

# CLASE DE JAVA

Una clase de Java es una plantilla donde se definen los atributos y los métodos para los objetos que se van a crear a partir de esa clase.

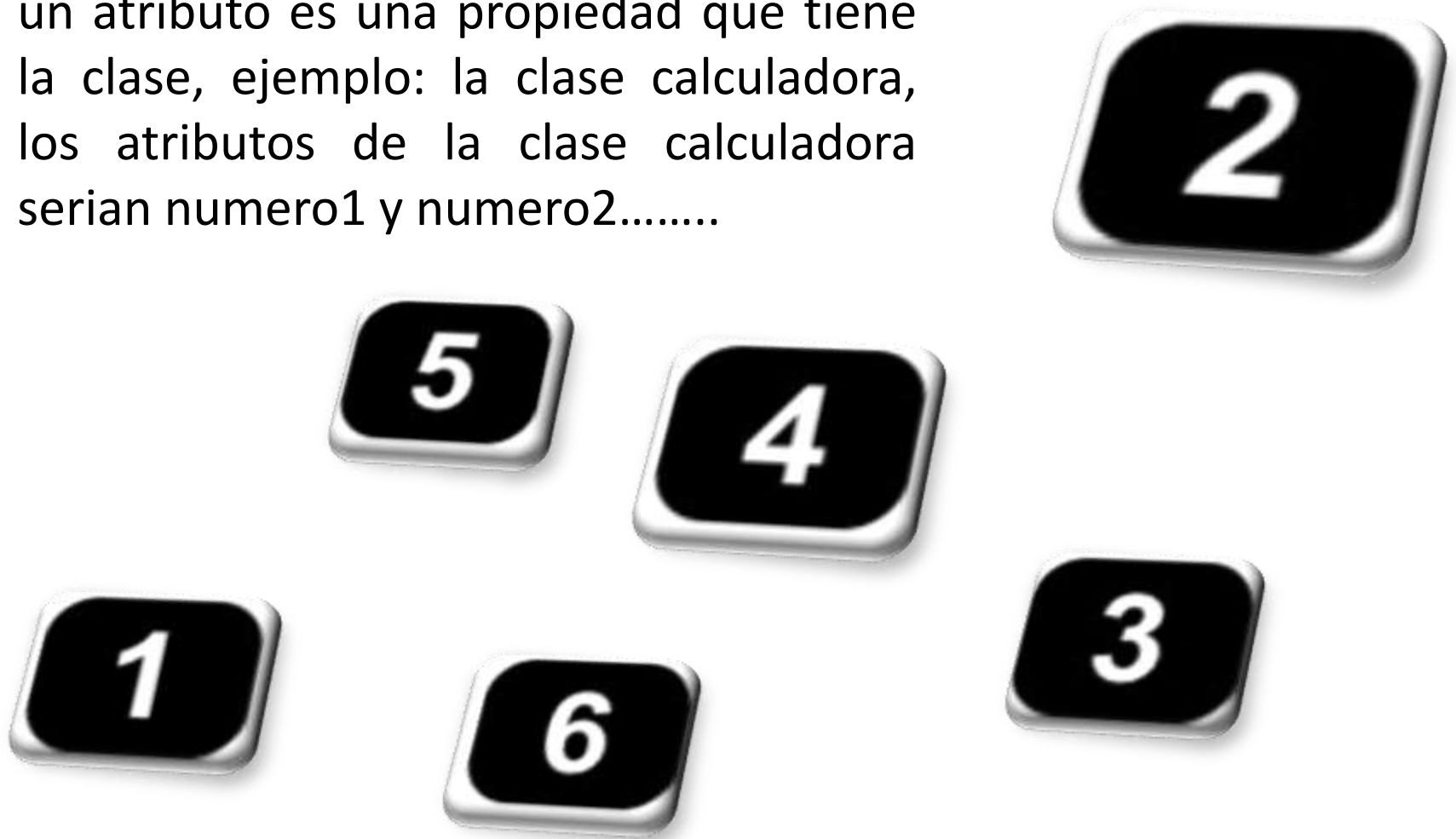


# Clase calculadora



# 1) ¿QUE ES UN ATRIBUTO?

un atributo es una propiedad que tiene la clase, ejemplo: la clase calculadora, los atributos de la clase calculadora serian numero1 y numero2.....



## 2) ¿QUE ES UN METODO?

Un método define una función de la clase, lo que hace una clase, ejemplo los métodos que podría tener la clase calculadora serian sumar, restar, multiplicar y dividir. Sencillamente los métodos definen las tareas o las acciones de la clase.



# ENCAPSULAMIENTO

También conocido como ocultación, permite cambiar la visibilidad de los elementos miembros de la clase, tanto para atributos como para métodos. De forma que los elementos miembro de la clase pueden tener tres niveles así:



```
import com.lauchenauer.istockhelper.  
import com.lauchenauer.lib.ui.Vertic  
import com.lauchenauer.lib.util.Brow  
  
public class AboutDialog extends JDia  
    protected CardLayout mLayout;  
    protected JButton mCredits;  
    protected JPanel mMainPanel;  
  
    public AboutDialog(JFrame owner) {  
        super(owner);  
        setModal(true);  
        setUndecorated(true);  
        initUI();  
    }  
  
    protected void initUI() {  
        setSize(440, 600);  
        Container cont = getContentPane  
        JPanel p =
```

## **Privados:**

Son los elementos que solo pueden ser accedidos directamente por la clase que los define.

En java la palabra reservada es **private**.

## **Protegidos:**

Los elementos protegidos son aquellos que pueden ser accedidos por las clases descendientes o clases que compartan el mismo espacio físico "paquete"

En java la palabra reservada es **protected**

## **Públicos:**

Estos son los elementos en los cuales no hay restricción alguna y pueden ser accedidos por cualquier clase y objeto del modelo.

En java la palabra reservada es **public**

## **El encapsulamiento es interno y externo.**

Para el interno manejamos los elementos privados y los protegidos.

Para el externo manejamos los elementos públicos.

## **Una analogía propuesta para entender el concepto:**

Piense en las cosas personales que no se pueden contar y eso debe ser privado nadie lo puede conocer.

Piense en las cosas que en la familia se sabe pero no se puede enterar nadie fuera de la familia esto debe ser protegido.

Lo que es de mostrar es todo lo bueno eso es lo público.

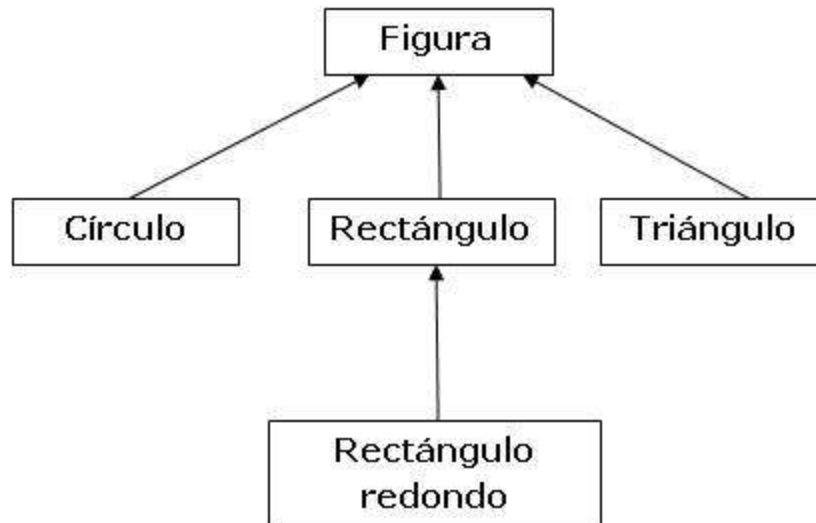
### Modificadores de acceso

	<b>La misma clase</b>	<b>Otra clase del mismo paquete</b>	<b>Subclase de otro paquete</b>	<b>Otra clase de otro paquete</b>
public	X	X	X	X
protected	X	X	X	
private	X			



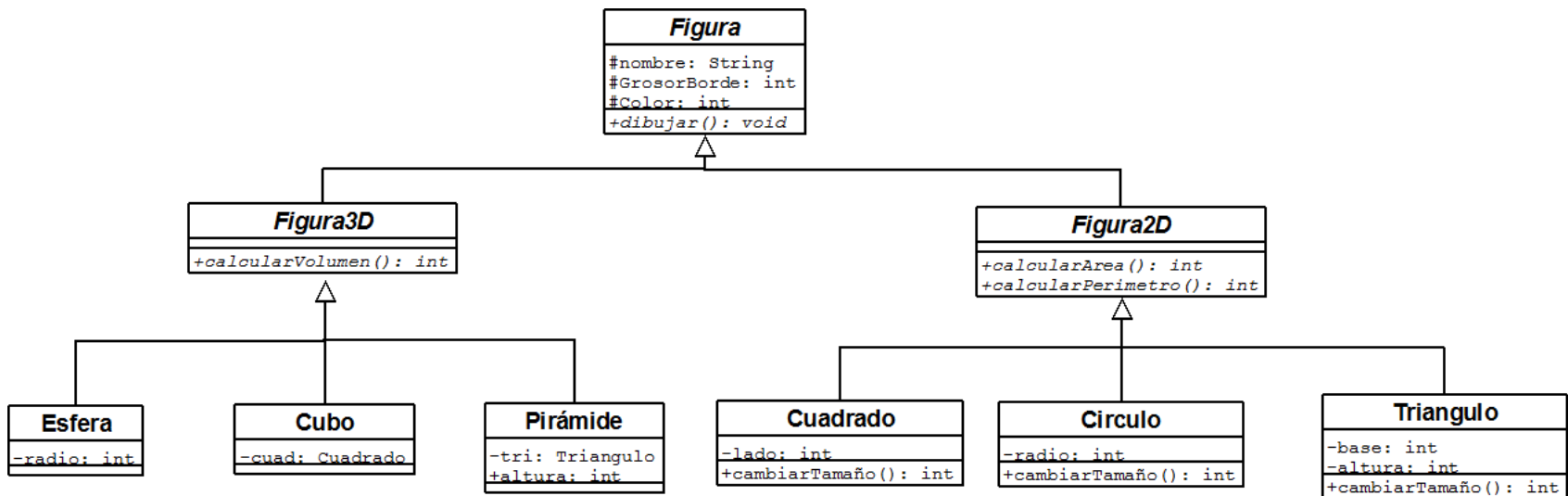
# Herencia

La idea de la herencia es permitir la creación de nuevas clases basadas en clases existentes. Cuando heredamos de una clase existente, rehusamos (o heredamos) métodos y campos, y agregamos nuevos campos y métodos para cumplir con la situación nueva.



# Polimorfismo

En general nos sirve para programar objetos con características comunes y que todos estos compartan la misma superclase en una jerarquía de clases, como si todas fueran objetos de la superclase. Esto nos simplifica la programación.

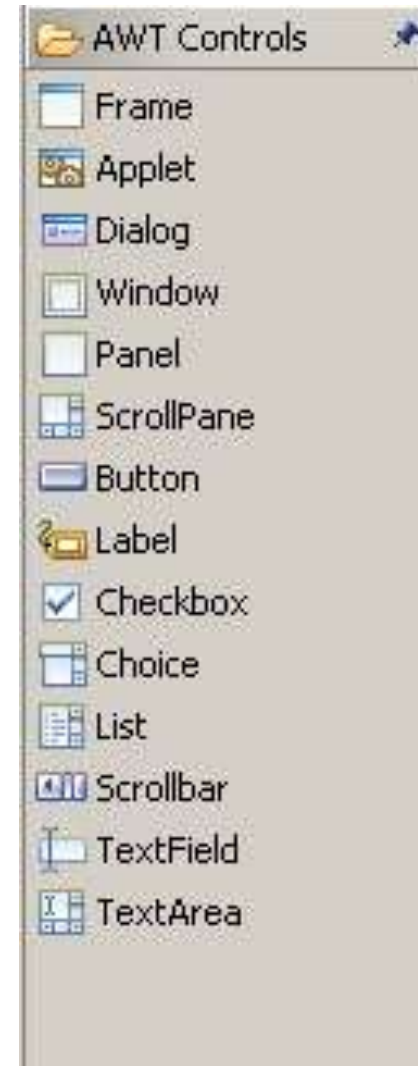


# AWT

## (caja de herramientas de las ventanas)



Todos los componentes que podemos visualizar en pantallas son subclases de clases componentes

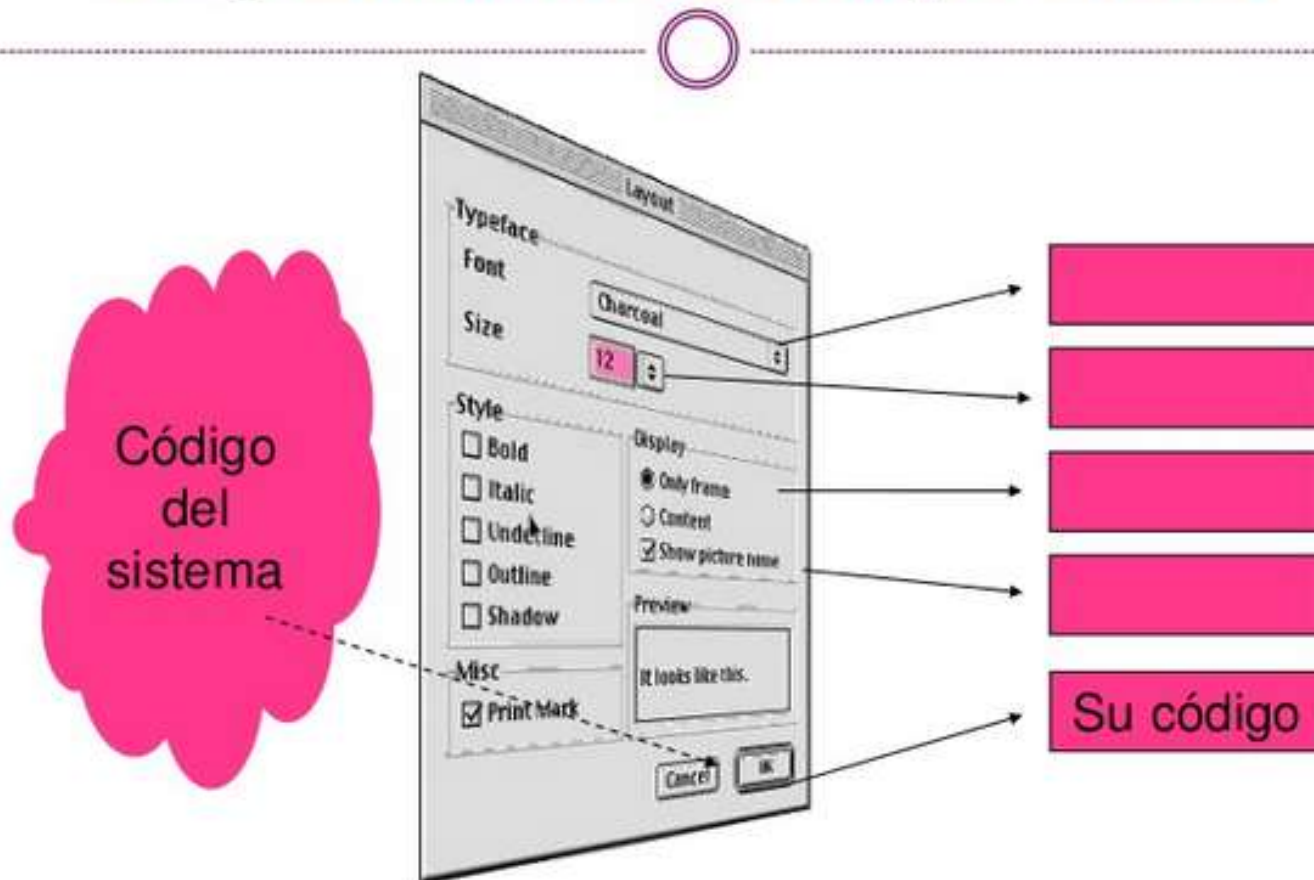


# Swing (biblioteca gráfica)

interfaz gráfica de usuario es  
una biblioteca grafica donde  
podemos encontrar caja de  
textos botones  
tablas .....



# Programación conducida por eventos



Imprimir

General Configurar página Aspecto

Servicio de Impresión

Nombre: LaserJet\_1200 Propiedades...

Estado: Aceptando trabajos

Escriba:

Información: ☐ Imprimir en archivo

Rango de Impresión

☒ Todo

☐ Páginas  A

Copias

Número de copias:

☐ Intercalar

Imprimir Cancelar