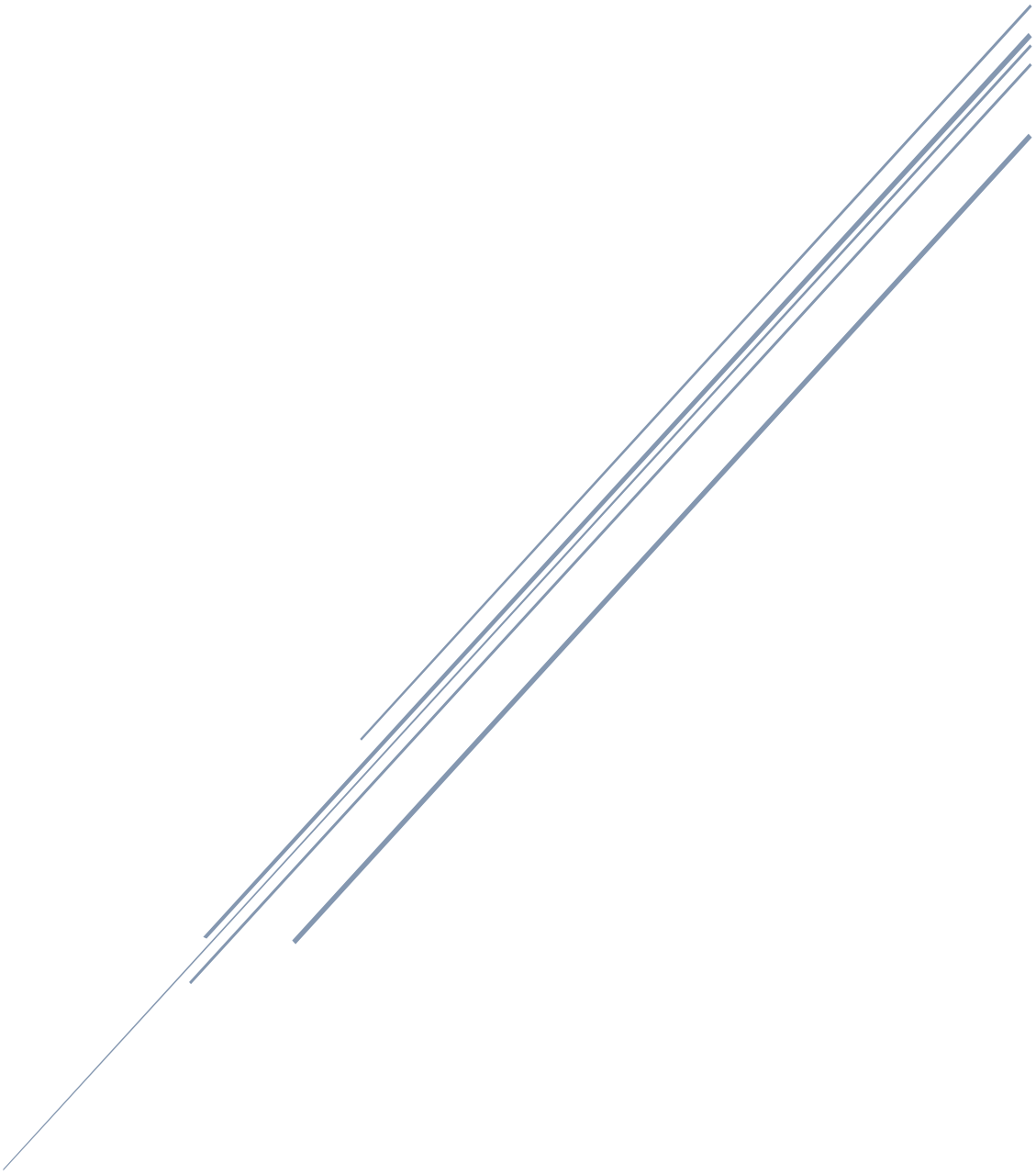
# CLASES ABSTRACTAS

## PROGRAMACIÓN



# Clases abstractas e interfaces

## Clases abstractas

Una clase abstracta...

Es una clase que no se puede instanciar se usa únicamente

para definir subclases

*¿Cuándo es una clase abstracta?*

En cuanto uno de sus métodos no tiene implementación (en Java, el método abstracto se etiqueta con la palabra reservada abstract).

*¿Cuándo se utilizan clases abstractas?*

Cuando deseamos definir una abstracción que englobe objetos de distintos tipos y queremos hacer uso del polimorfismo.

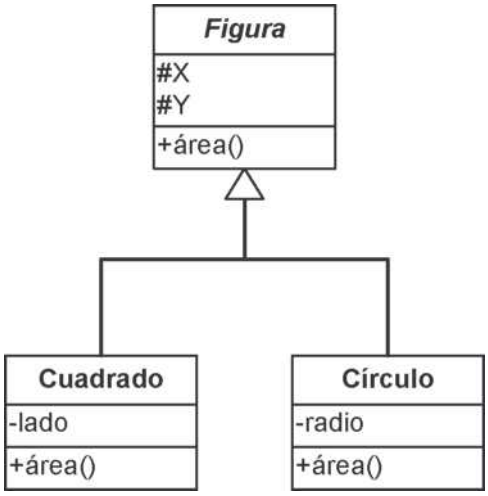



Figura es una clase abstracta (nombre en cursiva en UML) porque no tiene sentido calcular su área, pero sí la de un cuadrado o un círculo. Si una subclase de Figura no redefine area(), deberá declararse también como clase abstracta.

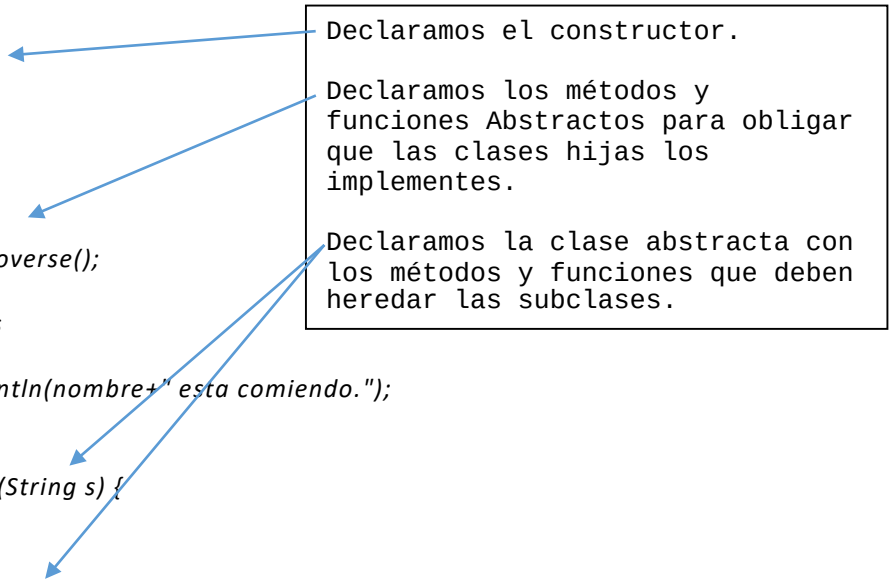
## Ejemplo 1:

```
public abstract class Figura {
    protected double x;
    protected double y;
    public Figura (double x, double y) {
        this.x = x; this.y = y;
    }
    public abstract double area () ;
}

public class Circulo extends Figura {

    private double radio;

    public Circulo (double x, double y, double radio) {
        super(x,y); this.radio = radio;
```

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

public class Cuadrado extends Figura {

    prívate double lado; public Cuadrado (double x, double y, double lado) {
        super(x,y); this.lado = lado;
    }
    public double area () {
        return lado*lado;
    }
}
```

## Ejemplo 2:

```
//Ejercicio  Interfaces y clases Abtractas – Archivo Animales.java
public abstract class Animales{   //declaramos la clase Abstracta
        //Declaracones
        private String nombre;

        //Constructor
        public Animales() {
                //nombre = s;
        }

        //Metodo Abstracto
        public abstract void moverse();

        //Metodos y Funciones
        public void comer() {
                System.out.println(nombre+" esta comiendo.");
        }

        public void setNombre(String s) {
                nombre = s;
        }

        public String getNombre() {
                return nombre;
        }
}
```



```
//Ejercicio  Interfaces y clases Abtractas – Archivo Ave.java
public class Ave extends Animales{

        public Ave(){
                super();
                setNombre("AveLoro");
        }

        public void moverse() {
                System.out.println("El ave esta volando");
        }

}
```

Declaramos el constructor.

Declaramos la clase abstracta que viene de su clase padre y la implementamos.

```
//Ejercicio  Interfaces y clases Abtractas – Archivo Marinos.java
public class Marinos extends Animales {

        public Marinos(String s){
                super();
                setNombre(s);
        }

        public void moverse() {
                System.out.println("El pescadito esta nadando");
        }

}
```

Declaramos el constructor.

Declaramos la clase abstracta que viene de su clase padre y la implementamos.

```
//Ejercicio  Interfaces y clases Abtractas – Archivo Terrestres.java
public class Terrestres extends Animales {

        public Terrestres(String s){
                super();
                setNombre(s);
        }

        public void moverse() {
                System.out.println("El animalito esta caminando");
        }

}
```

Declaramos el constructor.

Declaramos la clase abstracta que viene de su clase padre y la implementamos.