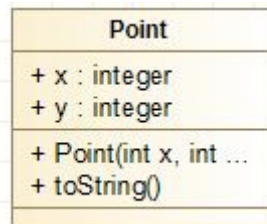


SLAM 4 Exercice 2

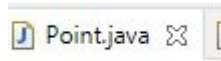
Dans un premier temps, nous avons créé le diagramme de classe pour la classe **Point** comme ci-dessous :



x et y en variable avec comme type “**int**” ensuite le constructeur **x** et **y** avec le type en paramètre.

Ensuite la méthode **toString()** pour faire appel aux accesseurs.

Une fois que le diagramme a été fait, nous allons commencer à programmer dans “**ECLIPSE**”. Pour cela, nous allons commencer à créer la classe “**Point**” comme ci-dessous :

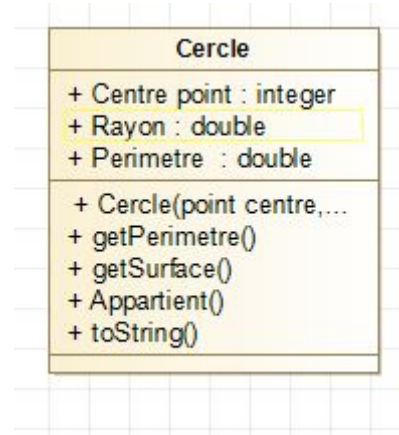


et ainsi nous allons ajouter son contenu comme ci-dessous :

```
Point.java Cercle.java
1 package exercice2;
2
3 public class Point {
4     private int x;
5     private int y;
6
7     public Point (int x, int y) {
8         this.x=x;
9         this.y=y;
10    }
11
12    public String toString() {
13        return "Le point est de"+x+"et"+y+".";
14    }
15    public double distance (Point p) {
16        int a =this.x-p.x;
17        int b=this.y-p.y;
18        return (Math.sqrt(a*a+b*b));
19    }
20
21
22
23 }
24
```

Ici nous avons initialisé les variables x et y avec comme type “**int**”. Par la suite nous avons créé le constructeur Point en lui affectant **x** et **y** avec type **int** dans les paramètres. Ensuite nous avons créé la méthode “**toString**” pour afficher la phrase avec les valeurs. Par la suite, nous avons fait appel à de nouvelles variables **a** et **b** avec type **int** et ces valeurs seront des formules pour la distance, le périmètre et de nous retourner le résultat.

Nous allons faire pour la classe “**CERCLE**” comme ci dessous :



Nous avons créé **int** pour le centre point ensuite d’affecter “double” pour le rayon et périmètre.

Puis nous avons créé le constructeur Cercle avec **point**, **rayon**, **périmètre** ensuite il s’agit des **méthodes**.

JH

Une fois que nous avons fait ceci, nous allons commencer à faire les méthodes dans la classe Cercle comme ci dessous :

```

1 package exercice2;
2
3 public class Cercle {
4
5     protected Point centre;
6     protected double rayon;
7     protected double perimetre;
8
9     public Cercle ( Point centre, double rayon) {
10         this.centre=centre;
11         this.rayon=rayon;
12     }
13
14     public Cercle (Point c, Point p) {
15         this.centre=c;
16         //int a = p.x-centre.p;
17         //int b = p.y-centre.y;
18         //this.rayon=Math.sqrt((a*a)+(b*b));
19         this.rayon=centre.distance(p);
20     }
21
22
23     public double getSurface() {
24         return Math.PI*rayon*rayon;
25     }
26     public double getperimetre () {
27         return 2*Math.PI*rayon*rayon;
28     }
29     public boolean appartient(Point p) {
30         double d=centre.distance(p);
31
32         if(centre.distance(p)<=rayon) {
33             return true;
34         }else return false;
35     }
36
37     public String toString() {
38         return ("X :" +centre.x + "Y:" +centre.y+"Rayon"+rayon);
39     }
40 }
41

```

Une fois que nous avons fait ceci, nous allons partir à la partie application. Pour cela nous allons commencer à créer une nouvelle classe qui se nommera “**main**” :

```

1 package exercice2;
2
3 public class main {
4
5     public static void main(String[] args) {
6         Point p = new Point (100,100);
7         Point p2 = new Point (200,200);
8         Cercle c1=new Cercle(p,40);
9         Point p3= new Point (130,100);
10        Cercle c2= new Cercle (p3,40);
11        System.out.println("Le périmètre du"+c1.toString()+"est :"+c1.getperimetre());
12        System.out.println("La surface du"+c1.toString()+"est :"+c1.getSurface());
13        System.out.println("Le périmètre du"+c1.toString()+"est :"+c1.getperimetre());
14        System.out.println("-----");
15        System.out.println("Le périmètre du"+c2.toString()+"est :"+c2.getperimetre());
16        System.out.println("La surface du"+c2.toString()+"est :"+c2.getSurface());
17        System.out.println("-----");
18        Point p4=new Point (120,100);
19        if((c1.appartient(p4)==true)&&(c2.appartient(p4)==true))
20            System.out.println(p4+"appartient à l'intersection de "+c1+ "et"+c2);
21        else
22            System.out.println(p4+"n'appartient pas à l'intersection de "+c1+ "et"+c2);
23
24
25
26

```

Ici dans cette classe, nous avons dans un premier temps créer les 3 points qui sont "p , p2 et p3" en leur affectant les valeurs énoncées dans l'énoncé. Par la suite, nous avons créé un nouveau cercle qui se nomme "c1" en lui affectant la valeur 40 comme rayon.

Par la suite nous avons créé un "sysout" pour récupérer les méthodes getPérimètre et toString() se situant dans la classe "**Cercle**". Ensuite nous avons créé un nouveau qui se nomme "**p4**" et on lui a affecté la valeur "120 et 130". Par la suite nous avons créé une condition "if" dans laquelle on dit que si le cercle "c1" est égal au point "p4" et si le cercle "c2" est égal au point 4 "**p4**" alors on affichera que le point p4 appartient à l'intersection du cercle 1 et le cercle 2.

sinon

on affichera que le point p4 n'appartient pas à l'intersection du cercle 1 et le cercle 2.

Voici le résultat :

```

Problems @ Javadoc Declaration Console
<terminated> main [Java Application] C:\Program Files\Java\jre1.8.0_251\bin\javaw.exe (11 sept. 2020 10:35:10 - 10:35:10)
Le périmètre duX :100Y:100Rayon40.0est :10053.096491487338
La surface duX :100Y:100Rayon40.0est :5026.548245743669
Le périmètre duX :100Y:100Rayon40.0est :10053.096491487338
-----
Le périmètre duX :130Y:100Rayon40.0est :10053.096491487338
La surface duX :130Y:100Rayon40.0est :5026.548245743669
-----
Le point est de120et100.appartient à l'intersection de X :100Y:100Rayon40.0etX :130Y:100Rayon40.0

```

