

## PRÁCTICA I: Clases, Herencia y Polimorfismo en Java

**OBJETIVOS:** Aplicar los conceptos de herencia y polimorfismo en JAVA.

### TEMPORIZACIÓN:

**Publicación del Enunciado:** 04 de Abril de 2022.

**Entrega:** 08 de Abril de 2022.

---

### Introducción

El objetivo general de esta práctica es repasar el uso de herencia en Java y su aplicación a arrays de objetos. La herencia la basaremos en el uso de *abstract* y *extends*, con las que crearemos una clase genérica y varias específicas que deriven de ella. Se implementarán sus métodos abstractos y se sobrescribirán otros que permitirán demostrar el polimorfismo a la hora de usar los objetos de la jerarquía de clases. Por último, usaremos un array de objetos de la anterior jerarquía que permita demostrar todo lo anterior.

En particular, en esta práctica se va a calcular el área total de un conjunto de figuras que hayamos creado previamente.

### Ejercicio 1.- Herencia. Las clases *Figura* y *Triangulo*

La clase *Figura* representa una figura genérica que se materializará posteriormente en una específica (*Triangulo*, *Cuadrado*, etc.). En Java, esto lo representamos mediante las clases abstractas que se representan mediante la palabra reservada ***abstract***.

**Nota:** Una clase abstracta es la que declara uno o más métodos abstractos cuya implementación se realiza en las clases derivadas. El primer ejercicio de la práctica consiste en definir la siguiente clase abstracta *Figura*, que representa una figura genérica:

```
public abstract class Figura
{
    String nombre;

    public Figura(String nombre);

    public abstract double calculaArea();
    public String getNombre();
    public abstract String toString();
}
```

**Nota:** Observa que se han declarado dos métodos abstractos que se implementarán en las clases derivadas correspondientes. Y un método propio de la clase *Figura*.

### Apartado 1: Implementar el constructor de *Figura*

Implementa el constructor de la clase. El valor por defecto de nombre es "figura". Recuerda que un atributo puede inicializarse en más sitios que el constructor.

Piensa en una cosa: ¿qué sentido tiene implementar el constructor de una clase abstracta?

### Apartado 2: Implementar la clase *Triangulo* para que herede de *Figura*

La clase *Triangulo* ahora deriva de *Figura* y debes de tener en cuenta los siguientes aspectos:

- Declaración de la clase *Triangulo* como derivada de *Figura*.
- Constructores: modifica los constructores para que en la primera línea llamen al constructor de *Figura* mediante la referencia **super**. La clase *triangulo* debe recibir como parámetros, la base y la altura.
- Implementa los siguientes métodos que se especifican a continuación:

```
public double calculaArea()
{
    /* Devuelve el área de un triángulo: Area = ( base X altura ) / 2 */
}

public String toString()
{
    /* Devuelve una cadena con el nombre del triangulo y el valor del área. */
}
```

### Apartado 3: Implementar el método *main* de *Triangulo*.

Ahora debes crear un método para probar el código anterior. Crea el método *main* en la clase *Test* que haga lo siguiente:

1. Crea un objeto de la clase *triangulo*.
2. Imprimir por pantalla una cadena descriptiva con los valores del triángulo.
3. Por último, imprime el valor del área del triángulo.

## Ejercicio 2.- Herencia. La clase Cuadrado

La clase *Cuadrado* hereda también de la clase *Figura* y debe de definir los mismos métodos que la clase *triangulo*. Este tiene como dato miembro aparte del nombre, el lado.

**Nota:** Observa también que, al derivar de una clase abstracta como *Figura*, la clase *Cuadrado* está obligada a implementar TODOS sus métodos abstractos... ¿cuáles son?

### Apartado 3: el método *main* de *Cuadrado*

Ahora debes crear un método para probar el código anterior. Crea el método *main* en la clase *Test* que haga lo siguiente:

1. Crear un cuadrado.
2. Imprimir por pantalla una cadena descriptiva con los valores del cuadrado.
3. Por último, imprime el valor del área del cuadrado.

## Ejercicio 3.- Arrays. Cálculo del área de varias figuras

La clase *AreaFiguras*, que contiene el array de figuras con el que trabajaremos en este ejercicio, es la que nos va a permitir calcular el área total de un conjunto de figuras. Tiene el siguiente aspecto:

```
public class AreaFiguras
{
    Figura figuras[] = null;

    public AreaFiguras( int numFiguras ) { /* .... */ }
    public void añadeFigura ( Figura f ) { /* .... */ }
    public double calculaAreaTotal() { /* .... */ }
    public void imprime() { /* .... */ }

    public static void main( String args[] ) { /* .... */ }
}
```

El constructor de la clase inicializa el array con el número máximo de figuras. El método *añadeFigura* permite añadir un objeto *Figura* en el primer hueco libre del array. El método *calculaAreaTotal* es el que irá sumando las áreas de las figuras que contiene el array y devolverá la suma total de ellas. El método *imprime* saca por pantalla el área de cada uno de los elementos de array así como el área total.

### **Apartado 3: el método *main* de *AreaFiguras***

Crea el programa principal que tiene que realizar lo siguiente:

1. Crear un objeto *AreaFiguras* con dos figuras como máximo.
2. Crea un triángulo y un cuadrado y añádelos al array utilizando los métodos antes definidos.
3. Imprime el resultado del cálculo del área total de las figuras. Para ello, invoca al método correspondiente de la clase.