

TD n°1 – Découvrir la programmation orientée objet avec Java

L'objectif des exercices ci-dessous est de mettre en œuvre le JDK 1.8 par lignes de commandes et de créer les premières classes Java.

Ces travaux doivent être exécutés sur les postes de travail du Département Informatique. Le JDK support est pré installé dans l'espace des logiciels de développement.

Création d'une arborescence de TD&TP

Sur votre espace personnel sur P:, créer une arborescence physique de sous-répertoires dédiée au module M213. Cette arborescence devra permettre de localiser de façon aisée : l'énoncé, les annexes et tous les fichiers sources qui seront manipulés pour chaque exercice de chaque séance de TD/TP.

Enregistrer dans cette arborescence l'énoncé de la présente feuille d'exercice. Cette action devra être reproduite ensuite pour chaque séance de TD/TP.

Options de compilation Java

Dans une "invite de commandes" lancée au préalable, exécuter le compilateur Java, sans aucun argument.

- 1) Quel est le rôle de l'option `-classpath` ?
- 2) Quel est le rôle de l'option `-version` ?
- 3) Quel est le rôle de l'option `-d` ?

Compilation et exécution d'une première classe

Dans un éditeur tel Notepad++, créer votre première classe *HelloWorld.java* en recopiant le code donné en Cours. Enregistrer cette classe dans le répertoire src de la séance de TD/TP d'aujourd'hui.

Dans une "invite de commandes" lancée au préalable, placez-vous dans le bon répertoire et compiler avec succès la classe HelloWorld. Identifier le nom du fichier bytecode produit par le compilateur.

Modifier le nom du fichier source et vérifier les contraintes de nommage imposées par le langage.

Finalement, exécuter ce programme dans l'invite de commandes.

Création d'une première classe

A partir du T.A.D. point vu au premier semestre :

```
typedef struct
{
    double x;
    double y;
}point;
```

Ecrire la classe *Point.java* définissant l'objet Point. Dans cette classe, vous ajouterez :

- les constructeurs adéquats (sans paramètre, valeurs des attributs en paramètre) :

```
public Point()
public Point(double x, double y)
```
- les accesseurs de consultation afin de récupérer la valeur de chaque attribut :

```
public double getAbscisse()
public double getOrdonnee()
```
- les méthodes appliquées à ces objets :

```
// --- Calcul de la distance entre le point courant et p
public double distanceP(Point p)

// --- Affichage d'un point
public String toString()
```

Dans la classe *Point.java*, ajouter une méthode **main** (`public static void main(String[] args)`) afin d'instancier des objets Point et tester les méthodes. Compiler et exécuter la classe Point en ligne de commandes.

Pour aller plus loin

- Externaliser la méthode main : vous créerez un deuxième fichier *TestPoint.java* qui contiendra uniquement la méthode main précédemment écrite dans *Point.java*. (Pensez à commenter ou effacer la méthode main de la classe Point !)
- Introduire dans la classe Point deux nouvelles méthodes *projX* et *projY* permettant respectivement de calculer le *projeté d'un point support* sur l'axe des abscisses et sur l'axe des ordonnées.
 Compiler avec succès cette nouvelle version. Modifier le programme de mise en œuvre de la classe Point (i.e. *TestPoint.java*) pour y introduire des exemples d'appel aux nouvelles méthodes et exécuter ce dernier avec succès.