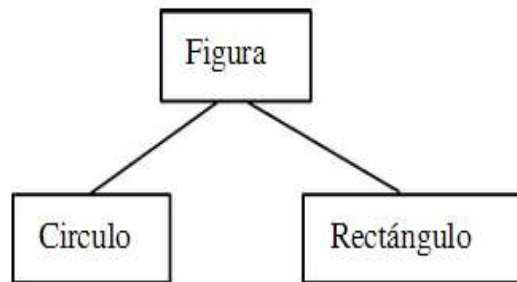


### Exemple d'herència : Figures geomètriques

En aquest primer exemple és pretén crear classes que representen diferents figures geomètriques bidimensionals. Nosaltres implementarem el cercle i el rectangle. Les figures geomètriques es caracteritzaran pel seu color i la seua posició. En els cercles, a més, s'estableix un radi; en els rectangles és necessari definir la longitud dels seus costats. L'estructura de les classes presenta el següent diagrama:



#### **CLASSE FIGURA**

```
1  import java.awt.Color;
3  public class Figura {
4      private Color ColorFigura;
5      private int[] Posicion = new int[2];
6
7      Figura() {
8          EstableceColor(Color.black);
9          Posicion[0] = 0;
10         Posicion[1] = 0;
11     }
12
13     Figura(Color color) {
14         EstableceColor(color);
15         Posicion[0] = 0;
16         Posicion[1] = 0;
17     }
18
19     Figura(Color color, int[] Posicion) {
20         EstableceColor(color);
21         EstableceCentro(Posicion);
22     }
23
24     public void EstableceColor(Color color) {
25         ColorFigura = color;
26     }
27
28     public Color DimeColor() {
29         return ColorFigura;
30     }
31
32     public void EstableceCentro(int[] Posicion) {
33         this.Posicion[0] = Posicion[0];
34         this.Posicion[1] = Posicion[1];
35     }
36
37     public int[] DimeCentro() {
38         return Posicion;
39     }
40 }
```

## CLASSE CERCLE

```
1    import java.awt.Color;
2
3    public class Circulo extends Figura {
4        private double Radio;
5
6        Circulo(double Radio) {
7            super();
8            EstableceRadio(Radio);
9        }
10       Circulo(double Radio, Color color) {
11           super(color);
12           EstableceRadio(Radio);
13       }
14
15       Circulo(double Radio, Color color, int[] Posicion) {
16           super(color, Posicion);
17           EstableceRadio(Radio);
18       }
19
20       public void EstableceRadio(double Radio) {
21           this.Radio = Radio;
22       }
23
24       public double DimeRadio() {
25           return Radio;
26       }
27     }
```

### **CLASSE RECTANGLE**

```
1  import java.awt.Color;
2
3  public class Rectangulo extends Figura {
4      private double[] Lados = new double[2];
5
6      Rectangulo(double[] Lados) {
7          EstableceLados(Lados);
8      }
9
10     Rectangulo(double[]Lados, Color color) {
11         super(color);
12         EstableceLados(Lados);
13     }
14
15     Rectangulo(double[] Lados, Color color, int[] Posicion) {
16         super(color, Posicion);
17         EstableceLados(Lados);
18     }
19
20     public void EstableceLados(double[] Lados) {
21         this.Lados[0] = Lados[0];
22         this.Lados[1] = Lados[1];
23     }
24
25     public double[] DimeLados() {
26         return Lados;
27     }
28 }
```

### **CLASSE PROVA FIGURES**

```
1    import java.awt.Color;
2
3    public class PruebaFiguras {
4        public static void main(String[] args) {
5            int[] Posicion = {10,20};
6            double[] Lados = {50d,100d};
7            Circulo MiCirculo = new Circulo(3d,Color.red,Posicion);
8            Rectangulo MiRectangulo = new Rectangulo(Lados,Color.blue,Posicion);
9
10           System.out.println(MiCirculo.DimeRadio());
11           int[] Centro = MiCirculo.DimeCentro();
12           System.out.println(Centro[0]);
13           System.out.println(Centro[1]);
14
15           System.out.println(MiRectangulo.DimeColor());
16
17       }
18 }
```