

Sujet B

Exercice 1 – Cercle

Q 1.1 Donner le code de la classe `Cercle` qui possède :

- un attribut rayon du type double.
- un constructeur à un paramètre pour initialiser le rayon.
- un constructeur sans argument qui initialise le rayon avec une valeur aléatoirement entre [12, 35[. Attention à la réutilisation de constructeurs existants.
- une méthode de copie qui renvoie un cercle ayant les mêmes caractéristiques.

```

1 public class Cercle {
2     private double rayon;
3
4     public Cercle(double rayon) {
5         this.rayon = rayon;
6     }
7
8     public Cercle() {
9         this(Math.random() * 23 + 12);
10    }
11
12    public Cercle copie(){
13        return new Cercle(rayon);
14    }
15 }
```

Q 1.2 Ajouter les méthodes suivantes :

- L'ascenseur et modificateur pour la variable d'instance rayon.
- `toString` qui renvoie une chaîne de caractère avec la valeur du rayon.
- `area` qui calcule l'aire du Cercle ($rayon^2 * \pi$), vous pouvez utiliser la constante `Math.PI` de la classe `Math`.
- `perimetre` qui calcule le périmètre du cercle ($2 * rayon * \pi$).
- `egalite` qui prend en argument un autre `Cercle` et renvoie true si ils sont égaux ou false si ils sont différents.

```

1     public double getRayon(){
2         return rayon;
3     }
4
5     public void setRayon(double rayon){
6         this.rayon = rayon;
7     }
8
9     public String toString(){
10        return "Cercle: rayon=" + rayon;
11    }
12    public double area(){
13        return (rayon * rayon * Math.PI);
14    }
15    public double perimetre(){
16        return (2 * rayon * Math.PI);
17    }
18    public boolean egalite(Cercle r){
19        return (rayon == r.rayon);
20    }
21 }
```

```
22 }
```

Exercice 2 – TestCercle

Q 2.1 Écrire une classe `TestCercle` contenant une méthode main dans laquelle on crée deux cercles, à chaque fois utiliser un constructeur différent. Créer un troisième cercle en utilisant la méthode de copie.

Q 2.2 Calculer et afficher l'aire d'un cercle créé à la question précédente.

Q 2.3 Calculer et afficher le périmètre d'un cercle créé à la question précédente.

Q 2.4 Afficher les caractéristiques des trois cercles.

```
1 public class TestCercle(){
2     public static void main(String [] args){
3         Cercle r1 = new Cercle();
4         Cercle r2 = new Cercle(10);
5         Cercle r3 = r2.copie();
6
7         System.out.println(r2.area());
8         System.out.println(r2.perimetre());
9         System.out.println(r1.toString());
10        System.out.println(r2);
11        System.out.println(r3);
12    }
13
14
15
16 }
```

Exercice 3 – Modification de la classe Cercle

Nous voudrions insérer un nouveau cercle dans un cercle existant, comme dans la figure 1. Les conditions à respecter sont :

- 1) le cercle inséré doit avoir moins que la moitié d'aire du cercle qui le héberge.
- 2) un cercle ne peut pas être inséré dans plusieurs cercles, et chaque cercle contient au max un cercle à son intérieur.

Q 3.1 Crée le(s) variable(s) d'instance(s) nécessaire(s) pour pouvoir insérer un cercle.

Q 3.2 Crée la méthode `insérer` pour insérer un cercle dans un autre cercle. La méthode affichera "L'insertion est pas possible" si une de conditions n'est pas satisfaite.

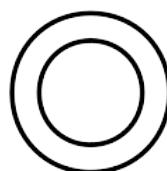


FIGURE 1 – Cercle inseré

```
1  private Cercle cercRef;
2
3  public void inserer(Cercle in){
4      if (cercRef != null || r1.area()/2 >= area() || in.cercRef != null){
5          System.out.println("L'insertion n'est pas possible")
6      }
7      else{
8          cercRef = in;
9          in.cercRef = this;
10     }
11 }
12 }
```