Java implanta otro mecanismo que resulta parecido al de herencia múltiple que es el de las interfaces que se verá más adelante.

Clases Abstractas

En ocasiones definimos clases de las que no pretendemos crear objetos, su único objetivo es que sirvan de superclases a las clases “reales” Por ejemplo:

- nunca crearemos objetos de la clase Figura - lo haremos de sus subclases Círculo, Cuadrado, ...
- nunca crearemos un Vehículo - crearemos un Coche, un Barco, un Avión, ...

La razón es que no existen “figuras” o “vehículos” genéricos, ambos conceptos son abstracciones de los objetos reales, tales como círculos, cuadrados, coches o aviones, a ese tipo de clases las denominaremos **clases abstractas.**

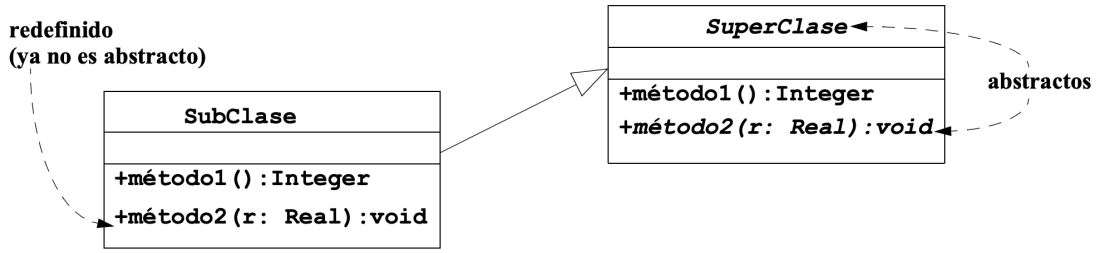
Una clase abstracta puede tener **métodos abstractos,** se trata de métodos sin cuerpo que es obligatorio redefinir en las subclases no abstractas, estas permiten declarar en la superclase un comportamiento que deberán verificar todas sus subclases, pero sin decir nada sobre su implementación.

Ejemplo de método abstracto en Java:

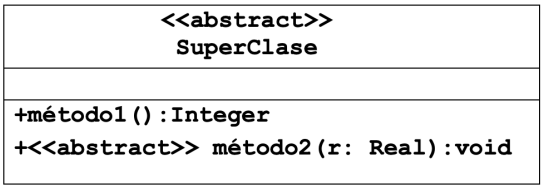
```
public abstract int métodoAbstracto(double d);
```

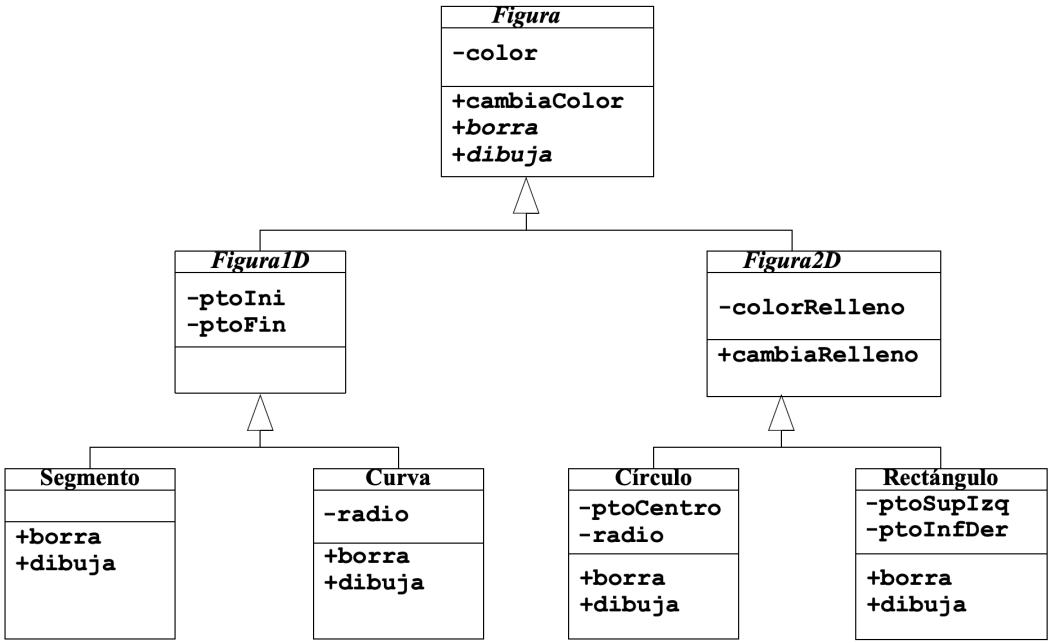
Observe que el método no tiene cuerpo.

Las clases y los métodos abstractos se escriben en cursiva



También puede utilizarse el estereotipo <<abstract>>





Las clases abstractas en Java se identifican mediante la palabra reservada abstract

```
public abstract class Figura { ... }
```

Es un error tratar de crear un objeto de una clase abstracta

```
Figura f = new Figura(...);
```

Pero NO es un error utilizar referencias a clases abstractas, que pueden apuntar a objetos de cualquiera de sus subclases

```
Figura f1 = new Círculo(...); // correcto

Figura f2 = new Cuadrado(...); // correcto
```

Interfaces

En java, las interfaces se declaran con la palabra reservda interface de manera similar a como se declaran las clases abstractas.

En la declaración de una ointerfaz, lo unico que puede aparecer son declaraciones de metodos (sin implementación) y definición de constantes simbolicas.

Una interfaz no encapsula datos, solo define cuales son los metodos que han de implementar lo objetos de aquellas cclases que implementen la interfaz.

En java, para indicar que una clase emplementa una interfaz se utiliza la palabra reservada implements.

La clase debe entonces emplementar todos los metodos definidos por la interfaz o declararse, a su vez como una clae abstracta.

```
public class Circulo implements Figura
{
  private double radio;

  public Circulo (double radio)
  {
    this.radio = radio;
  }

  public double area ()
  {
    return Math.PI*radio*radio;
  }
}


public class Cuadrado implements Figura
{
  private double lado;

  public Cuadrado (double lado)
  {
    this.lado = lado;
  }

  public double area ()
  {
    return lado*lado;
  }
}


public abstract class Figura
{
  public abstract double area ();
}

public interface Dibujable
{
  public void dibujar ();
}

public interface Rotable
{
  public void rotar (double grados);
}

public class Circulo extends Figura
                    implements Dibujable
…

public class Cuadrado extends Figura
                     implements Dibujable, Rotable
…
```