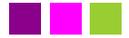


# Tp1 - Premiers Tps en java



## 1 Tests de classes java

1. Créez le répertoire de nom **java** qui va servir à ranger tous les programmes que vous allez écrire pour **BPO**.
2. Sous ce nouveau répertoire, créez un répertoire nommé **essai**. Placez le fichier **Test.java** donné sur arche dans votre répertoire **essai**. Vous obtenez l'arborescence de fichiers suivante :



3. le package **geometrie** se trouve sous : **/opt/depot/BPO**

**Ne copiez pas ce package chez vous !**

Vous pouvez visualiser la documentation des classes de **geometrie** à partir des fichiers **html** disponibles sous **/opt/depot/BPO/javadoc/geometrie**.

4. compilez et exécutez le fichier **Test.java** :

■ pour compiler un fichier java avec la commande **javac** :

- se placer dans le répertoire qui contient le package de la classe à compiler
- préciser l'emplacement des autres packages utilisés et le répertoire courant (noté **.**) avec l'option **classpath**
- intégrer les accents avec l'option **encoding**

exemple :

```
javac -classpath /opt/depot/BPO:. -encoding "iso-8859-1" essai/Test.java
```

■ pour demander l'exécution d'une classe avec la commande **java** :

- se placer dans le répertoire qui contient le package de la classe à compiler
- préciser l'emplacement des autres packages utilisés et le répertoire courant (noté **.**) avec l'option **classpath**
- donner le nom complètement spécifié de la classe à exécuter (classe contenant la fonction statique **main**)

exemple :

```
java -classpath /opt/depot/BPO:. essai.Test
```

## 2 Utilisation de la classe Polygone

---

Dans le même répertoire **essai**, créez un fichier de test de la classe **Polygone**. Pensez à nommer correctement le fichier (suffixe **.java**) et à commencer ce texte par la clause **package** et les clauses **import** adéquates. Ce fichier sera à déposer sur arche à la fin de la séance, après avoir programmé les points suivants :

1. Écrire une séquence d'instructions qui crée un polygone de 4 points. Afficher le polygone et son périmètre. Vérifier le résultat.
2. Écrire une séquence d'instructions qui translate chaque point. Les coefficients de translation, différents pour chaque point, sont lus sur l'entrée standard. Afficher le polygone et son périmètre. Vérifier le résultat.
3. Écrire une séquence d'instructions qui translate un point sur deux, sous réserve que le polygone soit convexe et que tous ses points soient situés dans les abscisses/ordonnées positives.
4. Écrire une séquence d'instructions qui identifie le premier côté d'une longueur supérieure ou égale à une longueur donnée, lue sur l'entrée standard. Afficher les deux points concernés (s'ils existent).
5. Écrire une séquence d'instructions qui crée un second polygone avec quatre autres points et le compare au premier. Afficher un message indiquant si les coordonnées des points des deux polygones sont identiques ou non.
6. Écrire une séquence d'instructions qui compare deux polygones, à une rotation près dans l'ordre des points, en faisant l'hypothèse que tous les points d'un même polygone sont différents.

À la fin de la séance, déposez votre fichier java de test sur arche.