

# CLASE ABSTRACTA

TEMA:  
**CLASE ABSTRACTA**  
Se basa en la herencia

MÉTODOS  
ABSTRACTOS

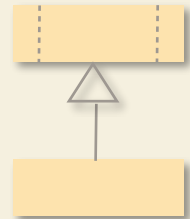
ATRIBUTOS Y  
MÉTODOS

Agosto, 2018

Guadalajara, Jalisco. México

Se utiliza en la herencia, sirve como un molde

Simbología



## CLASE ABSTRACTA

### UNA CLASE ABSTRACTA ES

Una clase que no tiene instancias directas, pero sus descendientes sí poseen instancias directas. Hace más fácil extender y modificar programas que requieren diferentes representaciones.

### CARACTERÍSTICAS

- Se define para que el resto de las clases puedan ampliarlas y hacerlas concretas implementando los métodos abstractos.
- Esta clase no se utiliza para crear objetos.
- Es un molde o modelo y se utiliza con la herencia

### ELEMENTOS DE LA CLASE ABSTRACTA

La clase abstracta puede tener declarados atributos, métodos y debe tener por lo menos un método abstracto.

### MÉTODO ABSTRACTO

Es un método que no tienen ninguna implementación en la clase donde se declara (clase abstracta) sino que se implementa (define) hasta las subclases.



### Sintaxis:

```
abstract class <IdAbstracta> {  
    <atributos>  
    <métodos>  
    <métodos abstractos>  
}  
class <subClase> extends <IdAbstracta> {  
    <atributos>  
    <métodos>  
}
```



# Aplicación de la clase abstracta en un problema de programación

## DETERMINAR EL ÁREA PARA DIFERENTES FIGURAS GEOMÉTRICAS

Para aplicar la metodología orientada a objetos en un problema es necesario realizar varios pasos para llegar a la solución del problema.

Es importante empezar a visualizar el problema desde el punto de vista de esta metodología.

Supongamos que queremos desplegar en pantalla las coordenadas en x y y donde inicia un triángulo o un cuadro o un círculo así como su área.

**Paso1.** Identificar los objetos dentro del problema: “desplegar en **pantalla** las coordenadas en x y y donde inicia un **triángulo** o un **cuadro** o un **cículo** así como su área”. Para este caso tenemos los objetos: pantalla, usuario, triángulo, cuadro, círculo.

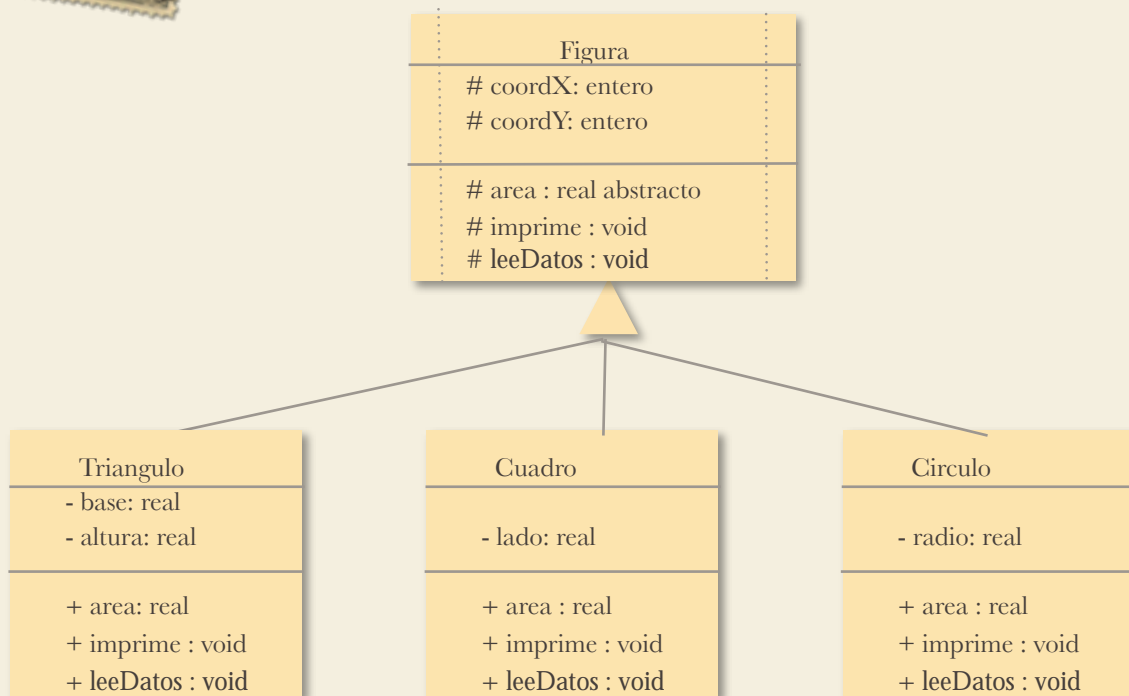
**Paso 2.** Una vez identificados los objetos tenemos que analizar cuales vamos a programar, para ello podemos hacernos la siguiente pregunta ¿Necesito programar algo en este objeto para solucionar el problema?, en el caso de la pantalla y para el usuario la respuesta es no, pero en el caso del triángulo, círculo y rectángulo la respuesta es sí. Por lo que tenemos 3 objetos a programar.

**Paso 3.** Después de saber qué objetos vamos a programar entonces tenemos que encapsularlos en una clase y hacer el proceso de abstracción para cada clase. Para esto necesitamos saber la representación gráfica de las clases para iniciar el diseño en un diagrama y que sea más sencillo realizar este paso.

Diseñaremos cada clase en el símbolo correspondiente a la clase y después veremos que relaciones hay entre ellas para también representarlas en el diagrama.

**Paso 4.** Por último se necesita pasar el diagrama que nos resultó en el diseño al código siguiendo la sintaxis correspondiente en el lenguaje que vayamos a trabajar (que utilice la metodología orientada a objeto) en nuestro caso será java





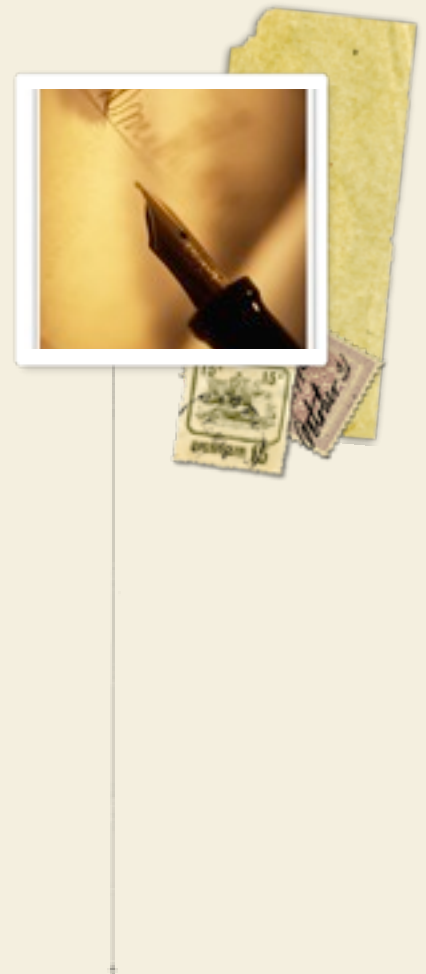
**abstract** class Figura

```

{
    protected int coordX ;
    protected int coordY;
    protected abstract double area();
    protected void imprime() {
        System.out.println("COORDENADAS DE LA FIGURA");
        System.out.println("Coordenada en X= " + coordX );
        System.out.println("Coordenada en Y= " + coordY );
    }
    protected void leeDatos () {
        System.out.print ("Escribe el numero de pixel de la coordenada X: ");
        coordX = EntradaDatos.entero();
        System.out.print ("Escribe el numero de pixel de la coordenada Y: ");
        coordY = EntradaDatos.entero();
    }
}
  
```

```

class Triangulo extends Figura
    private double base;
    private double altura;
    public double area() {
        return (base*altura)/2;
    }
    public void imprime() {
        super.imprime();
        System.out.println("Area Triangulo= " + area());
    }
}
  
```



```

        public void leeDatos () {
            super.leeDatos();
            System.out.print ("Escribe la base del triangulo : ");
            base = EntradaDatos.real \(\); //se usa el archivo EntradaDatos.java
            System.out.print ("Escribe la altura del triangulo : ");
            altura = EntradaDatos.real \(\);
        }
    }

class Cuadro extends Figura {
    private double lado;
    public double area() {
        return lado*lado;
    }
    protected void imprime() {
        super.imprime();
        System.out.println("Area del cuadro = " + area());
    }
    public void leeDatos () {
        super.leeDatos();
        System.out.print ("Escribe el lado del cuadro : ");
        lado = EntradaDatos.real \(\);
    }
}

class Circulo extends Figura {
    private double radio;
    public double area() {
        return 3.1416*(radio*radio);
    }
    public void imprime() {
        super.imprime();
        System.out.println("Area del circulo = " + area());
    }
    public void leeDatos () {
        super.leeDatos();
        System.out.print ("Escribe el radio del circulo : ");
        radio = EntradaDatos.real \(\);
    }
}

```

**Nota:** En este ejemplo se está usando el archivo [EntradaDatos.java](#), lo puedes encontrar en el archivo **U2-a0 DatosDeEntrada.pdf** en la **diapositiva 21**

```

class AppClaseAbstracta {
    public static void main ( String arg[ ] ) {
        Triangulo fig1 = new Triangulo ( );
        Cuadro fig2 = new Cuadro ( );
        Circulo fig3 = new Circulo ( );

        fig1.leeDatos();
        fig1.imprime();

        fig2.leeDatos();
        fig2.imprime();

        fig3.leeDatos();
        fig3.imprime();
    }
}

```

## EJERCICIO PARA LA ACTIVIDAD 3

### I. PROBLEMÁTICA

- Se quiere imprimir en pantalla el importe a pagar por un vehículo que circula por una autopista. El vehículo puede ser una moto, un auto o un camión. Los datos que el programa leerá para cualquiera de los tres vehículos son las placas y el proceso será para calcular el importe. Este importe se calculará de forma diferente para cada vehículo.
- Para el vehículo moto el importe a pagar será de \$74.00 + IVA y se cobrará el 2% más si no trae casco el conductor, en pantalla se imprimirá todos los datos por separado y el total a pagar. También tendrá otro proceso para imprimir en pantalla si el conductor trae casco o no en un mensaje.
- Para el vehículo auto el importe a pagar será de \$107.50 + IVA y se imprimirá en pantalla todos los datos por separado y el total a pagar.
- Para el vehículo camión el importe a pagar será de \$125.00 + \$15.50 por cada tonelada + IVA y se imprimirá en pantalla todos los datos por separado y el total a pagar
- El programa debe preguntar a través de un menú si se va pagar el importe de una moto, un auto o un camión. También se debe considerar que al final del programa se pregunte si quiere salir del programa en caso de que la respuesta sea no el programa debe regresar al menú.