

Java DUT 1 Feuille TP2  
Université Paris-Est Marne-la-Vallée

**Exercice 1.**—

Reprendre la classe `Segment` vue en cours (voir les slides du cours 2) qui sert à représenter des segments de droites dans un plan définis par deux points du plan.

- a) Écrire la classe `Segment`. Chaque segment contient deux points nommés `start` et `stop`. Mettre les "getters" et les "setters". Écrire deux constructeurs. L'un prendra en argument deux objets `Pixel` et l'autre quatre variables entières.
- b) Écrire une classe `SegmentTest` destinée à tester les classes `Segment` et `Pixel`. Elle contiendra une méthode `main` où on crée un segment `s1` avec le premier constructeur et un segment `s2` avec le second constructeur.
- c) On essaye de créer un segment `s3` qui va du point  $(0,0)$  au point  $(0,0)$  avec

```
public class SegmentTest {
    public static void main(String[] args) {
        Segment s3 = new Segment();
    }
}
```

Que se passe-t-il ? Pourquoi ?

- d) Écrire dans la classe `Segment` une méthode `toString` permettant d'afficher les deux points d'un segment. On utilisera la méthode `toString` de `Pixel` selon la technique de délégation. Dans `SegmentTest` affichez `s1` et `s2`.
- e) Écrire dans la classe `Segment` une méthode `translate` permettant de déplacer un segment dans le plan. On utilisera une méthode `translate` de `Pixel` selon la technique de délégation.
- f) Écrire une méthode `connectedTo` qui teste si deux segments ont une extrémité (un point) en commun. Dans quelle classe faut-il la mettre ? Quel est le type de retour de cette méthode ?
- g) Testez vos méthodes dans `SegmentTest`.

### Exercice 2.—

Dans la méthode `main` de la classe `PixelTest` on crée tout d’abord une liste vide destinée à contenir des objets `Pixel` avec le code ci-dessous (première ligne du `main`). Le type des objets contenus dans la liste est indiqué entre les crochets. En Java, tous les éléments de la liste doivent être du même type contrairement à Python. On ajoute ensuite chaque point dans la liste avec la méthode `add`.

```
import java.util.ArrayList;
public class PixelTest {
    public static void main(String[] args) {
        ArrayList<Pixel> list = new ArrayList<Pixel>();
        Pixel p1 = new Pixel(0,0);
        Pixel p2 = new Pixel(2,5);
        list.add(p1);
        list.add(p2);
    }
}
```

- a) Écrire cette méthode `main` en ajoutant quelques points dans la liste et afficher la liste `list`. Que remarque-t-on ?
- b) Créez une liste de points `list2` qui contient tous les points de coordonnées  $(i, i)$  pour  $i$  allant de 0 à 10.
- b) On crée un nouveau point `p3`. Écrire une instruction qui permet de tester si la liste `list2` contient le point `p3`. On consultera la documentation pour voir quelle méthode de la classe `ArrayList` on peut utiliser.