

## 1 Classe Clip

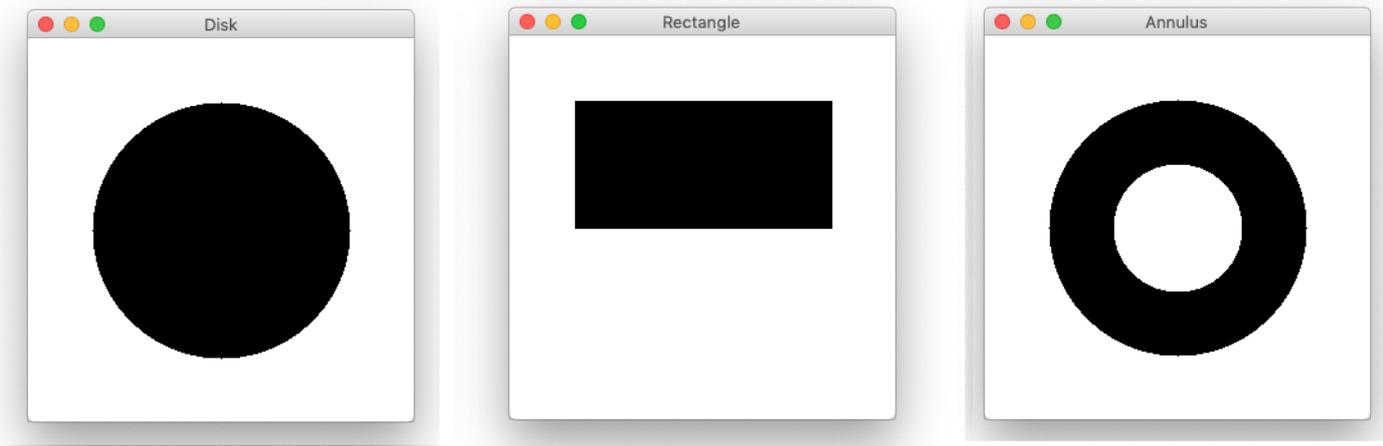
La classe suivante permet de définir des zones en 2D.

```
public class Clip {
    public enum Type { RECTANGLE, DISK, ANNULUS}

    private Type type;
    private int x,y,w,h,r;

    public Clip(Type type, double x, double y, double w, double h, double r) {
        this.type = type;
        this.x = x;
        this.y = y;
        this.w = w;
        this.h = h;
        this.r = r;
    }

    public boolean contains(double x, double y){
        switch(type) {
            double d = (this.x-x)*(this.x-x)+(this.y-y)*(this.y-y);
            case RECTANGLE:
                return this.x < x && x <= this.x+w
                    && this.y < y && y <= this.y+h;
                /* Vérifie si l'abscisse du point est entre x et x+w et si l'ordonnée
                du point est entre y et y+w */
            case DISK:
                return d <= r*r;
                /* Vérifie que la distance du point au centre (x,y) est inférieure à r */
            case ANNULUS:
                return d <= r*r && d >= w*w;
                /* Vérifie que la distance du point au centre (x,y) est comprise entre w et r */
            default:
                return false;
        }
    }
}
```



Affichage (pixels noir = pixels dans le Clip) de `new Clip(Clip.Type.DISK, 150, 150, 0, 0, 100)`, `new Clip(Clip.Type.RECTANGLE, 50, 50, 200, 100, 0)` et `new Clip(Clip.Type.ANNULUS, 150, 150, 50, 0, 100)` sur des fenêtres de 300 sur 300 pixels.

## 2 Questions

1. Décrivez les problèmes que vous avez avec les variables de ce code. Quels sont les autres problèmes de ce code ?
2. Écrivez le code d'une interface `ClipInterface` ayant une unique méthode `public boolean contains(double x, double y)`.
3. Écrivez le code d'une classe `Rectangle` ayant quatre attributs `int x`, `int y`, `int width` et `int height` et qui implémente l'interface `ClipInterface`. Cette classe représente un rectangle dont le coin en haut à gauche a pour coordonnées  $(x, y)$  avec une largeur égale à `width` et une hauteur égale à `height`.
4. Écrivez le code d'une classe `Disk` ayant trois attributs `int x`, `int y` et `int radius` et qui implémente l'interface `ClipInterface`. Cette classe représente un disque dont le centre a pour coordonnées  $(x, y)$  avec un rayon égal à `radius`.
5. Écrivez le code d'une classe `Annulus` ayant deux attributs `Disk innerDisk` et `Disk outerDisk` et qui implémente l'interface `ClipInterface`. Cette classe représente une couronne, c'est-à-dire une région du plan comprise entre deux cercles de même centre. Le plus grand cercle de la couronne correspond à l'attribut `outerDisk` alors que le plus petit cercle (le trou) correspond à `innerDisk`.
6. Lors de la construction d'un `Annulus`, on souhaiterait lever une exception `IllegalArgumentException` si les deux disques n'ont pas le même centre. Comment faut-il changer le code pour cela ?
7. On souhaiterait définir une forme `BitmapClip` définie par une matrice de booléens. Un point  $(x, y)$  est dans la forme si la case `[x][y]` de cette matrice est égal à `true`. Écrivez le code de la classe `BitmapClip`.
8. On souhaiterait avoir une classe `IntersectionClips` qui représente une intersection de `ClipInterface`. Cette classe devra implémenter l'interface `ClipInterface`. Le constructeur de la classe prendra en argument un tableau `ClipInterface[] clips`. Écrivez le code d'une telle classe.
9. On souhaiterait avoir maintenant une classe `UnionClips` qui représente une union de `ClipInterface`. Comment faire pour factoriser une partie du code des classes `ClipsIntersection` et `ClipsUnion` afin d'éviter la duplication de code ?
10. De quelle manière pourrait-on améliorer le code des classes que vous avez produit ?