

Bases de programmation - TD 4 : Programmation objet

IUT Aix-Marseille / DUT R&T 1^{ière} année
J. Seinturier (<http://www.seinturier.fr>)

1. Objets simples et Java

Exercice 1.1: Définir en Java une classe `Point` qui permet de représenter des points dans un espace à 2 dimensions (couple de coordonnées `x`, `y`). Les coordonnées d'un point sont des nombres réels.

```
public class Point {  
  
    float x;  
    float y;  
}
```

Exercice 1.2: Ajouter à classe `Point` une méthode `init()` qui prend en paramètre deux nombres réels et les affectent respectivement aux coordonnées `x` et `y` du point courant.

```
void init(float a, float b){  
    x = a;  
    y = b;  
}
```

Exercice 1.3: Ajouter à classe `Point` une méthode `dist()` qui prend en paramètre un autre point et retourne la distance entre le point actuel et le point donné. **Rappel** : si vous avez besoin de calculer la racine carrée d'un nombre `x` en Java, il faut utiliser l'instruction `Math.sqrt(x)`.

```
float dist(Point p){  
    float x2 = (p.x - x) * (p.x - x);  
    float y2 = (p.y - y) * (p.y - y);  
    return Math.sqrt(x2 + y2);  
}
```

Exercice 1.4: Définir en Java une classe `Rectangle` permettant de manipuler de tels objets. Un rectangle est défini par un `Point` représentant son coin inférieur droit (origine) ainsi qu'une longueur et une largeur. Les positions et les longueurs doivent être exprimées sous forme de nombres réels.

```
public class Rectangle {  
  
    Point origine;  
  
    float longueur;  
  
    float largeur;  
  
}
```

Exercice 1.5: Ajouter à classe `Rectangle` une méthode `init()` qui prend en paramètre un `Point` et deux nombres réels et les affectent respectivement à l'origine et à la longueur et largeur du rectangle.

```
void init(Point p, float lon, float lar){  
    origine = p;  
  
    longueur = lon;  
  
    largeur = lar;  
}
```

Exercice 1.6: Ajouter à la classe `Rectangle` une méthode `aire()` qui retourne l'aire du rectangle.

```
float aire(){  
    return longueur * largeur;  
}
```

Exercice 1.7: Ajouter à la classe `Rectangle` une méthode `translate()` qui prend en paramètre un `Point` et translate le rectangle selon le vecteur associé au point donné.

```
void translate(Point p){  
    origine.x = origine.x + p.x;  
    origine.y = origine.y + p.y;  
}
```

Exercice 1.8 : Ajouter à la classe `Rectangle` une méthode `contient()` qui teste si un `Point` donné en paramètre est à l'intérieur du rectangle.

```
boolean contient(Point p){  
    return (p.x >= origine.x) && (p.x <= origine.x + longueur)  
           && (p.y >= origine.y) && (p.y <= origine.y + largeur);  
}
```

Exercice 1.9 : Ajouter à la classe `Rectangle` une méthode `contientRect()` qui teste si un `Rectangle` donné en paramètre est entièrement à l'intérieur du `Rectangle` courant.

```
boolean contientRect(Rectangle r){  
  
    Point p1 = new Point();  
    p1.init(r.origine.x+r.longueur, r.origine.y + r.largeur);  
  
    return contient(r.origine) && contient(p1) ;  
}
```

Exercice 1.10 : Ajouter à la classe `Rectangle` une méthode `egaux()` qui teste l'égalité de deux rectangles. Deux rectangles sont égaux s'ils sont positionnés au même endroit et s'ils ont les mêmes dimensions.

```
boolean egal(Rectangle r){  
    return (origine.x == r.origine.x) && (origine.y == r.origine.y)  
           && (longueur == r.longueur) && (largeur == r.largeur);  
}
```

Exercice 1.11 : Définir en Java une classe `Cercle` qui permet de représenter des cercles dans un espace à 2 dimensions. Un cercle est caractérisé par un `Point` représentant son centre et une valeur réelle représentant son rayon.

Exercice 1.12 : Ajouter à la classe `Cercle` une méthode `aire()` qui calcule l'aire du cercle. ***Rappel :*** La valeur π est obtenue en Java par l'instruction `Math.PI`.

Exercice 1.13: Ajouter à la classe `Cercle` une méthode `translate()` qui prend en paramètre un `Point` et translate le cercle selon le vecteur associé au point donné.

Exercice 1.14 : Ajouter à la classe `Cercle` une méthode `contient()` qui teste si un `Point` donné en paramètre est à l'intérieur du cercle.