Travaux Dirigés de Java n°2

Cours de Java

—*Maîtrise d'informatique*—

Octobre 2001

▸ **Exercice 1.** *Essayer le code*

```
Point p1 = new Point(1,1);
Point p2 = new Point(1,1);
System.out.println(p1.equals(p2));
```

Expliquer. Que faudrait-il faire dans la classe `Point` *pour avoir la réponse qu'on imagine ?*

▸ **Exercice 2.** *Essayer le code*

```
Object o1 = new Point(1,1);
Object o2 = new Point(1,1);
System.out.println(o1.equals(o2));
```

Expliquer. Que faudrait-il faire dans la classe `Point` *pour avoir la réponse qu'on imagine ?*

▸ **Exercice 3.** *On souhaite avoir un type* `Point` *possédant les méthodes suivantes:*

```
boolean equals(Object)
String toString()
void translate(double dx, double dy)
double getX()
double getY()
void setX(double x)
void setY(double y)
```

On souhaite également, pour ce type, avoir deux implémentations possibles: une où les coordonnées des points sont des entiers (de type `int`*) et l'autre où les coordonnées des points sont des réels (de type* `double`*). Proposez une solution, et implémentez-là. Que faut-il faire pour écrire le moins de code possible ?*

▸ **Exercice 4.** *On ordonne les points en les projetant sur l'axe des abscisses. S'ils ont la même abscisse, on compare les ordonnées. Écrire une méthode dans laquelle on déclare un ensemble de points, qu'on triera. On se servira de la classe* `java.util.Arrays` *et de l'interface* `java.util.Comparator`*.*

▸ **Exercice 5.** *On souhaite pouvoir comparer des points suivants plusieurs ordres différents (en comparant les abscisses ou bien en comparant les ordonnées). On pourra, par exemple, faire deux méthodes de la classe* `Point` *retournant les comparateurs. On ne souhaite pas que les classes implémentant les comparateurs soient visibles de l'extérieur de la classe* `Point` *autrement que via ces méthodes.*

▸ **Exercice 6.** *Créez une classe d'objets, qui, en mourrant, affichent un message. Essayez.*

▸ **Exercice 7.** *Qu'affiche le code suivant?*

```
package a;
public class A {
  public void m() {System.out.println("A.m");}
  void m1() {System.out.println("A.m1");}
  public void m2(A a) {System.out.println("A.m2");}
  public static void main(String[] args) {
    A a = new b.B();
    a.m();
    a.m1();
    a.m2(a);
    a.m2((b.B)a);
    ((b.B)a).m2(a);
    ((b.B)a).m2((b.B)a);
  }
}

package b;
public class B extends a.A {
  public void m(){
    System.out.println("B.m");
  }
  public void m1(){
    System.out.println("B.m1");
  }
  public void m2(b.B a){
    System.out.println("B.m2");
  }
}
```