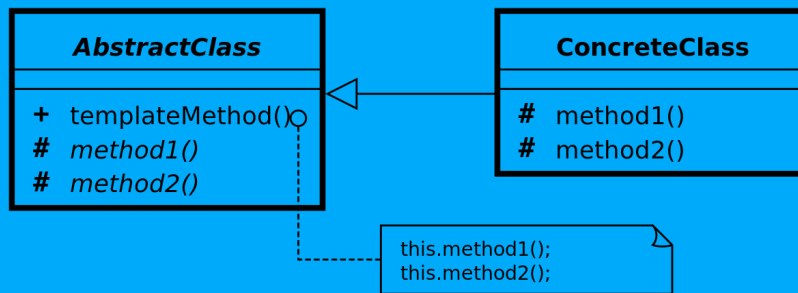


Java

## //Clases abstractas



## Clases Abstractas

Se dice que son clases abstractas aquellas clases base o padre (superclases) de las que no se permite la creación de objetos. Para ello, se utiliza la palabra reservada **abstract**. Es como un molde que utilizan otras clases, para heredar sus atributos y métodos. Como dice la frase anterior no se puede instanciar o sea, no se puede usar new para crear un objeto. Las clases hijas si lo pueden hacer.

En una clase abstracta es posible definir métodos abstractos. Para eso debemos colocar el modificador **abstract** y debe usarse solo en métodos de instancia. No se puede aplicar este modificador a métodos estáticos ni a constructores.

Una clase que contenga uno o más métodos abstractos también debe declararse como abstracta precediendo su declaración de clase con el modificador abstract, los cuales se caracterizan por el hecho de que no pueden ser implementados en la clase base o padre. De ellos, solo se escribe su firma (la firma de los métodos), en la superclase, y su funcionalidad –polimórfica– tiene que indicarse en las clases hijas (subclases).

Veamos un ejemplo:

Dadas las siguientes clases, Figura, Triángulo y Cuadrado.

```
Java
public abstract class Figura {
    private String color;
    public Figura(String color){
        this.color = color;
    }

    public abstract double calcularArea();

    public String getColor(){
        return color;
    }
}
```

En la clase Figura se define un atributo de tipo String llamado color, con un constructor y dos métodos uno abstracto calcularArea, que vemos que le colocamos la palabra reservada abstrac para indicar que este método es abstracto y se debe implementar en la clase hija.

Tiene otro método, el getColor para obtener los datos del atributo de la clase (un getter)

```
Java
public class Cuadrado extends Figura{
    private double lado;

    public Cuadrado(String color, double lado) {
        super(color);
        this.lado = lado;
    }

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

public class Triangulo extends Figura{
    private double base;
    private double altura;

    public Triangulo(String color, double base, double altura) {
        super(color);
        this.base = base;
        this.altura = altura;
    }

    public double calcularArea() {
        return base*altura/2;
    }
}
```

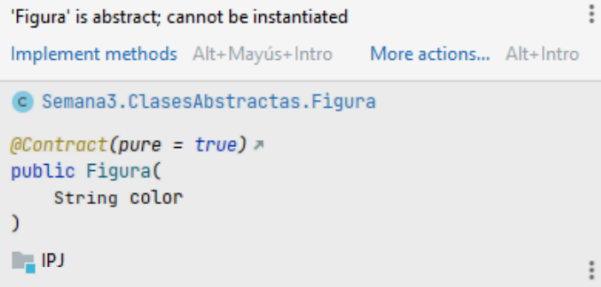
Si se fijan en estas clases que son hijas se le agrega la palabra reservada extends y luego el nombre de la clase de cual hereda en este caso Figura. Una de las cosas que no se permite en Java es la herencia múltiple. También si notan que el método abstracto de la clase Figura está implementado en las clases hijas Cuadrado y Triangulo, de acuerdo a cada necesidad. Se adecua a cada Clase este método.

Si queremos hacer el test de cómo funciona en el main(), podemos hacer la siguiente prueba:

Java

```
public class Principal {
    public static void main(String[] args){
        //Probando el método abstracto

        Figura f1 = new Figura("rojo");
    }
}
```



Que no nos deja instanciar la clase abstracta Figura, por lo que comentamos al principio, esta clase no se puede instanciar, solo sus hijas o subclases.

Java

```
import java.util.Scanner;
public class Principal {
    public static void main(String[] args){
        //Probando el método abstracto
        String color="color";
        double lado,base,altura;
        Scanner input = new Scanner(System.in);
        System.out.println("Introduzca lado del cuadrado :");
        lado=input.nextDouble();
        System.out.println("Introduzca la base del triángulo :");
        base=input.nextDouble();
        System.out.println("Introduzca la altura del triángulo :");
        altura=input.nextDouble();
        Cuadrado c1 = new Cuadrado(color, lado);
        Triangulo t1 = new Triangulo("verde", base, altura);
        System.out.printf("Área cuadrado %s es: %.2f\n", c1.getColor(), c1.calcularArea());
        System.out.printf("Área triángulo %s es: %.2f\n", t1.getColor(), t1.calcularArea());
    }
}
```

Ingresando los datos por consola, vemos la siguiente salida:

Java

```
Introduzca lado del cuadrado:
3,3
Introduzca la base del triángulo:
5
Introduzca altura del triángulo:
2
Área cuadrado color es: 10,89
Área triángulo verde es: 5,00
```

Ahí vemos cómo utilizamos la instancia de cada clase, y hacen sus cálculos de área respectivos, utilizando el método `calcularArea()` de acuerdo a la clase que lo utiliza. Ven el `printf` es una forma diferente de indicar la salida por consola, el `%s` indica que va un `String`, el `%f` que va un dato del tipo `float` o `double`, el `.2f` indica los decimales que queremos ver, y el `%n` al final indica que queremos que haga un salto de línea.



Video de ejemplo