

Atelier Programmation Orientée Objet Avancée –JAVA-

TP 3

Les classes en Java

- Les constructeurs -

ISET Zaghuan



Enseignant

Boukchim_mossaab@yahoo.fr

Problème 1

Voici un autre exercice relativement simple qui vous permettra de faire vos premiers pas en Java : utilisation de l'éditeur de texte, compilation des sources (commande `javac`), exécution du bytecode sur la JVM (commande `java`), compréhension des mécanismes d'instanciation d'objets, d'appels de méthodes, . . . bref, le B-A-BA de la programmation Java.

Nous allons pour cela prendre l'exemple des comptes bancaires. Il s'agit de la classe `Compte` disposant de deux attributs `numéro` et `solde`, d'un constructeur, ainsi que de trois méthodes `déposer`, `retirer` et `donnerSolde`.

Il vous est tout d'abord demandé d'écrire et de compiler cette classe en Java. Comme vous pourrez le constater, le compilateur Java est très rigoureux. . . ;-) Vous apprendrez néanmoins assez rapidement à interpréter les messages d'erreurs les plus couramment renvoyés par le compilateur.

Une fois ceci fait, il vous faut écrire un "programme principal" pour pouvoir tester votre nouvelle classe. En Java, nous réalisons généralement ceci en écrivant une nouvelle classe ne disposant que d'une unique méthode "main". C'est dans cette méthode un peu particulière que nous allons instancier un ou plusieurs objets à partir de la classe `Compte`, puis que nous allons leur envoyer des messages : les appels de méthodes.

1. Ecrire la classe `Compte` via l'éditeur. Sauvegardez dans le fichier `Compte.java`
 2. Compiler cette classe : `C :...>javac Compte.java`
- Sauf erreur, vous obtenez un fichier `Compte.class`
3. Ecrire la classe `TestCompte` contenant le "programme principal" (fichier `TestCompte.java`)
 4. Compiler la classe `TestCompte`
 5. Exécuter votre programme via la commande `C :...>java TestCompte` (sans le `.class`)

Voilà ce qui s'affiche à l'exécution du programme `TestCompte` si ça peut vous aider.

<pre>1. Créer un compte 2. Afficher un compte 'Solde' 3. Retirer Argent 4. Déposer Argent 5. Sortir Votre choix :</pre>

Problème 2

On va modéliser une application devant servir à l'inventaire d'une bibliothèque. Elle devra traiter des documents de nature diverse : des livres, des dictionnaires, et autres types de documents qu'on ne connaît pas encore précisément mais qu'il faudra certainement ajouter un jour (bandes dessinées, dictionnaires bilingues,...).

On définira une classe Bibliothèque réduite à la seule méthode main pour tester les différentes classes de ce problème.

Exercice 1

Tous les documents possèdent un titre. Quand un document est créé, son titre est donné à la création et par la suite, il ne change plus.

1. Définir la classe Document avec son constructeur public, et la propriété titre privée et son accesseur. (**Les accesseurs sont des méthodes fournissant ou modifiant certaines des caractéristiques de la classe.**)
2. On veut attribuer un numéro d'enregistrement unique dès que l'on crée un objet Document. On veut que le premier document créé ait le numéro 0, et que ce numéro s'incrémente de 1 à chaque création de document. Quelles propriétés faut-il ajouter à la classe Document ? Lesquelles doivent être static ? Les ajouter sans leurs accesseurs. Puis ajouter une méthode getNumero renvoyant le numéro d'enregistrement du document. (**Certains attributs peuvent être communs à tous les objets de la classe et exister indépendamment de tout objet de la classe. Ces attributs sont déclarés static. On peut trouver des variables static et des constantes static.**)
3. Ajouter un second constructeur public permettant de définir aussi, en plus du titre, le numéro d'enregistrement (supposé positif ou nul) lors de la création d'un document. Quelles restrictions peut-on prendre afin d'éviter que deux documents créés aient le même numéro ? Dans les cas de restriction, on prendra le numéro suivant prévu, à la place du numéro indiqué par l'utilisateur.
4. Redéfinir la méthode toString renvoyant la chaîne de caractères constituée du numéro d'enregistrement et du titre du document.

Exercice 2

On veut implémenter une liste de documents dans laquelle on veut pouvoir ajouter des documents petit à petit, et y accéder par leur numéro d'enregistrement. Mais plutôt que d'utiliser les structures de collections Java (ArrayList, Vector, etc.), on veut programmer soi-même une structure de données.

Pour implémenter cette structure, on va créer une classe **ListeDeDocuments** avec un tableau de documents tableau contenant les documents de la liste. Cependant, pour éviter d'avoir à augmenter la taille du tableau à chaque ajout, on va fixer initialement sa taille à 1 et doubler sa taille dès qu'on veut ajouter un document dont le numéro d'enregistrement est plus grand que la taille du tableau.

1. Définir la classe **ListeDeDocuments** avec la propriété privée tableau et la méthode ajouter qui ajoute un document à la place indiquée par son numéro d'enregistrement, ainsi qu'une méthode getDocument qui prend en argument un numéro et renvoie le document de la liste ayant ce numéro (null s'il n'est pas dans la liste).
2. Dans cette classe, redéfinir la méthode toString pour qu'elle renvoie les descriptions de tous les documents de la liste, séparées par un saut de ligne "\n".

Exercice 3

Concevoir et codifier une classe Bibliothèque réduite à la seule méthode main pour tester les différentes classes de ce problème (menu d'utilisation).