تحويلها الى سلسلة حروف، ولن تحتاج هذه الوظيفة المنتسبة الى مدخل، فالمتغيرة المهيكلة التي تنفذ من خلالها الوظيفة المنتسبة تعرفها هذه الأخيرة، وهذا ما نراه في السطر 32 من خطأ: لم يُعثر على مصدر المرجع اذ يُطلب تنفيذ الوظيفة المنتسبة **الى_ساسلة** من خلال المتغيرة المهيكلة **حم1** التي تمثل **عددا مركبا**،
  - اما فيما يخص الوظيفة المستقلة **حول_الى_سلسلة** ، فيجب وصف المتغيرة المهيكلة في مدخلها، وهذا ما نراه جليا في السطر 30، فقد وضعت المتغيرة المهيكلة **حم1** في مدخل الوظيفة **حول_الى_سلسلة**.



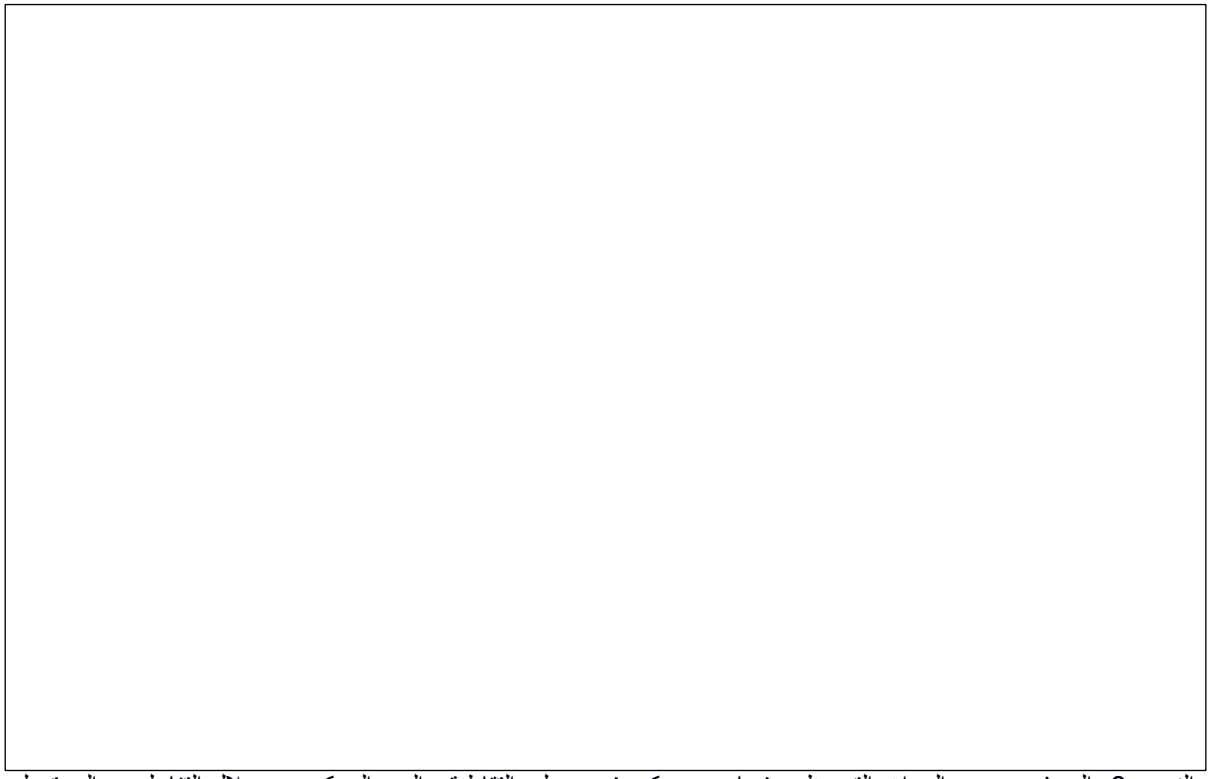
النص 7:البحث عن عدد المرات التي يظهر فيها عدد مركب في جدول باستعمال وظيفة عادية و وظيفة منتسبة لتحويل العدد المركب الى سلسلة

4 – 1 – 4: نسخة اخرى للإجراء اساسي في خوارزم خطأ: لم يُعثر على مصدر المرجع

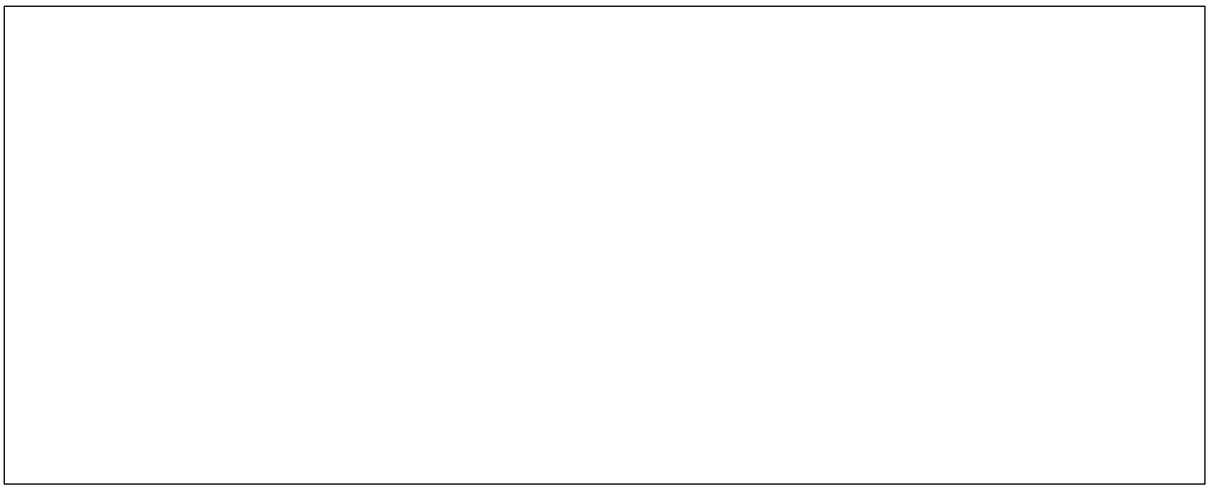
في هذه النسخة، عوض تحميل المتغيرة **عم1** بقيم ثابتة من خلال المشيد كما هو الحال في السطر 27 من خطأ: لم يُعثر على مصدر المرجع، نستعمل التفاعل مع المستعمل للحصول على القيم التي توضع في المتغيرة المهيكلة **عم1**، لبلوغ هذا الهدف ودون احداث تغيير كبير في الإجراء اساسي، ننشأ وظيفة تقوم بالتقاط القيم من خلال لوحة المفاتيح ثم نطلب تنفيذ هذه الوظيفة لشحن القيم في المتغيرة المهيكلة **عم1**.

في خطأ: لم يُعثر على مصدر المرجع، نرى تعريف الوظيفة **'اقبض_عدد_مركب'** من السطر 18 الى السطر 25 ، ونرى تغييرا طفيفا جدا قد طرأ على الإجراء اساسي في السطر 33، اما في السطر 35 فنرى استعمال الوظيفة المنتسبة لتحويل العدد المركب الى سلسلة من الحروف.

تذكير و ملاحظة: من خلال الوظيفة **اقبض_عدد_مركب** نلاحظ استعمال وظيفة جديدة هي الوظيفة **اقرأ_حقيقي،** وتهدف هذه الوظيفة الى الحصول على عدد حقيقي يكتب على لوحة الحروف، وتوجد وظائف أخرى مشابها لها هي: **اقرأ_طبيعي()، اقرأ_سلسلة()، اقرأ _منطقي()**، وكل هذه الوظائف لها صيغة اخرى اين نجد في مدخلها سلسلة من الحروف تخرج على الشاشة قبل عملية القبض، فمثلا نجد **اقرأ_حقيقي(سلسلة س) ، اقرأ_طبيعي(سلسلة س)، اقرأ_سلسلة(سلسلة س)، اقرأ_منطقي(سلسلة س)،** وهذه الأخيرة مستعملة في النسخة الثانية للوظيفة **اقبض_عدد_مركب** (خطأ: لم يُعثر على مصدر المرجع)



النص 8: البحث عن عدد المرات التي يظهر فيها عدد مركب في جدول، التقاط قيم العدد المركب من خلال التفاعل مع المستعمل

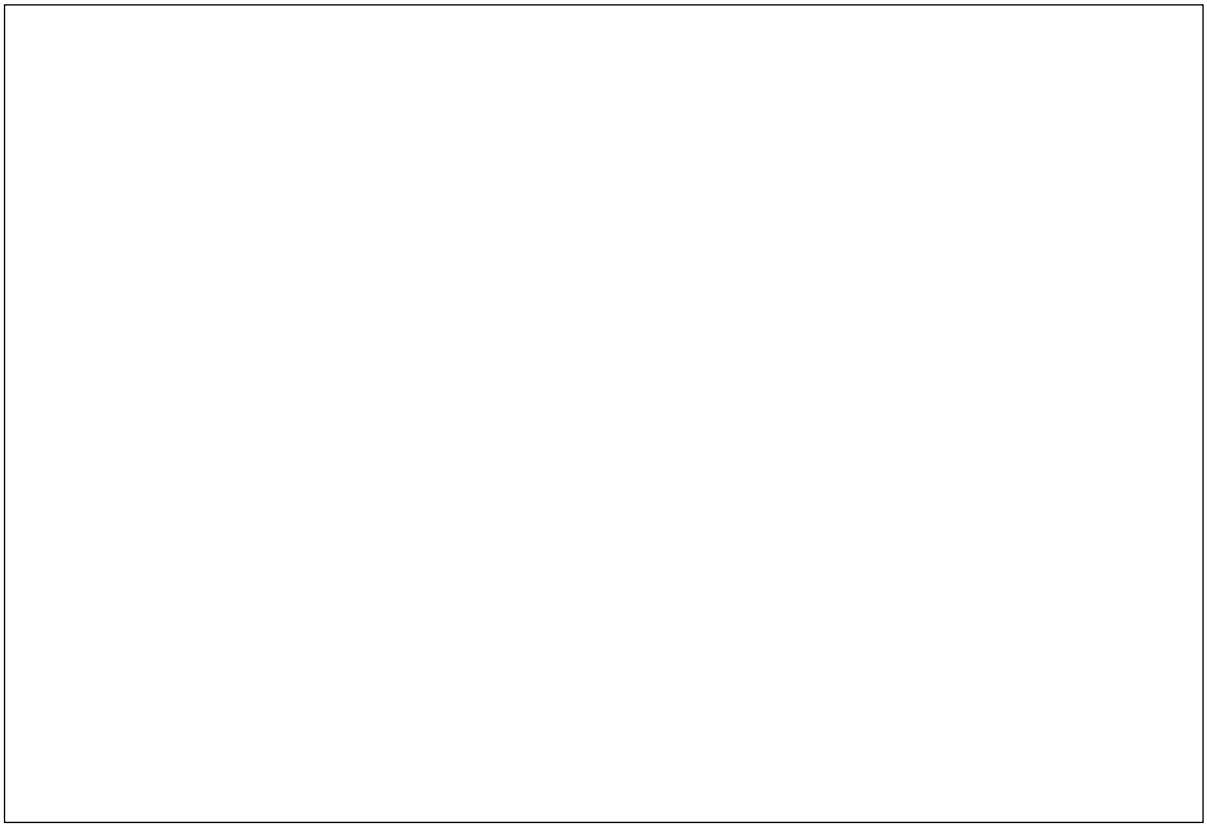


'النص 9: النسخة الثانية للوظيفة 'اقبض_عدد_مركب

4 – 2 المثال الثاني: البحث عن عدد المرات التي يظهر فيها شخص معين في جدول الأشخاص.

4–2 – 1 ضرورة انشاء وظيفة لمقارنة متغيرتين من نمط يمثل شخصا:

لكون هذا الخوارزم يتعامل مع الأشخاص، فلا بد لنا من التعريف بنمط يمثل الأشخاص (خطأ: لم يُعثر على مصدر المرجع)، وبصفة ادق يمثل السمات التي تهمنا من الأشخاص، ومن بين هده السمات نجد الجنس (ذكر ام انثى) اللقب، الاسم ، اسم الأب، لقب و اسم الأم، اسم الجد، تاريخ الولادة، الحالة المدنية (متزوج او لا)، الطول والوزن (خطأ: لم يُعثر على مصدر المرجع).



النص 10: تعريف النمط شخص

حتى يتمكن الخوارزم من حساب عدد المرات التي يظهر فيه شخص ما في جدول للأشخاص، عليه ان يقارن بين معلومتين مهيكلتين كل واحدة تمثل شخصا، وقد توصلنا فيما سبق الى ان المقارنة بين معلومتين (او متغيرتين) مهيكلتين لا تتم باستعمال عملية المساواة (==) التي تستعمل لمقارنة قيمتين بدائيتين، فللمقارنة بين متغيرتين مهيكلتين لا بد لنا من انشاء وظيفة تقوم بعملية المقارنة بينهما، وفيما يلي نميل ونفضل ان تكون الوظيفة التي نريد انجازها وظيفة منتسبة للنمط شخص (الوظيفة يساوي في السطر 16 من خطأ: لم يُعثر على مصدر المرجع).

في هذا الخوارزم نبحث عن عدد المرات الذي ذكر فيها نفس الشخص في الجدول، اي ان عملية البحث تنطلق من معرفة شخص ما من خلال متغيرة مهيكلة تصفه، ونسمي **شخ** هذه المتغيرة، ثم نبدأ بالتجوال في الجدول، فكلما وجدنا خانة تحتوي على نفس المعلومات التي تحتويها المتغيرة **شخ** نقوم بإضافة واحد الى عداد قد انشأ بقيمة اولية تساوي 0 قبل بدئ التجوال في الجدول، ومهمة الوظيفة التي تقارن بين شخصين هي المقارنة بين سمات المتغيرتين المهيكلتين، فاذا كانت النتيجة هي التساوي بين سمات المتغيرتين المهيكلتين فمعنى هذا ان المتغيرتين تمثلان نفس الشخص(خطأ: لم يُعثر على مصدر المرجع).

الإشكال الذي يطرح هنا هو: ما هي السمات التي تكفي لتمثيل شخص ولا نجدها في غيره، والإجابة هنا مرتبطة حقا بالواقع الذي يعالجه الخوارزم، وفي العموم نختار السمات الثابتة التي لا تتغير، كالاسم، واللقب، واسم الأب ، ولقب واسم الأم وتاريخ الميلاد ومكان الولادة، اما السمات التي

تتغير فلا نستعملها، فمثلا سمة الحالة المدنية تتغير، فيكون الشخص في وقت اعزب، ثم يتزوج، ثم مطلق، ثم يتزوج الخ، و نفس الشيء بالنسبة للطول والوزن.

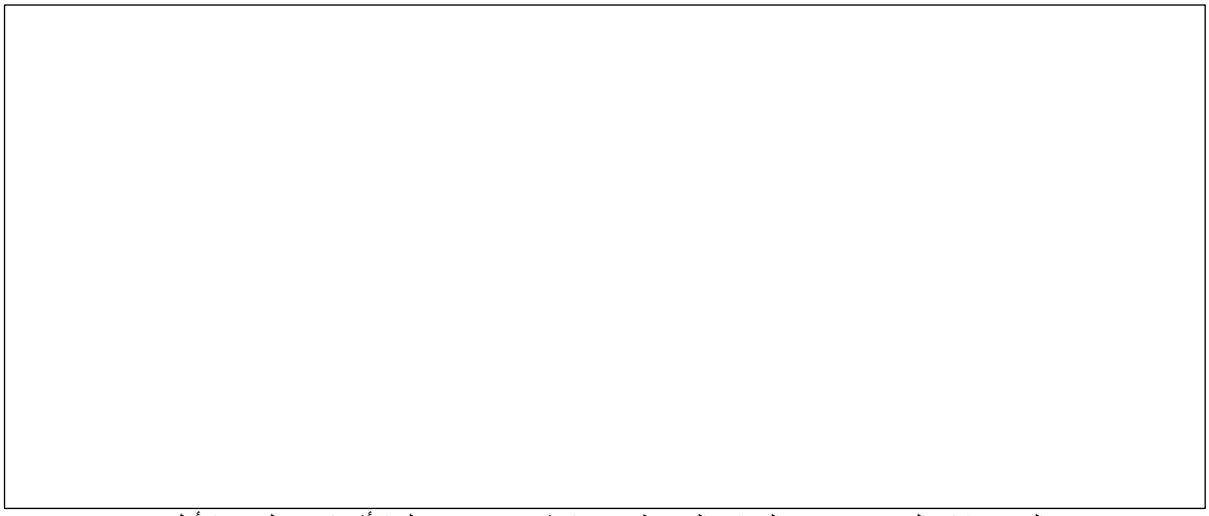
يُعرف خطأ: لم يُعثر على مصدر المرجع بالنمط شخص، ويحتوي هذا النص على مشيد واحد والوظيفتان المنتسبتان **يساوي** و **الى_سلسلة**، وتقوم الوظيفة **يساوي** بمقارنة المتغيرة التي تنفذ من خلالها هذه الوظيفة، ويرمز اليها في النص بالصلة **'انا'**، والمتغيرة التي توضع في المدخل الوحيد لهذه الوظيفة، اما الوظيفة **الى_سلسلة** فهدفها اعطاء المعلومات الممثلة للشخص على شكل سلسلة واحدة من الحروف، وفي النص تتكون هذه السلسلة من الاسم و اللقب.

في الوظيفة **يساوي** المنتسبة للنمط **شخص**، نلاحظ انه ما عدا السمة المسماة **الجنس**، كل السمات التي تقارن انما هي سمات من انماط غير بدائية، اي لا يصح معها استعمل عملية المقارنة البدائية (==)، فمن هذه السمات ما هي من النمط سلسلة، كالأسماء والألقاب، ومنها ما هي من النمط **تاريخ_دقيق** الذي نرى تعريفه في خطأ: لم يُعثر على مصدر المرجع.

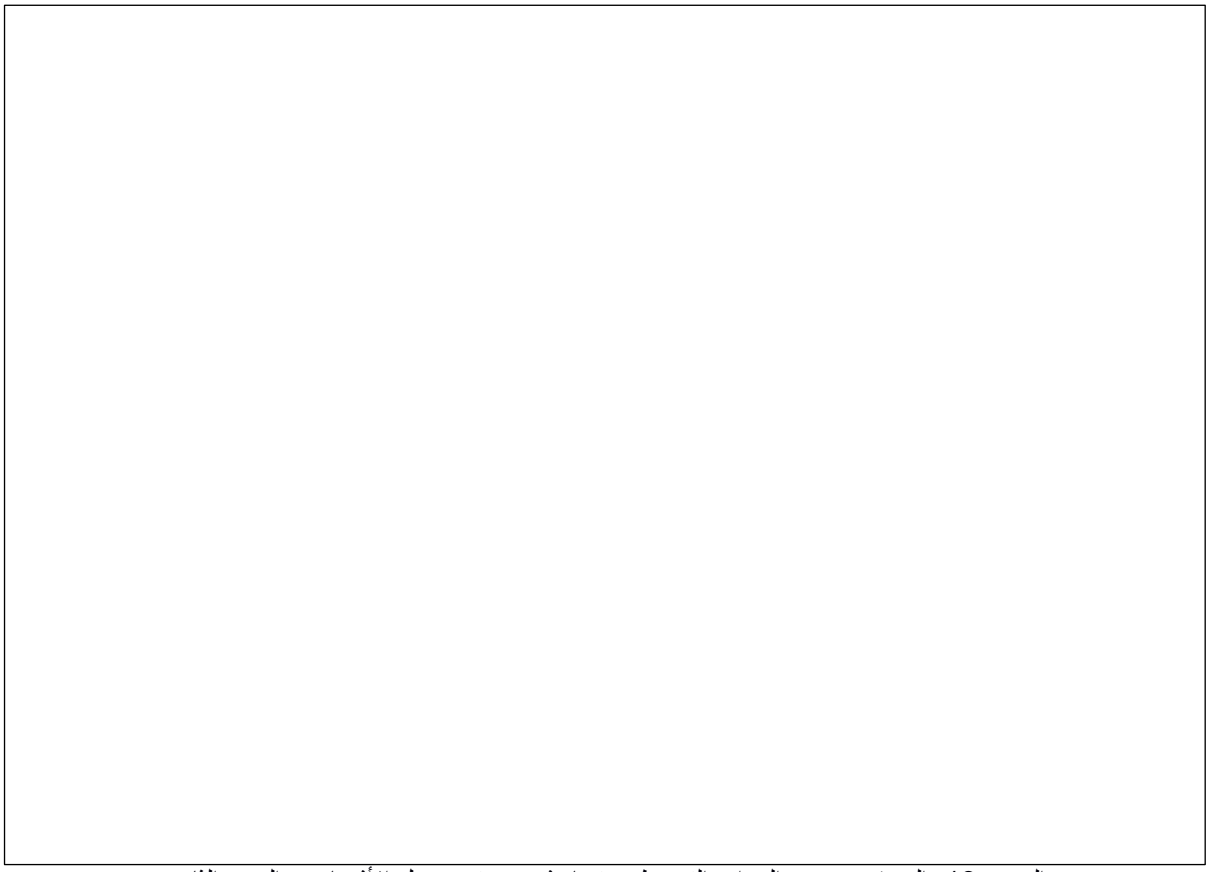
كما تعلمناه من قبل في الفصل الثالث عشر، فان النمط **سلسلة** يحتوي في الأساس على وظيفة منتسبة اسمها **تساوي** وهدفها مقارنة سلسلتين، السلسلة الأولى هي التي تنفذ من خلالها الوظيفة **تساوي** والسلسلة الثانية هي التي توضع في مدخل الوظيفة **تساوي**.

النص 11: تعريف النمط تاريخ_دقيق

اما بالنسبة للنمط **تاريخ_دقيق**، فنرى في السطر 11 من خطأ: لم يُعثر على مصدر المرجع ان هذا النمط مزود بالوظيفة المنتسبة **يساوي** التي تقارن بين متغيرتين من النمط **تاريخ_دقيق**.



النص 12: البحث عن عدد المرات التي يظهر فيها شخص في جدول الأشخاص، الجزء الأول



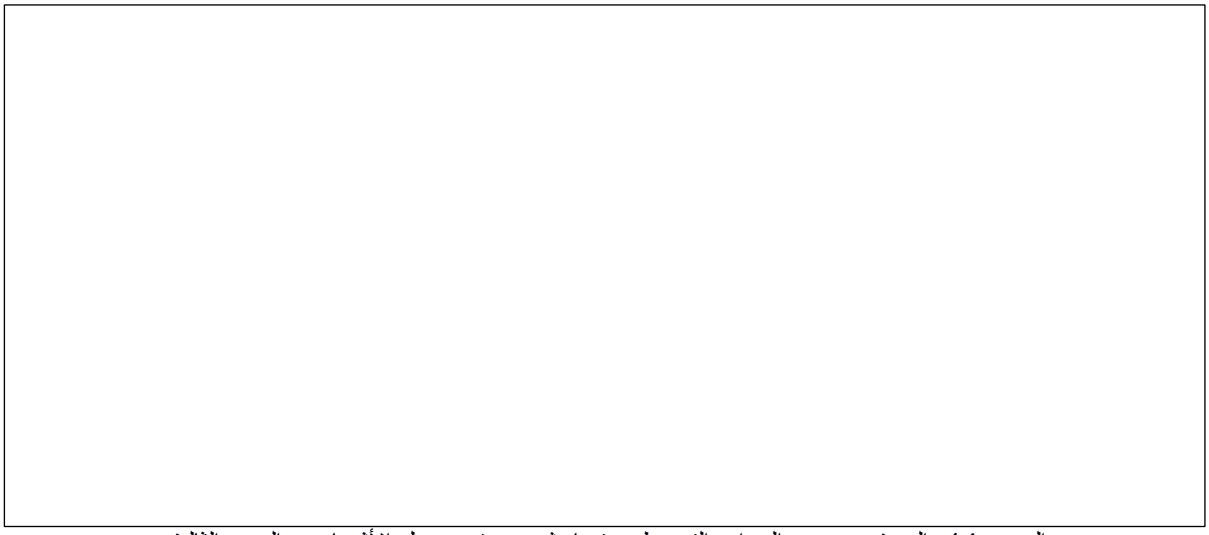
النص 13: البحث عن عدد المرات التي يظهر فيها شخص في جدول الأشخاص، الجزء الثاني

2-4 – 2 انجاز خوارزم البحث عن عدد المرات التي يظهر فيها شخص معين في جدول الأشخاص:

نص هذا الخوارزم يشبه الى حد كبير خوارزم خطأ: لم يُعثر على مصدر المرجع الذي يبحث عن عدد المرات التي يظهر فيها عدد مركب، والفرق بينهما يكمن في النمط وفي الوظيفة التي ترجع عددا مركبا قد جلبت معطياته من قبل المستعمل عبر لوحة الحروف، فاذا اردنا ان ننجز خوارزما يخبر عن عدد المرات التي يظهر فيها شخص ما في جدول الأشخاص، نأخذ خطأ: لم يُعثر على

مصدر. المرجع فنستبدل اولا النمط **عدد_مركب** بالنمط **شخص**، ثم نسدد اسم الوظيفة **اقبض_عدد_مركب** ليصبح **اقبض_شخص**، ثم نسدد محتوى الوظيفة **اقبض_شخص** حتى نمكن المستعمل من ادخال سمات الشخص الذي نبحث عنه في الجدول، وننتج بذلك نصا كاملا تمثله الأجزاء الموجودة في خطأ: لم يُعثر على مصدر المرجع وخطأ: لم يُعثر على مصدر المرجع وخطأ: لم يُعثر على مصدر المرجع.

في اطار انجاز محتوى الوظيفة **اقبض_شخص** (خطأ: لم يُعثر على مصدر المرجع)، وبالتحديد عند قبض سمات تاريخ الولادة، يمكننا استحداث وظيفة اخرى هدفها قبض تاريخا دقيقا من خلال لوحة المفاتيح، وهذا ما نراه في السطر 29 من خطأ: لم يُعثر على مصدر المرجع، الذي فيه يبدأ تعريف الوظيفة **'اقبض_تاريخ_دقيق'**، ونرى استعمال الوظيفة **اقبض_تاريخ_دقيق** من قبل الوظيفة **اقبض_شخص** في السطر 24 من خطأ: لم يُعثر على مصدر المرجع.



النص 14: البحث عن عدد المرات التي يظهر فيها شخص في جدول الأشخاص، الجزء الثالث

الفصل السادس عشر

الأوعية: العمليات المعتادة على الجداول: ترتيب المعلومات

الجزء الثاني:الفصل السادس عشر: الأوعية: العمليات المعتادة على الجداول: عمليات ترتيب المعلومات

1 – مقدمة:

تعد عمليات ترتيب المعلومات في الجداول من اهم العمليات، فجدول مرتب يُسهل بشكل كبير باقي العمليات المتعلقة بالجداول خاصة عمليات البحث، فجدول غير مرتب يجعل عمليات البحث تستغرق وقتا طويلا قبل الوصول الى نتيجة ما، فعلى سبيل المثال:

- البحث في الجدول 15 عن القيمة 5 يتطلب زيارة كل خانات الجدول، ونفس البحث في الجدول 16 لن يتطلب الا زيارة 4 خانات.
- البحث في الجدول 15 عن القيمة 3 يتطلب زيارة كل خانات الجدول دون جدوى، ونفس البحث في الجدول 16 لن يتطلب الا زيارة 4 خانات.
- البحث في الجدول 15 عن اصغر قيمة يتطلب زيارة كل خانات الجدول، ونفس البحث في الجدول 16 لن يتطلب الا زيارة الخانة الأولى.
- البحث في الجدول 15 عن اكبر قيمة يتطلب زيارة كل خانات الجدول، ونفس البحث في الجدول 16 لن يتطلب الا زيارة الخانة الأخيرة.

5	1	12	23	98	76	34	564	237	19	0	8	0	12	233

الجدول 15

564	237	233	98	76	76	23	19	12	12	8	5	1	0	0

الجدول 16

القيم التي يمكن ترتيبها:

هي كل القيم التي يمكن ان نحدد لها علاقة ترتيب، مثل : اصغر، اكبر، اسبق، اولى الخ ... فهي القيم التي يمكن ان نجري عليها عمليات المقارنة :اصغر، اكبر، يساوي، وفي العموم هذه القيم بدائية :كالقيم الطبيعية، والقيم الحقيقية، والقيم الممثلة للحروف، والترتيب يكون اما **تصاعدي** (من اصغر القيم الى اعلاها) او **تنازلي** (من اكبر القيم الى ادناها).

1 – 1 – ترتيب الحروف

فيما يخص الحروف، فعلينا الرجوع الى جداول الرموز التي تبين قيمة كل حرف، ومن هذه الجداول نستخلص مثلا:

- قيمة الحرف **الف** اصغر من قيمة الحرف **باء**، و قيمة الحرف **باء** اصغر من الحرف **تاء** و هكذا، اي ان القيم الممثلة للحروف الأبجدية تحترم ترتيبها.
- قيمة الحرف '0' اصغر من قيمة الحرف '1' ، و قيمة الحرف '1' اصغر من قيمة الحرف '2' و هكذا.

- قيمة الحرف '9' اقل من قيمة الحرف **الف**

1 – 2 – ترتيب سلاسل الحروف

كما اشرنا اليه من قبل، فالنمط سلسلة ليس بنمط بدائي، فهو نمط مركب، لا يمكن ان نجري عليه عمليات المقارنة التي نستعملها مع القيم البدائية، ومع هذا فان سلاسل الحروف كما تعارف عليه الناس يمكن ترتيبها، وترتيبها يستند الى ترتيب مكوناتها، اي الحروف، وهكذا على اساس قيم الحروف يمكننا ايجاد طريقة لترتيب سلاسل الحروف، فمثلا:

- السلسلة "**احمد**" اصغر من السلسة "**عمر**" لأن الحرف الأول من السلسلة "**احمد**" اصغر من الحرف الأول في السلسلة "**عمر**"
- السلسلة "**سليم**" اصغر من السلسلة "**سليمة**"، فالسلسلتين متساويتين حتى الحرف **'م'**، بعده تنته السلسلة الأولى، اما السلسلة الثانية فتحتوي على حرف اضافي هو **'ة'**
- والسلسة "**1– البداية**" اصغر من السلسلة "**أ– البداية** "، لكون الحرف الأول في السلسلة "**1– البداية**" اصغر من الحرف **'أ'** الذي تبدأ به السلسلة "**أ– البداية** ".
- والسلسة "**5–نجوم**" اكبر من السلسلة "**3–نجوم**".

كما تعلمناه في الفصل 13 من هذا الجزء الثاني، فان النمط سلسلة مزود بالوظيفة المنتسبة "**طبيعي قارن(سلسلة س)**" التي تمكن من مقارنة سلسلتين من الحروف.

1 – 3 – ترتيب القيم المعقدة

اما القيم المعقدة المحمولة في متغيرات مهيكلة، فلا يمكن ترتيبها الا بعد ان نُعرف معنى الترتيب، فمثلا اذا تحصلنا على قائمة فيها اسماء والقاب طلبة، و طولهم، ووزنهم، وعمرهم، ومعدلهم السنوي، ثم قيل لنا "**رتب قائمة الطلبة**"، فالسؤال البديهي الذي نطرحه هو "**على اي اساس نرتب القائمة**"، هل هي على اساس اللقب، او الاسم او العمر او الوزن او المعدل العام الخ... فاذا قيل لنا على اساس الاسم او اللقب نستعمل ترتيب سلاسل الحروف، واذا قيل لنا على اساس افضل معدل عام، نستعمل ترتيب الأعداد الحقيقية.

في مواقع اخرى، يجب علينا اجراء عمليات يمكن ان تكون معقدة، من اجل ايجاد القيم البدائية التي على اساسها ترتب القيم المعقدة، فمثلا اذا كانت لدينا اشكال مختلفة، من ثلاثيات ومستطيلات ودائرات ثم طلب منا ترتيبها، فعلينا اولا ايجاد الأساس الذي عليه نرتب هذه الأشكال، اهي المساحة، اهو المحيط، اهو الموقع، وبعدها نقوم بحساب القيمة الممثلة لأساس الترتيب ثم نقارنها بغيرها.

ملاحظة: فيما يلي سوف نستعمل جداول تحتوي على قيم طبيعية في اطار شرح خوارزميات الترتيب، والترتيب يكون تصاعديا، اي من الأصغر الى الأكبر.

2 – خوارزميات الترتيب:

هناك عدد لا بأس به من خوارزميات الترتيب، كل واحد ينفرد بطريقته الخاصة ليبلغ الهدف، ويمكن ان يجد المرء طرق جديدة فيؤسس عليها خوارزما جديدا للترتيب.

2 – 1 : الترتيب بالانتخاب التدريجي للقيم الأصغر ونقلها الى مواقعها النهائية في نفس الجدول:

يعمل على الخوارزم على مراحل، وفي بداية كل مرحلة:

- يحدد المجال الذي يجب ان يرتب (كل الجدول في المرحلة الأولى، في المرحلة الثانية كل الجدول ما عدا قيمة واحدة وهي اصغر قيمة وجدت في المرحلة الأولى وحدد مكانها النهائي في الجدول).
- يحدد الموقع الذي نضع فيه اصغر قيمة وجدت في مجال المرحلة، وهي دائما اول موقع في مجال المرحلة.
- ايجاد القيمة الأصغر.
- وضع القيمة الأصغر في الموقع الذي حدد لها باستبدال محتوى الخانات: القيمة الأصغر. تذهب الى الخانة التي خصصت لأصغر. قيمة، والقيمة التي كانت في الموقع الذي خصص لأصغر. قيمة تذهب الى الموقع التي وجدت فيه اصغر قيمة وحولت منه.
- تحديد المجال الجديد للمرحلة التالية، ومنه تستثنى الخانة التي وضعت فيها اصغر قيمة في المرحلة الحالية.
- ينته الخوارزم من عمله عندما يصبح المجال مكونا من خانة واحدة فقط.

ونشرح فيما يلي عمل هذا الخوارزم في كيفية ترتيبه الجدول 17 .

- **البداية: هي المرحلة الأولى:**
  - تحديد مجال المرحلة الأولى: كل الجدول من الموقع 0 الى الموقع 14.
  - موقع لأصغر قيمة: هي اول خانة في المجال، اي الخانة 0، وتحتوي على القيمة 233
  - ايجاد اصغر. قيمة: هي القيمة 0 الموجودة في الخانة رقم 2
  - نقل اصغر قيمة الى موقعها: هنا علينا ان لا نُتلف القيمة التي كانت في الموقع المحدد لأصغر قيمة، ولتفادي اتلافها نضعها في الموقع الذي وجدنا فيه اصغر قيمة و نحول اصغر قيمة الى موقعها النهائي، وبهذا يصبح الجدول على الشكل الجديد (الجدول 18) بعد هذه المرحلة الأولى.

5	1	15	23	98	76	34	564	237	19	30	8	0	12	233

14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

الجدول 17

5	1	15	23	98	76	34	564	237	19	30	8	233	12	0
14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

الجدول 18

- **المرحلة الثانية:**
  - تحديد المجال: الجدول من الموقع 1 الى الموقع 14.
  - موقع لأصغر قيمة: هي اول خانة في المجال، اي الخانة 1، وفيها القيمة 12.
  - ايجاد اصغر قيمة في المجال: هي القيمة 1 الموجودة في الخانة رقم 13
  - نقل اصغر قيمة الى موقعها النهائي: هنا علينا ان لا نُتلف القيمة التي كانت في الموقع المحدد لأصغر قيمة، ولتفادي اتلافها نضعها في الموقع الذي وجدنا فيه اصغر قيمة ونحول اصغر قيمة الى موقعها، وبهذا يصبح الجدول على الشكل الجديد (الجدول 19) في آخر المرحلة الثانية.

5	12	15	23	98	76	34	564	237	19	30	8	233	1	0
14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

الجدول 19

- **المرحلة الثالثة:**
  - تحديد مجال المرحلة الثالثة: الجدول من الموقع 2 الى الموقع 14.
  - موقع المخصص لاحتواء اصغر قيمة في المجال : هي دائما اول خانة في المجال، اي الخانة 2، و تحتوي على القيمة 233.
  - ايجاد اصغر قيمة في المجال : هي القيمة 5 الموجودة في الخانة رقم 14
  - نقل اصغر قيمة الى موقعها باستبدال قيم الخانات وبهذا يصبح الجدول على الشكل الجديد (الجدول 20الجدول 18) بعد هذه المرحلة الثانية.

233	12	15	23	98	76	34	564	237	19	30	8	5	1	0
14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

الجدول 20

- **المرحلة ما قبل الأخيرة:**

يكون الجدول على حال الجدول 21.

- تحديد المجال: الجدول من الموقع 13 الى الموقع 14.
- موقع اصغر قيمة: هي اول خانة في المجال، اي الخانة 13، وتحتوي على القيمة 564
- ايجاد اصغر قيمة: هي القيمة 237 الموجودة في الخانة رقم 14
- نقل اصغر قيمة لموقعها باستبدال قيم الخانات وبهذا يصبح الجدول على الشكل الجديد(الجدول 22).

237	564	233	98	76	34	30	23	19	15	12	8	5	1	0
14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

الجدول 21

564	237	233	98	76	34	30	23	19	15	12	8	5	1	0
14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

الجدول 22

- **المرحلة الأخيرة:**

لم يبق في المجال الا خانة واحدة، في هذه الحالة ينته الخوارزم و تكون النتيجة هي الجدول على حاله في المرحلة الأخيرة(الجدول 22).

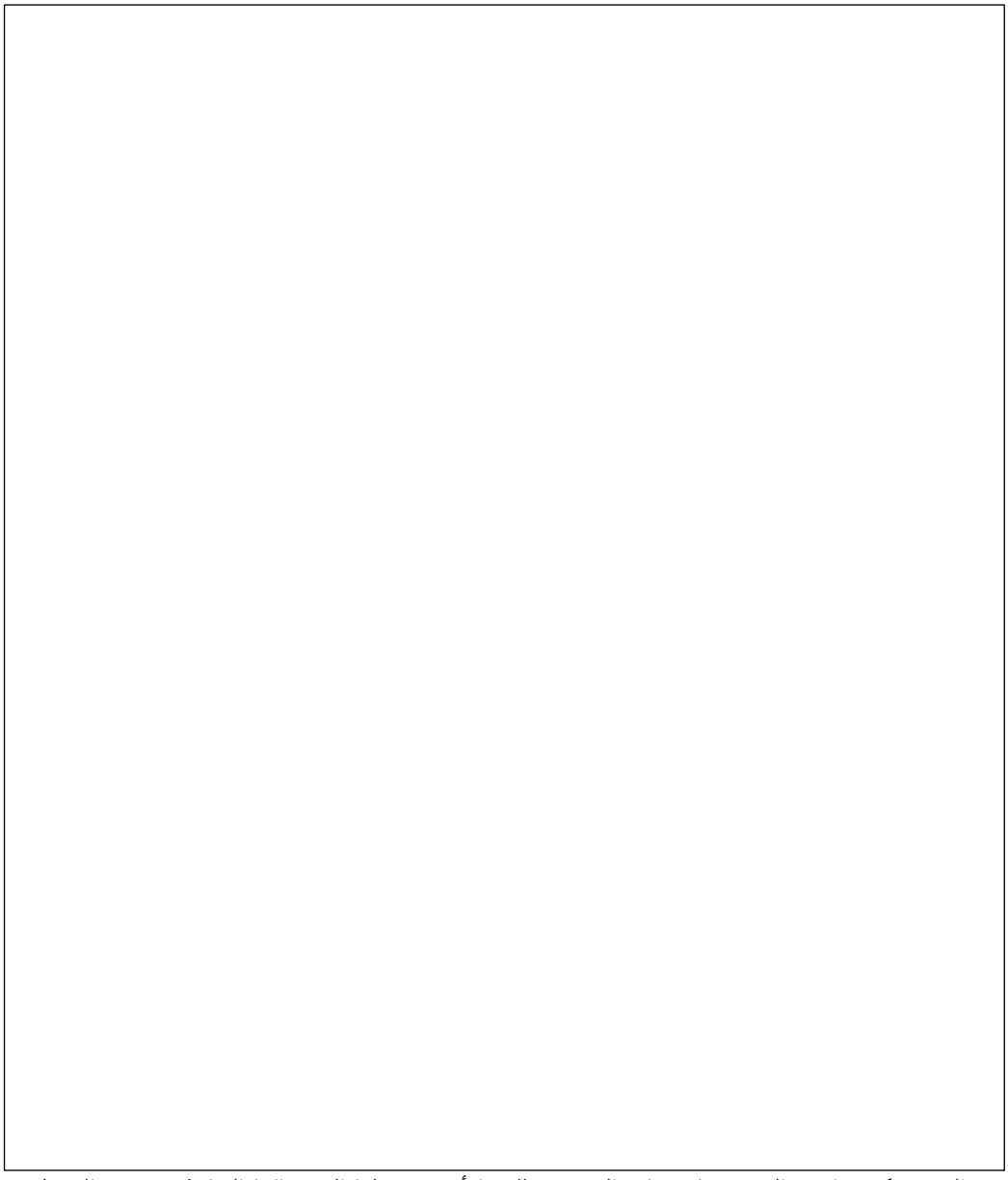
2-1-1 نص الخوارزم:

لترتيب جدولا ما نستعمل الخوارزميات الثلاث التالية(خطأ: لم يُعثر على مصدر المرجع، من السطر 1 الى السطر 32):

- الوظيفة **موقع_الأصغر.** (السطر 2) التي تبحث عن موقع (رقم خانة) لأصغر قيمة في مجال بحث ما، ونذكر. ان مجال البحث مكون من مجموعة من خانات جدول نعرف رقم اول وآخر خانة فيه، فيبدأ البحث عن الأصغر من الخانة الأولى لمجال البحث وينته عند آخر خانة من مجال البحث ، وترجع الوظيفة **موقع_الأصغر** رقم الخانة التي تحتوي على اصغر قيمة في مجال البحث.
- الإجراء **استبدل** (السطر. 10) الذي يقوم باستبدال قيم خانتين، ويحتاج هذا الإجراء في مداخله الى: جدول طبيعي، رقم الموقع الأول ورقم الموقع الثاني.
- الإجراء **رتب** (السطر 17) الذي يقوم بترتيب جدول طبيعي وُصف في مدخله، ويستعمل هذا الإجراء الوظيفة **موقع_الأصغر** والإجراء **استبدل**، وفي كل مرحلة يقوم هذا الإجراء بتحديد مجال البحث في جدول يُراد ترتيبه، فكلما كان هناك مجال للبحث (اي انه يحتوي

على الأقل على خانتين) يطلب الإجراء **رتب** من الإجراء **موقع_الأصغر.** تحديد اصغر قيمة في المجال، وبعد الحصول على موقع اصغر قيمة في المجال من قبل الوظيفة **موقع_الأصغر.**، وفي حال ما اذا لم يكن اول موقع في المجال هو الذي يحتوي فعليا على اصغر قيمة (السطر 29)، يطلب الإجراء **رتب** من الإجراء **استبدل** استبدال القيم بين اول خانة في المجال والخانة التي وجدت فيه اصغر قيمة في المجال، وبعد هذا الاستبدال تظهر آخر اصغر قيمة وجدت في الخانة الأولى للمجال، وبعدها يقوم الخوارزم بتقليص المجال (السطر 30)، فيخرج منه الخانة الأولى للمجال السابق التي اصبحت تحتوي على آخر اصغر قيمة وجدت في مجال البحث السابق.

في (خطأ: لم يُعثر على مصدر المرجع) تظهر الخوارزميات الثلاث (**موقع_الأصغر، استبدل** و **رتب**) اصافة الى الإجراء اساسي الذي يقوم باختبار خوارزم الترتيب، فيجهز اولا الجدول **جدول_طب** (السطر 37) بعد التصريح به بقيم اولية، ثم يطلب في السطر 38 من الوظيفة **رتب** ترتيب الجدول الطبيعي **جدول_طب**.



النص 1: خوارزم الترتيب بالانتخاب التدريجي للقيم الأصغر ونقلها الى مواقها النهائية في نفس الجدول

بعد الإنتهاء من تنفيذ الوظيفة **رتب**، يقوم الإجراء **اساسي** بإظهار محتوى الجدول المرتب على الشاشة كما يظهر ذلك في خطأ: لم يُعثر على مصدر المرجع.

```
الجدول بعد الترتيب:
718 222 81 78 65 56 56 54 23 21 9
3
الى اللقاء...
```

الشكل 1: نتيجة تنفيذ خوارزم النص 1

2 – 2 : الترتيب بدفع القيمة الأثقل الى آخر خانة:

كما حدث مع الخوارزم السابق، فان هذا الخوارزم يمر بمراحل عدة، و في كل مرحلة يحدد مجال عمله، ومجال العمل هو مجموعة الخانات التي تُستهدف من قبل عملية الترتيب، وهذه المرة، وفي كل مرحلة، ندفع بأكبر قيمة نحو آخر خانة من المجال، وفي المرحلة التالية يقلص مجال الترتيب فلا يشمل الخانة التي زحزحت اليها اثقل قيمة.

الفرق بين الخوارزميين يكمن في ايجاد القيمة التي ستوضع في مكانها في كل مرحلة (أصغر قيمة بالنسبة الخوارزم السابق و اكبر قيمة فيما يخص هذا الخوارزم)، فبينما في الخوارزم الأول نبحث عن اصغر قيمة، في هذا الخوارزم، وفي اطار التجوال، لا ننتظر زيارة آخر خانة لاتخاذ تدبير ما، فكلما زرنا خانة، نرى محتوى الخانة التالية، فاذا وجدنا ان الخانة التالية تحتوي على قيمة اقل من الخانة الحالية، ندفع اليها القيمة الأكبر باستبدال قيم خانتين، ثوم نتابع الزيارة حتى نصل الى الخانة الأخيرة وقد ثبتت فيها اثقل قيمة، وفيما يلي نشرح كيفية ترتيب هذا الخوارزم الجدول 23.

3	4	35	2	9	5	17	23
7	6	5	4	3	2	1	0

الجدول 23

المرحلة الأولى: خانة بداية المجال: 0 ، خانة نهاية المحال: 7

- زيارة الخانة 0 :
  - مقارنة الخانة 0 مع الخانة الي تليها: اي 1
  - محتوى الخانة 0 اكبر من محتوى الخانة 1: اذا ندفع الثقيل في اتجاه آخر خانة باستبدال محتوى الخانتين، ونحصل على الجدول 24

3	4	35	2	9	5	23	17
7	6	5	4	3	2	1	0

الجدول 24

- زيارة الخانة 1:
  - مقارنة الخانة 1 مع الخانة التي تليها: اي 2
  - محتوى الخانة 1 اكبر من محتوى الخانة 2: اذا ندفع الثقيل في اتجاه آخر خانة باستبدال محتوى الخانتين، ونحصل على الجدول 25
  -

3	4	35	2	9	23	5	17

7	6	5	4	3	2	1	0

الجدول 25

- زيارة الخانة 4: عند زيارة الخانة 4، يكون الجدول على حال الجدول 26.

3	4	35	23	2	9	5	17
7	6	5	4	3	2	1	0

الجدول 26

  - مقارنة الخانة 4 مع الخانة الي تليها: اي 5
  - محتوى الخانة 4 اصغر من محتوى الخانة 2: لا داعي لدفع محتوى الخانة 4، فيبق الجدول على حاله ونذهب لزيارة الخانة 5

- زيارة الخانة 6: عند زيارة الخانة 6، يكون الجدول على حال الجدول 27

3	35	4	23	2	9	5	17
7	6	5	4	3	2	1	0

الجدول 27

  - مقارنة الخانة 6 مع الخانة الي تليها: اي 7
  - محتوى الخانة 6 اكبر من محتوى الخانة 7: وجب الاستبدال لنحصل على الجدول 28.

35	3	4	23	2	9	5	17
7	6	5	4	3	2	1	0

الجدول 28

- زيارة الخانة 7: لا توجد بعدها خانة، اي ان المرحلة الأولى قد انتهت

المرحلة الثانية: خانة بداية المجال: 0 ، خانة نهاية المحال: 6

في آخر هذه المرحلة نتحصل على الجدول 29.

35	23	3	4	17	2	9	5
7	6	5	4	3	2	1	0

الجدول 29

المرحلة الثالثة: خانة بداية المجال: 0 ، خانة نهاية المحال: 5

في آخر هذه المرحلة نتحصل على الجدول 30

35	23	17	3	4	9	2	5
7	6	5	4	3	2	1	0

الجدول 30

المرحلة الرابعة: خانة بداية المجال: 0 ، خانة نهاية المحال: 4

آخر هذه المرحلة نتحصل على الجدول 31

35	23	17	9	3	4	5	2
7	6	5	4	3	2	1	0

الجدول 31

المرحلة الخامسة: خانة بداية المجال: 0 ، خانة نهاية المحال: 3

في نهاية هذه المرحلة نتحصل على الجدول الجدول 32.

35	23	17	9	5	3	4	2
7	6	5	4	3	2	1	0

الجدول 32

المرحلة السادسة: خانة بداية المجال: 0 ، خانة نهاية المحال: 2

في نهاية هذه المرحلة نتحصل على الجدول الجدول 33.

35	23	17	9	5	4	3	2
7	6	5	4	3	2	1	0

الجدول 33

المرحلة السادسة: خانة بداية المجال: 0 ، خانة نهاية المحال: 1

في نهاية هذه المرحلة نتحصل على الجدول الجدول 34.

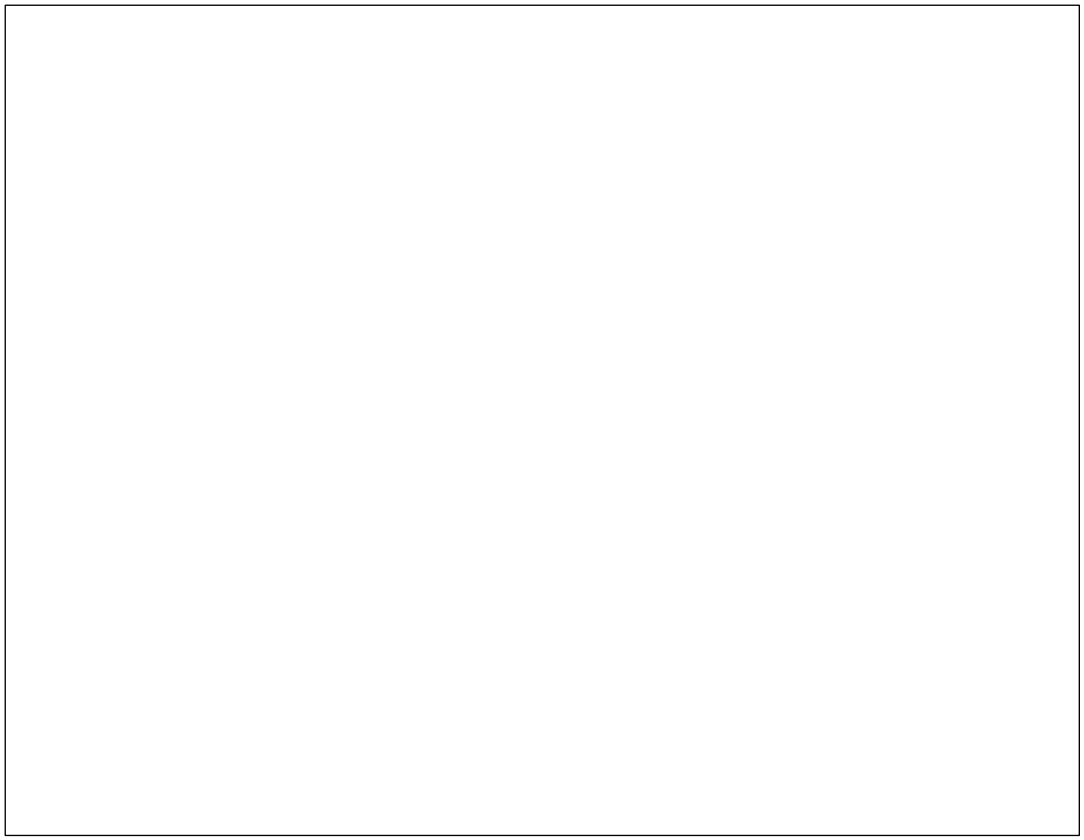
35	23	17	9	5	4	3	2
7	6	5	4	3	2	1	0

الجدول 34

المرحلة السابعة: خانة بداية المجال: 0 ، خانة نهاية المحال: 0

المجال فيه خانة واحة فقط، وبهذا ينته الخوارزم بالنتيجة الظاهرة في الجدول 34.

1-2-2 نص الخوارزم: (خطأ: لم يُعثر على مصدر المرجع)



النص 2: خوارزم الترتيب بدحرجة القيم الثقيلة الى الخانات الأخيرة

2 – 3 : الترتيب السريع:

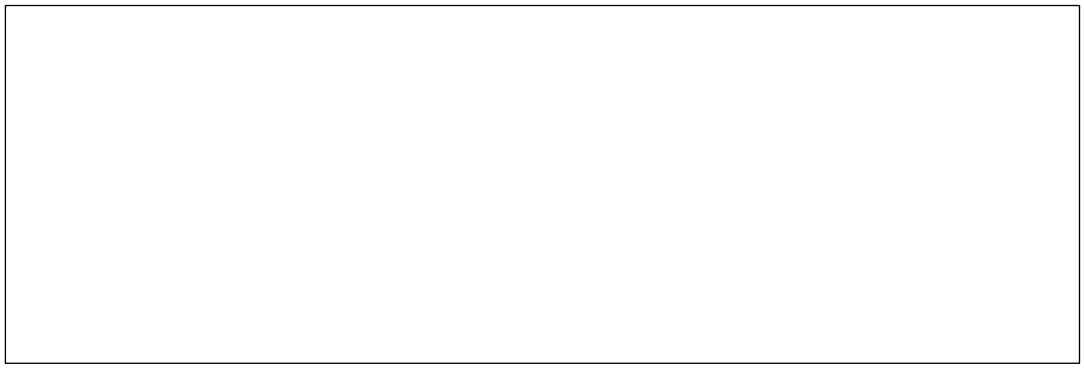
يرتكز خوارزم الترتيب السريع على مفهوم الخوارزميات التراجعية، ولذا سنقوم اولا بتعريف هذا النوع من الخوارزميات قبل الشروع في تقديم خوارزم الترتيب السريع.

3-2 -1: الخوارزميات التراجعية (او ذاتية التردد): هي خوارزميات ترجع الى نفسها (او ان شئت قل: تستعين بنفسها)، لحل اشكال ما، فتنادي نفسها من اجل حل جانب جزئي من مشكل عام، فعندما ينطلق الخوارزم يكون امام مشكل كلي، فبدلا من مواجهة المشكل الكلي، يُجزئ الخوارزم المشكل الكلي ثم ينادي نفسه ليحل مشكلا جزئيا من المشكل الكلي، ويمكن ان تكون عملية التجزئة عميقة جدا، ففي مرحلة اولى يُجزأ المشكل، وفي مرحلة ثانية يجزأ المجزأ الذي تحصلنا عليه في المرحلة الأولى، وفي مرحلة ثالثة يجزئ ما جزء في المرحلة السابقة، حتى نصل الى حالة لا يمكن فيها التجزئة ويكون الإشكال هنا بسيطا جدا فيحل، ثم يرجع الخوارزم الى الوراء، وكلما رجع الى الوراء يجد جزءا قد حل وعليه حل الجزء الباقي وهكذا، والأمثلة التالية تعين على فهم ميزات الخوارزميات التراجعية.

2-3-2 خوارزم حساب "العاملي" (خطأ: لم يُعثر على مصدر المرجع)

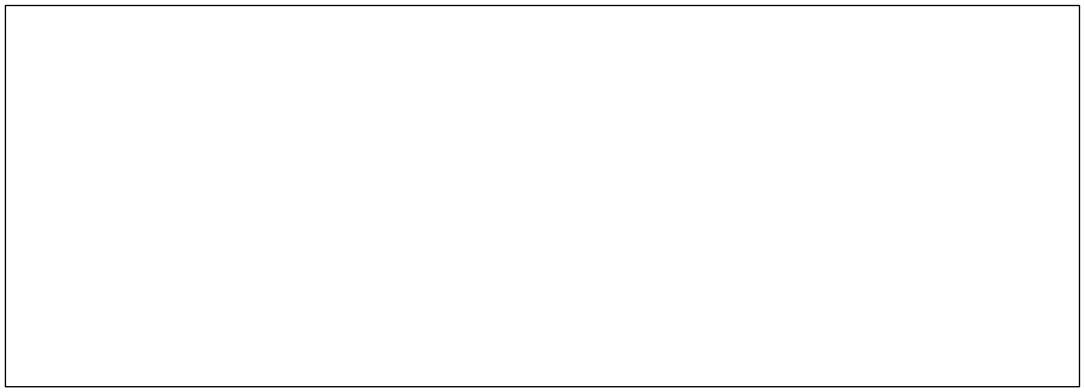
تذكير: العاملي لعدد طبيعي **ط**، والذي يكتب **ط!** والذي يقرأ "**عاملي ط**"، هو جداء كل الأعداد الطبيعية (الأعداد الصحيحة الموجبة قطعا) المساوية أو الأصغر من **ط**، ما عدا الصفر.

نلاحظ من خلال **خطأ: لم يُعثر على مصدر المرجع** ان الوظيفة "**عاملي**" اذا صادفت إشكالا بسيطا. تقوم بحله (السطر 3)، وهذا ما يحصل عندما تكون في مدخلها القيمة 1 ، اما اذا كان المشكل غير بسيط فتجزؤه، فجزء يحل في الفترة الحالية ويتمثل في ضرب القيمة المتغيرة **ط** (وهي المدخل) بما ترجعه الوظيفة التراجعية حين تواجه مشكلا اقل تعقيدا، اي حساب العاملي للقيمة ط- 1.



النص 3: خوارزم فيه وظيفة لحساب العاملي

استعمال "التراجع" في الخوارزميات هي في حقيقة الأمر ليست الطريقة الوحيدة لحل اشكال ما، فيمكن، بدل بناء خوارزم يستعمل التراجع، بناء خوارزم يستعمل عمليات التكرار، وفي هذه الحالة تكون الكتابة نوعا ما معقدة كما يظهر. ذلك في خطأ: لم يُعثر على مصدر المرجع.



النص 4: خوارزم فيه وظيفة لحساب العاملي

2- 3 -3 اصل استعمال التراجع بدل التكرار:

يكون استعمال التراجع بعد ملاحظة الملامح الكبرى للحل، فمثلا عوض رؤية **عاملي ط** على الشكل:

ط! = ط*(ط-1)*(ط-2).... *1

نراه على الشكل التالي:

0! = 1

ط! = ط * (ط – 1) !

ونفس الشكل يمكن ان نراه في سياق انجاز خوارزم يقوم بجمع عدد من الأعداد الطبيعية الأولى، ويكون هذا الشكل كالتالي:

جمع (1) = 1

جمع (ن) = ن + جمع (ن – 1)

4-3-2 : الخوارزم التراجعي لجمع عدد من الأعداد الطبيعية الأولى

النص 5: خوارزم فيه وظيفة ترلجعية لجمع عدد من الأعداد الطبيعية الأولى

2 – 3 – 5 سلوك خوارزم الترتيب السريع:

يستعمل هذا الخوارزم منهجية "**فرق تسود**" لبلوغ هدفه وهو ترتيب الجدول، فيقوم اولا بتجزئة الجدول الى جزئين (او مجالين) ثم يعاود الكرة مع كل مجال مجزأ حتى لا يبق في المجال المجزأ الا خانة واحدة، ويكون سلوكه على النحو التالي:

- في البداية، يكون مجال العمل مكون من الجدول كله.
- نختار من بين قيم المجال قيمة، فنجعلها قيمة محورية، وفي العموم نختار قيمة آخر خانة في المجال، وفي نهاية هذه المرحلة تجد القيمة المحورية نفسها في خانة نسميها الخانة المحورية.
- نحدث في المجال جزئين (مجالين) تتوسطهما الخانة المحورية.

- قيم المجال الجزئي الأول تكون اصغر من القيمة المحورية، واول خانة من المجال الجزئي هي الخانة الأولى للمجال الغير المجزء وآخر خانة في المجال الجزئي الأول هي الخانة التي تسبق خانة القيمة المحورية في المجال الغير المجزء.
- قيم الجزء الثاني تكون اكبر من القيمة المحورية، واول خانة من هذا المجال الجزئي هي الخانة التي تأتي مباشرة بعد الخانة المحورية وآخر خانة هي آخر خانة من المجال الغير المجزء.

- بعد الحصول على المجالين المجزئين اثر هذه العملية، تجد القيمة المحورية مكانها النهائي في الجدول، ويبق علينا ترتيب المجالين الجزئين الذين تحصلنا عليهما، وهما على يمين ويسار الخانة المحورية.
- يمكن ان نرى كل مجال جزئي على انه مجال يجب ترتيبه، فهنا نعاود نفس المنهجية في اختيار المحور ثم تجزئة كل مجال جزئي الى مجالين جزئيين اصغر، فننفذ على كل جزء نفس الخوارزم.
- تنته عملية التجزئة عندما لا يصبح في المجال الا خانة واحدة، فلا ضرورة لترتيب قيمة واحدة فهي في مكانها.

وفيما يلي، نفسر بعض المراحل الهامة من عمل خوارزم الترتيب السريع لترتيب الجدول 35.

- **البداية:**
  - تعريف المجال: من الخانة رقم 0 الى الخانة الأخيرة رقم 7 (الجدول 35).
  - القيمة المحورية في الخانة 7.

11	35	4	23	2	9	5	17
7	6	5	4	3	2	1	0

الجدول 35

التجزءة الأولى:

- تكوين الجزئين (الجدول 36)
- ثم نقل القيمة المحورية بين الجزئين (الجدول 37)

11	35	17	23	4	2	9	5
7	6	5	4	3	2	1	0

الجدول 36

23	35	17	11	4	2	9	5
7	6	5	4	3	2	1	0

الجدول 37

- **التجزءة الثانية:** تُطبق على كل جزء وجد في المرحلة السابقة (اي الجزء الأول من الخانة 0 الى الخانة 3 والجزء الثاني من الخانة 5 الى الخانة 7) لتكوين الأجزاء ثم نقل القيم المحورية ما بين الأجزاء.
  - **تجزئة الجزء الأول:** المجال من الخانة 0 الى الخانة 3 ، القيمة المحورية في الخانة 3، والنتيجة : جزئين(الجدول 39) : الجزء من الخانة 0 الى الخانة 0، و هذا مجال لن يجزأ، والجزء من الخانة 2 الى الخانة 3، وننقل القيمة المحورية الى الخانة المحورية رقم 1.
  - **تجزئة الجزء الثاني:** المجال من الخانة 5 الى الخانة 7 ، القيمة المحورية في الخانة 7، والنتيجة: جزئين (الجدول 39): الجزء من الخانة 5 الى الخانة 5، و هذا مجال لن يجزأ، الجزء من الخانة 7 الى الخانة 7، و هذا مجال لن يجزأ، وتنقل القيمة المحورية الى الخانة المحورية رقم 6.

23	35	17	11	4	2	9	5
7	6	5	4	3	2	1	0

الجدول 38

35	23	17	11	9	5	4	2
7	6	5	4	3	2	1	0

الجدول 39

- **التجزءة الثالثة:** تطبق فقط على الجزء المكون من الخانتين رقم 2 و رقم 3

2 – 3 – 6 هيكلة الخوارزم:

من خلال السلوك العام للخوارزم نلاحظ ما يلي:

- بروز نشاط هام وهو نشاط تجزئة المجال، ولذا نخصص لهذا النشاط خوارزما خاصا به، ونسميه **جزء.**

- في اطار التجزئة سوف نحول ما تحتويه خانات الى خانات اخرى باستبدال ما في الخانات، اي اننا نحتاج الى الإجراء **استبدل** الذي انشأناه في الخوارزميات السابقة.

2 – 3 – 7 سلوك خوارزم التجزئة:

لتجزئة مجال ما يقوم الخوارزم "**جزء**" بما يلي:

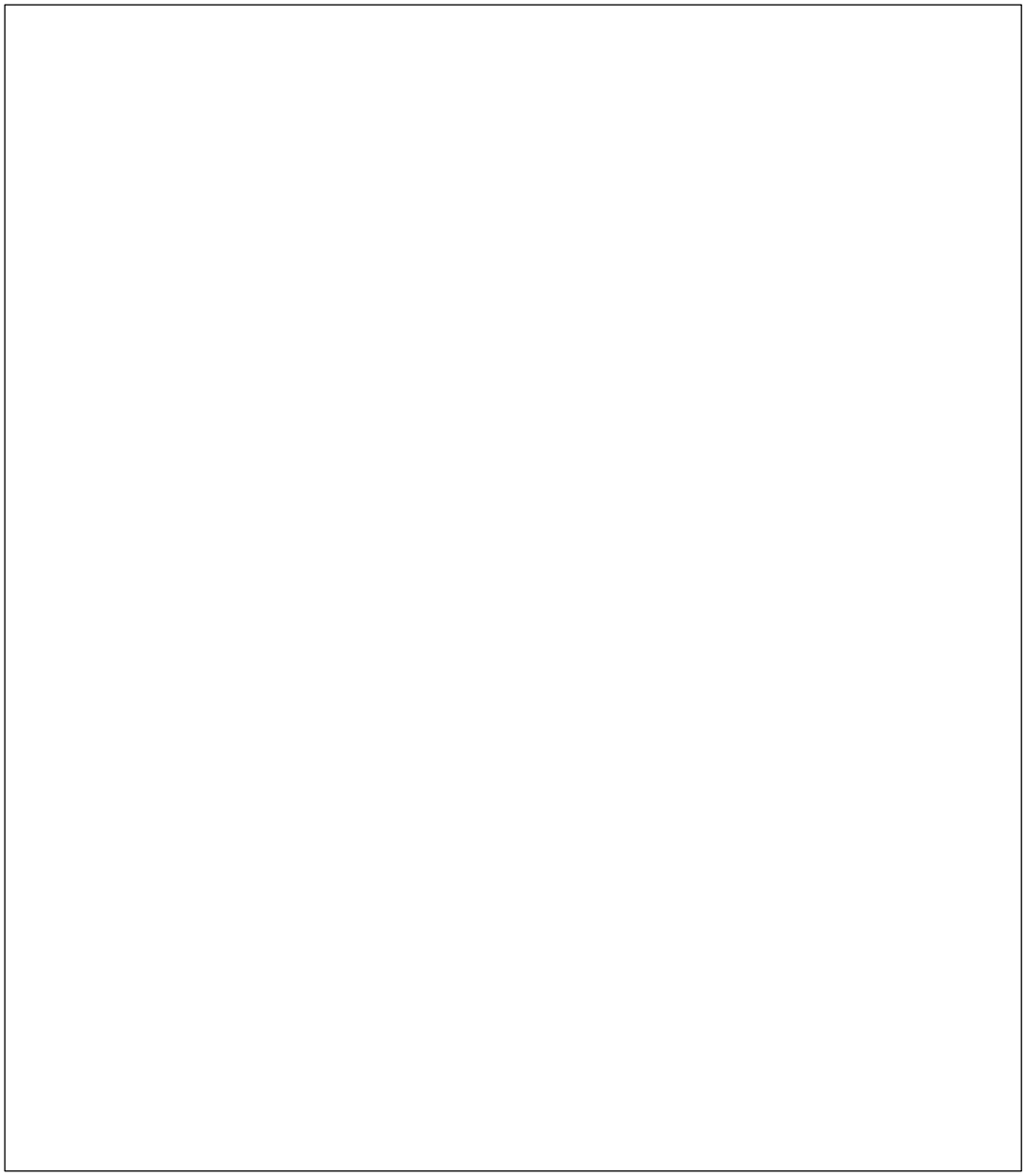
- اولا، اختيار القيمة المحورية، وفيما يخصنا، تكون آخر خانة في المجال هي التي تحتوي على القيمة المحورية.
- ثانيا: التجوال في المجال من اول خانة في المجال مستعملين متغيرتين : "**م**" و "**ج**" :
  - بالمتغير "**م**" يزور الخوارزم تباعا كل الخانات ما عدا الخانة المحتوية على القيمة المحورية (الخانة الأخيرة).
  - بالمتغير "**ج**" يزور فقط الخانات التي ستصبح الجزء المحتوي على القيم الأصغر من القيمة المحورية (ونسميه ايضا بالجزء الأيمن)، فبهذه المتغيرة "**ج**" يبني الخوارزم الجزء المحتوي على قيم اصغر من القيمة المحورية، وطبعا باق القيم تكون اكبر او تساوي القيمة المحورية.
- كلما زار الخوارزم خانة بالمتغيرة "**م**"، قام بمقارنة قيمة الخانة المزارة بالقيمة المحورية:
  - ان كانت قيمة الخانة المزارة اكبر من القيمة المحورية او تساويها لا يحدث اي شيئ، يستمر الخوارزم "**جزء**" فقط في زيارة الخانات، فينتقل الى الخانة التالية.
  - اما ان كانت القيمة المزارة اصغر من القيمة المحورية، ينقل الخوارزم هذه القيمة الى منطقة الجزء الأصغر، اي الى الخانة المشار اليها بقيمة المتغيرة "**ج**"، وهنا يقوم باستبدال قيم الخانات، ثم ينتقل الى الخانة التالية في منطقة الجزء الأصغر. (يضيف 1 الى "**ج**") وتستمر الزيارة على اساس المتغيرة "**م**".
- عند الانتهاء من زيارة كل الخانات باستعمال المتغيرة "**م**" (ما عدا الخانة الأخيرة الحاملة للقيمة المحورية)، يتحصل الخوارزم على ما يلي: قيمة المتغيرة **ج** تصبح هي رقم الخانة المحورية، وبناءا على "**ج**" تُنقل القيمة المحورية اليها باستعمال عملية الاستبدال.
- بعد عملية التجزئة يجب على الخوارزم "**جزء**" الإخبار بالحالة الجديدة للمجال المجزئ، ويكفي ان يخبر بموقع الخانة المحورية، فكل ما على يمينها مجال مجزئ فيه قيم اصغر

من القيمة المحورية ويجب ترتيبه وكل على ما يسارها مجال مجزئ فيه قيم اكبر من القيمة المحورية ويجب ترتيبه.

2 – 3 – 8 دراسة نص خوارزم الترتيب السريع (خطأ: لم يُعثر على مصدر المرجع) :

لإنجاز الترتيب السريع على جدول ما نحتاج الى 3 خوارزميات جزئية :

- الإجرا **ترتيب_سريع** (السطر 25)، وهو اجراء تراجعي، له 3 مداخل تمثل المجال الذي سوف تتم فيه عملية الترتيب: جدول طبيعي، رقم اول خانة من المجال ، ورقم آخر خانة من المجال، ويستعين هذا الأجراء بالوظيفة **جزء** لتحديد القسمين اللذين سوف يعاد عليهم تنفيذ نفس الخوارزم اي **ترتيب_سريع.**
- الوظيفة **جزء** (السطر 8)، التي تقسم مجالا ما الى قسمين وبينهما خانة محورية تحمل القيمة المحورية، وتنتته هذه الوظيفة بإرجاع موقع الخانة المحورية، وتستعمل هذه القيمة لتحديد الجزئين (او القسمين) اللذين سوف يعاد عليهما تنفيذ الأجراء **ترتيب_سريع.**
- الإجراء **استبدل** (السطر 2) التي شرحناها من قبل.



النص 6: خوارزم الترتيب السريع

الملحق الأول

الكلمات الشائعة الاستعمال بالعربية والإنجليزية

ترجمة الكلمات الشائعة الاستعمال الى الإنجليزية[4]

عربي	انجليزي
المكون المرن	Software
المكون الصلب	Hardware
أدا لوفلاس	Ada Lovelace
يو أم أل	UML
خارطة الانسياب	Organigram
متغيرة	Variable
قيمة	Value
ثابتة	Constant
متغيرة ثابتة	Constant Variable
تعليمة	instruction
كتلة	Bloc
ارجع	Return
اجرج	Exit
البداية	Begin
النهاية	End
اجراء	Procedure
وظيفة	function
المنافذ	Input / output
المداخل	Input
المخارج	output
المعامل	Parameter
التعليمة غادر	Break instruction
التعليمة واصل	Continue instruction
الصلة	Reference, Pointer
متغيرة ظرفية	Temporary variable
متغيرة طارئة	Dynamic variable
اقرأ	Read
اكتب	write
الإعلام الآلي	Computer Science
الذاكرة	Memory
الذاكرة النشطة	Dynamic memory
الشاشة	Screen

4 لمزيد من المصطلحات تصفح كتاب نبراس: دليل المصطلحات التقنية للأستاذ طه زروقي، قسم الإعلام الألي، جامعة البويرة

الجزء الثاني: الملحق الأول: الكلمات الشائعة الاستعمال بالعربية والإنجليزية

Printer	الطابعة، الكاتبة
type	نمط ، نوع
Variable	المتغيرات
Structured type	نمط المهيكلة
Structured variable	المتغيرات المهيكلة
Computer System	انظمة الإعلام الألي
Comments	تعليقات
Instructions	تعليمات
Algorithm	خوارزم
Static memory	ذاكرة المتغيرات العامة
execution stack, run-time stack	ذاكرة المتغيرات المحلية
Algorithm behavior	سلوك الخوارزم
Expression	عبارة
operations	عمليات
Block	كتلة
Keyboard	لوحة المفاتيح لوحة الحروف
Outputs	مخارج
Inputs	مداخل
Inputs/outputs	منافذ
Call	ناد
Perform, execute	نفذ
Starting point	نقطة الدخول او نقطة الانطلاق

مفردات لتحويل اللغة الشرمزية الى لغة جافا، سي و سي++

لغة سي	لغة سي++	لغة جافا	اللغة الشبه رمزية
exit	exit	exit	اخرج
if	if	if	اذا
return	return	return	ارجع
case	case	case	الخيار
pointer	reference, pointer	reference	الصلة
switch	switch	switch	تحول الى، حول
char	char	char	حرف
float	float	float, double	حقيقي
char *	char *	string	سلسلة
int	int	int	طبيعي
break	break	break	غادر
continue	continue	continue	واصل
for	for	for	منحنى
		for	لكل
while	while	while	مادام
----	----	boolean	منطقي
else	else	else	والا
		final	ثابت

الملحق الثاني

ترجمة خوارزميات الجزء الثاني الى لغة "جافا".

الفصل الأول

المعلومات في الواقع المعاش

ترجمة النص 2 بإضافة تحسين النص 3 الى لغة جافا

```
package part_02_chapter_01;
import java.util.Scanner;
public class SchoolClassResults {
  static Scanner keyBoard = new Scanner(System.in);

// بداية ترجمة الإجراء نتائج_القسم_في_مادة
  public static void ClassCourseResults(){
    double
       average1=0.0, average2 =0.0, average3=0.0, //average4=0.0...average10=0.0,
       //average11=0.0,. . . average20=0.0,
       //average21=0.0,. . . average30=0.0,
       //average31=0.0,. . . average39=0.0,
       average40 = 0.0;

    double average, average_sum= 0.0, class_average, minimum_average;
    int number_of_average=0;
    int number_of_failer=0;
    System.out.println("هذا الخوارزم يحسب معدل القسم في مادة ما، و يعطيك عدد الطلبة الراسبين في المادة");
    System.out.println("1- ادخل تباعا معدلات الطلبة، وعند اتمام ادخال كل المعدلات ادخل ");
    System.out.println(" ادخل المعدل");
    average = keyBoard.nextDouble();
    while (average != -1) {
       average_sum = average_sum + average;
       number_of_average++;
        /* بعد التحصل على المعدل يجب علينا ان نخزنه في المتغيرة المناسبة */
             switch (number_of_average) {
                   case 1: average1 = average;
                         break;
                   case 2: average2 = average;
                         break;
                   case 3: average3 = average;
                         break;
//....
// ....
             case 40: average40 = average;
                         break;
             case 41: System.out.println("لا يمكن لهذا الخوارزم التعاطي مع اكثر من 40 معدلا");
                   return; // نهاية مبكرة للإجراء
             }
             System.out.println(" ادخل المعدل");
       average = keyBoard.nextDouble();
    } //نهاية جسم التعليمة مادام
    class_average = average_sum/number_of_average;
    minimum_average = class_average*0.8;

    if (number_of_average > 0)
       if (average1 < minimum_average) number_of_failer++;
    if (number_of_average > 1)
       if (average2 < minimum_average) number_of_failer++;
    if (number_of_average > 2)
       if (average3 < minimum_average) number_of_failer++;
    //....
    // ...
    if (number_of_average > 39)
       if (average40 < minimum_average) number_of_failer++;

    System.out.println("عدد المعدلات التي ادخلت هي :" + number_of_average);
    System.out.println("معدل القسم هو :" + class_average);
    System.out.println("ادنى معدل مقبول هو :" + minimum_average);
    System.out.println("عدد الراسبين في المادة :" + number_of_failer);

  }
// نهاية ترجمة الإجراء نتائج_القسم_في_مادة
```

```
        public static void main(String[] argv) {
                ClassCourseResults();
        }
}
```

ترجمة النص 5: نص عير كامل للخوارزم على الطريقة الثانية

```
package part_02_chapter_01;
public class SchoolClassResultsGlobalSteps {
        static void readAverages(){System.out.println("تحصل_على_المعدلات");}
        static void commputeClassAverage(){System.out.println("احسب_معدل_القسم");};
        static void computeMinimumClassAverage(){System.out.println("احسب_ادنى_معدل");};
        static void commputeNumberOfFails(){System.out.println("عد_معدلات_الراسبين");};
        static void printResults(){System.out.println("اخرج_النتائج");};

//      بداية الترجمة للنص 5
        public static void ClassCourseResults(){
          readAverages(); // تحصل_على_المعدلات
          commputeClassAverage();  // احسب_معدل_القسم
          computeMinimumClassAverage();  // احسب_ادنى_معدل
          commputeNumberOfFails();   //  عد_معدلات_الراسبين
          printResults(); //  اخرج_النتائج
        }
//      نهاية الترجمة للنص 5

        public static void main(String[] argv) {
                ClassCourseResults();
        }
}
```

ترجمة النص 6: نص عير كامل للخوارزم نتائج_القسم_في_مادة على الطريقة الثانية، مستعملا المتغيرات العامة//

```
package part_02_chapter_01;
public class SchoolClassResultsGlobalStepsV2 {
        static double
        average1=0.0, average2 =0.0, average3=0.0, //average4=0.0...average10=0.0,
        //average11=0.0,. . . average20=0.0,
        //average21=0.0,. . . average30=0.0,
        //average31=0.0,. . . average39=0.0,
         average40 = 0.0;

        static int readAverages(){System.out.println("تحصل_على_المعدلات"); return 0;}
        static double commputeClassAverage(int number_of_average ){
                System.out.println("احسب_معدل_القسم");return 0.0;
        }
        static double computeMinimumClassAverage(double class_average){
                System.out.println("احسب_ادنى_معدل");
                return 0.0;
        }
        static int commputeNumberOfFails(double minimum_average,int
number_of_average){
                System.out.println("عد_معدلات_الراسبين");
                return 0;
        }
        static void printResults(int number_of_average, double class_average,
                                 double minimum_average, int number_of_failer){
                System.out.println("اخرج_النتائج");
        }

//      بداية الترجمة للنص 6

  public static void ClassCourseResults(){
```

```
        int number_of_average=0, number_of_failer=0;
        double class_average, minimum_average;

        number_of_average=readAverages(); // تحصل_على_المعدلات
        class_average=commputeClassAverage(number_of_average);  // احسب_معدل_القسم
        minimum_average=computeMinimumClassAverage(class_average);  // احسب_ادنى_معدل
        number_of_failer=commputeNumberOfFails(minimum_average,number_of_average );
         printResults(number_of_average,class_average,
minimum_average,number_of_failer);
   }
//      نهاية الترجمة للنص 6

  public static void main(String[] argv) {
              ClassCourseResults();
   }
}
```

ترجمة النص 6: نص كامل للخوارزم نتائج_القسم_في_مادة بعد انجازمحتوى كل الخوارزميات الجزئية//

```
package part_02_chapter_01;
import java.util.Scanner;
public class SchoolClassResultsGlobalStepsV3 {
        static Scanner keyBoard = new Scanner(System.in);
        static double
        average1=0.0, average2 =0.0, average3=0.0, //average4=0.0...average10=0.0,
        //average11=0.0,. . . average20=0.0,
        //average21=0.0,. . . average30=0.0,
        //average31=0.0,. . . average39=0.0,
         average40 = 0.0;

        static int readAverages(){
              int  number_of_average=0;
              double average;
              System.out.println("تحصل_على_المعدلات");
              System.out.println("1- ادخل تباعا معدلات الطلبة، وعند اتمام ادخال كل المعدلات ادخل  ");
              System.out.println(" ادخل المعدل");
              average = keyBoard.nextDouble();
              while (average != -1) {
                    number_of_average++;
                switch (number_of_average) {
                    case 1: average1 = average;
                                  break;
                    case 2: average2 = average;
                                  break;
                    case 3: average3 = average;
                                  break;
//....
// ....
                    case 40: average40 = average;
                                  break;
                    case 41: System.out.println(" 40 لا يمكن لهذا الخوارزم التعاطي مع اكثر من
معدلا");
                                  System.exit (0);
                }
                System.out.println(" ادخل المعدل");
                    average = keyBoard.nextDouble();
              }
              return number_of_average;
        }
        static double commputeClassAverage(int number_of_average ){
              System.out.println("احسب_معدل_القسم");
              double average_sum = 0.0;
              if (number_of_average > 0) average_sum = average_sum + average1;
              else return average_sum/number_of_average;
              if (number_of_average > 1) average_sum = average_sum + average2;
```

```
                else return average_sum/number_of_average;
                if (number_of_average > 2) average_sum = average_sum + average3;
                else return average_sum/number_of_average;
                //....
                /// ...
                if (number_of_average > 39) average_sum = average_sum + average40;
                return average_sum/number_of_average;
        }
        static double computeMinimumClassAverage(double class_average){
                System.out.println("احسب_ادنى_معدل");
                return class_average*0.8;
        }
        static int commputeNumberOfFails(double minimum_average,int
number_of_average){
                int number_of_failer =0;
                System.out.println("عد_معدلات_الراسبين");
                if (number_of_average > 0)
                        if (average1 < minimum_average) number_of_failer++;
                if (number_of_average > 1)
                        if (average2 < minimum_average) number_of_failer++;
                if (number_of_average > 2)
                        if (average3 < minimum_average) number_of_failer++;
                //....
                // ...
                if (number_of_average > 39)
                        if (average40 < minimum_average) number_of_failer++;
                return number_of_failer;
        }
        static void printResults(int number_of_average, double class_average,
                                 double minimum_average, int number_of_failer){
                System.out.println("اخرج_النتائج");
                System.out.println("عدد المعدلات التي ادخلت هي :" + number_of_average);
                System.out.println("معدل القسم هو :" + class_average);
                System.out.println("ادنى معدل مقبول هو :" + minimum_average);
                System.out.println("عدد الراسبين في المادة :" + number_of_failer);
        }

        public static void ClassCourseResults(){
                int number_of_average=0, number_of_failer=0;
                double class_average, minimum_average;

                number_of_average = readAverages(); // تحصل_على_المعدلات
                class_average = commputeClassAverage(number_of_average);  // احسب_معدل_القسم
                minimum_average = computeMinimumClassAverage(class_average);  // احسب_ادنى_معدل
                number_of_failer =
commputeNumberOfFails(minimum_average,number_of_average );
        printResults(number_of_average, class_average,
                             minimum_average, number_of_failer);
        }

        public static void main(String[] argv) {
                ClassCourseResults();
        }
}
```

libUtils ترجمة النصوص من 7 الى 12 مع استعمال تسهيلات

```
package part_02_chapter_01;
import java.util.Scanner;

import libUtils.*;
public class TheYoungest {
        public static Scanner keyBoard = new Scanner(System.in);
    ////////////////////////////////////////////////////////////////////////
```

```
    ///             اجراء مقدمة_عن_الخوارزم()                                    /////
    ////////////////////////////////////////////////////////////////////////////////
       public static void aboutTheProgram(){ // مقدمة_عن_الخوارزم
              System.out.println("هذا الخوارزم  يقبض معلومات عن شخصين ثم يبين اصغرهما");
              String dateOfTheDayInHijri = MyDate.toDay("hijri");
              String dateOfTheDayInGrego = MyDate.toDay("grego");
              String nameOfTheDay = MyDate.nameOfTheDay(dateOfTheDayInGrego);
              System.out.println("اليوم هو "+ nameOfTheDay + " " + dateOfTheDayInHijri +
"الموافق ل" + dateOfTheDayInGrego);
       }
    ////////////////////////////////////////////////////////////////////////////////
       ///      اجراء معلومات_الشخص(سلسلة &الاسم، &اللقب & العنوان؛ طبيعي &السنة، &الشهر، &اليوم؛ حقيقي &طول)
/////
    ////////////////////////////////////////////////////////////////////////////////
       public static void  personInfo(StringBuffer firstName, StringBuffer lastName,
                           StringBuffer adress, MyInteger year, MyInteger month,
                           MyInteger day, MyDouble  height){//  معلومات_الشخص


              if (keyBoard.hasNextLine()) keyBoard.nextLine();
              System.out.println("ادخل اسم الشخص");
              firstName.insert(0, keyBoard.nextLine());
              System.out.println("ادخل اسم الشخص ");
              lastName.insert(0, keyBoard.nextLine());
              System.out.println("ادخل سنة ميلاد الشخص  ");
              year.value = keyBoard.nextInt();
              System.out.println("ادخل شهر الميلاد الشخص   ");
              month.value = keyBoard.nextInt();
              System.out.println("ادخل يوم الميلاد الشخص   ");
              day.value = keyBoard.nextInt();
              keyBoard.nextLine();
              System.out.println("ادخل عنوان الشخص   ");
              adress.insert(0, keyBoard.nextLine());
              System.out.println("ادخل طول الشخص    ");
              height.value = keyBoard.nextDouble();
       }

    ////////////////////////////////////////////////////////////////////////////////
       ///             قارن_تاريخين(طبيعي سنة1، سهر1، يوم1، سنة2، سهر2، يوم
2                    /////
    ////////////////////////////////////////////////////////////////////////////////

       public static int compareDate(int year1, int month1, int day1, int year2, int
month2, int day2){
              if (year1 > year2) return 1;
              if (year1 < year2) return -1;

              if (month1 > month2) return 1;
              if (month1 < month2) return -1;

              if (day1 > day2) return 1;
              if (day1 < day2) return -1;

              return 0;
       }
    ////////////////////////////////////////////////////////////////////////////////
       ///             اجراء اخرج_النتيجة(سلسلة القب، الاسم)                        /////
       ////////////////////////////////////////////////////////////////////////////////


       public static void printTheResult(String firstName, String lastName){  // اخرج
النتيجة_
              System.out.println("اصغر شخص هو " + firstName + " " + lastName);
       }
```

```
	////////////////////////////////////////////////////////////////////////
	///                                اجراء  اساسي                        /////
	////////////////////////////////////////////////////////////////////////

	public static void main(String[] args) {
		StringBuffer
		 firstName1=new StringBuffer(),lastName1=new StringBuffer(),
		 adress1=new StringBuffer(),
		 firstName2=new StringBuffer(),lastName2=new StringBuffer(),
		 adress2=new StringBuffer();
		MyInteger
		 year1=new MyInteger(), month1=new MyInteger(),day1=new MyInteger(),
		 year2=new MyInteger(), month2=new MyInteger(), day2=new MyInteger();
		MyDouble height1=new MyDouble(), height2=new MyDouble();

		//        المرحلة الأولى  //  ؛ مقدمة_عن_الخوارزم()
		aboutTheProgram();

		//       // الحصول على معلومات الشخص الأول
	      // فضلا ادخل تباعا معلومات الشخص الأول "اكتب
		System.out.println("فضلا ادخل تباعا معلومات الشخص الأول");
		personInfo(firstName1, lastName1, adress1, year1, month1,
day1,height1);
		System.out.println("فضلا ادخل تباعا معلومات الشخص الثاني");
		personInfo(firstName2, lastName2, adress2, year2, month2,
day2,height2);
		 int result = compareDate(year1.value, month1.value, day1.value,
		                          year2.value, month2.value,  day2.value);
		 if (result == 0) System.out.println("التاريخين متساويين");
		 else if (result == 1)
		         printTheResult(firstName1.toString(), lastName1.toString());
		      else
		         printTheResult(firstName2.toString(), lastName2.toString());
	}
}
```

محتوى libUtils

في لغة جافا لا يمكن لإجراء او وظيفة ان يكون لها مخارج نمطها من الأنماط البدائية، ولذا لا يمكن الترجمة المباشرة من اللغة الشرمزية الى لغة جافا في مثل هذه الحالات، ولتفادي هذا الإشكال علينا ان ننجز في جافا ا نماط تمثل الأنماط ال بدائية و يمكن ان نضعها في المخارج، فالنمط MyInteger يمثل القيم الطبيعية والنمط MyDouble يمثل النمط حقيقي

```
//////////////////////   File MyDouble.java    ////////////////////////////
package libUtils;
public class MyDouble {
	public double  value;
	public String toString() {return ""+value;}
}
//////////////////////   File MyInteger.java    ////////////////////////////
package libUtils;
public class MyInteger {
	public int  value;
	public String toString() {return ""+value;}
}
//////////////////////   File MyDate.java    ////////////////////////////
package libUtils;
import java.util.Locale;
import org.joda.time.*;
import org.joda.time.chrono.IslamicChronology;

public class MyDate {
	/*
	 * this class use a single date format: dd/mm/yyy
	 */
	public static String toDay(String dateType){
```

```
            Chronology hijri = IslamicChronology.getInstanceUTC();

        LocalDate todayIso = LocalDate.now();
        LocalDate todayHijri = new LocalDate(todayIso.toDateTimeAtStartOfDay(),
                                             hijri);
        if (dateType.equals("hijri"))
            return (todayHijri.toString("dd/MM/yyyy"));
        // Any other String value of dateType means Gregorian date
        else return (todayIso.toString("dd/MM/yyyy"));
    }
    public static String nameOfTheDay(){
       LocalDate todayIso = LocalDate.now();
        return (todayIso.dayOfWeek().getAsText(Locale.ENGLISH));
    }

    public static String nameOfTheDay(int d, int m, int y){
       LocalDate todayIso = new LocalDate(y, m, d);
        return (todayIso.dayOfWeek().getAsText(Locale.ENGLISH));
    }
    public static String nameOfTheDay(String date){
       // date must be in dd/mm/yyyy/ format
       int d = Integer.parseInt(date.substring(0,2));
       int m = Integer.parseInt(date.substring(3,5));
       int y = Integer.parseInt(date.substring(6,10));
       LocalDate todayIso = new LocalDate(y, m, d);
        return (todayIso.dayOfWeek().getAsText(Locale.ENGLISH));
    }
}
```

الفصل الثاني

المعلومات المهيكلة

ترجمة محتوى النص 1
نلاحظ غيب ال كلمة static في تعريف محتوى النمط المهيكل

الملف SimpleDate.java
```
package part_02_chapter_02;
public class SimpleDate {    //    عرف نمط تاريخ
   int year, month, day;
}
```

الملف DateWithTime.java
```
package part_02_chapter_02;
public class DateWithTime {     // عرف نمط تاريخ_دقيق
   int year, month, day;
   int hour, minute, second = 0;
}
```

الملف Person.java
```
package part_02_chapter_02;
public class Person { //  عرف نمط شخص
     String lastName, firstName, adress;
     double height, weight;
     boolean married;
}
```

ترجمة محتوى النص 2
النمطين شخص و تاريخ المستعملين في النص2 تم انجازهما في ترجمة النص 1 و كل في ملفه

```
package part_02_chapter_02;
public class Example_01 {
      /* التصريح بمتغيرات مهيكلة عامة، و تنشأ على صورة النمط المهيكل */
    static Person p1 = new Person(), p2 = new Person(); // شخص ش1، ش2 ؛
    public static void main(String[] args) {
        Person p3=new Person(), p4=new Person();  // شخص ش3، ش4 ؛
        SimpleDate d1=new SimpleDate(), d2=new SimpleDate();// تاريخ ت1، ت2 ؛
    }
}
```

ترجمة محتوى النص 3
النمط تاريخ_دقيق المستعمل في النص 3 تم انجازهما في ترجمة النص 1 و كل في ملفه

```
package part_02_chapter_02;
public class Example_02 {
    public static DateWithTime birthDate= new DateWithTime(),
                               appointment = new DateWithTime();
    public static void main(String[] args) {
    /* جعل المتغيرة تاريخ_الولادة تمثل التاريخ 12 اوت 2011 */
      birthDate.year = 2011;
      birthDate.month = 8;
      birthDate.day = 12;
        /* جعل المتغيرة الموعد تمثل التاريخ 2 اكتوبر 2015 ، الساعة 1 الا الربع */
      appointment.year = 2015;
      appointment.month = 10;
      appointment.day = 2;
      appointment.hour = 12;
      appointment.minute = 45;
   }
}
```

ترجمة محتوى النص 4
اعادة اثراء النمط شخص الذي انجز في ترجمة النص 1

الملف Person.java : اضافة تاريخ الميلاد بالنسبة لل نمط شخص

```
package part_02_chapter_02;
public class Person {
     String lastName, firstName, adress;
     SimpleDate birthDate = new SimpleDate();
     double height, weight;
     boolean married;
{
```

الملف Example_03.java

```
package part_02_chapter_02;
public class Example_03 {
       public static void main(String[] args) {
              Person p1= new Person();
              p1.firstName= "عمر";
              p1.lastName = "عمر";
              p1.birthDate.year = 2001;
              p1.birthDate.month = 8;
              p1.birthDate.day = 12;

       }
{
```

ترجمة محتوى النص 5
مع استعمال النمط تاريخ_دقيق الذي انجز في ترجمة النص 1

```
package part_02_chapter_02;
public class Example_04 {
    static DateWithTime  dt1=new DateWithTime(), dt2=new DateWithTime();
    public static void main(String[] args) {
       dt1.day = 12;
       dt1.year = 2012;
       dt1.month = 8;
       dt1.year = 2015;
       dt1.day = 1;
       dt1.month = 3;
    }
}
```

ترجمة محتوى النص 6

الملف SpecificRectangle.java

```
package part_02_chapter_02;
public class SpecificRectangle {
    int x =200, y = 300;
    int length = 100, width=50;
}
```

الملف Example_05.java

```
package part_02_chapter_02;
public class Example_05 {
    public static SpecificRectangle r1 = new SpecificRectangle();
    public static int perimeter, area;
       public static void main(String[] args) {
              perimeter = (r1.length + r1.width)*2;
              area = r1.length * r1.width;
       }
}
```

ترجمة محتوى النص 7

```
package part_02_chapter_02;
public class Example_06 {
    static DateWithTime  dt1, dt2;
       public static void main(String[] args) {
         dt1 =new DateWithTime();
         dt2 =new DateWithTime();
               dt1.day = 12;
               dt1.year = 2012;
               dt1.month = 8;
               dt1.year = 2015;
               dt1.day = 1;
               dt1.month = 3;
       }
}
```

الفصل الثالث

تهيئة المتغيرات المهيكلة

ترجمة محتوى النص3

Date.java الملف

```
package part_02_chapter_03;
public class Date {
    int year, month, day;
    int hour, minute, second;
}
```

NumberOfDaysFromBirthToNow_Code_03.java الملف

```
package part_02_chapter_03;
import java.util.Scanner;
public class NumberOfDaysFromBirthToNow_Code_03 {
      public static Scanner   in = new Scanner(System.in);
      public static Date d1= new Date(), toDay=new Date();//التصريح بمتغيرتين عامتين من النمط تاريخ
      public static void main(String[] args) {
            //  التصريح بمتغيرات محلية من النمط طبيعي
            int numberOfDays, numberOfDays1, numberOfDays2;
            System.out.println(" خوارزم حساب عدد الأيام بين تارخين");
            System.out.println("ادخل تباعا سمات تاريخ ميلادك: اليوم، الشهر و السنة  ");
            d1.day= in.nextInt();
            d1.month= in.nextInt();
            d1.year= in.nextInt();
            System.out.println("ادخل تباعا سمات تاريخ اليوم: اليوم، الشهر و السنة  ");
            toDay.day= in.nextInt();
            toDay.month= in.nextInt();
            toDay.year= in.nextInt();
            numberOfDays1 = d1.year*365 + d1.month*30+ d1.day;
            numberOfDays2 = toDay.year*365 + toDay.month*30+ toDay.day;
            numberOfDays =numberOfDays2 - numberOfDays1;
            System.out.println("عدد الأيام التي خلت بين التاريخين هو :" + numberOfDays);
      }
}
```

ترجمة محتوى النص 4

اعادة استعمال النمط تاريخ (Date) الذي عرف في النص 3

```
package part_02_chapter_03;
import java.util.Scanner;
public class NumberOfDaysFromBirthToNow_Code_04 {
   public static Scanner   in = new Scanner(System.in);
   public static Date  d1 = new Date(), toDay = new Date();
   public static void main(String[] args) {
      int numberOfDays, numberOfDays1, numberOfDays2;
      System.out.println(" خوارزم حساب عدد الأيام بين تارخين");
      System.out.println("ادخل تباعا سمات تاريخ ميلادك: اليوم، الشهر و السنة  ");
      while(true){
         System.out.println("اليوم:");
         d1.day= in.nextInt();
         if(d1.day > 0 && d1.day < 31) break;
         System.out.println("القيمة " + d1.day + " التي ادخلتها غير صحيحة، يجب ان تكون من 1 الى 30
");
      }
      while(true){
         System.out.println("الشهر:");
         d1.month= in.nextInt();
         if(d1.month > 0 && d1.month < 13) break;
          System.out.println("القيمة " + d1.month + " التي ادخلتها غير صحيحة، يجب ان تكون من 1 الى
   12  ");
      }
      while(true){
         System.out.println("السنة:");
         d1.year= in.nextInt();
         if(d1.year >0) break;
         System.out.println("القيمة " + d1.year + " التي ادخلتها غير صحيحة  ");
```

```
      }

      System.out.println("ادخل تباعا سمات تاريخ اليوم: اليوم، الشهر و السنة  ");
      while(true){
         System.out.println("اليوم:");
         toDay.day= in.nextInt();
         if(toDay.day > 0 && toDay.day < 31) break;
         System.out.println("القيمة "+toDay.day +"30 الى 1 التي ادخلتها غير صحيحة، يجب ان تكون من
   ");
      }
      while(true){
         System.out.println("الشهر:");
         toDay.month= in.nextInt();
         if(toDay.month > 0 && toDay.month < 13) break;
         System.out.println("القيمة "+toDay.month+" الى 1 التي ادخلتها غير صحيحة، يجب ان تكون من
   12");
      }
      while(true){
         System.out.println("السنة:");
         toDay.year= in.nextInt();
         if(toDay.year >0) break;
         System.out.println("القيمة " + toDay.year + " التي ادخلتها غير صحيحة  ");
      }

      numberOfDays1 = d1.year*365 + d1.month*30+ d1.day;
      numberOfDays2 = toDay.year*365 + toDay.month*30+ toDay.day;
      numberOfDays =numberOfDays2 - numberOfDays1;
      System.out.println("عدد الأيام التي خلت بين التاريخين هو :" + numberOfDays);
   }
}
```

ترجمة محتوى النص 5

اعادة استعمال النمط تاريخ (Date) الذي عرف في النص 3

```
package part_02_chapter_03;
import java.util.Scanner;
public class NumberOfDaysFromBirthToNow_Code_05 {
   public static Scanner   in = new Scanner(System.in);
   public static Date  d1 = new Date(), toDay = new Date();
   public static  int readAnInteger(String msg, int min, int max){
      int anInteger=0;
      while(true){
         System.out.println(msg);
         anInteger = in.nextInt();
         if(anInteger >= min && anInteger <= max) break;
         System.out.println("القيمة " +anInteger+
                            "التي ادخلتها غير صحيحة، يجب ان تكون من "+min+" الي "+ max);
      }
      return anInteger;
   }

   public static void main(String[] args) {
      int numberOfDays, numberOfDays1, numberOfDays2;
      System.out.println("خوارزم حساب عدد الأيام بين تارخين ");
      System.out.println("ادخل تباعا سمات تاريخ ميلادك: اليوم، الشهر و السنة  ");
      d1.day = readAnInteger("1,30 ,":اليوم);
      d1.month = readAnInteger("1,12 ," :الشهر);
      d1.year = readAnInteger("1,3000 ,":السنة);
```

```
        System.out.println("ادخل تباعا سمات تاريخ اليوم: اليوم، الشهر و السنة  ");
       toDay.day = readAnInteger("1,30 ,":اليوم);
       toDay.month = readAnInteger("1,12 ," :الشهر);
       toDay.year = readAnInteger("1,3000 ,":السنة);

        numberOfDays1 = d1.year*365 + d1.month*30+ d1.day;
        numberOfDays2 = toDay.year*365 + toDay.month*30+ toDay.day;
        numberOfDays =numberOfDays2 - numberOfDays1;
        System.out.println("عدد الأيام التي خلت بين التاريخين هو :" + numberOfDays);
    }
}
```

ترجمة محتوى النص 6

```
package part_02_chapter_03;
import java.util.Scanner;
public class NumberOfDaysFromBirthToNow_Code_06 {
       public static Scanner   in = new Scanner(System.in);
    public static Date  d1 = new Date(), toDay = new Date();

    public static  int readAnInteger(String msg, int min, int max){
            int anInteger=0;
            while(true){
                  System.out.println(msg);
                  anInteger = in.nextInt();
                  if(anInteger >= min && anInteger <= max) break;
                  System.out.println("القيمة "+anInteger+
                           "التي ادخلتها غير صحيحة، يجب ان تكون من "+min+" الي "+ max);
            }
            return anInteger;
      }

      public static int compareDate(Date d1, Date d2){
            // المقارنة مبنية فقط على اليوم والشهر والسنة
            // تجاهل سمات التوقيت
            if (d1.year > d2.year) return 1;
            if (d1.year < d2.year) return -1;

            if (d1.month > d2.month) return 1;
            if (d1.month < d2.month) return -1;

            if (d1.day > d2.day) return 1;
            if (d1.day < d2.day) return -1;

            return 0;
      }
    public static void main(String[] args) {
            int numberOfDays, numberOfDays1, numberOfDays2;
            System.out.println("خوارزم حساب عدد الأيام بين تارخين ");
            System.out.println("ادخل تباعا سمات تاريخ ميلادك: اليوم، الشهر و السنة  ");
            d1.day = readAnInteger("1,30 ,":اليوم);
            d1.month = readAnInteger("1,12 ," :الشهر);
            d1.year = readAnInteger("1,3000 ,":السنة);

      System.out.println("ادخل تباعا سمات تاريخ اليوم: اليوم، الشهر و السنة  ");
            toDay.day = readAnInteger("1,30 ,":اليوم);
            toDay.month = readAnInteger("1,12 ," :الشهر);
            toDay.year = readAnInteger("1,3000 ,":السنة);

            if (compareDate(toDay, d1)<0)
                  System.out.println("خطأ في التواريخ: تاريخ الميلاد غير معقول، فهو احدث من تاريخ
اليوم");
```

```
            else {
                  numberOfDays1 = d1.year*365 + d1.month*30+ d1.day;
                  numberOfDays2 = toDay.year*365 + toDay.month*30+ toDay.day;
                  numberOfDays =numberOfDays2 - numberOfDays1;
                  System.out.println("عدد الأيام التي خلت بين التاريخين هو :" + numberOfDays);
            }
            System.out.println("شكرا لاستعمالك الخوارزم... الى اللقاء ");
      }
}
```

ترجمة محتوى النص 7

الملف Date.java

```
package part_02_chapter_03;
public class Date {
    int year=2016, month=1, day=1;
    int hour=0, minute=0;
}
```

الملف Person.java

```
package part_02_chapter_03;
public class Person {
      public String firtsName, lastName, adress;
      Date birthDay;
      double heigth, weigth;
      boolean married;
}
```

الملف Wheel.java

```
package part_02_chapter_03;
public class Wheel {
      String producer;
    double thikness, width, ray;
}
```

الملف Car.java

```
package part_02_chapter_03;
public class Car {
      Person owner;
      String regitrationNumber, previousRegitrationNumber;
      String carType, serialNumber;
      String structure;
      Wheel w1= new Wheel(), w2= new Wheel(), w3= new Wheel(), w4= new Wheel();
      Date manufacturingDate=new Date(), firstUseDate=new Date();
}
```

الملف Example_02_Code_07.java

```
package part_02_chapter_03;
public class Example_02_Code_07 {
      public static void main(String[] args) {
            Car c1= new Car(), c2= new Car(), c3= new Car(), c4= new Car(),
                c5= new Car(), c6= new Car(), c7= new Car();
      }
}
```

ترجمة محتوى النص 10

الملف Rectangle.java

```
package part_02_chapter_03;

public class Rectangle {
    public int x, y, heigth, width ;
}
```

الملف Example_05_Code_10.java

```
package part_02_chapter_03;
public class Example_05_Code_10 {
    public static Rectangle r1 = new Rectangle(), r2 = new Rectangle();
       public static void main(String[] args) {
              r1.x=100;
              r1.y = 200;
              r1.heigth = 60;
              r1.width = 60;

              r2.x=160;
              r2.y = 220;
              r2.heigth = 60;
              r2.width = 20;

       }
}
```

ترجمة محتوى النص 11
مع إعادة استعمال النمط مستطيل الذي عرف في النص 10

```
package part_02_chapter_03;
public class Example_06_Code_11 {
       public static void initializeRectangle(Rectangle r,int x, int y,int
heigth,int width ){
              r.x = x;  r.y = y;
              r.heigth =heigth;
              r.width =width;
       }
       public static void main(String[] args) {
              Rectangle r1=new Rectangle(), r2=new Rectangle();
              initializeRectangle(r1, 100, 200, 60, 20);
              initializeRectangle(r2, 160, 220, 60, 20);
       }
}
```

ترجمة محتوى النص 12
مع إعادة استعمال النمط مستطيل الذي عرف في النص 10

```
package part_02_chapter_03;
public class Example_07_Code_12 {

       public static Rectangle buildRectangle(int x, int y, int heigth, int width ){
              Rectangle r=new Rectangle();
              r.x = x;  r.y = y;
              r.heigth =heigth;
              r.width =width;
              return r;
       }
       public static void main(String[] args) {
              Rectangle r1=buildRectangle(100, 200, 60, 20);
              Rectangle r2=buildRectangle( 160, 220, 60, 20);
       }
}
```

ترجمة محتوى النص 13

مع إعادة استعمال النمطين تاريخ و شخص اللذين عرفا في النص 7

```
package part_02_chapter_03;
public class Example_08_Code_13 {

	public static Date createDate(int day, int month, int year){
		Date d=new Date();
		d.day = day;
		d.month =month;
		d.year =year;
		return d;
	}
	public static void initializePerson(Person p, String lName, String fName,
String adress,
		Date birthDay, boolean married ){
			p.lastName = lName;
			p.firtsName = fName;
			p.adress =  adress;
			p.married = married;
	}
	public static void main(String[] args) {
		Date d1 = createDate(15,7,2016);
		Person p1 = new Person();
		initializePerson(p1, "1000 ،البويرة" ,"عبد الحق" ,"عمر", d1, false);
	}
}
```

الفصل الرابع

العمليات الأساسية على المتغيرات المهيكلة

ترجمة محتوى النص 19

Rectangle.java الملف

```
package part_02_chapter_04;
public class Rectangle {
    public int x, y, heigth, width ;
}
```

Example_13_Code_19.java الملف

```
package part_02_chapter_04;
public class Example_13_Code_19 {
	public static Rectangle initializeRectangle(Rectangle r, int x, int y,
                                                    int heigth, int width ){
		r.x = x;  r.y = y;
		r.heigth =heigth;
		r.width =width;
		return r;
	}

	public static Rectangle r1, r2;
	public static void main(String[] args) {
		r1 = initializeRectangle(r1, 10,20, 40,60 );
		r2 = initializeRectangle(r2, 10,20, 40,60 );
	}
}
```

ترجمة محتوى النص 20
مع اعادة استعمال النمط مستطيل الذي عرف سابقا

```
package part_02_chapter_04;
public class Example_13_Version2_Of_IntializeRectangle_Code_20 {
	public static void initializeRectangle(Rectangle r, int x, int y,
                                                int heigth, int width ){
		r.x = x;  r.y = y;
		r.heigth =heigth;
		r.width =width;
	}
	public static Rectangle r1, r2;
	public static void main(String[] args) {
		initializeRectangle(r1=new Rectangle(), 10,20, 40,60 );
		initializeRectangle(r2=new Rectangle(), 10,20, 40,60 );
	}
}
```

الفصل الخامس

الخوارزميات المنتسبة للأنماط المهيكلة

ترجمة النص 1

لاحظ عدم استعمال الكلمة static عندما تكون السمة سمة النمط وعندما يكون الخوارزم الجزئي منتسبا للنمط

```
package part_02_chapter_04;
public class Rectangle {
    public int x, y, heigth, width ;
    public void initialize(int x1, int y1, int h, int w ){
           x = x1;
           y = y1;
           heigth =h;
           width =w;
      }
}
```

ترجمة النص 2

لاحظ عدم استعمال الكلمة static عندما تكون السمة سمة النمط وعندما يكون الخوارزم الجزئي منتسبا للنمط

الملف Rectangle.java

```
package part_02_chapter_04;
public class Rectangle {
    public int x, y, heigth, width ;
    public void initialize(int x1, int y1, int h, int w ){
           x = x1;
           y = y1;
           heigth =h;
           width =w;
      }
    public int perimeter(){
      return (heigth + width)*2;
    }
    public int area(){
      return heigth*width;
    }
}
```

الملف Circle.java

```
package part_02_chapter_04;
public class Circle {
    public int x, y;
    public int radius;
    public final double PI = 3.1416;
    public void initialize(int x1, int y1, int rad ){
           x = x1;
           y = y1;
           radius =rad;
      }
    public double perimeter(){
      return 2*PI*radius;
    }
    public double area(){
      return PI*radius*radius;
    }
}
```

الملف Example_01_Code_02.java

```
package part_02_chapter_04;
public class Example_01_Code_02 {
      public static void main(String[] args) {
           Rectangle r1 = new Rectangle(), r2 = new Rectangle();
           Circle c1 = new Circle(), c2 = new Circle();
```

```
                r1.initialize(100, 200, 60, 20);
                r2.initialize(160, 220, 60, 20);

                c1.initialize(100, 200, 160);
                c2.initialize(160, 220, 602);
        }
}
```

ترجمة النص 3

```
package part_02_chapter_04;
public class Rectangle {
    public int x, y, heigth, width ;
    public void initialize(int x1, int y1, int h, int w ){
                x = x1;
                y = y1;
                heigth =h;
                width =w;
        }
    public int perimeter(){
        return (heigth + width)*2;
    }
    public int area(){
        return heigth*width;
    }
    public void zoom(int zoomFactor){
        initialize(x,y,heigth*zoomFactor,width*zoomFactor);
    }
}
```

ترجمة النص 7

```
package part_02_chapter_04;
public class Rectangle {
    public int x, y, heigth, width ;
    public void initialize(int x, int y, int height, int width ){
                this.x = x;
                this.y = y;
                this.heigth =height;
                this.width =width;
        }
    public int perimeter(){
        return (this.heigth + this.width)*2;
    }
    public int area(){
        return heigth*width;
    }
    public void zoom(int zoomFactor){
        initialize(x,y,heigth*zoomFactor,width*zoomFactor);
    }
}


package part_02_chapter_04;
public class Circle {
        public static final double PI = 3.1416;
    public int x, y;
    public int radius;
    public void initialize(int x, int y, int radius ){
        this.x = x;
                this.y = y;
                this.radius =radius;
        }
    public double perimeter(){
        return 2*PI*radius;
    }
    public double area(){
```

```
        return PI*radius*radius;
    }
}

package part_02_chapter_04;
public class Example_02_Code_07 {
	public static void main(String[] args) {
		Rectangle r1 = new Rectangle(), r2 = new Rectangle();
		Circle c1 = new Circle(), c2 = new Circle();
		r1.initialize(100, 200, 60, 20);
		r2.initialize(160, 220, 60, 20);

		c1.initialize(100, 200, 160);
		c2.initialize(160, 220, 602);
	}
}
```

ترجمة النص 8

```
package part_02_chapter_05;
public class Rectangle {
    public int x, y, heigth, width ;
    public Rectangle(int x, int y, int height, int width ){
       this.x = x;
       this.y = y;
       this.heigth =height;
       this.width =width;
    }
    public void initialize(int x, int y, int height, int width ){
       this.x = x;
       this.y = y;
       this.heigth =height;
       this.width =width;
    }
 }
package part_02_chapter_05;
public class Circle {
    public static final double PI = 3.1416;
    public int x, y;
    public int radius;
    public Circle(int x, int y, int radius ){
       this.x = x;
       this.y = y;
       this.radius =radius;
    }
    public void initialize(int x, int y, int radius ){
       this.x = x;
       this.y = y;
       this.radius =radius;
    }
}

package part_02_chapter_05;

public class Example_04_Code_08 {
	public static void main(String[] args) {
		Rectangle r1, r2, r3;
		Circle c1, c2, c3;
		r1 = new Rectangle();
		r2 = new Rectangle(10,20, 40,80);
		r3 = new Rectangle(50, 120,180,90);
		r1 = new Rectangle(30,30,60,80);
		c1 = new Circle();
		c2 = new Circle(50, 120, 70);
		c3 = new Circle(80,90,180);
		c1 = new Circle(150,30,70);
	}
```

```
}
```

ترجمة النص 9 و 10

```
package part_02_chapter_05;
public class Rectangle {
    public int x, y, heigth, width ;
       public void initialize(int x, int y, int height, int width ){
              this.x = x;
              this.y = y;
              this.heigth =height;
              this.width =width;
       }
    public Rectangle(int x, int y, int height, int width ){
       initialize(x,y,height,width );
    }
    public Rectangle() {

       }
 }


package part_02_chapter_05;
public class Circle {
       public static final double PI = 3.1416;
    public int x, y;
    public int radius;

       public void initialize(int x, int y, int radius ){
       this.x = x;
              this.y = y;
              this.radius =radius;
       }
          public Circle(int x, int y, int radius ){
                 initialize(x, y, radius );
          }
          public Circle() { }
}

package part_02_chapter_05;
public class Example_05_Code_09 {
    public static Rectangle recg1 = new Rectangle(8,6,4,9);
    public static Rectangle recg2 = new Rectangle(5,90,20,5);
    public static Rectangle recg3;
    public static void main(String[] args) {
       Rectangle rec1, rec2;
       Rectangle rec3 = new Rectangle(10,20,80,40);
       rec1 = new Rectangle(310,420,80,40);
       rec2 = new Rectangle(50,120,180,90);
       recg3 = new Rectangle(58,10,110,901);
    }
}
```

ترجمة النص 11

```
package part_02_chapter_05;
public class Rectangle {
    public int x, y, heigth, width ;
    private int height;
    public Rectangle(int x1, int y1, int height1, int width1 ){
       x = x1;
       y = y1;
```

```
        heigth =height1;
        width =width1;
    }
    public Rectangle(int width1, int height1){
        heigth =height1;
        width =width1;
        x=y=0; // not required in java since x and y are zeroed by default
    }

    public Rectangle() {
        x=y=height=width=0; not required in java since all properties are zeroed by
default
    }
 }



package part_02_chapter_05;
public class Example_06_Code_11 {
        public static void main(String[] args) {
            Rectangle r1 = new Rectangle(10,10,100,100); // استعمال المشيد الأول
            Rectangle r2 = new Rectangle(200,300); // استعمال المشيد الثاني
            Rectangle r3 = new Rectangle(); // استعمال المشيد الثالث
            Rectangle r4 ;
        }
}
```

ترجمة النص 12

في النص نستعمل مستودع الإجراءات والوظائف الموجهة للرسم المسمى DrawingLib
وهذا المستودع صمم من اجل الجزء الأول من الكتاب والأجزاء الأخرى
وسوف نتخلى عنه في حينه لصاح المستودعات الرسمية للغة جافا

```
package part_02_chapter_05;
public class Rectangle {
    public int x, y, heigth, width ;
}
package part_02_chapter_05;
import java.util.Scanner;
import proceduraljava.DrawingLib;;
public class TheSingleRectangle_Code_12 {
    public static Scanner in = new Scanner(System.in);
    public static final int backGroundColor = 0;
    public static final int rectangleColor = 5;

    public static void moveRectangle(Rectangle r){
       int x1, y1;
        System.out.println("أدخل احداثيات الموقع الجديد للمستطيل");
        System.out.println("ادخل س");
        x1 = in.nextInt();
        System.out.println("ادخل ج");
        y1 = in.nextInt();
        DrawingLib.setColor(backGroundColor);
        DrawingLib.drawRectangle(r.x, r.y, r.heigth, r.width);
        r.x = x1;
        r.y = y1;
        DrawingLib.setColor(rectangleColor);
        DrawingLib.drawRectangle(r.x, r.y, r.heigth, r.width);

    }
    public static void changeRectangleSize(Rectangle  r){
        int height1, width1;
       System.out.println("ادخل سمات الحجم الجديد للمستطيل");
       System.out.println("العلو الجديد: ");
       height1 = in.nextInt();
       System.out.println("العرض الجديد: ");
       width1 = in.nextInt();
        DrawingLib.setColor(backGroundColor);
```

```
        DrawingLib.drawRectangle(r.x, r.y, r.heigth, r.width);
        r.heigth = height1;
        r.width = width1;
        DrawingLib.setColor(rectangleColor);
        DrawingLib.drawRectangle(r.x, r.y, r.heigth, r.width);
    }
    public static int choice(){
       int theChoice;
       while (true){
              System.out.println("- - - - 1 ادخل – – – – لتحريك المستطيل");
              System.out.println("- - - - 2 ادخل –المستطيل– لتغيير حجم ");
              System.out.println("- - - - 0 ادخل –------- للإنتهاء");
              System.out.println("--- ما هو خيارك؟");
              theChoice = in.nextInt();
              if (theChoice >=  0 && theChoice <= 2) return theChoice;
              System.out.println("خطأ في الخيار.. الخيارات المتاحة هي : 0، 1 و 2");
       }
    }

    public static void main(String[] args) {
       Rectangle r = new Rectangle();
       int aChoice;
              DrawingLib.initDrawingArea("700 ,800 ,0 ,0 ,"1 النسخة الوحيد، المستطيل,
backGroundColor);
       System.out.println("  خوارزم رسم و تحريك مستطيل ");
       System.out.println("  ادخل علو و عرض المستطيل ");
       System.out.println("العلو :  ");
       r.heigth = in.nextInt();
       System.out.println("العرض :  ");
       r.width = in.nextInt();
        r.x = 0; // not necessary in java since x is equal to 0 by default
        r.y = 0; // not necessary in java since x is equal to 0 by default
        DrawingLib.setColor(rectangleColor);
        DrawingLib.drawRectangle(r.x, r.y, r.heigth, r.width);
        while (true){
            aChoice = choice();
            if (aChoice == 0) break;
            if (aChoice == 1) moveRectangle(r);
            if (aChoice == 2) changeRectangleSize(r);
        }
             System.out.println("مع السلامة ....  ");
    }
}
```

ترجمة النص 13

في النص نستعمل مستودع الإجراءات والوظائف الموجهة للرسم المسمى DrawingLib
وهذا المستودع صمم من اجل الجزء الأول من الكتاب والأجزاء الأخرى
وسوف نتخلى عنه في حينه لصاح المستودعات الرسمية للغة جافا

```
package part_02_chapter_05;
import proceduraljava.DrawingLib;
public class Rectangle {
    public int x, y, height, width ;
    public int drawingColor, erasingColor;
    public Rectangle(int x, int y, int height, int width,
                     int drawingColor, int erasingColor ){
       this.x = x;
       this.y = y;
       this.height =height;
       this.width =width;
       this.drawingColor = drawingColor;
       this.erasingColor =erasingColor;
   }
   public Rectangle() {   }
   public int perimeter(){
       return (this.width+ this.height)*2;
```

```
    }
    public int area(){
        return this.width*this.height;
    }
    public void resize(double factor){
        erase();
        this.width  *=  factor;
        this.height *=  factor;
        draw();
    }
    public void move(int x, int y){
        erase();
        this.x  =  x;
        this.y  =  y;
        draw();
    }
    public void resize(double factWidth, double factHeight){
        erase();
        this.height *= factHeight;
        this.width *= factWidth;
        draw();
    }
    public void rotate(){ //  90° only
        erase();
        int tmp = this.width;
        this.width = this.height;
        this.height = tmp;
        x -= this.width;
        y -= (this.height - this.width);
        draw();
    }
    public void draw(){
        DrawingLib.setColor(drawingColor);
        DrawingLib.drawRectangle(this.x, this.y, this.height, this.width);
    }
    public void erase(){
        DrawingLib.setColor(erasingColor);
        DrawingLib.drawRectangle(this.x, this.y, this.height, this.width);
    }
    public void erase(int aColor){
        DrawingLib.setColor(aColor);
        DrawingLib.drawRectangle(this.x, this.y, this.height, this.width);
    }
}
```

ترجمة النص 14 مع اعادة استعمال النمط مستطيل الذي عرف في النص 13

في النص نستعمل مستودع الإجراءات والوظائف الموجهة للرسم المسمى DrawingLib
وهذا المستودع صمم من اجل الجزء الأول من الكتاب والأجراء الأخرى
وسوف نتخلى عنه في حينه لصاح المستودعات الرسمية للغة جافا

```
package part_02_chapter_05;
import java.util.Scanner;
import proceduraljava.DrawingLib;;
public class TheSingleRectangle_Code_14 {
    public static Scanner in = new Scanner(System.in);
    public static void moveRectangle(Rectangle r){
       int x1, y1;
        System.out.println("أدخل احداثيات الموقع الجديد للمستطيل");
        System.out.println("ادخل س");
        x1 = in.nextInt();
        System.out.println("ادخل ج");
        y1 = in.nextInt();
        r.move(x1, y1);
    }
    public static void changeRectangleSize(Rectangle  r){
        int height1, width1;
```

```
        System.out.println("ادخل سمات الحجم الجديد للمستطيل");
        System.out.println("العلو الجديد: ");
        height1 = in.nextInt();
        System.out.println("العرض الجديد: ");
        width1 = in.nextInt();
         r.resize(width1, height1);
    }
    public static int choice(){
        int theChoice;
        while (true){
               System.out.println("- - - - - 1 ادخل - - - - - لتحريك المستطيل");
               System.out.println("- - - - - 2 ادخل -المستطيل- لتغيير حجم ");
               System.out.println("- - - - - 0 ادخل -------- للإنتهاء");
               System.out.println("--- ما هو خيارك؟");
               theChoice = in.nextInt();
               if (theChoice >=  0 && theChoice <= 2) return theChoice;
               System.out.println("خطأ في الخيار.. الخيارات المتاحة هي : 0، 1 و 2");
        }
    }

    public static void main(String[] args) {
        Rectangle r ;
        int h, w;
        int aChoice;
               DrawingLib.initDrawingArea("0 ,700 ,800 ,0 ,0 ,"1 النسخة ،المستطيل الوحيد");
        System.out.println("خوارزم رسم و تحريك مستطيل  ");
        System.out.println("ادخل علو و عرض المستطيل  ");
        System.out.println("العلو :  ");
        h = in.nextInt();
        System.out.println("العرض :  ");
        w = in.nextInt();
         r = new Rectangle(0,0, h, w,5,0);
         r.draw();
         while (true){
             aChoice = choice();
             if (aChoice == 0) break;
             if (aChoice == 1) moveRectangle(r);
             if (aChoice == 2) changeRectangleSize(r);
         }
               System.out.println("مع السلامة ....  ");
    }
}
```

الفصل السادس

الأوعية: الجداول

الجزء الثاني: الملحق الثاني: ترجمة خوارزميات الجزء الثاني الى لغة "جافا"

ترجمة كيفية كتابة الأنماط المختلفة للجداول الى لغة حافا

Java Language			اللغة الشرمزية	
Content Type	Array Type	تعليق	نمط الجدول	نمط المحتوى
double, float	double[ ], float[ ]	جدول ببعد واحد للأعداد الحقيقية	حقيقي[ ]	حقيقي
int, long, short, byte	int[ ][ ], long[ ][ ], short[ ][ ], byte[ ][ ]	جدول ببعدين للأعداد الطبيعية	طبيعي[ ][ ]	طبيعي
String	String[ ]	جدول ببعد واحد لسلاسل الحروف	سلسلة[ ]	سلسلة
char	char[ ][ ][ ]	جدول بثلاثة ابعاد للحروف	حرف[ ] [ ] [ ]	حرف
Rectangle	Rectangle[ ]	جدول ببعد واحد للمستطيلات	مستطيل[ ]	مستطيل
Date	Date[ ]	جدول ببعد واحد للتواريخ	تاريخ[ ]	تاريخ
Car	Car[ ]	جدول ببعد واحد للسيارات	سيارة[ ]	سيارة
Wheel	Wheel[ ]	جدول ببعد واحد للعجلات	عجلة[ ]	عجلة

ترجمة كيفية انشاء الجداول بالتصريح بها مهيأة بقيم اولية

اللغة الشرمزية	جافا
انشاء جدول طبيعي ذو بعد واحد وبقيم اولية، حجم الجدول 4، جط1 صلة تمكن من الوصول الى الجداول الطبيعية ذات بعد واحد	
طبيعي[ ] جط1 = { 10، 15، 20، 25 } ؛	`int[] iArray= {10,15,20,25} ;`
انشاء جدول طبيعي ذو بعدين و بقيم اولية، عدد الأسطر 3 و عدد الأعمدة 6، جط2 صلة تمكن من الوصول الى الجداول الطبيعية ذات بعدين	
طبيعي[ ] [ ] جط2 ={ { 10، 15، 20، 25 ، 30، 35}، { 11، 16، 21، 26 ، 31، 36}، { 12، 17، 22، 27، 32، 37} } ؛	`int[][] iArray2D = {` `{10, 15, 20, 25, 30, 35},` `{11, 16, 21, 26, 31, 36},` `{12, 17, 22, 27, 32, 37}` `};`
انشاء جدول منطقي ذو بعدين، 3 اسطر×3 اعمدة، جم2 صلة تمكن من الوصول الى الجداول المنطقية ذات بعدين	
منطقي[ ] [ ] جم2 ={ { صحيح، صحيح، خطأ}، { صحيح، خطأ، خطأ }، { خطأ ، خطأ، خطأ } } ؛	`boolean[][] bArray2D = {` `{ true, true, false},` `{ true, false, false },` `{ false, false, false }` `};`

انشاء الجداول باستعمال العملية شيد، ترجمة بعض التصاريح الى لغة جافا

اللغة الشرمزية	لغة جافا
التصريح بالصلة **جط1** التي تمكن من الوصول لجدول ذو بعد واجد قادر على احتواء الأعداد الطبيعية، الجدول **جط1** لم ينشأ بعد، ومحتوى **جط1** غير معلوم	
طبيعي[ ] جط1	`int[] intTab1;`
التصريح بالصلة جس1 التي تمكن من الوصول لجدول ذو بعد واجد قادر على احتواء سلاسل الحروف، الجدول جس1 لم ينشأ بعد، ومحتوى **جس1** غير معلوم	
سلسلة [ ] جس1	`String[] strTab1;`
إنشاء جدول ببعد واحد فيه 12 خانة، وكل خانقة في حقيقتها متغيرة من النمط حقيقي، وتمكن الصلة **ج1** من الوصول الى هذا الجدول	
حقيقي[ ] ج1 = شيد حقيقي[12]	`double[] dTab1 = new double[12];`
إنشاء جدول ببعدين، حجم كل بعد 5، ويحتوي الجدول على 25 خانة (او متغيرة من النمط طبيعي)، وفي الجدول نجد 5 سطور و 5 اعمدة	
طبيعي[ ] [ ] ج2 = شيد طبيعي[5][5]	`int[][] intTab2D = new int[5][5];`
إنشاء جدول ببعدين، حجم البعد الأول 12، ويمثل عدد الأسطر، وحجم البعد الثاني هو 4، و يمثل عدد الأعمدة و هو نفسه عدد الخانات في كل سطر، ويحتوي الجدول على44 خانة (او متغيرة من النمط منطقي)	
منطقي[ ] [ ] ج2 = شيد طبيعي[12][4]	`boolean[][] bTab2D = new boolean[12][4];`
إنشاء جدول ببعد واحد فيه 4 خانات، وكل خانقة قادرة على احتواء المعلومات المتعلقة بمستطيل ما	
مستطيل[ ] جمس1 = شيد مستطيل[4]	`Rectangle[] recTab1 = new Rectangle[4];`

ترجمة محتوى النص 1

	نص الخوارزم	تعليق	JAVA
	خوارزم المثال_1_الفصل_6 {		`public class Example_01_Code_01 {`
1	اجراء اساسي(){		`public static void main(String[] args) {`
2	طبيعي س=2، ع=0	التصريح بقيمة اولية لمتغيرتين من النمط طبيعي	`int x = 2, y =0`
3	طبيعي[ ] جط1_1 = شيد طبيعي[20] ؛	إنشاء جدول ببعد واحد يتسع لعشرين معلومة من النمط طبيعي	`int[] iTab1_1 = new int[20]`
4	طبيعي[ ] [ ] جط2_1 = شيد طبيعي[10] [15]	إنشاء جدول ببعدين يتسع لمئة وخمسين معلومة (10 اسطر على 15 عمود) من النمط طبيعي	`int[ ][ ] iTab2_1 = new int[10][15]`
5	طبيعي[ ] جط1_2 ؛	التصريح بصلة تستعمل مع الجداول من النمط طبيعي ببعد واحد	`int[] iTab1_2`
6	طبيعي[ ][ ] جط2_2 ؛	التصريح بصلة تستعمل مع الجداول من النمط طبيعي ببعدين	`int[ ][ ] iTab2_2`
7	جط1_1[0] = 10 ؛	شحن الخانة الأولى من الجدول جط1_1 بالقيمة 10،	`iTab1_1[0] = 10`

8	جط1_1[23%5] = 40 ؛	شحن الخانة الرابعة من الجدول جط1_1 بالقيمة 40، (رقم الخانة الرابعة هو 3)	iTab1_1[23%5] = 40
9	جط1_1[س] = 30 ؛	شحن الخانة الثالثة من الجدول جط1_1 بالقيمة 30، (قيمة س هي 2 و رقم الخانة الثالثة هو 2)	iTab1_1[x]=30
10	جط1_1[س - 1] = جط1_1[س -2] * 2 ؛	شحن الخانة الثانية بقيمة الخانة الأولى مضروبة في 2 اي 10 * 2، اي 20	iTab1_1[x-1]=iTab1_1[x-2]*2
11	جط1_1[19] = 100 ؛	شحن الخانة الأخيرة بالقيمة 100(الترقيم يبدأ من 0 و ينته في 19، اي حجم البعد - 1)	iTab1_1[19]= 100
12	س = جط1_1 [س]*2	تغيير قيمة س لتصبح 60 (ما في جط1_1[س] هو ما في جط1_1[2] اي 30)	x= iTab1_1[x]*2
13	جط1_1[س] = 0	خطأ خطير: محاولة الوصول الى الخانة الواحدة و الستون (الرقم 60)، و هي خانة غير موجودة	iTab1_1[x] = 0
14	جط1_2[0] = 5	خطأ خطير: محاولة الوصول لخانة من جدول غير معلوم، الصلة جط1_2 لم تربط باي جدول	iTab1_2[5] = 0
15	جط1_2 = جط2_1	خطأ خطير: عدم توافق الأبعاد، جط1_2 صلة لجدول ببعد واحد، وجط2_1 صلة لجدول ببعدين	iTab1_2 = iTab2_1
16	جط1_2 = جط1_1	الجدول جط1_1 له الآن صلتين: جط1_1 و جط 2_1	iTab1_2 = iTab1_1
17	جط1_2[س/6]= 16	شحن الخانة الحادية عشر بالقيمة 16 (س/6 تساوي 10 كون المتغيرة س تحتوي على القيمة 60)	iTab1_2[x/6] = 16
18	جط1_2[4][6] = 12	خطأ خطير: عدم توافق الأبعاد، الجدول جط1_2 ذو بعد واحد و ليس ذو بعدين	iTab1_2[4][6] = 12
19	جط2_2[4][6]= 18 ؛	خطأ خطير: الكتابة صحيحة لكنها غير واقعية، الصلة جط2_2 لم تربط بأي جدول حقيقي معلوم	iTab2_2[4][6] = 12
20	جط2_2 = جط2_1	الجدول الموجود جط2_1 له الآن صلتين: جط 2_1 و جط2_2	iTab2_2 = iTab2_1
21	جط2_2[4][6] = 12	الوصول الى خانة السطر 5 و العمود 7 من الجدول جط2_1 عبر جط2_2 و شحنها بالقيمة 12	iTab2_2[4][6] = 12
22	ع = س = جط2_1[4][6]	شحن المتغيرة س بما في الخانة (سطر رقم 4، عمود رقم 6) اي القيمة 12، ثم شحن ع بما في س	x=y= iTab2_1[4][6]
23	جط2_1[س][ع] = جط2_1[س][ع]*2	شحن الخانة (سطر رقم س اي 12، عمود رقم ع اي 12) بما فيها في 2، اي 12*2 اي 24	iTab2_1[x][y] = iTab2_1[x][y]*2
24	جط2_1[س+1][ع-1] = جط2_1[س][ع]*س- 10	شحن الخانة (سطر رقم س+1 اي 13، عمود رقم ع -1، اي 11) بما في الخانة (سطر رقم س اي 12، عمود رقم ع اي 12) اي 12، في س، اي 12*12، اي 144 ثم نقص 10، اي ان النتيجة هي 134 في الخانة (سطر 13، عمود 11)	iTab2_1[x+1][y-1] = iTab2_1[x][y]*x -10
25	جط2_2[س*3][6] = 12	خطأ خطير: السطر س*3 اي 12*3 اي 36 غير موجود في الجدول جط2_2 الذي هو ايضا الجدول جط2_1	**iTab2_2[x*3][6] = 12**
26	{ // نهاية الإجراء اساسي		}
{ // نهاية الخوارزم المثال_1_الفصل_6			}

ترجمة محتوى النص 2

```
package part_02_chapter_06;
import java.util.Scanner;
public class Example_02_Code_02 {
    public static Scanner in  = new Scanner(System.in);
    public static int readAnArrayLength(String message){
       int i;
       while (true){
              System.out.println(message);
              i = in.nextInt();
              if (i > 0) return i;
              System.out.println("خطأ، الحجم يجب ان يكون اكبر من 0");
       }
    }
    public static void main(String[] args) {
       int x=10, y=20;
       int x1,y1;
```

```
        int[]    tab1_1 = new int[x];
        int[][]  tab2_1 = new int[x][y];
        int[] tab1_2, tab1_3;
        int[][]  tab2_2, tab2_3;
        x = readAnArrayLength("اعطني حجم الجدول ");
        tab1_2 = new int[x];
        x = readAnArrayLength("اعطني حجم البعد الأول ");
        y = readAnArrayLength("اعطني حجم البعد الثاني ");
        tab2_2 = new int[x][y];
        //tab1_3=new int[x1];//The local variable x1 may not have been initialized
        x1 = readAnArrayLength("اعطني حجم الجدول ");
        //tab2_2=new int[x1][y1];//The local variable y1 may not have been
initialized
    }
}
```

ترجمة محتوى النص 3

```
package part_02_chapter_06;
import java.util.Scanner;
public class Example_03_Code_03 {
    public static Scanner in  = new Scanner(System.in);
    public static int readAnArrayIndex(String message, int length){
       int i;
       while (true){
              System.out.println(message);
              i = in.nextInt();
              if (i >= 0  && i < length) return i;
              System.out.println("خطأ، الرقم يجب ان يكون من 0 الى " + (length-1));
       }
    }
    public static void main(String[] args) {
       int x=10, choice, val;
       int[]    iTab1 = new int[x];
       System.out.println("هل تريد ان تملأ جدولا ذو بعد واحد حجمه "+ x);
       System.out.println("0 ادخل ،1 ادخل :لا  نعم: ");
       choice = in.nextInt();
       if (choice == 1){
              while (true){
                     System.out.println("ادخل القيمة التي تريد وضعها في الجدول ");
                     val= in.nextInt();
                     int i = readAnArrayIndex("ادخل رقم الخانة ", x);
                     iTab1[i] = val;
                     System.out.println("هل تريد ان تتابع العملية ");
                     System.out.println("0 ادخل ،1 ادخل :لا  نعم: ");
                     choice = in.nextInt();
                     if (choice == 0) break;
              } // end of while
       }// end of if
       System.out.println("شكرا على الزيارة... الى اللقاء.. ");
    }
}
```

ترجمة محتوى النص 4

```
package part_02_chapter_06;
import java.util.Scanner;

class Int2DArrayCellPosition{
       public int i, j;
}
public class Example_04_Code_04 {
    public static Scanner in  = new Scanner(System.in);
    public static int readAnArrayIndex(String message, int length){
       int i;
       while (true){
```

```
            System.out.println(message);
            i = in.nextInt();
            if (i >= 0  && i < length) return i;
            System.out.println("خطأ، الرقم يجب ان يكون من 0 الى " + (length-1));
        }
    }

      public static Int2DArrayCellPosition read2DArrayPosition(int numberOfLines,
                                                               int numberOfColumns){
            Int2DArrayCellPosition cellPosition = new Int2DArrayCellPosition();
            cellPosition.i = readAnArrayIndex("اعطني رقم السطر ", numberOfLines);
            cellPosition.j = readAnArrayIndex("اعطني رقم العمود ", numberOfColumns);
            return cellPosition;
      }
      public static void main(String[] args) {
            int lines=10, columns = 20, choice, val;
            Int2DArrayCellPosition position;
            int[][]    iTab2D = new int[lines][columns];
            System.out.println("هل تريد ان تملأ جدولا ذو بعدين:  "+ lines + "على" + columns);
            System.out.println("0 ادخل ،1 ادخل :لا  نعم:  ");
            choice = in.nextInt();
            if (choice == 1){
                  while (true){
                        System.out.println("ادخل القيمة التي تريد وضعها في الجدول ");
                        val= in.nextInt();
                        position = read2DArrayPosition(lines, columns);
                        iTab2D[position.i][position.j] = val;
                        System.out.println("هل تريد ان تتابع العملية ");
                        System.out.println("0 ادخل ،1 ادخل :لا  نعم:  ");
                        choice = in.nextInt();
                        if (choice == 0) break;
                  } // end of while
            }// end of if
            System.out.println("شكرا على الزيارة... الى اللقاء.. ");
      }
}
```

ترجمة محتوى النص 4

```
package part_02_chapter_06;
public class Example_05_Code_05 {
    public static Rectangle[] recTab1D= new Rectangle[6];
    public static Rectangle[][] recTab2D= new Rectangle[20][10];
    public static void main(String[] args) {
       int area, area2;
       recTab1D[4] = new Rectangle();
       recTab1D[4].x = 100;
       recTab1D[4].y = 40;
       recTab1D[4].heigth = 210;
       recTab1D[4].width = 70;
       area = recTab1D[4].heigth*recTab1D[4].width;

       recTab2D[5][9] = new Rectangle();
       recTab2D[5][9].x = recTab1D[4].x;
       recTab2D[5][9].y = recTab1D[4].y;
       recTab2D[5][9].heigth = recTab1D[4].heigth*2;
       recTab2D[5][9].width = recTab1D[4].width*3 + recTab2D[5][9].heigth;
        area2 = recTab2D[5][9].heigth*recTab2D[5][9].width;
   }
}
```

الفصل التاسع

الأوعية: التجوال في الجداول (1)

ترجمة النص 3

```
package part_02_chapter_09;
public class Example_03_Code_03 {
	public static void main(String[] args) {
		int arraySize = 100;
		int[] integerTab = new int[arraySize];
		int cell = 0;
		while (cell < arraySize){
			integerTab[cell] =0;
			cell++;
		}
	}
}
```

ترجمة النص4

```
package part_02_chapter_09;
import java.util.Scanner;
public class Example_04_Code_04 {
    public static Scanner in = new Scanner(System.in);
    public static void main(String[] args) {
       int size, iCell;
       int commonValue;
       int[] iTab;
       System.out.println("اهلا و سهلا في خوارزم تعمير جدول طبيعي ذو بعد واحد بقيمة طبيعية واحدة");
       System.out.println("فضلا، حدد القيمة التي تريد وضعها في كل خانات الجدول:");
       commonValue = in.nextInt();
       System.out.println("فضلا، حدد حجم الجدول الذي تريد تعميره بالقيمة " + commonValue);
       size = in.nextInt();
       iTab = new int[size];
       iCell = 0;
       while (iCell < size){
              iTab[iCell] = commonValue;
              commonValue++;
       }
   }
{
```

ترجمة النص 5

```
package part_02_chapter_09;
import java.util.Scanner;

public class Example_05_Code_05 {
	public static Scanner in = new Scanner(System.in);
    public static void fillArrayWithAValue(int[]  iArray, int val){
       int i = 0;
       while (i < iArray.length){
              iArray[i] = val;
              i++;
       }
    }
       public static void main(String[] args) {
              int size, iCell;
              int commonValue;
              int[] iTab;
              System.out.println("اهلا و سهلا في خوارزم تعمير جدول طبيعي ذو بعد واحد بقيمة طبيعية واحدة");
              System.out.println("فضلا، حدد القيمة التي تريد وضعها في كل خانات الجدول:");
              commonValue = in.nextInt();
              System.out.println("فضلا، حدد حجم الجدول الذي تريد تعميره بالقيمة " + commonValue);
              size = in.nextInt();
              iTab = new int[size];
              fillArrayWithAValue(iTab, commonValue);
```

```
	}
}
```

ترجمة النص 6

```
package part_02_chapter_09;
import java.util.Scanner;

public class Example_06_Code_06 {
	public static Scanner in = new Scanner(System.in);
    public static int[] createAndFillArrayWithAValue(int size, int val){
	int[]  iArray = new int[size];
	int i = 0;
	while (i < iArray.length){
		iArray[i] = val;
		i++;
	}
	return iArray;
    }
	public static void main(String[] args) {
		int size, iCell;
		int commonValue;
		int[] iTab;
		System.out.println("اهلا و سهلا في خوارزم تعمير جدول طبيعي ذو بعد واحد بقيمة طبيعية واحدة");
		System.out.println("فضلا، حدد القيمة التي تريد وضعها في كل خانات الجدول:");
		commonValue = in.nextInt();
		System.out.println("فضلا، حدد حجم الجدول الذي تريد تعميره بالقيمة " + commonValue);
		size = in.nextInt();
		iTab= createAndFillArrayWithAValue(size, commonValue);
	}
}
```

ترجمة النص 9

```
package part_02_chapter_09;
import java.util.Scanner;
public class Example_09_Code_09 {
	public static Scanner in = new Scanner(System.in);
	public static void main(String[] args) {
		int numberOfStudents = 25;
		double[] grades = new double[numberOfStudents];
		int iStudent = 1;
		System.out.println("أهلا و سهلا في خوارزم تعمير جدول المعدلات");
		while (iStudent <= numberOfStudents){
			System.out.println("أدخل معدل التلميذ رقم  " + iStudent);
			grades[iStudent-1] = in.nextDouble();
			iStudent++;
		}
	}
}
```

ترجمة النص 10

```
package part_02_chapter_09;
import java.util.Scanner;
public class Example_10_Code_10 {
	public static Scanner in = new Scanner(System.in);
	public static void main(String[] args) {
		int numberOfStudents;
		double[] grades;
		int iStudent = 1;
		System.out.println("أهلا و سهلا في خوارزم تعمير جدول المعدلات");
		System.out.println("فضلا، وفني بعدد التلاميذ في القسم");
		numberOfStudents = in.nextInt();
```

```
		grades = new double[numberOfStudents];
		while (iStudent <= numberOfStudents){
			System.out.println("أدخل معدل التلميذ رقم  " + iStudent);
			grades[iStudent-1] = in.nextDouble();
			iStudent++;
		}
	}
}
```

ترجمة النص 11

```
package part_02_chapter_09;
import java.util.Scanner;
public class Example_11_Code_11 {
	public static Scanner in = new Scanner(System.in);

	public static double[] createAndFillGradeTable(int studentNumber){
		double[] gradeTable =  new double[studentNumber];
		int iStudent = 1;
		while (iStudent <= studentNumber){
			System.out.println("أدخل معدل التلميذ رقم  " + iStudent);
			gradeTable[iStudent-1] = in.nextDouble();
			iStudent++;
		}
		return gradeTable;
	}

	public static void main(String[] args) {
		int numberOfStudents;
		double[] grades;
		int iStudent = 1;
		System.out.println("أهلا و سهلا في خوارزم تعمير جدول المعدلات");
		System.out.println("فضلا، وفني بعدد التلاميذ في القسم");
		numberOfStudents = in.nextInt();
		grades = createAndFillGradeTable(numberOfStudents);
	}
}
```

ترجمة النص 14

```
package part_02_chapter_09;
public class Example_14_Code_14 {
	public static void main(String[] args) {
		int lines=20, columns=9;
		int[][] iTab2D = new int[lines][columns];
		int common_value= 100;
		int i, j;
		i = 0;
		while (i < lines){
			j = 0;
			while ( j < columns){
				iTab2D[i][j] = common_value;
				j++;
			}
			i++;
		}
	}
}
```

ترجمة النص 15

```
package part_02_chapter_09;
import java.util.Scanner;
```

```
public class Example_15_Code_15 {
	public static Scanner in = new Scanner(System.in);

	public static int[][]  createAndFillIntegerTab2D(int lines, int columns, int
val){
		int[][] iTab2D =  new int[lines][columns];
	    int i, j;
		i = 0;
		while (i < lines){
			j = 0;
			while ( j < columns){
				iTab2D[i][j] = val;
				j++;
			}
			i++;
		}
		return iTab2D;
	}

	public static void main(String[] args) {
		int[][] iTab2D ;
		int lines, columns, common_value;
		System.out.println(" فضلا، ادخل على التوالي: عدد الأسطر، عدد الأعمدة و القيمة التي تريد ان تعمر
بها الجدول");
		lines = in.nextInt();
		columns = in.nextInt();
		common_value = in.nextInt();
		iTab2D = createAndFillIntegerTab2D(lines, columns, common_value);
	}
}
```

ترجمة النص 19

```
package part_02_chapter_09;
import java.util.Scanner;

public class Example_19_Code_19 {
	public static Scanner in = new Scanner(System.in);

	public static void main(String[] args) {
		int lineNumber, columnNumber;
		int numberOfLines = 25, numberOfColumns = 4;
		double[][] dTab2D = new double[numberOfLines][numberOfColumns];
		System.out.println("أهلا و سهلا في خوارزم تعمير جدول النقاط");
		lineNumber = 0;
		while (lineNumber <  numberOfLines){
			System.out.println("فضلا، وفني بنقاط التلميذ رقم "+ (lineNumber + 1));
			columnNumber = 0;
			while (columnNumber < numberOfColumns){
				System.out.println("فضلا، وفني بنقاط التلميذ رقم " + (columnNumber +
1));

				dTab2D[lineNumber][columnNumber] = in.nextDouble();
				columnNumber++;
			}
			lineNumber++;
		}
	}
}
```

ترجمة النص 20

```
package part_02_chapter_09;
import java.util.Scanner;

public class Example_20_Code_20 {
```

```
	public static Scanner in = new Scanner(System.in);
	public static void fillTableWithAStudentGrade(double[]  dTab){
		int i = 0;
		while (i < dTab.length){
			System.out.println("فضلا، وفني بنقاط التلميذ رقم " + (i + 1));
			dTab[i] = in.nextDouble();
			i++;
		}

	}
	public static void main(String[] args) {
		int lineNumber;
		int numberOfLines = 25, numberOfColumns = 4;
		double[][] dTab2D = new double[numberOfLines][numberOfColumns];
		System.out.println("أهلا و سهلا في خوارزم تعمير جدول النقاط");
		lineNumber = 0;
		while (lineNumber <  numberOfLines){
			System.out.println("فضلا، وفني بنقاط التلميذ رقم "+ (lineNumber + 1));
			 fillTableWithAStudentGrade(dTab2D[lineNumber]);
			lineNumber++;
		}
	}
}
```

ترجمة النص 21

```
package part_02_chapter_09;
import java.util.Scanner;

public class Example_21_Code_21 {
	public static Scanner in = new Scanner(System.in);
	public static void fillTableWithAStudentGrade(double[]  dTab){
		int i = 0;
		while (i < dTab.length){
			System.out.println("فضلا، وفني بنقاط التلميذ رقم " + (i + 1));
			dTab[i] = in.nextDouble();
			i++;
		}
	}
	public static double[][] fillStudentGradeArray(int nbStudent, int
nbGradeByStudent){
		double[][] studentGradeArray = new double[nbStudent]
[nbGradeByStudent];
		int line = 0;
		while (line < nbStudent){
			System.out.println("فضلا، وفني بنقاط التلميذ رقم "+ (line + 1));
			fillTableWithAStudentGrade(studentGradeArray[line]);
			line++;
		}
		return studentGradeArray;
	}
	public static void main(String[] args) {
		int numberOfStudent, numberOfGrades;
		double[][] dTab2D ;
		System.out.println("أهلا و سهلا في خوارزم تعمير جدول النقاط");
		System.out.println("فضلا، وفني بعدد تلاميذ القسم ");
		numberOfStudent = in.nextInt();
		System.out.println("فضلا، وفني بعدد النقاط في المادة  ");
		numberOfGrades = in.nextInt();
		dTab2D = fillStudentGradeArray(numberOfStudent, numberOfGrades);
	}
}
```

ترجمة النص 22

```
package part_02_chapter_09;
import java.util.Scanner;

public class Example_22_Code_22 {
```

```
	public static Scanner in = new Scanner(System.in);

	public static void fillTableWithAStudentGrade(double[]  dTab){
		int i = 0;
		while (i < dTab.length){
			System.out.println("وفني بالنقطة رقم " + (i + 1));
			dTab[i] = in.nextDouble();
			i++;
		}
	}

	public static double[][] fillStudentGradeArray(int nbStudent){
		double[][] studentGradeArray = new double[nbStudent][];
		int nbGradeByStudent;
		int line = 0;
		while (line < nbStudent){
			System.out.println("فضلا، وفني بعدد نقاط  التلميذ رقم "+ (line + 1));
			nbGradeByStudent = in.nextInt();
			studentGradeArray[line] = new double[nbGradeByStudent];
			System.out.println("فضلا، وفني بنقاط التلميذ رقم "+ (line + 1));
			fillTableWithAStudentGrade(studentGradeArray[line]);
			line++;
		}
		return studentGradeArray;
	}

	public static void main(String[] args) {
		int numberOfStudent;
		double[][] dTab2D ;
		System.out.println("أهلا و سهلا في خوارزم تعمير جدول النقاط");
		System.out.println("فضلا، وفني بعدد تلاميذ القسم ");
		numberOfStudent = in.nextInt();
		dTab2D = fillStudentGradeArray(numberOfStudent);
	}
}
```

الفصل العاشر

الأوعية: التجوال في الجداول (2)

ترجمة النص 1

```
package part_02_chapter_10;

import java.util.Scanner;

public class Example_01_Code_01 {
	public static Scanner in = new Scanner(System.in);
	public static void main(String[] args) {
		int num1, num2, num3;
		System.out.println("100 و 1 فضلا، ادخل رقما ما بين:");
		num1 = in.nextInt();
		System.out.print("200 و 101 فضلا، ادخل رقما ما بين:");
		num2 = in.nextInt();
		System.out.print("300 و 201 فضلا، ادخل رقما ما بين:" + "\n");
		num3 = in.nextInt();
		System.out.print(num1 + "," + num2 + "," + num3 +".");
		System.out.println(num1 + "--" + num2 + "--" + num3 +".");
		System.out.print("1 سطر: "+ num1 + "," + num2 + "----" + "\n" + ":2 سطر
500 + " + ","+num3 + "----");
	}
}
```

ترجمة النص 2

```
package part_02_chapter_10;
import java.util.Scanner;

public class Example_02_Code_02 {
	public static Scanner in = new Scanner(System.in);
	public static void horizontal_display(int[]  iTab){
		int num_cell=0;
		String deco_line = "-------------------------";
		char cell_limit = '|';
		System.out.println(deco_line);
		while (num_cell < iTab.length){
			System.out.print(cell_limit);
			System.out.print(iTab[num_cell]);
			num_cell++;
		}
		System.out.println(cell_limit);
		System.out.println(deco_line);

	}
	public static void main(String[] args) {
		int iTable[] = {12, 24, 48, 96, 10, 20, 40, 80};
		System.out.println("اهلا و سهلا، اختبار اجراء عرض جدول افقيا");
		horizontal_display(iTable);
		System.out.println("نأمل ان يكون العرض الأفقي للجدول قد نال اعجابك.....");
	}
}
```

ترجمة النص 3

```
package part_02_chapter_10;
import java.util.Scanner;

public class Example_03_Code_03 {
	public static Scanner in = new Scanner(System.in);
	public static void vertital_display(int[]  iTab){
		int num_cell=0;
		String deco_line = "----";
		char cell_limit = '|';
		System.out.println(deco_line);
		while (num_cell < iTab.length){
```

```
                System.out.println(""+cell_limit+ iTab[num_cell]+cell_limit);
                num_cell++;
            }
            System.out.println(deco_line);
        }
        public static void main(String[] args) {
            int iTable[] = {48, 96, 40, 80};
            System.out.println("اهلا و سهلا، اختبار اجراء عرض جدول افقيا");
            vertital_display(iTable);
            System.out.println("نأمل ان يكون العرض الأفقي للجدول قد نال اعجابك.....");
        }
}
```

ترجمة النص 4

```
package part_02_chapter_10;
import java.util.Scanner;

public class Example_04_Code_04 {
        public static Scanner in = new Scanner(System.in);
        public static void horizontal_display(int[]  iTab, String deco_line,char
cell_limit){
            int num_cell=0;
            System.out.println(deco_line);
            while (num_cell < iTab.length){
                System.out.print(cell_limit);
                System.out.print(iTab[num_cell]);
                num_cell++;
            }
            System.out.println(cell_limit);
            System.out.println(deco_line);
        }
        public static void main(String[] args) {
            int iTable[] = {12, 24, 48, 96, 10, 20, 40, 80};
            System.out.println("اهلا و سهلا، اختبار اول  للنسخة الثانية من اجراء عرض جدول افقيا");
            horizontal_display(iTable, "-------------------------", '|');
            System.out.println("اهلا و سهلا، اختبار ثان  للنسخة الثانية من اجراء عرض جدول افقيا");
            horizontal_display(iTable, "*************************", '*');
                System.out.println("نأمل ان يكون العرض الأفقي للجدول قد نال اعجابك.....");
        }
}
```

ترجمة النص5

```
package part_02_chapter_10;
import java.util.Scanner;

public class Example_05_Code_05 {
        public static Scanner in = new Scanner(System.in);
        public static void vertital_display(int[]  iTab, String deco_line, char
cell_limit){
            int num_cell=0;
            System.out.println(deco_line);
            while (num_cell < iTab.length){
                System.out.println(""+cell_limit+ iTab[num_cell]+cell_limit);
                num_cell++;
            }
            System.out.println(deco_line);
        }
        public static void main(String[] args) {
            int iTable[] = {48, 96, 40, 80};
            System.out.println("اهلا و سهلا، اختبار اول  للنسخة الثانية من اجراء عرض جدول عموديا");
            vertital_display(iTable, "----", '|');
            System.out.println("اهلا و سهلا، اختبار ثان  للنسخة الثانية من اجراء عرض جدول عموديا");
            vertital_display(iTable, "====", '*');
```

```
                System.out.println("نأمل ان يكون العرض الأفقي للجدول قد نال اعجابك.....");
        }
}
```

ترجمة النص 6

```
package part_02_chapter_10;
import java.util.Scanner;

public class Example_06_Code_06 {
        public static Scanner in = new Scanner(System.in);

        public static int[]  createAndFillIntTab(){
                int[] iTab;
                int size, numCell;
                System.out.println(" فضلا، وفني بعدد القيم التي تريد ادخالها ");
                size = in.nextInt();
                iTab = new int[size];
                numCell = 0;
                while (numCell < size){
                        System.out.println("أدخل قيمة : ");
                        iTab[numCell] = in.nextInt();
                        numCell++;
                }
                return iTab;
        }

        public static int[] sum_2_consecutive_cell(int[] iTab){
                int lengthOfResultingTab= iTab.length / 2;
                int[] resultingTab = new int[lengthOfResultingTab];
                int num1=0, num2=0;
                while (num2 < lengthOfResultingTab){
                        resultingTab[num2] = iTab[num1]+ iTab[num1+1];
                        num2++;
                        num1 += 2;  //equivalent to  num2 = num2 + 2
                }
                return resultingTab;
        }
        public static void horizontal_display(int[] iTab,String deco_line,char
cell_limit){
                int num_cell=0;
                System.out.println(deco_line);
                while (num_cell < iTab.length){
                        System.out.print(cell_limit);
                        System.out.print(iTab[num_cell]);
                        num_cell++;
                }
                System.out.println(cell_limit);
                System.out.println(deco_line);
        }
        public static void main(String[] args) {
                int[] iTab1, iTab2;
                iTab1 = createAndFillIntTab();
                iTab2 = sum_2_consecutive_cell(iTab1);
                horizontal_display(iTab1, "---------------------------------", '|');
                horizontal_display(iTab2, "----------------", '|');
        }
}
```

ترجمة النص 8 و 9

```
package part_02_chapter_10;
import java.util.Scanner;
```

```
public class Example_08_09_Code_08_09 {
	public static Scanner in = new Scanner(System.in);

	public static void display_line(char deco, int length){
		int counter = 0;
		while (counter < length){
			System.out.print(deco);
			counter++;
		}
		System.out.println();
	}
	public static int[]  createAndFillIntTab(){
		int[] iTab;
		int size, numCell;
		System.out.println("فضلا، وفني بعدد القيم التي تريد ادخالها ");
		size = in.nextInt();
		iTab = new int[size];
		numCell = 0;
		while (numCell < size){
			System.out.println("أدخل قيمة : ");
			iTab[numCell] = in.nextInt();
			numCell++;
		}
		return iTab;
	}


	public static int[] sum_2_consecutive_cell(int[] iTab){
		int lengthOfResultingTab= iTab.length / 2;
		int[] resultingTab = new int[lengthOfResultingTab];
		int num1=0, num2=0;
		while (num2 < lengthOfResultingTab){
			resultingTab[num2] = iTab[num1]+ iTab[num1+1];
			num2++;
			num1 += 2;  //equivalent to  num2 = num2 + 2
		}
		return resultingTab;
	}
	public static void horizontal_display(int[]  iTab, char deco,char cell_limit)
{
		int num_cell=0;
		int decoLineLength = iTab.length;
		display_line(deco, decoLineLength);
		while (num_cell < iTab.length){
			System.out.print(cell_limit);
			System.out.print(iTab[num_cell]);
			num_cell++;
		}
		System.out.println(cell_limit);
		display_line(deco, decoLineLength);
	}
	public static void main(String[] args) {
		int[] iTab1, iTab2;
		iTab1 = createAndFillIntTab();
		iTab2 = sum_2_consecutive_cell(iTab1);
		horizontal_display(iTab1, '-', '|');
		horizontal_display(iTab2, '-', '|');

	}

}
```

ترجمة النص 10

```
package part_02_chapter_10;
import java.util.Scanner;

public class Example_08_09_Code_08_09 {
```

```
	public static Scanner in = new Scanner(System.in);

	public static String build_deco_line(char deco, int length){
		String deco_line = "";
		int counter = 0;
		while (counter < length){
			deco_line = deco_line+ deco;
			counter++;
		}
		return deco_line;
	}
	public static int[]  createAndFillIntTab(){
		int[] iTab;
		int size, numCell;
		System.out.println("فضلا، وفني بعدد القيم التي تريد ادخالها ");
		size = in.nextInt();
		iTab = new int[size];
		numCell = 0;
		while (numCell < size){
			System.out.println("أدخل قيمة : ");
			iTab[numCell] = in.nextInt();
			numCell++;
		}
		return iTab;
	}

	public static int[] sum_2_consecutive_cell(int[] iTab){
		int lengthOfResultingTab= iTab.length / 2;
		int[] resultingTab = new int[lengthOfResultingTab];
		int num1=0, num2=0;
		while (num2 < lengthOfResultingTab){
			resultingTab[num2] = iTab[num1]+ iTab[num1+1];
			num2++;
			num1 += 2;  //equivalent to  num2 = num2 + 2
		}
		return resultingTab;
	}
	public static void horizontal_display(int[]  iTab, char deco,char cell_limit)
{
		int num_cell=0;
		int decoLineLength = iTab.length;
		String deco_line =  build_deco_line(deco, decoLineLength);
		System.out.println(deco_line);
		while (num_cell < iTab.length){
			System.out.print(cell_limit);
			System.out.print(iTab[num_cell]);
			num_cell++;
		}
		System.out.println(cell_limit);
		System.out.println(deco_line);
	}
	public static void main(String[] args) {
		int[] iTab1, iTab2;
		iTab1 = createAndFillIntTab();
		iTab2 = sum_2_consecutive_cell(iTab1);
		horizontal_display(iTab1, '-', '|');
		horizontal_display(iTab2, '-', '|');

	}
}
```

الفصل الحادي عشر

الأوعية: تمثيل واخراج جداول الواقع المعاش

رموز التحكم في الإجراج في ابسط صورها:

الشرمزية	تعليق	لغة جافا
%ط	المدخل المستهدف فيه قيمة من النمط طبيعي,	**%d**
%ح	المدخل المستهدف فيه قيمة من النمط حقيقي.	**%f**
%س	المدخل المستهدف فيه قيمة من النمط سلسلة حروف	**%s**
%م	المدخل المستهدف فيه قيمة من النمط منطقي	**%b**
%ف	المدخل المستهدف فيه قيمة من النمط حرف	**%c**

ترجمة رموز اخراج بعض الأمثلة

الأمثلة	الشرمزية	جافا
ما هو الرمز لكتابة القيمة الطبيعية 278 في مساحة قدرها 25 حرفا	**%25ط**	%25d
ما هو الرمز لكتابة القيمة الطبيعية 278 في مساحة قدرها 25 حرفا، ونريد ملء الفراغ الذي يسبق كتابة القيمة بالحرف 0	**%025ط**	%025d
ما هو الرمز لكتابة القيمة الطبيعية 278 في مساحة قدرها 25 حرفا، ونريد اظهار اشارة +	**%+25ط**	%+25d
ما هو الرمز لكتابة القيمة الطبيعية 278 في مساحة قدرها 20 حرفا، ونريد ملء الفراغ الذي يسبق كتابة القيمة بالحرف 0، و نريد اظهار اشارة +	**%+025ط**	%+025d
ما هو الرمز لكتابة القيمة الحقيقية 134,87654 في مساحة قدرها 20 حرفا، وبدقة ثلاثة اعداد (اي ثلاثة اعداد بعد الفاصلة)	**%20.3ح**	%20.3f
ما هو الرمز لكتابة القيمة الحقيقية 134,87654 في مساحة قدرها 20 حرفا، بدقة بأربعة اعداد، و نريد ان يكتب الرقم 0 بدل الفراغات	**%020.4ح**	%020.4f

ترجمة النص 1

```
package part_02_chapter_11;

public class Example_01_CodeSource_01 {

	public static void main(String[] args) {
		int a = 1234;
		double d =45.36;
		boolean b=true;
		char  c = ',';
		String s = " , ";
```

```
        System.out.printf("%d%f%b%n", a,d,b);   // %n for new line
        System.out.printf("%d%c%f%c%b%n", a,c,d,c,b); // %n for new line
        System.out.printf("%d%s%f%s%b%n", a,s,d,s,b); // %n for new line
        System.out.printf("a=%dd=%fb=%b%n", a,d,b);   // %n for new line
        System.out.printf("%d%n", a);   // %n for new line
        System.out.printf("%f%n",d);   // %n for new line
        System.out.printf("%b%n", b);   // %n for new line
      }
}
```

ترجمة النص 2

```
package part_02_chapter_11;
public class Example_01_CodeSource_01 {
	public static void main(String[] args) {
		int alif = 1234, ba = 256789;
		double d = 45.56789;
		String[] names = {"قط","عنكبوت","أسد"};

		System.out.printf("%10d%10d%n", alif,ba);   // %n for new line
		System.out.printf("| %10d| %10d%n", alif,ba);   // %n for new line
		System.out.printf("| %10f| %10f%n", d,d);   // %n for new line
		int counter = 0;
		while (counter < names.length){
			System.out.printf("| %7s", names[counter]);
			counter++;
		}
		System.out.println(" |");
	}
}
```

ترجمة النص 3

```
package part_02_chapter_11;

public class Example_03_CodeSource_03 {
  public static void main(String[] args) {
    int i = 12;
    double d =24.76;
    int s_i = 20, s_d = 40;//s_i: space for an int - s_d: space for a double
    System.out.printf("int is 20 char %20d - double is 40 char%40f%n", i,d);
    System.out.printf("int is 20 char "+ "%20d" + " - double is 40 char"+ "%40f%n",
i,d);
    System.out.printf("int is 20 char %"+ "20"+"d" + " - double is 40 char%"+
"40"+"f"+"%n", i,d);
    System.out.printf("int is 20 char %"+ 20+"d" + " - double is 40 char%"+ 40
+"f"+"%n", i,d);
    System.out.printf("int is 20 char %"+ s_i +"d" + " - double is 40 char%"+ s_d
+"f"+"%n", i,d);
  }
   System.out.printf("int is " +s_i+ " char %"+ s_i +"d" + " - double is "+s_d+"
char%"+ s_d +"f"+"%n", i,d);

}
```

ترجمة النص 4

```
package part_02_chapter_11;

public class Example_04_CodeSource_04 {

	public static void printTitles(String[] titlesTab){
		int titleSpace = (80/titlesTab.length) - 4;
		int counter = 0;
		while (counter < titlesTab.length -1 ){
			System.out.printf("|%"+ titleSpace + "s ", titlesTab[counter]);
			counter++;
```

```
                }
                System.out.printf("|%"+ titleSpace + "s |%n", titlesTab[counter]);
        }
        public static void printLine(char c , int length){
                int counter = 0;
                while (counter < length ){
                        System.out.print(c);
                        counter++;
                }
                System.out.println();
        }
        public static void main(String[] args) {
                String[] titles_01 = { "قسم رابع" , "قسم ثالث" ,"قسم ثاني" ,"قسم اول" };
                String[] titles_02 = { " ","06 ق" ,"05 ق" ,"04 ق" ,"03 ق" ,"02 ق" ,"01 ق
08 ق" ,"07 ق"" };
                printLine('-' , 80);
                printTitles(titles_01);
                printLine('-' , 80);
                printTitles(titles_02);
                printLine('-' , 80);
        }
}
```

ترجمة النص 5

```
package part_02_chapter_11;
public class Example_05_CodeSource_05 {
        public static void printTitles(String[] titlesTab, char border, int
lineLength){
                int titleSpace = (lineLength/titlesTab.length) - 4;
                int counter = 0;
                while (counter < titlesTab.length -1 ){
                        System.out.printf(border +"%"+ titleSpace + "s ",
titlesTab[counter]);
                        counter++;
                }
                System.out.printf(border +"%"+ titleSpace + "s |%n",
titlesTab[counter]);
        }
        public static void printLine(char c , int length){
                int counter = 0;
                while (counter < length ){
                        System.out.print(c);
                        counter++;
                }
                System.out.println();
        }
        public static void main(String[] args) {
                String[] titles_01 = { "قسم رابع" , "قسم ثالث" ,"قسم ثاني" ,"قسم اول" };
                String[] titles_02 = { " ","06 ق" ,"05 ق" ,"04 ق" ,"03 ق" ,"02 ق" ,"01 ق
08 ق" ,"07 ق"" };
                int lineLen = 120;
                printLine('-' , lineLen);
                printTitles(titles_01, '*',lineLen);
                printLine('-' , lineLen);
                printTitles(titles_01, '*',lineLen);
                printLine('-' , lineLen);
        }
}
```

ترجمة النصوص من 6 الى 13

```
package part_02_chapter_11;
import java.util.Scanner;
class InstitutionInfo{
```

```
	String full_name,  adress_01, adress_02;
}
public class Example_06_CodeSource_06 {
	public static Scanner in = new Scanner(System.in);
	/*
	 * Using String variable instead of using String value in the program
	 */
	// English version
	public static String total_stdudent_in_a_level_msg = "Total";
	public static String total_stdudent_in_institution_msg = "Total";
	public static String introducingInstitution ="Collecting global Institution
Inf ormation ";
	public static String institutionNameMsg = "Please enter complete institution
name";
	public static String adressPart01Msg = "PLease enter part 1 of instittion
adress";
	public static String adressPart02Msg = "PLease enter part 2 of instittion
adress";
	public static String welcome = "Welcome to Student Management System";
	public static String numberOfInstitutionLevelMsg = "Please, enter the number
of teaching level in your institution ";
	public static String numberOfClassInlevelMsg = "Please enter the number of
class in  level ";
	public static String numberOfStudentInClassMsg = "Please enter the number of
student in class ";
	public static String nextClassMsg = "Going to next class ...";
    public static String definingLevelsNamesMsg = "Starting defining names of levels
...";
	public static String levelWordMsg = "levels..";
	public static String enterLevelNameMsg =  "Please, Enter the name of level N°
";
    public static String maxNumberOfClassInLevelMsg = "Please the maximùuùmnumbrr of
class a level may have";
    public static String startGettingClassNameMsg = "Starting getting class names";

	// Arabic
	/*
	 public static String total = "المجموع";
	 public static String introducingInstitution = "تعريف المؤسسة التعليمية";
	 public static String institutionNameMsg = "فضلا، ادخل الاسم الكامل لمؤسستكم:  ";
	 public static String adressPart01Msg = " فضلا، ادخل الفقرة الأولى من العنوان (الشارع، الحي،
الأرقام الخ :";
	 public static String adressPart02Msg = "فضلا، ادخل الفقرة الثانية من العنوان (المدينة و الولاية
:";
     public static String welcome = "أهلا و سهلا بكم في برنامج تسيير التلاميذ في المؤسسة";
     public static String numberOfInstitutionLevelMsg = "فضلا ما هو عدد مستويات  مؤسستكم";
     public static String levelNameMsg = "فضلا ادخل عدد أقسام المستوى ";
     public static String numberOfStudentInClassMsg = "ضلا ادخل عدد التلاميذ في القسم";
     public static String nextClassMsg = "ننتقل الآن الى القسم التالي";
	   public static String definingLevelsNamesMsg = "عرف اسماء المستويات في مؤسستكم، هناك ";
	   public static String levelWordMsg = "levels..";
	   public static String levelNameMsg = "فضلا، ادخل ا سم المستوى رقم"  "
     public static String maxNumberOfClassInLevelMsg = " فضلا، افدنا بأكبرعدد من الإقسام يمكن ان
يحتويه مستوى ما في مؤسستكم ";
     public static String startGettingClassNameMsg = "سنبدأ الآن تعريف اسماء الأ قسم في مؤسستكم
";

	 */
	public static void printLine(char c , int length){
		int counter = 0;
		while (counter < length ){
			System.out.print(c);
			counter++;
		}
		System.out.println();
	}
    public static InstitutionInfo  takeInstitutionInfo(){
```

```
        InstitutionInfo iif = new InstitutionInfo();
        System.out.println(introducingInstitution);
        System.out.print(institutionNameMsg);
        iif.full_name = in.nextLine();
        System.out.print(adressPart01Msg);
        iif.adress_01 = in.nextLine();
        System.out.print(adressPart01Msg);
        iif.adress_02 = in.nextLine();
        return iif;
    }

    public static String[] takeLevelNames(int numberOfLevel){
        String[] levelNames =  new String[numberOfLevel];
        System.out.println(definingLevelsNamesMsg + numberOfLevel + levelWordMsg);
        int counter=0;
        in.nextLine();
        while (counter < numberOfLevel){
               System.out.print(enterLevelNameMsg + (counter + 1) + ":");
               levelNames[counter] = in.nextLine();
               counter++;
        }

        return levelNames;
    }
    public static String[] takeClassesNames(){

        String[] classesNames;
        int numberOfClass, counter=0;
        System.out.print(maxNumberOfClassInLevelMsg);
        numberOfClass = in.nextInt();
        classesNames =  new String[numberOfClass];
        System.out.print(startGettingClassNameMsg);
        in.nextLine();
        while (counter <numberOfClass){
               System.out.print("فضلا، ادخل ا سم القسم رقم   "+ (counter+1)+ ":" );
               classesNames[counter] = in.nextLine();
               counter ++;
        }
        return classesNames;
    }

    public static void printInstitutionInfo(InstitutionInfo iif, int lineWidth, char
spaceChar){
         printLine('-', lineWidth);
        // System.out.println(iif.full_name);

         int spaceBeforeTitle = (lineWidth - iif.full_name.length())/2;
         System.out.printf("%"+spaceBeforeTitle+"c%s%n",spaceChar, iif.full_name);

         spaceBeforeTitle = (lineWidth - iif.adress_01.length())/2;
         System.out.printf("%"+spaceBeforeTitle+"c%s%n",spaceChar, iif.adress_01);

         spaceBeforeTitle = (lineWidth - iif.adress_02.length())/2;
         System.out.printf("%"+spaceBeforeTitle+"c%s%n",spaceChar, iif.adress_02);
         printLine('-', lineWidth);

    }
       public static void printTitlesLine(String[] titlesTab, char border, char
spaceChar, int lineLength, int leadingSpace){

               int dataZone = ((lineLength - leadingSpace)/(titlesTab.length +1)) -4;
               System.out.printf("%"+ leadingSpace+"c", spaceChar);
               int counter = 0;
               while (counter < titlesTab.length){
                      System.out.printf(border+"%"+dataZone + "s    ",
titlesTab[counter]);
```

```
                counter++;
            }

            System.out.printf(border + "%"+ dataZone+ "s    "+border+ "%n" ,
total_stdudent_in_a_level_msg);
      }
      public static void printALevel(int maxNumberOfClasses,
int[]studentInALevelclasses, String levelName, int levelNameWidth,
                                       int numberOfCells, int cellSpace,
                                       char border, char spaceChar){
            int spaceForNumberOfStudentInClass = 2;
            int spaceForTotalNumberOfStudentInLevel = 3;
            int spaceBeforeNumberOfStudent  = (cellSpace-
spaceForNumberOfStudentInClass)/2;
            int spaceBeforeTotalInLevel  = (cellSpace-
spaceForTotalNumberOfStudentInLevel)/2;
            int spaceAfterNumberOfStudent = cellSpace -
spaceForNumberOfStudentInClass - spaceBeforeNumberOfStudent;
            int spaceAfterTotalInLevel= cellSpace -
spaceForTotalNumberOfStudentInLevel - spaceBeforeTotalInLevel;


            System.out.printf(border+"%"+(levelNameWidth -1) +"s", levelName);
            int total = 0;
            int counter = 0;
            while (counter < maxNumberOfClasses ){

                  if (counter < studentInALevelclasses.length){
                        System.out.printf(border+"%"+ spaceBeforeNumberOfStudent
+ "c%"+spaceForNumberOfStudentInClass+
                                    "d%"+ (spaceAfterNumberOfStudent-1) +
"c",spaceChar,studentInALevelclasses[counter],spaceChar);
                        total = total + studentInALevelclasses[counter];
                  }
                  else
                        System.out.printf(border+"%"+ spaceBeforeNumberOfStudent
+ "c%"+spaceForNumberOfStudentInClass+
                                    "s%"+ (spaceAfterNumberOfStudent-1) +
"c",spaceChar,"--",spaceChar);

                  counter++;
            }
            System.out.printf(border+"%"+ spaceBeforeTotalInLevel + "c
%"+spaceForTotalNumberOfStudentInLevel+
                        "d%"+spaceAfterTotalInLevel + "c%n",
spaceChar,total,border);

      }
      public static void printTotalLine(int total, int lineWidth, int cellSize, int
numberOfClasses, char border, char spaceChar){
            //System.out.println("CellSize "+(lineWidth - cellSize*2));
            int spaceForTotalStudent = 4;
            int spaceBeforeTotal =(cellSize - spaceForTotalStudent)/2;
            int spaceAfterTotal = cellSize - spaceForTotalStudent-
spaceBeforeTotal;


            System.out.printf("%"+ (lineWidth - cellSize*2 -1) +"c", '|'  );
            System.out.printf("%"+ cellSize+"s", total_stdudent_in_institution_msg
+ border);
            System.out.printf("%" + spaceBeforeTotal + "c%" + spaceForTotalStudent
+ ("d%" + spaceAfterTotal)+"c" +"%n",spaceChar,total,border);
      }
      public static void printLevLine(int lineWidth){

       printLine('-',lineWidth );
```

```
	}
	public static void printEndTotalLine(int lineWidth, int cellSize, char
spaceChar){
		int totalEnLineSize = lineWidth - cellSize*2 -3;
	  System.out.printf("%"+totalEnLineSize +"c",spaceChar );
	  printLine('-',cellSize*2+3 );
	}

    public static void printInstitutionClasses(InstitutionInfo iif, int[][]
studentInClassTab, int[] totalStudentInLevelsTab,
                                    String[] levelNamesTab, String[]
classesNames, int totalNumberOfStudent ){
	// Adjusted after execution
	  final int lineWidth = 80;
	  final char spaceChar = ' ';
	  final char border = '|';
	  final int leadingSpace = 15;
	  int maxNumberOfClasses = classesNames.length;
	  int cellSpace = (lineWidth-leadingSpace)/(maxNumberOfClasses+1);
	  // Step 1
	  printInstitutionInfo(iif, lineWidth, spaceChar);

	  // Step 2

	  System.out.printf( "%" +  leadingSpace+"c", spaceChar);
	  printLine('-', lineWidth - leadingSpace );
	  printTitlesLine(classesNames, border, spaceChar,  lineWidth, leadingSpace);
	  printLine('-', lineWidth );

	  // Step 3
	  int counter = 0;
	  while (counter <studentInClassTab.length){
		printALevel(maxNumberOfClasses, studentInClassTab[counter],
levelNamesTab[counter],leadingSpace, maxNumberOfClasses, cellSpace, border,
spaceChar);
		printLevLine(lineWidth);
		counter++;
	  }

	  // step 4
	  printTotalLine(totalNumberOfStudent, lineWidth,
cellSpace,maxNumberOfClasses+1,border, spaceChar);

	  //Step 5
	  printEndTotalLine(lineWidth, cellSpace, spaceChar);

	}


	public static void main(String[] args) {
		int[][] allLevelsClassesTab;
		int[] studentInLevelTab;
		String[] levelNames;
		String[] classNames;
		int totalNumberOfStudents=0;
		int  numberOfLevels,numberOfClassesInLevel, levelCounter,
classCounter, numberOfStudentInClass, numberOfStudentInLevel;
		System.out.println(welcome+": ");
		InstitutionInfo institution = takeInstitutionInfo();
		System.out.print(numberOfInstitutionLevelMsg+": ");
		numberOfLevels = in.nextInt();

		allLevelsClassesTab = new int[numberOfLevels][];
		levelNames = takeLevelNames(numberOfLevels);

		studentInLevelTab = new int[numberOfLevels];
```

```
                classNames = takeClassesNames();

                levelCounter = 0;
                while (levelCounter < numberOfLevels){
                        System.out.print(numberOfClassInlevelMsg +
levelNames[levelCounter] + " : ");
                        numberOfClassesInLevel = in.nextInt();
                        allLevelsClassesTab[levelCounter] = new
int[numberOfClassesInLevel];
                        classCounter = 0;
                        numberOfStudentInLevel = 0;
                        while (classCounter < numberOfClassesInLevel ){
                                System.out.print(numberOfStudentInClassMsg +
classNames[classCounter] + ":");
                                numberOfStudentInClass = in.nextInt();
                                allLevelsClassesTab[levelCounter][classCounter] =
numberOfStudentInClass;
                                numberOfStudentInLevel = numberOfStudentInLevel +
numberOfStudentInClass;
                                classCounter++;
                                if (classCounter < numberOfClassesInLevel)
System.out.println();
                        }
                        studentInLevelTab[levelCounter] = numberOfStudentInLevel;
                        totalNumberOfStudents = totalNumberOfStudents +
numberOfStudentInLevel;
                        System.out.println(nextClassMsg);
                        levelCounter++;
                }
                printInstitutionClasses(institution, allLevelsClassesTab,
studentInLevelTab,levelNames, classNames, totalNumberOfStudents );

        }
}
```

الفصل الثاني عشر

الأوعية: سلاسل الحروف

ترجمة النص 2

```
package part_02_chapter_12;
public class Example_01_CodeSource_02 {
	public static void examples(){
		String str1 = "سلسلة ما";
		String str2 = new String(str1);
		String str3 = "سلسلة ما";
		String str4 = "سلسلة ما";
		String str5 = str4 = str1= "سلسلة ما";
		str4 = "سلسلة ما";
		str5 = "سلسلة ما";
	}
	public static void main(String[] args) {
		examples();
	}
{
```

ترجمة النص 3

```
package part_02_chapter_12;
public class Example_02_CodeSource_03 {
   public static void examplesOfStringConcatenation(){
        String salam = "السلام عليكم";
        String rahma = "و رحمة الله ";
        String salamWaRahma = salam + rahma;
        String salamWaRahmaWaBaraka_01 = salamWaRahma + " وبركاته" ;
        String salamWaRahmaWaBaraka_02 = salam + rahma + " وبركاته" ;
        double unitPrice =  678953.65;
        int quantity = 78;
       String totalPrice = "السعر الإجمالي للكمية " + quantity +  " هو: " +
unitPrice*quantity;
       String fifty = "" + 50;
       String oneHundredFifty_1 = 100 + fifty;
       String oneHundredFifty_2 = 100 + 50 + "";
       String oneHundredFifty_3 = ""+ 50 + 100 ;
    }
    public static void main(String[] args) {
       examplesOfStringConcatenation();
  }
}
```

الفصل الثالث عشر

الأوعية: اهم وظائف النمط سلسلة

الجزء الثاني: الملحق الثاني: ترجمة خوارزميات الجزء الثاني الى لغة "جافا"

String java class	1 عرف نمط سلسلة {
// Constructors	2 // المشيدات
String(String s)	3 سلسلة (سلسلة س) ؛
غير متاح، استعمل المشيد السابق بعبارة فيها السلسلة الفارغة ملصقة بالقيمة الطبيعية `String x = new String(""+2);`	4 سلسلة (طبيعي ط) ؛
غير متاح، استعمل المشيد السابق بعبارة فيها السلسلة الفارغة ملصقة بالقيمة الحقيقية `String x = new String(""+2.7);`	5 سلسلة (حقيقي ح) ؛
غير متاح، استعمل المشيد السابق بعبارة فيها السلسلة الفارغة ملصقة بالقيمة المنطقية `String x = new String(""+ true);`	6 سلسلة (منطقي م) ؛
// methods	7 // الوظائف المنتسبة الأكثر استعمالا
int length()	8 طبيعي حجم() ؛
char charAt(int index)	9 حرف حرف_في_موقع(طبيعي موقع)
int indexOf(char c)	10 طبيعي موقع_حرف(حرف ح) ؛
int indexOf(char c, int fromIndex)	11 طبيعي موقع_حرف(حرف ح، طبيعي موقع_بداية_البحث) ؛
boolean equals(String s)	12 منطقي تساوي (سلسلة س)
int compareTo(String s)	13 طبيعي قارن(سلسلة س)
	14 {

ترجمة النص 1

```
package part_02_chapter_13;
public class Example_01_CodeSource_01 {
	public static void main(String[] args) {
		//////  ملاحظة هامة  //////////////
		/*
في لغة جافا، عندما يصادف المترجم سلاسل ثابتة مكتوبة بين حرفي الاقتباس ولها نفس الأحرف تمتما ، كما هو الشأن بالتصريحين التاليين

		  String basmala_1  = "بسم الله الرحمان الرحيم" ;
		 String basmala_2  = "بسم الله الرحمان الرحيم" ;

فمن اجل استغلال الفعال للذا كرة، فان المترجم جافا لا يخزن هذه الثابتات في اماكن مختلفة، بل في نفس المكان ، ففي السطرين السابقين يقوم المترجم جافا بتوقير الكان للسلسلة basmala_1 وعندما يجد ان السلسلة basmala_2 متساوية تماما مع السلسلة basmala_1 فإنه لن يحجز ذاكرة جديدة كما هو الحال مع اللغة الشرمزية، بل يعاود استغلال الذاكرة التي فيها السلسلة basmala_1 ، وبهذا يكون محتوى basmala_1 يساوي محتوى basmala_2 ، فالترجمة الصحيحة لا يمثلها السطرين السابقين، بل يجب إنشاء انشاء سلاسل حروف باستعمال المشيدلت

		*/
		////////////////////////////////////  الترجمة التالية غير دقيقة
		//String basmala_1  = "بسم الله الرحمان الرحيم" ;
		//String basmala_2  = "بسم الله الرحمان الرحيم" ;
		//System.out.println(basmala_1 == basmala_2);
		////////////////////نهاية الملاحظة ////////////////////
		////////////////////////////////////  الترجمة التالية دقيقة تتوافق مع اللغة الشرمزية
		String basmala_1  = new String( "بسم الله الرحمان الرحيم" );
		String basmala_2  = new String("بسم الله الرحمان الرحيم" );
		System.out.println(basmala_1 == basmala_2);
	}
}
```

ترجمة النص 2

```
package part_02_chapter_13;
public class Example_02_CodeSource_02 {
    public static void main(String[] args) {
       String basmala_1  = new String( "بسم الله الرحمان الرحيم" );
       String basmala_2  = new String("بسم الله الرحمان الرحيم" );
       System.out.println(basmala_1.equals(basmala_2));
```

```
        System.out.println(basmala_1.equals("بسم الله الرحمان الرحيم"));
        System.out.println("بسم الله الرحمان الرحيم".equals(basmala_2));
        System.out.println(new String("بسم الله الرحمان الرحيم").equals(new String(" بِسم الله الرحمان
الرحيم")));
    }
}
```

ترجمة النص 3

```
package part_02_chapter_13;
import java.util.Scanner;

public class Example_03_CodeSource_03 {
  public static Scanner in = new Scanner(System.in);
  public static void main(String[] args) {
        String sentence_1, sentence_2;
        System.out.print("فضلا ادخل جملة اولى  او كلمة نهاية ان اردت الخروج: ");
        sentence_1 = in.nextLine();
        while (!sentence_1.equals("end")){
              System.out.print( "فضلا ادخل جملة ثانية ");
              sentence_2 = in.nextLine();
              int result = sentence_1.compareTo(sentence_2);
              if (result == 0){
                    System.out.println("الجملتين متساويتين..");
              }
              else if (result == 1){
                System.out.println("الجملة:" + sentence_1 + " -- اكبر من الجملة --"+
sentence_2);
                    }
              else System.out.println("الجملة:" + sentence_1 + " اصغر من الجملة"+
sentence_2);

        System.out.println("-------------------------------------------------");
                    System.out.print("فضلا ادخل جملة اولى  او كلمة نهاية ان اردت الخروج: ");
                    sentence_1 = in.nextLine();
              }

        System.out.println("-------------------------------------------------");
              System.out.println("الى اللقاء...");

        System.out.println("-------------------------------------------------");
        }
}
```

ترجمة النص 4

```
public class Example_04_CodeSource_04 {
  public static Scanner in = new Scanner(System.in);
  public static void main(String[] args) {
        String assalam = "السلام عليكم";
        String rahma = "ورحمة الله";
        int salamLength = assalam.length();
        int rahmaLength = rahma.length();
        System.out.println("طول سلام =" + salamLength);
        System.out.println("طول سلام و الرحمة =" + rahmaLength+ salamLength);
        System.out.println("طول سلام و الرحمة الحقيقي =" + (rahmaLength+ salamLength));
        int completeSalamLength = "السلام عليكم و رحمة الله و بركاته".length();
        System.out.println("طول السلام كاملا =" + completeSalamLength);
        System.out.println("السلام عليكم" + "= طول سلام مرة ثانية".length());
        System.out.println(" ورحمة " + "السلام عليكم") + "= طول سلام و الرحمة مرة ثانية
("الله.length());
        System.out.println("طول السلسلة الفارغة هو :" + "".length());
```

```
        System.out.println(" فضلا ادخل جملة مفيدة:");
        String txt;
        txt = in.nextLine();
        System.out.println("الجملة التي ادخلت مكونة من " + txt.length() + " حرفا");

  }
}
```

ترجمة النص 5

```
package part_02_chapter_13;
import java.util.Scanner;
public class Example_05_CodeSource_05 {
  public static Scanner in = new Scanner(System.in);
  public static void main(String[] args) {
        String txt;
        System.out.println(" فضلا ادخل جملة مفيدة:");
        txt = in.nextLine();
        int txtLen =  txt.length();
        System.out.println("الجملة التي ادخلت مكونة من " + txtLen  + " حرفا.");
        System.out.println("اعطني موقعا حقيقيا في الجملة التي ادخلت اعطك حرفه.");
        System.out.println("انتبه... الموقع يجب ان يكون من 0 و لا يتجاوز " + (txtLen -1));
        System.out.println("انتبه...في حالة اعطائك لموقع غير حقيقي، ينته الخوارزم...");
        System.out.println("الموقع :");
        int charPosition;
        charPosition = in.nextInt();
        while (charPosition  >= 0   &&  charPosition < txtLen){
              System.out.println("الحرف الموجود في الموقع " + charPosition + "هو: " +
txt.charAt(charPosition));
              System.out.println("الموقع :");
              charPosition = in.nextInt();
        }
        System.out.println("الموقع" + charPosition + "غير حقيقي... الى اللقاء في المرة القادمة.");
  }
}
```

ترجمة النص 6

```
package part_02_chapter_13;
import java.util.Scanner;
public class Example_06_CodeSource_06 {
        public static Scanner in = new Scanner(System.in);
        public static void main(String[] args) {
              String str ;
              System.out.println(" فضلا ادخل جملة مفيدة لا تحتوي على اي حرف ممثل للرقم صفر:");
              str = in.nextLine();
              int len = str.length() ;
              System.out.println("الجملة التي ادخلت مكونة من " + len + " حرفا.");
              System.out.println("اعطني حرفا ما ابلغك بأول ظهور له في الجملة التي ادخلت.");
              System.out.println("ادخل الحرف 0 ان اردت توقيف تنفيذ الخوارزم");
              String s;
              char c;
              System.out.println("ادخل الحرف الذي تريد معرفة موقعه في الجملة : ") ;
              s = in.nextLine(); c = s.charAt(0);
              while (c != '0'){
                    int charPosition = str.indexOf(c);
                    if (charPosition == -1 ){
                          System.out.println("الحرف " + c + "غير موجود في الجملة التي ادخلت.
");
                    }
                    else {
                          System.out.println("موقع الحرف " + c + " هو: " +
charPosition);
```

```
                }
                System.out.println("ادخل الحرف الذي تريد معرفة موقعه في الجملة : ") ;
                s = in.nextLine(); c = s.charAt(0);
            }
            System.out.println(".الى اللقاء في المرة القادمة") ;

        }
}
```

ترجمة النص 7

```
package part_02_chapter_13;
public class Example_07_CodeSource_07 {
        public static void main(String[] args) {
                String str = ".لا تسقني كأس الحياة بذلة بل واسقني في العزة كأس الحنضل";
                System.out.println("موقع الحرف س انطلاقا من بداية الجملة  هو: "+str.indexOf('س'));
                System.out.println("موقع الحرف س انطلاقا من بداية الجملة  هو: "+ str.indexOf('س',
0));
                System.out.println("موقع الحرف س انطلاقا من بداية الجملة  هو: "+ str.indexOf('س',
5));
                System.out.println("موقع الحرف س انطلاقا من بداية الجملة  هو: "+ str.indexOf('س',
12));
                System.out.println("موقع الحرف س انطلاقا من بداية الجملة  هو: "+ str.indexOf('س',
31));
                System.out.println("موقع الحرف س انطلاقا من بداية الجملة  هو: "+ str.indexOf('س',
47));

        }
}
```

ترجمة النص 8

```
package part_02_chapter_13;
public class Example_08_CodeSource_08 {
        public static void main(String[] args) {
                int startingPosition = 0, currentPosition;
                String str = ".لا تسقني كأس الحياة بذلة بل واسقني في العزة كأس الحنضل";
                currentPosition = str.indexOf('س', startingPosition);
                System.out.println("موقع الحرف س انطلاقا من الموقع "+ startingPosition + " في الجملة
هو: "+ currentPosition) ;

                startingPosition= currentPosition+1;
                currentPosition = str.indexOf('س', startingPosition);
                System.out.println("موقع الحرف س انطلاقا من الموقع "+ startingPosition + " في الجملة
هو: "+ currentPosition) ;

                startingPosition= currentPosition+1;
                currentPosition = str.indexOf('س', startingPosition);
                System.out.println("موقع الحرف س انطلاقا من الموقع "+ startingPosition + " في الجملة
هو: "+ currentPosition) ;

                startingPosition= currentPosition+1;
                currentPosition = str.indexOf('س', startingPosition);
                System.out.println("موقع الحرف س انطلاقا من الموقع "+ startingPosition + " في الجملة
هو: "+ currentPosition) ;

                startingPosition= currentPosition+1;
                currentPosition = str.indexOf('س', startingPosition);
                System.out.println("موقع الحرف س انطلاقا من الموقع "+ startingPosition + " في الجملة
هو: "+ currentPosition) ;
```

```
        }
}
```

ترجمة النص 9

```
package part_02_chapter_13;
import java.util.Scanner;
public class Example_09_CodeSource_09 {
  public static Scanner in = new Scanner(System.in);
  public static char readChar(){
        String s = in.nextLine();
        return s.charAt(0);
  }
  public static int countCharInString(char c, String s){
        int startingPosition = 0, currentPosition;
        int counter = 0;
        currentPosition = s.indexOf(c, startingPosition);
        while (currentPosition != -1){
              counter++;
              startingPosition= currentPosition+1;
              currentPosition = s.indexOf(c, startingPosition);
        }
        return counter;
  }
  public static void main(String[] args) {
        int startingPosition = 0, currentPosition;
        String str ;
        System.out.println( "خوارزم يخبر عن عدد المرات التي يظهر فيها حرف ما في جملة ما") ;
        System.out.println( "اكتب جملة:") ;
        str = in.nextLine();
        System.out.println("ادخل الأن الحرف الذي تريد معرف عدد المرات الذي يظهر في الجملة التي ادخلت") ;
        System.out.println( "او ادخل 0 ان اردت الخروج ");
        char c = readChar();
        while (c != '0'){
              System.out.println("عدد المرات التي يظهر فيها الحرف "+ c + "هو: "+
                                 countCharInString(c, str));
              System.out.println("ادخل الأن الحرف الذي تريد معرف عدد المرات الذي يظهر في الجملة التي
ادخلت");
              System.out.println( "او ادخل 0 ان اردت الخروج ");
              c = readChar();
        }
        System.out.println("الى اللقاء ....") ;
   }
}
```

ترجمة النص 10

```
package part_02_chapter_13;
import java.util.Scanner;
public class Example_10_CodeSource_10 {
  public static Scanner in = new Scanner(System.in);
  public static void main(String[] args) {
        String str1, str2 ;
        System.out.println( "ادخل نصا صغيرا:")  ;
        str1 = in.nextLine();
        System.out.println( "ادخل جزءا من النص الذي ادخلت سالفا لنخبرك عن موقعه في النص:");
        str2 = in.nextLine();
        int indexOfStr2InStr1 = str1.indexOf(str2);
        if (indexOfStr2InStr1 != -1){
          System.out.println("السلسلة -"+ str2 +  "- موجودة في الموقع رقم " + indexOfStr2InStr1
);
        }
        else {
```

```
            System.out.println("السلسلة -"+ str2 +  "- الذي ادخلت النص في موجودة غير ") ;
        }
    }
}
```

الفصل الرابع عشر

الأوعية: العمليات المعتادة على الجداول

ترجمة النص 1

```
package part_02_chapter_14;
public class Example_01_CodeSource_01 {
	/// Magnify version 01
	public static void magnify_v1(int[] iTab, int magnifier){
		int i = 0;
		while (i <  iTab.length){
			iTab[i] = iTab[i]*magnifier;
			i++;
		}
	}

	/// Magnify version 02
	public static void magnify_v2(int[] iTab, int magnifier){
		for (int i = 0; i <  iTab.length; i++){
			iTab[i] = iTab[i]*magnifier;
		}
	}

	public static void horizontalDisplay(int[] iTab){
		String line = "--------------------------";
		char  borderChar = '|';
		System.out.println(line);
		for (int i = 0; i < iTab.length; i++){
			System.out.print(borderChar);
			System.out.print(iTab[i]);
		}
		System.out.println(borderChar);
		System.out.println(line);
	}
	public static void main(String[] args) {
	  int[]   t = {6, 12, 24, 48, 5, 10, 25, 25};
	  magnify_v2(t, 3);
	  horizontalDisplay(t);
	}
}
```

ترجمة النص 2

```
package part_02_chapter_14;
public class Example_02_CodeSource_02 {

	public static void printTab(int[] iTab){
		for (int val : iTab){
			System.out.println(("قيمة الخانة التي وصلتها الزياره هي:  " + val));
		}
	}
	// The main is not part of translated code source 02.
	// Introduced here to test prinTab procedure
	public static void main(String[] args) {
	  int[]   t = {6, 12, 24, 48, 5, 10, 25, 25};
	  printTab(t);
	}
}
```

ترجمة النص 3

```
package part_02_chapter_14;
import java.util.Scanner;
public class Example_03_CodeSource_03 {
	public static Scanner in = new Scanner(System.in);
	public static void printTab(int[] iTab){
		for (int val : iTab){
			System.out.println( val);
		}
	}
	public static void fillIntTabFromKeyBoard(int[] iTab){
		if (iTab == null){
			System.out.println(" خطأ خطير ... الصلة غير مرتبطة بجدول ما ");
			return;
		}
		if (iTab.length == 0){
			System.out.println(" خطأ ... الجدول لا يحتوي على اي خانة، حجمه 0 ");
			return;
		}
		System.out.println("بداية تعمير جدول حجمه هو:" + iTab.length);
		int i =0;
		while (i < iTab.length){
			System.out.println("ادخل قيمة طبيعية توضع في الخانة رقم " + i + " : ");
			iTab[i] = in.nextInt();
			i++;
		}
	}
	public static void main(String[] args) {
		int[] t = new int[20];
		fillIntTabFromKeyBoard(t);
		printTab(t);

	}

}
```

ترجمة النص 4

```
package part_02_chapter_14;
import java.util.Scanner;
public class Example_04_CodeSource_04 {
	public static Scanner in = new Scanner(System.in);
	public static void printTab(int[] iTab){
		for (int val : iTab){
			System.out.println( val);
		}
	}
	public static void fillIntTabFromKeyBoard(int[] iTab){
		if (iTab == null){
			System.out.println(" خطأ خطير ... الصلة غير مرتبطة بجدول ما ");
			return;
		}
		if (iTab.length == 0){
			System.out.println(" خطأ ... الجدول لا يحتوي على اي خانة، حجمه 0 ");
			return;
		}
		System.out.println("بداية تعمير جدول حجمه هو:" + iTab.length);
		for (int i =0; i < iTab.length; i++){
			System.out.println("ادخل قيمة طبيعية توضع في الخانة رقم " + i + " : ");
			iTab[i] = in.nextInt();
		}
	}
	public static void main(String[] args) {
		int[] t = new int[20];
		fillIntTabFromKeyBoard(t);
		printTab(t);
```

```
	}

}
```

ترجمة النص 5

```
package part_02_chapter_14;
import java.util.Scanner;
public class Example_05_CodeSource_05 {
	public static Scanner in = new Scanner(System.in);
	public static void printTab(int[] iTab){
		for (int val : iTab){
			System.out.println( val);
		}
	}
	public static int[] fillIntTabFromKeyBoard(int size){
		if (size <= 0 ){
			System.out.println("الحجم .. خطأ خطير "+ size + "لا. غير واقعي بالنسبة للجداول يمكن انشاء و تعمير جدول") ;
			return null;
		}
		int[] iTab = new int[size];
		System.out.println("بداية تعمير جدول حجمه هو:" + size);
		for (int i =0; i < iTab.length; i++){
			System.out.println("ادخل قيمة طبيعية توضع في الخانة رقم " + i + " : ");
			iTab[i] = in.nextInt();
		}
		return iTab;
	}
	public static void main(String[] args) {
		int[] t = fillIntTabFromKeyBoard(15);
		printTab(t);

	}
}
```

ترجمة النص 6

```
package part_02_chapter_14;
import java.util.Scanner;
public class Example_06_CodeSource_06 {
	public static Scanner in = new Scanner(System.in);
	public static void printTab(int[] iTab){
		String line = "-------------------------";
		char  borderChar = '|';
		System.out.println(line);
		for (int i = 0; i < iTab.length; i++){
			System.out.print(borderChar);
			System.out.print(iTab[i]);
		}
		System.out.println(borderChar);
		System.out.println(line);
	}
	public static void copyTab(int[] source, int[] destination){
		if (source == null ){
			System.out.println(" خروج ...الجدول منبع المعطيات غير موجود...  ");
			return;
		}
		if (destination == null ){
			System.out.println(" خروج ...االجدول هدف التعمير غير موجود... ...  ");
			return;
		}
		int size = source.length;
		if (destination.length  < source.length)
			size = destination.length;
```

```
                System.out.println("بداية تعمير الجدول المستهدف انطلاقا من جدول منبع:");
                for (int i = 0; i < size; i++) destination[i] = source[i] ;


        }
        public static void main(String[] args) {
                int[] tSource = {1, 3, 6, 9, 12,15 };
                int[] tDestination = new int[8];
                copyTab(tSource,tDestination);
                printTab(tDestination);
        }
}
```

ترجمة النص 7

```
package part_02_chapter_14;
import java.util.Scanner;
public class Example_07_CodeSource_07 {
        public static Scanner in = new Scanner(System.in);
        public static void printTab(int[] iTab){
                String line = "--------------------------";
                char  borderChar = '|';
                System.out.println(line);
                for (int i = 0; i < iTab.length; i++){
                        System.out.print(borderChar);
                        System.out.print(iTab[i]);
                }
                System.out.println(borderChar);
                System.out.println(line);
        }
        public static int[] copyTab(int[] source, int[] destination){
                if (source == null ){
                        System.out.println(" خروج ...الجدول منبع المعطيات غير موجود...  ");
                        return  null;
                }
                if (destination == null )  destination = new int[source.length];

                if (destination.length <source.length) destination=new
int[source.length];

                System.out.println("بداية تعمير الجدول المستهدف انطلاقا من جدول منبع:");
                for (int i = 0; i < source.length; i++) destination[i] = source[i] ;
                return destination;
        }
        public static void main(String[] args) {
                int[] tSource = {1, 3, 6, 9, 12,15 };
                int[] tDestination =  copyTab(tSource,null);
                printTab(tDestination);
        }
}
```

ترجمة النص 8

```
package part_02_chapter_14;
public class Example_08_CodeSource_08 {
        public static void printTab(int[] iTab){
                String line = "--------------------------";
```

```
		char  borderChar = '|';
		System.out.println(line);
		for (int i = 0; i < iTab.length; i++){
			System.out.print(borderChar);
			System.out.print(iTab[i]);
		}
		System.out.println(borderChar);
		System.out.println(line);
	}
	public static int[] copyTab(int[] source){
		if (source == null ){
			System.out.println(" خروج ...الجدول منبع المعطيات غير موجود...  ");
			return  null;
		}
		int[] destination = new int[source.length];

		System.out.println("بداية تعمير الجدول المستهدف انطلاقا من جدول منبع:");
		for (int i = 0; i < source.length; i++) destination[i] = source[i] ;
		return destination;
	}
	public static void main(String[] args) {
		int[] tSource = {1, 3, 6, 9, 12,15 };
		int[] tDestination =  copyTab(tSource);
		printTab(tDestination);
	}
}
```

ترجمة النص 9

```
package part_02_chapter_14;
public class Example_09_CodeSource_09 {
	public static void printTab(int[] iTab){
		String line = "---------------------------";
		char  borderChar = '|';
		System.out.println(line);
		for (int i = 0; i < iTab.length; i++){
			System.out.print(borderChar);
			System.out.print(iTab[i]);
		}
		System.out.println(borderChar);
		System.out.println(line);
	}
	public static void magnifyTab(int[] tab, int magnifier){
		if (tab == null ){
			System.out.println(" خروج ...الجدول  غير موجود...  ");
			return  ;
		}
		for (int i = 0; i < tab.length; i++) tab[i] = tab[i]*magnifier ;

	}
	public static void main(String[] args) {
		int[] t = {1, 3, 6, 9, 12,15 };
		magnifyTab(t, 5);
		printTab(t);
	}
}
```

ترجمة النص 10

```
package part_02_chapter_14;
public class Example_10_CodeSource_10 {
	public static void printTab(double[] dTab){
		String line = "-------------------------";
		char  borderChar = '|';
		System.out.println(line);
		for (int i = 0; i < dTab.length; i++){
			System.out.print(borderChar);
			System.out.print(dTab[i]);
		}
		System.out.println(borderChar);
		System.out.println(line);
	}
//////////////   Average of values in  a one dimensional tab
	public static double average1DTab(double[] tab){
		if (tab == null ) return -1.0  ;
		double av = 0.0;
		for (int i = 0; i < tab.length; i++) av = av + tab[i] ;
		return av/tab.length;

	}
//////////////   Average of values in  a two dimensional tab
	public static double[] average2DTab(double[][] gradesTab){
		if (gradesTab == null ) return null  ;
		double[] avTab = new double[gradesTab.length];
		System.out.println("بداية تعمير الجدول المستهدف انطلاقا من جدول منبع:");
		for (int i = 0; i < gradesTab.length; i++) avTab[i] =
average1DTab(gradesTab[i]) ;
		double av = 0.0;

		return avTab;
	}
	public static void main(String[] args) {
		double[][] grades = {
					{12.0, 12, 12.0, 6, 18 },
					{10, 5, 15 },
					{16, 8, 4,12 }
				};

		double[] av = average2DTab(grades);
		printTab(av);
	}
}
```

الفصل الخامس عشر

الأوعية: العمليات المعتادة على الجداول: عمليات البحث

ترجمة النص 1

```
package part_02_chapter_15;

public class Example_01_CodeSource_01 {

	public static int search(int value,  int[] tab){
		if (tab == null )return -1; // القيمة غير موجودة في جدول غير موجود
		for (int i = 0; i < tab.length; i++)
			if (tab[i] == value) return i;
		return -1;
	}

	public static void main(String[] args) {
		int[] t = {12, 23, 56, 78, 65, 54, 56, 78, 81, 3,  9, 21 } ;
		int position = search(81,t);
		System.out.println ("رسالة :1 موقع القيمة 81 في الجدول هو: " + position);

		position = search(56,t);
		System.out.println ("رسالة :2 موقع القيمة 56 في الجدول هو: " + position);


		int a = 54;
		position = search(a,t);
		System.out.println ("رسالة :3 موقع القيمة  : "+ a +"في الجدول هو: " + position);

		a++;
		position = search(a,t);
		System.out.println ("رسالة :4 موقع القيمة  : "+ a +"  في الجدول هو: " +
position);

	}
}
```

ترجمة النص 2

```
package part_02_chapter_15;
public class Example_02_CodeSource_02 {
  public static int occurence(int value,  int[] tab){
	if (tab == null )return -1; // القيمة غير موجودة في جدول غير موجود
	int counter = 0;
	for (int i = 0; i < tab.length; i++)
		if (tab[i] == value) counter ++;
	return counter;
  }

  public static void main(String[] args) {
	int[] t = {12, 23, 56, 78, 65, 54, 56, 78, 81, 3,  9, 21 } ;
	int position = occurence(81,t);
	System.out.println ("رسالة :1 عدد المرات التي تظهر فيها القيمة 81 في الجدول هو: " +
position);

	position = occurence(56,t);
	System.out.println ("رسالة :2 عدد المرات التي تظهر فيها القيمة 56 في الجدول هو: " +
position);


	int a = 54;
	position = occurence(a,t);
	System.out.println("رسالة :3 عدد المرات التي تظهر فيها القيمة  : "+a+"في الجدول هو:
"+position);
```

```
        a++;
        position = occurence(a,t);
        System.out.println("القيمة  عدد المرات التي تظهر فيها :4 رسالة : "+a+"في الجدول هو:
"+position);

    }
}
```

ترجمة النص 3

```
package part_02_chapter_15;

public class Example_03_CodeSource_03 {

        public static int min(int[] tab){
                if (tab == null )return -1; // القيمة غير موجودة في جدول غير موجود
                if (tab.length == 0 )return -1;
                int imin = 0;
                for (int i = 0; i < tab.length; i++)
                        if (tab[i] < tab[imin]) imin = i;
                return imin;
        }

        public static void main(String[] args) {
                int[] t = {12, 23, 56, 78, 65, 54, 56, 78, 81, 3,  9, 21 } ;
                int position = min(t);
                System.out.println ("موقع اصغر قيمة في الجدول هو : :1 رسالة: " + position);
                System.out.println ("اصغر قيمة في الجدول هي :2 رسالة: " + t[position]);
        }
}
```

ترجمة النص 4

```
package part_02_chapter_15;

public class Complex {
  double r, i;
  public Complex(double r, double i){
        this.r = r;
        this.i = i;
  }
  public boolean equals(Object o){
        Complex c = (Complex)o;
        if (this.r == c.r  && this.i == c.i) return true;
        else return false;
  }
}
```

ترجمة النص 5

```
package part_02_chapter_15;
public class Example_04_CodeSource_05 {
  public static int occurence(Complex c, Complex[] tabComplex){
        if (tabComplex == null )return -1; // القيمة غير موجودة في جدول غير موجود
        if (c == null )return -2;

        int counter = 0;
        for (int i = 0; i < tabComplex.length; i++)
                if (tabComplex[i].equals(c) ) counter ++;
        return counter;
  }
```

```
  public static void main(String[] args) {
        Complex[] tc = {new Complex(12.6, 23.7),  new Complex(16.6, 83.7),
                        new Complex(8.5, 16.0) ,  new Complex(96.5, 76.87),
                        new Complex(8.5, 16.0) ,  new Complex(6.0, 12.5)
                       };


        int n = occurence(new Complex(8.5, 16.0),tc);
        System.out.println ("عدد المرات التي تظهر فيها  القيمة المركبة (8.5، 16.0) في الجدول هو: " + n);
        Complex c1 = new Complex(96.5, 76.87);
        n = occurence(c1,tc);
        System.out.println ("عدد المرات التي تظهر فيها  القيمة المركبة (96.5، 76.87) في الجدول هو: " +
n);
  }
}
```

ترجمة النص 6

```
package part_02_chapter_15;

public class Complex {
  double r, i;
  public Complex(double r, double i){
        this.r = r;
        this.i = i;
  }
  public boolean equals(Object o){
        Complex c = (Complex)o;
        if (this.r == c.r  && this.i == c.i) return true;
        else return false;
  }
  public String toString(){
        return "("+r+","+i+")";
  }
}
```

ترجمة النص 7، 8، 9

```
package part_02_chapter_15;

import java.awt.image.ColorConvertOp;
import java.util.Scanner;

public class Example_05_CodeSource_07 {
      public static Scanner in = new Scanner(System.in);

      public static Complex readComplex(){
            Complex c = new Complex();
            System.out.println ("أخل فيما يلي اجزاء عددا مركبا");
            System.out.print ("الجزء الحقيقي :");
            c.r = in.nextDouble();
            System.out.print ("الجزء الخيالي  :");
            c.i = in.nextDouble();
            return c;
      }
      public static int occurence(Complex c, Complex[] tabComplex){
            if (tabComplex == null )return -1; // القيمة غير موجودة في جدول غير موجود
            if (c == null )return -2;

            int counter = 0;
            for (int i = 0; i < tabComplex.length; i++)
                  if (tabComplex[i].equals(c) ) counter ++;
            return counter;
      }

      public static String convertToString(Complex c){
             return "("+c.r+","+c.i+")";
```

```
        }
        public static void main(String[] args) {
          Complex[] tc = {new Complex(12.6, 23.7),  new Complex(16.6, 83.7),
                                new Complex(8.5, 16.0) ,  new Complex(96.5, 76.87),
                                new Complex(8.5, 16.0) ,  new Complex(6.0, 12.5)
                              };

                int n = occurence(new Complex(8.5, 16.0),tc);
                System.out.println ("عدد المرات التي تظهر فيها  القيمة المركبة (8.5، 16.0) في الجدول هو:
" + n);
                Complex c1 = readComplex();
                n = occurence(c1,tc);
                System.out.println ("رسالة 2: عدد المرات التي تظهر فيها القيمة المركبة "+
convertToString(c1) + " في الجدول هو: " + n);
                System.out.println ("رسالة 2: عدد المرات التي تظهر فيها القيمة المركبة "+ c1.toString()
+ " في الجدول هو: " + n);
                System.out.println ("رسالة 2: عدد المرات التي تظهر فيها القيمة المركبة "+ c1 + " في الجدول
هو: " + n);
        }
}
```

ترجمة النص 10

```
package part_02_chapter_15;

public class Person {
        String lastName, firstName, fatherFirstName;
        String motherLastName, motherFirstName;
        char  sex;
        AccurateDate birthDay;
        String birthPlace;
        double height, weight;
        int familySituation; //0:single 1: Married 2: divorced

        public Person(){ }
        public Person(String lastName, String firstName, String fatherFirstName,
                              String motherLastName, String motherFirstName, char sex,
                              AccurateDate  birthDay, String birthPlace,
                              double height, double weight){

                this.lastName= lastName;
                this.firstName= firstName;
                this.fatherFirstName = fatherFirstName;
                this.motherLastName = motherLastName;
                this.motherFirstName = motherFirstName;
                this.sex= sex;
                this.birthDay = birthDay;
                this.birthPlace= birthPlace;
                this.height = height;
                this.weight = weight;
        }
        public boolean equals (Object obj){
                Person p  = (Person)obj;
                if (this.sex == p.sex   &&
                        this.lastName.equals(p.lastName) &&
                        this.firstName.equals(p.firstName) &&
                        this.fatherFirstName.equals(p.fatherFirstName) &&
                        this.motherLastName.equals(p.motherLastName)&&
                        this.motherFirstName.equals(p.motherFirstName)&&
                        this.birthDay.equals(p.birthDay) &&
                    this.birthPlace.equals(p.birthPlace)
                        )
                        return true;
                else // this line may be deleted
                        return false;
```

```
	}
	public String toString(){
		return this.firstName + "   " + this.lastName;
	}
}
```

ترجمة النص 11

```
package part_02_chapter_15;
public class AccurateDate {
	int day, month, year;
	int hour, minute, second;// second is automatically set to 0
	// Constructors
	public AccurateDate(){}
	public AccurateDate(int day, int month, int year){
		this.day = day;
		this.month = month;
		this.year = year;
		// hour, minute, second are automatically set to 0
	}
	public AccurateDate(int day, int month, int year, int hour, int minute, int
second){
		this(day,month, year); // calling previous constructor
		this.hour =hour;
		this.minute = minute;
		this.second = second;
	}
	public boolean equals(Object obj){
		AccurateDate  acd = (AccurateDate)obj;
		if (this.year == acd.year && this.month == acd.month && this.day ==
acd.day &&
			this.hour == acd.hour && this.minute == acd.minute &&
this.second == acd.second )
			return true;
		else   // this else may be deleted
			return false;
	}
}
```

ترجمة النص 12، 13، 14

```
package part_02_chapter_15;
import java.util.Scanner;
public class Example_07_CodeSource_10_11_12 {
	public static Scanner in = new Scanner(System.in);
	public static  char readChar(){
		String l = in.nextLine();
		return l.charAt(0);
	}
	public static AccurateDate readAccurateDate(){
		AccurateDate  acd = new AccurateDate();
		System.out.print("ادخل السنة:")          ;
		acd.year = in.nextInt();

		System.out.print("ادخل الشهر:")       ;
		acd.month = in.nextInt();

		System.out.print("ادخل اليوم:")         ;
		acd.day = in.nextInt();

		System.out.print("ادخل الساعة:")         ;
		acd.hour = in.nextInt();

		System.out.print("ادخل الدقيقة:")        ;
		acd.minute = in.nextInt();

		System.out.print("ادخل الثانية:")       ;
```

```
            acd.second = in.nextInt();
            in.nextLine();
            return acd;
      }
      public static Person readPerson(){
            Person p = new Person();
            System.out.print("ادخل الجنس:")  ;
            p.sex = readChar();

            System.out.print("ادخل الإسم:")     ;
            p.firstName = in.nextLine();

            System.out.print("ادخل اللقب:")      ;
            p.lastName = in.nextLine();

            System.out.print("ادخل إسم الأب:")      ;
            p.fatherFirstName = in.nextLine();


            System.out.print("ادخل اسم الأم:")      ;
            p.motherFirstName = in.nextLine();

            System.out.print("ادخل لقب الأم:")      ;
            p.motherLastName = in.nextLine();

            System.out.println("ادخل تاريخ الولادة:")        ;
            p.birthDay = readAccurateDate();

            System.out.print("ادخل مكان الولادة:")        ;
            p.motherLastName = in.nextLine();

            in.nextLine();
            return p;
      }
      public static int occurence(Person p, Person[] tabPerson){
            if (tabPerson == null )return -1; // القيمة غير موجودة في جدول غير موجود
            if (p == null ) return -2;

            int counter = 0;
            for (int i = 0; i < tabPerson.length; i++)
                  if (tabPerson[i].equals(p) ) counter ++;
            return counter;
      }
      public static void main(String[] args) {
          Person[]  tabPerson = {
            new Person('ذ',"بن حسن" ,"سلمى" ,"موسى" ,"سعيد","عمر', new AccurateDate(8 ,
9, 1980, 15,  30, 0), "1.7,40,"الجزائر),
            new Person('ذ',"يحياوي" ,"زهرة" ,"صالح" ,"محمدي","عثمان', new AccurateDate(11,
9 , 1998, 10,  45, 0), "1.9,70,"وهران),
             new Person('أ',"عربي" ,"ريم" ,"احمد","مستعانمي","سميرة', new AccurateDate(26,
6 , 2006, 02,  15, 0), "1.2,37,"بويرة),
             new Person('أ',"جوامع" ,"حليمة" ,"عمر","تواتي","صباح', new AccurateDate(12,
3 , 2009, 07,  45, 0), "1.9,90,"تلمسان),
             new Person('ذ',"بن حسن" ,"سلمى" ,"موسى" ,"سعيد","عمر', new
AccurateDate(8,9, 1980, 15,  30, 0), "1.85,84,"الجزائر),
          };
        // Next line for rapid testing
        //Person p = new Person('ذ',"بن حسن" ,"سلمى" ,"موسى" ,"سعيد","عمر',new
AccurateDate(8 ,  9, 1980, 15,  30, 0), "1.7,40,"الجزائر);
        Person p = readPerson();
        int n = occurence(p, tabPerson);

        System.out.println("في الجدول  "+p.toString()+" رسالة 2: عدد المرات التي يظهر فيها الشخص
هو: " + n);
```

```
        // الكتابة التالية تساوي الكتابة التالية
        System.out.println (  "في الجدول هو " + p + " رسالة 2: عدد المرات التي يظهر فيها الشخص" + n);

    }
}
```

الفصل السادس عشر

الأوعية: العمليات المعتادة على الجداول: ترتيب المعلومات

ترجمة النص 1

```
package part_02_chapter_16;

public class Example_01_CodeSource_01 {
	public static int min(int[] tab, int start, int end){
		if (tab == null )return -1; // القيمة غير موجودة في جدول غير موجود
		if (tab.length == 0 )return -2;
		int imin = start;
		for (int i = start; i <= end; i++)
			if (tab[i] < tab[imin]) imin = i;
		return imin;
	}

	public static void swap(int[] tab, int pos1, int pos2){
		int tmp = tab[pos1];
		tab[pos1] = tab[pos2];
		tab[pos2] = tmp;
	}

	public static int sort(int[] tab){
		if (tab == null )return -1; // القيمة غير موجودة في جدول غير موجود
		if (tab.length == 0 )return -2;  // this line not present in algorithm
code
		int iStart = 0, iLast = tab.length -1;
		int iMin = iStart;
		int width  = iLast - iStart ;
		while (width > 0){
			iMin = min(tab, iStart, iLast);
			if (iMin != iStart){
				swap(tab, iStart, iMin);
			}
			iStart++;
			iMin = iStart;
			width  = iLast - iStart;
		}
		return 0; // No problem.. function successfully finished
	}

	public static void main(String[] args) {
		int[] t = {222, 23, 56, 78, 65, 54, 56, 718, 81, 3,  9, 21 }  ;
		sort(t);
		System.out.println ("الجدول بعد الترتيب:");

		for (int i = 0; i < t.length; i++)
			System.out.print(t[i]+ "    ");
		System.out.println();
		System.out.println("الى اللقاء...");
	}
}
```

ترجمة النص 2

```
package part_02_chapter_16;

public class Example_02_CodeSource_02 {
	public static int min(int[] tab, int start, int end){
		if (tab == null )return -1; // القيمة غير موجودة في جدول غير موجود
		if (tab.length == 0 )return -2;
		int imin = start;
		for (int i = start; i <= end; i++)
			if (tab[i] < tab[imin]) imin = i;
		return imin;
	}
```

```
	public static void swap(int[] tab, int pos1, int pos2){
		int tmp = tab[pos1];
		tab[pos1] = tab[pos2];
		tab[pos2] = tmp;
	}

	public static int bubbleSort(int[] tab){
		if (tab == null )return -1; // القيمة غير موجودة في جدول غير موجود
		if (tab.length == 0 )return -2;  // this line not present in algorithm
code
		for (int bubblePosition = tab.length -1;bubblePosition > 1;
bubblePosition-- )
			for (int i = 0; i < bubblePosition; i++)
				if (tab[i] > tab[i+1])
					swap(tab, i, i+1);

		return 0; // No problem.. function successfully finished
	}

	public static void main(String[] args) {
		int[] t = {222, 23, 56, 78,1, 65,18, 54, 56, 718, 81, 2, 18, 3,  9,
21, 19 }  ;
		bubbleSort(t);
		System.out.println ("الجدول بعد الترتيب :");

		for (int i = 0; i < t.length; i++)
			System.out.print(t[i]+ "    ");
		System.out.println();
		System.out.println("الى اللقاء...");
	}
}
```

ترجمة النص 3

```
package part_02_chapter_16;
public class Example_03_CodeSource_03 {
	public static int factorial( int n){
		if (n == 0 )return 1;
		return n*factorial(n-1);
	}

	public static void main(String[] args) {
		System.out.println ("هو 6 تعاملي: " + factorial(6));
	}
}
```

ترجمة النص 4

```
package part_02_chapter_16;
public class Example_04_CodeSource_04 {
	public static int factorial( int n){
		int res = 1;
		for (int f = n; f > 1; f--) res = res*f;

		return res;
	}

	public static void main(String[] args) {
		System.out.println ("هو 6 تعاملي: " + factorial(6));
	}
}
```

ترجمة النص 5

```
package part_02_chapter_16;
public class Example_05_CodeSource_05 {
	public static int sumFirstIntegerNumber( int n){
		if (n == 1 )return 1;
		return n + sumFirstIntegerNumber(n-1);
	}

	public static void main(String[] args) {
		System.out.println ("نتيجة جمع 10 اعداد اولى هي: " +
sumFirstIntegerNumber(10));
	}
}
```

ترجمة النص 6

```
package part_02_chapter_16;

public class Example_06_CodeSource_06 {
	public static void swap(int[] tab, int pos1, int pos2){
		int tmp = tab[pos1];
		tab[pos1] = tab[pos2];
		tab[pos2] = tmp;
	}
	public static int divideSortSpace(int[] tab, int start, int end){
		if (tab == null )return -1; // القيمة غير موجودة في جدول غير موجود
		if (tab.length == 0 )return -2;
		int s = start;
		for (int i = start; i < end; i++){
			if (tab[i] < tab[end]){
				swap (tab, i, s);
				s++;
			}
		}
		swap(tab, s, end);
		return s;
	}

	public static void rapidSort(int[] tab, int start, int end){
		if (end > start){
			int pivot = divideSortSpace(tab, start, end);
			rapidSort(tab, start, pivot -1);
			rapidSort(tab, pivot +1, end);
		}

	}

	public static void main(String[] args) {
		int[] t = {222, 23, 56, 78,1, 65,18, 54, 56, 718, 81, 2, 18, 3,  9,
21, 19,0,0,1,1,2,2,1222 }  ;
		rapidSort(t,0,t.length -1);
		System.out.println ("الجدول بعد الترتيب :");

		for (int i = 0; i < t.length; i++)
			System.out.print(t[i]+ "   ");
		System.out.println();
		System.out.println("الى اللقاء...");
	}
}
```