

# DESARROLLO PROFESIONAL DE APPS PARA ANDROID

Gerardo Andres Gutiérrez Rodríguez

# NORMAS PARA ESCRIBIR CÓDIGO EN JAVA

- Indentación de código para facilitar lectura.
- Usar CamelCase como estándar para la codificación.
- Documentación de código usando comentarios

Indentación: Anglismo que significa mover un bloque de código hacia la derecha insertando espacios en blanco o tabulaciones para separar del margen izquierdo generando un sangrado en el texto.

CamelCase: No espacios, primeras letras de cada palabra en mayúscula, si es clase inicia en mayúscula, si es variable o método inicia con minúscula la primera palabra, si es un valor constante se escribe en mayúscula

Documentar el código de un programa es añadir suficiente información como para explicar lo que hace, punto por punto, de forma que no sólo los ordenadores sepan qué hacer, sino que además los humanos entiendan qué están haciendo y por qué.

Porque usar normas de codificación:

- El 80% del coste del código de un programa va a su mantenimiento.
- Casi ningún software lo mantiene toda su vida el autor original.
- Las convenciones de código mejoran la lectura del software, permitiendo entender código nuevo mucho más rápidamente y más a fondo
- Si distribuyes tu código fuente como un producto, necesitas asegurarte de que esta bien hecho y presentado como cualquier otro producto.

# PROGRAMACIÓN ORIENTADA OBJETOS USANDO JAVA

La programación orientada a objetos o POO (OOP según sus siglas en inglés) es un paradigma de programación que usa objetos para diseñar aplicaciones y software.

# ¿QUE ES PROGRAMACIÓN?

- La programación es el proceso de diseñar, codificar, depurar y mantener el código fuente de programas computacionales.
- El código fuente es escrito usando un lenguaje de programación de alto nivel.
- El propósito de la programación es crear programas que realicen un comportamiento deseado.
- El proceso de escribir código requiere conocimientos en algoritmos y lógica formal.
- De este proceso se genera software que tambien se define como un conjunto de programas.

# ¿QUE ES UN OBJETO?

- Es un elemento de software que cuenta con un estado y un comportamiento.
- El estado de un objeto esta dado por diferentes variables y su comportamiento por diversas acciones.
- También a un objeto se le conoce como instancia de una clase.
- Un objeto es capas de interactuar con otros objetos

# ¿QUE ES UNA CLASE?

- Es una plantilla para la creación de objetos según un modelo predefinido.
- Se utilizan para representar entidades.
- Cada clase es un modelo que define un conjunto de variables y métodos apropiados para el comportamiento del modelo.

# DEFINIR UNA CLASE

```
// Paquete al que pertenece la clase
package com.vivelabhuila.ejemplo;

// Elementos a usar
import java.lang.Integer;

// Definición de clase
public class Main {

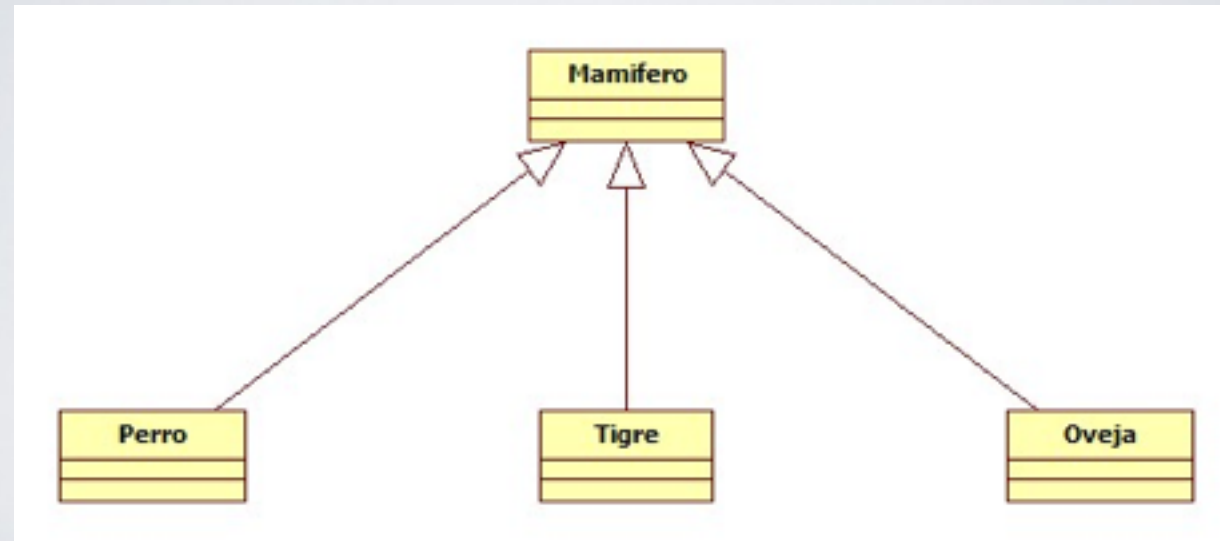
    // Atributos o Propiedades
    private Integer id;

    // Constructor
    public Main(){
        id = new Integer(0);
    } // Cierre de constructor

    // Métodos de clase
    public Integer getId(){
        return this.id;
    } // Cierre de método

} // Cierre de clase
```

# QUE ES HERENCIA



Es la relación entre una clase general y otra clase más específica. Por ejemplo: Si declaramos una clase párrafo derivada de una clase texto, todos los métodos y variables asociadas con la clase texto, son automáticamente heredados por la subclase párrafo.



# QUE ES POLIMORFISMO

```
class Figura {  
    float alto;  
    float ancho;  
}
```




Diagram illustrating inheritance: Two arrows point from the Rectangulo and Triangulo class definitions below to the Figura class definition above, indicating that both Rectangulo and Triangulo inherit from the Figura base class.

```
class Rectangulo extends Figura {  
    public float area(){  
        return alto * ancho;  
    }  
}
```

```
class Triangulo extends Figura {  
    public float area(){  
        return (alto * ancho) / 2;  
    }  
}
```

# ENCAPSULAMIENTO

```
class Persona {  
  
    private double cedula;  
    private String nombre;  
  
    public String getNombre() {  
        return this.nombre;  
    }  
  
    public void setNombre(String nombre) {  
        this.nombre = nombre;  
    }  
  
}
```

Encapsulamiento es el ocultamiento del estado de una clase a las clases que acceden a las propiedades de la misma.



¡Manos a la moto!