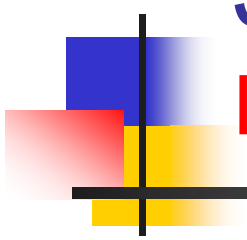

Java

Licence Professionnelle CISII, 2009-2010

Cours 4 : Programmation structurée

Exercice 1-2

- Créer une classe ListeDeRectangles avec une méthode qui affiche la surface de chaque rectangle
- Les méthodes
  - ListeDeRectangles() constructeur sans paramètre,
  - boolean estVide() qui indique si la liste est vide,
  - void inserer(Rectangle r) qui ajoute un rectangle au début de la liste,
  - String toString() qui donne des informations sur la liste de Rectangles sous forme de chaîne de caractères.
- Rajoutez un compteur pour compter le nombre de rectangles dans une ListeDeRectangles. Rajoutez un compteur pour compter le nombre de rectangles dans toutes les ListeDeRectangles. Rajoutez un compteur pour compter le nombre de ListeDeRectangles
- Testez l'utilisation.

La classe Point

```
public class Point {
        private int x;
        private int y;

    public Point(int x, int y) {
          this.x = x;
          this.y = y;
        }

    public void translater(int x, int y) {
          this.x = this.x + x;
          this.y = this.y + y;
        }

public int compareTo(Point p){
    if ((this.x == p.x) && (this.y
    == p.y)){return 0; }
        else if (this.x < p.x){
                return -1;}
        return 1;}

public void setX(int x) {this.x = x;}

public void setY(int y) {this.y = y;}

public int getX() {return x;}

public int getY() {return y;}

public String toString() {
   return ("("+ x +"," + y + ")");
      }
}
```

La classe Rectangle

```
public class Rectangle {
    private static int nbRectangles;
    private Point infGauche;
    private Point supDroit;

    public Rectangle() {
      infGauche = new Point(0,0);
       supDroit = new Point(1,1);
       nbRectangles++; //garde sa valeur jqa la gfin du programme car elle est static
    }

    public Rectangle(int x1, int y1, int x2, int y2) {
      infGauche = new Point(x1,y1);
      supDroit = new Point(x2,y2);
      nbRectangles++;
    }

public Rectangle(Point p1, Point p2) {
      infGauche = p1;
      supDroit = p2;
      nbRectangles++;
    }

public static int getNbRectangles() {
       return nbRectangles;}

public int surface() {
    return (supDroit.getX() - infGauche.getX())*
    (supDroit.getY() - infGauche.getY());
    }
```

```
public int compareTo(Rectangle r) {
    if ((this.infGauche.compareTo(r.infGauche) == 0 ) &&
        (this.supDroit.compareTo(r.supDroit) == 0)){return 0;}
        return (this.infGauche.compareTo(r.infGauche) );
      }

      public void translater (int a, int b){
        infGauche.setX(infGauche.getX() + a);
        infGauche.setY(infGauche.getY() + b);
        supDroit.setX(supDroit.getX() + a);
        supDroit.setY(supDroit.getY() + b);
      }

      public String toString(){
        return ("inferieur gauche : " + infGauche + " superieur droit
    : " + supDroit);
      }
}
```

```
public class ListeDeRectanglesNoeud {

  private Rectangle element;
  private ListeDeRectanglesNoeud
     suivant;

  public
     ListeDeRectanglesNoeud(Rectangle
     r,ListeDeRectanglesNoeud l){
    element = r;
    suivant = l;
  }

  public ListeDeRectanglesNoeud
     (Rectangle r){this(r,null);}

  public ListeDeRectanglesNoeud
     getSuivant(){return suivant;}

  public String toString(){
    return element.toString();}
}
```

```
public class ListeDeRectangles {

 private static int nbListes; //utilisable partout et
    garde sa valeur
 private ListeDeRectanglesNoeud premier;
 private int nbRectangles;

 public ListeDeRectangles(){
   premier = null;
   nbListes++;}

 public static int getNbListes () {
       return nbListes;}

 public int getNbRectangles() {
       return nbRectangles;}

 public boolean estVide(){
   return premier == null;}

 public void inserer(Rectangle r){
   if (estVide())
     premier = new ListeDeRectanglesNoeud (r);
   else
     premier = new ListeDeRectanglesNoeud
     (r,premier);
   nbRectangles++;
 }
```

```
public String toString(){
    String s="";
    if (estVide())
      return "Vide";
    ListeDeRectanglesNoeud  p = premier;
    while (p!=null){
      s = s + " " + p.toString() + "\n";
      p = p.getSuivant();
    }
    return s;
  }
}
```

```
public class Test {


  public static void main(String[] args) {
    Rectangle r = new
     Rectangle(1,2,4,5);
    ListeDeRectangles l = new
     ListeDeRectangles();
    System.out.println("on a cree la liste,
     est-elle vide ? "+ l.estVide());


     l.inserer(r);
     l.inserer(new Rectangle(0,0,1,1));
     l.inserer(new Rectangle(0,0,2,1));
     l.inserer(new Rectangle(0,0,3,1));
     System.out.println(l);


     ListeDeRectangles l2 = new
     ListeDeRectangles();
     l2.inserer(new Rectangle(0,0,4,1));
     l2.inserer(new Rectangle(0,0,5,1));
     System.out.println("Nombre total de
        rectangles :" +
        Rectangle.getNbRectangles());
        //Ceci est possible car il est déclaré
        en static
      System.out.println("Nombre de
        rectangles dans liste l :" +
        l.getNbRectangles());
      System.out.println("Nombre de
        rectangles dans liste 2 :" +
        l2.getNbRectangles());
      System.out.println("Nombre de listes
        :" +
        ListeDeRectangles.getNbListes());
  }
}
```