TP1 : Classes et Méthodes

1 Installation

Téléchargez le Java SE Development Kit (JDK) : un environnement complet qui contient plusieurs outils pour la programmation Java et notamment un compilateur et une machine virtuelle. Le choix du programme d'installation à télécharger dépend du système d'exploitation sur lequel vous comptez l'utiliser : Linux ou Windows (normalement, la JDK est installé par défaut sur les Mac OS récents).
Le lien à partir duquel vous pouvez télécharger le JDK est :
http ://www.oracle.com/technetwork/java/javase/downloads/index.html

Vous pouvez écrire des classes Java avec n'importe quel éditeur de texte. Cependant, nous vous recommendons l'installation d'Eclipse :
https ://www.eclipse.org/downloads/

Merci de respecter les conventions d'écriture suivante :
- Les noms de classes commencent toujours par une Majuscule.
- Les noms d'instances ou de variables commencent par une minuscule.

2 Exercice 1

1. A l'aide d'un éditeur de texte sous windows ou sous linux, créer un fichier `Bonjour.java` contenant :

```
public class Bonjour  {
   public void affiche(){
       System.out.println("bonjour");
    }
} // Bonjour
```

   Sous n'importe quelle systéme s'exploitation, compiler le fichier Bonjour.java à l'aide de la commande `javac Bonjour.java.`
   Vérifier bien que le fichier Bonjour.class est créé.
   Pourquoi la commande `> java Bonjour` renvoie-t-elle un message d'erreur ?

2. Modifier le fichier `Bonjour.java` comme suit :

```
public class Bonjour  {

   public void affiche(){
       System.out.println("bonjour");
```

```
    }

    public static void main(String args[]){
        Bonjour x=new Bonjour();
        x.affiche();
    }
} // Bonjour
```

3. Compiler à nouveau cette classe. Corriger les erreurs éventuelles signalées.
4. Vérifier que le fichier `Bonjour.class` a été bien créé, et l'exécuter en tapant : `java Bonjour`
5. Modifier le programme précédent pour qu'il affiche Bonjour suivi d'un prénom et d'un nom.
6. Exécuter le programme

3 Classe Point : partie1

Un point est représenté par deux coordonnées x et y, avec x est l'abscisse du point et y son ordonnée.

— Déclarer une classe Point et ses attributs.
— Écrire une méthode **initialise** pour attribuer des valeurs aux coordonnées d'un points.
— Écrire une méthode **deplace** pour modifier les coordonnées d'un point. dx et dy sont des réels qui sont placés en paramètre de la méthode.
— Écrire une méthode **affiche**, pour afficher les coordonnées d'un point.

Appelez toute les méthodes dans la fonction main.

4 Classe Point : partie2

Réécrire la même classe **Point** et transformer la méthode **initialise** en un constructeur avec deux arguments. Appeler ce constructeur dans la fonction main.

5 Classe Point : partie3

Nous avons crée une classe Point dans l'exercice 1, copier cette classe et répondre aux différentes questions :

— Écrire un constructeur qui prend en paramètre les valeurs de x et de y.
— Écrire un constructeur qui est un constructeur de copie. Il a pour but de créer un Point qui a les mêmes attributs que le Point passé en paramètre.

— Écrire un nouveau constructeur qui donne la valeur (1,2) aux coordonnées du Point. Appeler ce constructeur dans la fonction main.
— Écrire une méthode qui affiche le Point courant (le Point qui appellera la méthode).
— Écrire une méthode qui translate le Point courant. La coordonnée sera déplacée de dx et l'abscisse de dy. dx et dy sont des réels qui sont placés en paramètre de la méthode.
— Écrire une méthode qui donne une valeur à l'abscisse du Point courant. Cette valeur sera passée en paramètre de la méthode.
— Écrire une méthode qui retourne la valeur de l'ordonnée du Point courant.

Appeler toute les méthodes dans la fonction main.
Toutes les méthodes seront testées au fur et à mesure dans une classe executable.

6 La somme

Écrire une classe utilitaire **CalcTableau** disposant des méthodes statiques suivantes :
— une méthode **somme** qui fournit la somme des valeurs d'un tableau de réels (double) de taille quelconque,
— une méthode **increment** qui incrémente d'une valeur donnée toutes les valeurs d'un tableau de réels (double).
— une méthode qui affiche des valeurs d'un tableau de réels
— Écrire un petit programme d'essai TstCalcTableau.

7 La moyenne

Le but de cet exercice est de programmer une classe **MoyenneTable** pour faire la moyenne de nombres d'un tableau.

— Déclarer la classe et son attribut.
— Écrire un constructeur qui prend en paramètre un tableau de String. Dans la classe Double de la bibliothèque java la méthode *double Double.parseDouble(String a) permet de changer le String a en double.*
— Écrire une méthode permettant l'affichage de l'attribut.
— Écrire une méthode qui retourne la somme des éléments du tableau.
— Écrire une méthode utilisant la méthode précédente (somme) et qui retourne la moyenne des éléments du tableau.
— Écrire une fonction main permettant d'exécuter le programme.

8 Un tableau de Points

Dans l'exercice precedent, nous avons crée une classe Point. Dans le même repertoire,
— Déclarer une classe *tableaudepoints* et son attribut "un tableaux de Points".

— Écrire un constructeur qui prend en paramètre la taille du tableau. Générer aléatoirement des points et les mettre dans le tableau.
— Écrire une méthode permettant d'afficher le contenu du tableau.
— Écrire une méthode main pour tester votre programme.