

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

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

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

I. Création d'une arborescence de TD&TP

Sur votre espace personnel, 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.

II. 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` ?

III. 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.

- 1) Dans une "invite de commandes" lancée au préalable, placez-vous dans le bon répertoire et compiler avec succès la classe `HelloWorld`.
- 2) Identifier le nom du fichier bytecode produit par le compilateur.
- 3) Modifier le nom du fichier source et vérifier les contraintes de nommage imposées par le langage.
- 4) Finalement, exécuter ce programme dans l'invite de commandes.

IV. Bonnes pratiques de programmation : mise en œuvre d'une classe `Point.java`

1) Création d'une classe `Point.java`

A partir du T.A.D. point suivant :

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

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

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

```
public Point()  
public Point(double x, double y)
```
- les méthodes appliquées à ces objets :

```
// --- Calcul de la distance entre le point courant et p  
public double getDistance(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.

2) Utiliser l'IDE Eclipse

Lire le tutoriel disponible à l'adresse : <http://www.eclipsetotale.com/articles/premierPas.html>

Plus précisément :

- Sur votre machine, ouvrir la dernière version disponible d'Eclipse (Neon ou Photon)
- Choisir comme *workspace* (espace de travail), le répertoire M213 créé sur votre espace personnel
- Créer un nouveau projet Java (*Java Project*) intitulé TD1
- Dans ce projet (voir le « *Package Explorer* » sur la gauche de l'écran d'Eclipse), cliquer avec le bouton droit de la souris sur le répertoire *src* et choisir « *New > package* ». Créer 1 package par exercice de TD. *Le nom des packages commence par une lettre minuscule.*

Une fois le package créé, clic droit sur le package pour ajouter une classe (*New > Class*)

Créer un package « point » et y ajouter la classe *Point.java* précédemment créée. Exécuter le main directement dans Eclipse.

3) Encapsulation

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* !)

- Essayer d'accéder aux attributs des objets instanciés. Est-ce possible ?
- Tester l'accès en changeant l'encapsulation des attributs : *rien*, *public*, *protected*, *private*.
- Créer un nouveau package « point2 » dans lequel vous créerez une classe permettant de voir les effets de l'encapsulation des attributs de *point.Point*.

Les bonnes pratiques de programmation recommandent de mettre les attributs en mode *private* afin de les protéger de modifications intempestives.

- Retourner dans la classe *Point* et ajouter les accesseurs de consultation afin de récupérer la valeur de chaque attribut :

```
public double getX()  
public double getY()
```

Vous pouvez ajouter les getters/setters directement dans Eclipse à partir du menu *Source > Generate Getters/Setters*.

- Vérifier dans les programmes de mise en œuvre (*main*) créés précédemment que vous pouvez accéder à la valeur des attributs par ces accesseurs.
- 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.
Modifier `TestPoint.java` pour y introduire des exemples d'appel aux nouvelles méthodes et exécuter ce programme avec succès.

4) Utiliser l'API Java

Consulter <http://docs.oracle.com/javase/10/docs/api/>.

- Décrire la page de l'API : à quoi peut-on accéder ?
- Comment rechercher une classe ? un package ?
- Quelles informations trouvez-vous dans le descriptif d'une classe ?

Rechercher l'objet `Point` existant dans l'API,

- Dans quel package se situe la classe ?
- Quelles sont les différences avec la classe que l'on a créée ?
- Ajouter les constructeurs et méthodes suivantes dans votre classe `Point` suivant le descriptif de la classe de l'API :

```
public Point(Point p)  
public boolean equals(Object obj)  
public Point getLocation()  
public void setLocation(Point p)  
public void setLocation(double x, double y)  
public void translate(int dx, int dy)
```

5) Documentation

Javadoc est un outil fourni avec le JDK pour permettre la génération d'une documentation technique à partir du code source. Dans le code source Java, les commentaires utilisent des balises *@tag* reconnues par *Javadoc* pour générer la documentation technique dans des pages html.

Afin de générer la doc dans Eclipse, suivre le tutoriel suivant : <http://www.objjs.com/formation-java/tutoriel-java-creation-javadoc-eclipse.html>

- Commenter votre classe `Point` en utilisant les balises reconnues,
- Générer la documentation en ligne dans votre Java Project Eclipse.

Guide javadoc officiel d'Oracle : <https://docs.oracle.com/en/java/javase/11/javadoc/>