



Héritage, Mutabilité, redéfinition, polymorphisme, varargs

Exercice 1 - Circle

Pour tout l'exercice, pour vos tests, vous écrirez votre code une la classe `Main` dans le package `fr.uml.v.geom.main`.

1. Le code de la classe `fr.uml.v.geom.Point` ne compile pas.
Expliquer pourquoi ?
Discuter des choix d'implantation et de leurs influences sur la signature de la méthode `translate`.
2. Pour la suite de l'exercice, on choisira la version mutable de la classe `fr.uml.v.geom.Point`.
Écrire une classe `fr.uml.v.geom.Circle`, un cercle étant défini par un `Point` correspondant au centre et un rayon de type `int`. Que doit-on préciser en déclarant ces deux champs?
3. Écrire le constructeur du `Circle`.
4. Écrire la méthode `toString()` qui affiche le centre et le rayon.
5. Écrire la méthode `translate(int dx, int dy)` qui translate le cercle.
6. Qu'affiche le code suivant sur la console :

```
var point = new Point(1, 2);
var circle = new Circle(point, 1);

var circle2 = new Circle(point, 2);
circle2.translate(1, 1);

System.out.println(circle + " " + circle2);
```

Que doit-on faire pour que cela n'arrive pas ?

7. Quel est le problème avec l'accessor `getCenter()` codé de la façon suivante :

```
public Point getCenter() {
    return center;
}
```

Pour vous aider, regardez le code suivant :

```
var p = new Point(1, 2);
var c = new Circle(p, 1);
var p2 = c.getCenter();
p2.translate(1,1);
System.out.println(c);
```

Que doit-on faire pour que le code soit correct ?

8. Écrire la méthode `surface()` qui renvoie la surface du disque.
Modifier la méthode `toString()` pour quelle affiche aussi la surface.
9. Écrire la méthode `contains(Point p)` qui renvoie vrai si un point est contenu dans un disque.
10. Écrire la méthode `contains(Point p, Circle... circles)` qui renvoie vrai si un point est contenu dans un des disques.
Que veut dire `"..."` dans la signature de la méthode ?

Exercice 2 - One Ring for ...

Le but de cet exercice est de construire un anneau comme étant un cercle dont on a évidé une zone circulaire définie par son rayon interne.

Pour tout l'exercice, pour vos tests, réutilisez la classe `Main` vue précédemment.

1. Rappeler dans un premier temps, dans quel cas il est judicieux de faire de l'héritage.
2. Écrire la classe `Ring` qui hérite de la classe `Circle`.
3. Écrire un constructeur de la classe `Ring` prenant en paramètre, un centre, un rayon et un rayon interne. Faites attention à ce que le rayon interne soit inférieur au rayon de l'anneau.
Que faire dans le cas contraire?
Note: Tous les champs doivent être privés.
4. On souhaite maintenant qu'un anneau soit affiché comme un cercle, avec en plus au bout de la ligne "InternalRadius: ir" où ir est le rayon interne de l'anneau".
Modifiez ce qu'il faut pour que cela fonctionne. Essayez avec le code suivant (est ce que tout vous semble correct? Dans le cas contraire, que doit-on faire pour corriger?):

```
var point = new Point(1, 2);  
var circle = new Circle(point, 2);  
System.out.println(circle);  
var ring = new Ring(point, 2, 1);  
System.out.println(ring);
```

Ensuite...

1. Implantez une méthode `contains(Point)` en évitant d'allouer des objets ou de dupliquer du code.
PS: il existe deux solutions dont une plus élégante que l'autre.
2. Écrire la méthode `contains(Point p, Ring... rings)` qui renvoie vrai si un point est contenu dans l'un des anneaux.
3. Que peut-on conclure de ce TP?