

Atelier Programmation Orientée Objet Avancée –JAVA–

TP2 Les classes en Java

ISET Zaghouan



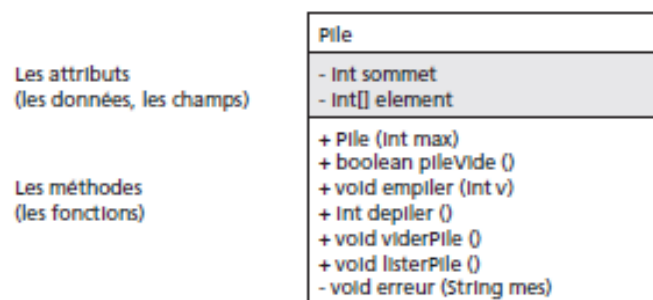
Enseignant

Boukchim_mossaab@yahoo.fr

Une classe est une extension de la notion de module. Les données et les fonctions traitant les données sont réunies ensemble dans une classe qui constitue un nouveau type. On peut déclarer des variables du type de cette classe. Cela revient à avoir la possibilité de dupliquer les modules en leur attribuant des noms différents, chaque module ayant ses propres variables locales visibles ou non de l'extérieur.

L'exemple de la pile peut être schématisé comme indiqué sur la figure suivante qui suit les conventions de la modélisation objet UML (Unified Modeling Language). On distingue les données de la classe appelées attributs en PO (Programmation Objet) et les fonctions appelées méthodes. Un signe moins(-) devant un élément indique un élément privé, visible seulement dans la classe ; un signe plus(+) indique un élément public, accessible par tous les utilisateurs de la classe ; l'absence de signe indique une visibilité ¹ seulement pour les classes du paquetage (du répertoire).

Les attributs sommet et element, et la méthode erreur() sont privés ; les autres méthodes sont publiques.



La classe est un modèle qui s'apparente à la déclaration d'un nouveau type de données. On peut déclarer une variable (on parle d'objet ou d'instance de la classe en PO) du type de cette classe.

Exemple :

```
Pile p1 = new Pile (5); // p1 et p2 sont des instances de la classe Pile
Pile p2 = new Pile (100); // p1 et p2 sont des objets de type Pile
```

Exercice 1 « La classe Pile » :

Il s'agit d'écrire le code en Java de la classe Pile. Le mot clé "private" indique que les attributs sommet et element, et la méthode erreur() sont privés.

¹ En Java, il y a quatre types d'accès à un attribut ou à une méthode. Devant un attribut ou une méthode on peut déclarer : private, public ou ne rien déclarer, ce qui signifie un droit de paquetage. Un autre type d'accès (protected) sera vu aux TP's suivants en relation avec le concept d'héritage.

Exercice 2 « La mise en œuvre de la classe Pile » :

Il s'agit d'écrire le programme PPPile qui crée l'objet pile1 de type Pile sur lequel on applique, en fonction des réponses au menu de l'utilisateur, des méthodes de la classe Pile.

Exemples de résultats d'exécution de PPPile.java :

```
GESTION D'UNE PILE
0 - Fin
1 - Initialisation de la pile
2 - La pile est-elle vide ?
3 - Insertion dans la pile
4 - Retrait de la pile
5 - Vidage de la pile
6 - Listage de la pile
Votre choix ? 1
Taille de la pile ? 4
Initialisation d'une pile vide
Menu
Votre choix ? 2
La pile est vide
Menu
Votre choix ? 4
***erreur : Pile vide
Valeur depilee : 0
Menu
Votre choix ? 3
Valeur a empiler ? 12
Menu
Votre choix ? 4
Valeur depilee : 12
```

Si la pile est saturée :

```
Menu
Votre choix ? 3
Valeur a empiler ? 14
***erreur : Pile saturee
```

Remarque :

- ✓ Les cas d'erreurs pourraient être gérés différemment en utilisant la notion d'exceptions pour les détecter et les prendre en compte.
- ✓ Les données sont encapsulées. L'accès à l'entier sommet de la Pile pile1 serait autorisé si sommet était déclaré public dans la classe Pile ; dans ce cas pile1.sommet serait valide dans PPPile.java pour accéder à la variable sommet de la pile. En vertu du principe d'encapsulation, il est préférable que cet accès soit interdit.