الصفوف والكائنات
Classes and objects

مقدمة:

تستخدم الصفوف في لغة الجافا كأنواع يمكن أن نشتق منها كائنات، وبما أن برامج لغة الجافا هي مجموعة من الكائنات التي تتعاون على حل مسألة معينة فإن مفهوم الصف يشكل الركن الأول في البرمجة كائنية (غرضية) التوجه، كونه يسمح بتعريف مواصفات هذه الكائنات ويسمح باشتقاق الكائنات المطلوبة لحل المسألة.

الصف: هو عبارة عن أداة برمجية تمكننا من تعريف أنواع جديدة في اللغة غير معرفة مسبقاً، وكذلك تسمح بتعريف العمليات التي نرغب بتعريفها على هذه الأنواع.

- أي إن الصف هو نوع يعرف الصفات المشتركة لمجموعة من الكائنات والعمليات عليها.

ملاحظة: لقد كان لدينا في لغة C++ مفهوم السجل الذي يحوي على متحولات من أنواع مختلفة، ولكن لا يمكن تعريف عمليات على تلك السجلات.

قاعدة هامة:

* يكتب الصف في ملف مستقل يحمل اسم الصف نفسه ما عدا حالة خاصة وهي: عندما يكون هناك مجموعة من الصفوف المرتبطة فيما بينها بعلاقة وراثة فإننا نكتب جميعاً في الملف ذاته على أن نختار اسم أحد هذه الصفوف ليكون اسماً للملف.

* يجب ألا يحوي الملف الواحد (في حال تعدد الصفوف فيه) إلا على صف عام (public) واحد.

وسندرس مفهوم الوراثة لاحقاً.

ملاحظة: يمكن أن يحوي الصف على صفوف داخلية Inner Classes ولكن سنؤجل الحديث عنها.

Scanned by CamScanner

* إن كتابة صف جديد تعني تعريف نوع جديد، ولايمكن استعمال هذا النوع الجديد مباشرة في البرنامج، بل نستعمل ممثلين عن هذه الأنواع، أي يوجب التصريح عن متحولات من هذا النوع، ويكون هذه المتحولات مركبة.
- كل متحول من نوع صف يدعى كائن (غرض object)

* يتم التصريح عن الكائن حسب القاعدة الأساسية للتصريح عن متحولات في لغة الجافا أي:

اسم المتحول ; فراغ النوع

⟹ اسم الكائن ; فراغ اسم الصف

* وكما نعلم، فالتصريح عن المتحولات المركبة لايعني إنشاءها، وبالتالي حتى ننشئ متحول من نوع صف (حجز مكان في الذاكرة وإعطاء قيم بدائية) يجب تنفيذ التعليمة:

classname فراغ objectname = new classname();

الشكل العام للصف في لغة الجافا:

- إن اسم الصف يبدأ بحرف كبير عُرفاً لا قاعدة.

```
class فراغ classname{
  Data members definition
  ===
  ===
  ===

  Constructors definition
  ===
  ===
  ===

  Methods members definition
  ===
  ===
  ===
}
```


Scanned by CamScanner

/ /

مكونات الصف في لغة الجافا:

يحتوي الصف على مجموعة من الأعضاء:

1 المعطيات الأعضاء (الحقول): Data Members (Fields)

* يمكن للصف أن يصرّح عن مجموعة (قد تكون خالية) من الحقول التي تُستخدم لتخزين بيانات من أنواع معقدة بسيطة أو معرّفة مسبقاً.
* تُستخدم هذه البيانات من قبل الصف نفسه (دواله وبياناته تحديداً) للقيام بالاختصاص المحدد لهذا الصف.
* يمكن أن تُعطى هذه الحقول قيماً بدائية ويمكن أن تُترك بدون قيم بدائية. في الحالة الأخيرة يقوم المترجم بوضع قيم افتراضية للحقول حسب أنواعها.
* إن هذه المعطيات هي التي تميّز غرض object ما عن غيره.
* يتم استدعاء المعطيات الأعضاء لكائن ما في البرنامج بالعبارة:

```
objetname . data name
```

2 الدوال (الطرق) الأعضاء: Methods Members

يعرّف كل صف مجموعة من الطرق التي تقوم بتنفيذ مهام محددة على البيانات، وأي عمل يقوم به الصف هو عبارة عن دالة فيه.

* البنية العامة للطريقة في لغة الجافا:

```
Access-modifier  return-type  methodname (List-of-arguments)
  {   Local-variables
      Statements
  }
```


Scanned by CamScanner

/ /

* محدّد الوصول: Acces_modifier
  يحدّد سماحية استدعاء الدالة ، ويكون أحد الكلمات التالية:
  public , private , protected
  وسوف نقوم بشرح محددات الوصول بالتفصيل فيما بعد.

* نوع الإرجاع: return_type
  يحدّد نوع القيمة التي تعيدها الطريقة بعد الانتهاء من التنفيذ ، ويتم ذلك من خلال التعليمة return (...);
  يمكن لنوع الإرجاع أن يكون أياً من الأنواع المعرفة مسبقاً أو حتى من نوع الصف نفسه.
  ويمكن للطريقة ألا تعيد قيمة ، وفي هذه الحالة يكون نوع الإرجاع void.

* اسم الطريقة: method_name
  هو الاسم المعطى للطريقة ، وتتبع تسمية الطرق للقواعد نفسها الناظمة لتسمية المتحولات.

* قائمة الوسطاء: List_of_arguments
  تعرّف عن مجموعة من الوسطاء التي تُستخدم لتمرير قيم تحتاجها الطريقة للقيام بعملها.

* المتحولات المحلية: Local_variables
  هي متحولات محلية تُعرّف ضمن الطريقة وتُستخدم للتخزين المؤقت للبيانات ضمن الطريقة.

* التعليمات: Statements
  هي مجموعة تعليمات برمجية تقوم بتنفيذ العمل المحدد للطريقة.
  تستطيع التعليمات البرمجية هنا الوصول إلى القيم المخزنة في كل من:
  1) وسطاء الطريقة.
  2) المتحولات المحلية.
  3) متحول الصف الذي يحوي الطريقة (المتحولات الأعضاء).

Scanned by CamScanner

/ /

✳ استدعاء الدالة في البرنامج: Invocation

يمكن استدعاء الطريقة بعد تعريفها في صنف معين من قبل أي طريقة أخرى (دالة main الرئيسية أو غيرها) بهدف الحصول على الخدمة التي تقدمها هذه الطريقة وذلك من خلال العبارة:

(وسطاء الدالة إن وجدت) اسم الدالة . اسم الكائن

ولا يمكن استدعاء أي دالة ضمن الصنف إلا عن طريق متحول من نوع الصنف نفسه (كائن).

٣ البانيات: Constructors

* الباني هو طريقة خاصة لها اسم الصنف المعرف فيه نفسه ويمكن أن تقبل مجموعة من الوسطاء.
* يختلف الباني عن الطرق العادية في أنه لا يملك نوع إرجاع (ولا حتى void) لأنه لا يعيد ظاهرياً أية قيمة، لكنه ضمنياً يعيد كائناً من نوع الصنف.
* تعتمد فكرة تعدد البانيات على مبدأ التحميل الزائد للدوال، أي إنه لا يمكن أن يكون لباني نفس عدد وأنواع وترتيب الوسطاء.
* تقوم البانيات بما يلي:
1) استحقاق كائنات objects من صنف معين، وذلك باستخدام الكلمة المفتاحية new.
2) إعطاء قيم بدائية للحقول المعرفة في صنف معين.
* لا تشترط لغة الجافا كتابة باني بشكل صريح في الصنف، فلغة الجافا تعرّف باني افتراضياً لوسطاء له إذا لم يكتب المبرمج بانياً فعلياً لاستحقاق الكائنات، ويكون له الشكل:

```
classname ( ) { }
```

ولكن ذلك لا يعني أن المترجم سيضيف تعليمة جديدة، بل سيقوم بإنشاء الباني الافتراضي في ملف التنفيذ byte code الذي لاحقته (.class) وليس في ملف التعليمات الذي لاحقته (.java)

- ولكن إذا قام المبرمج بتعريف باني ما ضمن الصنف، فإن المترجم لن يعرّف بانياً افتراضياً لهذا الصنف.

وهنا إذا أردنا استخدام الباني الافتراضي فيجب تعريفه بشكل صريح.

Scanned by CamScanner

تمرين (وظيفة): لم يقم الدكتور بحله

- اكتب صفاً يقوم بتمثيل طالب جامعي في كلية العلوم - قسم الرياضيات على أن يحوي سجله على علامات 48 مقرر دراسي فقط، علماً أن المقررات الدراسية مرقمة بالأرقام من 0 إلى 47.
- ومن ثم اكتب برنامجاً يقوم باستخدام الصف السابق لإدخال بيانات طالب واحد فقط وطباعتها على الشاشة.

الحل: - كتابة الصف:

```
class Student {
  String name;
  String Lastname;
  int degree[] = new int [48];
}
```

المعطيات الأعضاء

- كتابة البرنامج:

```
class UseStudent {
  static void main (String args[])
  {  Student S = new Student ();
     System.out.println ("Enter student's name:");
     S.name = Stdin.readString ();
     System.out.println ("Enter student's Last name:");
     S.Lastname = Stdin.readString ();
    System.out.println ("Enter student's marks:");
    for (int i=0 ; i<48 ; i++)
     { System.out.print ("mark" + (i+1) + "=");
       S.degree[i] = Stdin.readInt (); }
  System.out.print (S.name + " " + S.lastname + "'s marks are:");
  for (int i=0 ; i<48 ; i++)
     System.out.print (S.degree[i] + " ");
}}
```

ALADDIN.net
Scanned by CamScanner