Programmation orientée objet
TD8

Romain Marie & Hervé Gaudin

Octobre 2019

3iLINGENIEURS
</les sciences informatiques>

1 Exercice 1

Le programme qui vous est fourni contient 7 erreurs de syntaxe ou de logique. Pour chacune d'entre elles :

(a) Rayez les éléments incorrects,

(b) Expliquez ce qui ne va pas,

(c) Proposez une correction pour parvenir au même résultat sans erreur. De la place est volontairement laissée si vous souhaitez ajouter de nouvelles méthodes.

2 Exercice 2

On souhaite maintenant ajouter une classe Rectangle à notre programme. Un rectangle sera représenté par une largeur, une hauteur, et un Point désignant son coin supérieur gauche.

1. En respectant l'encapsulation, écrivez une classe Rectangle qui respecte le cahier des charges ci-dessus, et implémente les méthodes suivantes :
   (a) Un constructeur avec 3 paramètres qui permette d'entièrement définir les propriétés d'un rectangle
   (b) Une méthode translater() qui applique un déplacement (en x et en y) au rectangle
   (c) Une méthode toString() qui génère une chaine de caractère décrivant le rectangle
2. Soit le code ci-dessous. A votre avis, qu'est-ce qui sera affiché à l'écran ? Proposez une amélioration pertinente.

```
public class TD1 {
    public static void main(String[] args) {
        Point p = new Point(12,13);

        Rectangle r = new Rectangle(p,2,3);
        r.translater(5, 6);

        System.out.println(r);
        System.out.println(p);
    }
}
```

3. Ajoutez à votre Rectangle ;
    - une méthode tourner() qui lui applique une rotation de Π/2 dans le sens anti-horaire.
    - une méthode contient() qui prend en paramètre un Point, et indique si le Point est dans le Rectangle
    - une méthode contient() qui prend en paramètre un Rectangle, et indique si le Rectangle (en paramètre) est dans le Rectangle
    - une méthode intersection() qui calcule l'aire de l'intersection du Rectangle avec un autre Rectangle passé en paramètre.
4. Complétez le main pour faire utilisation de toutes ces méthodes.

```java
public class TD1 {
    public static void main(String[] args) {
        Point p;                    // Point p construit par défaut aux coordonnées (0,0)
        Point p2 = new Point(12,13); // Point p2 construit aux coordonnées (12,13)

        // Place le point p aux coordonnées (12,13)
        p.x = 12;
        p.y = 13;

        // Affiche les coordonnées du point p sous la forme (x,y)
        System.out.println(p);

        // Affiche les coordonnées du point p2 sous la forme (x,y)
        System.out.println("("+p2.x+","+p2.y+")");

        // Affiche un message si p et p2 ont les mêmes coordonnées
        if(p == p2)
            System.out.println("Les deux points ont les mêmes coordonnées");

    }
}
```

```java
public class Point {
    private int x,y; // Coordonnées d'un point

    // Constructeur par défaut pour initialiser le point en (0,0)
    public void Point() {
        x = 0;
        y = 0;
    }

    // Constructeur pour initialiser le point aux coordonnées souhaitées
    public void Point(int x,int y) {
        x = this.x;
        y = this.y;
    }
```

```java
}
```