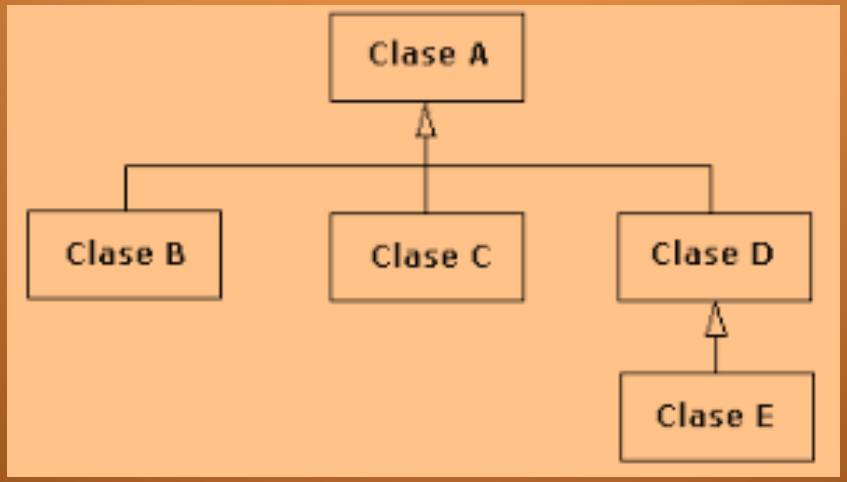
Programación Orientada a Objetos

Fundamentos de la POO

- Abstracción
- Encapsulamiento y ocultación de datos
- Herencia (generalización o especialización)
- Polimorfismo
- Reutilización

Java
Herencia
(Generalización o Especialización)



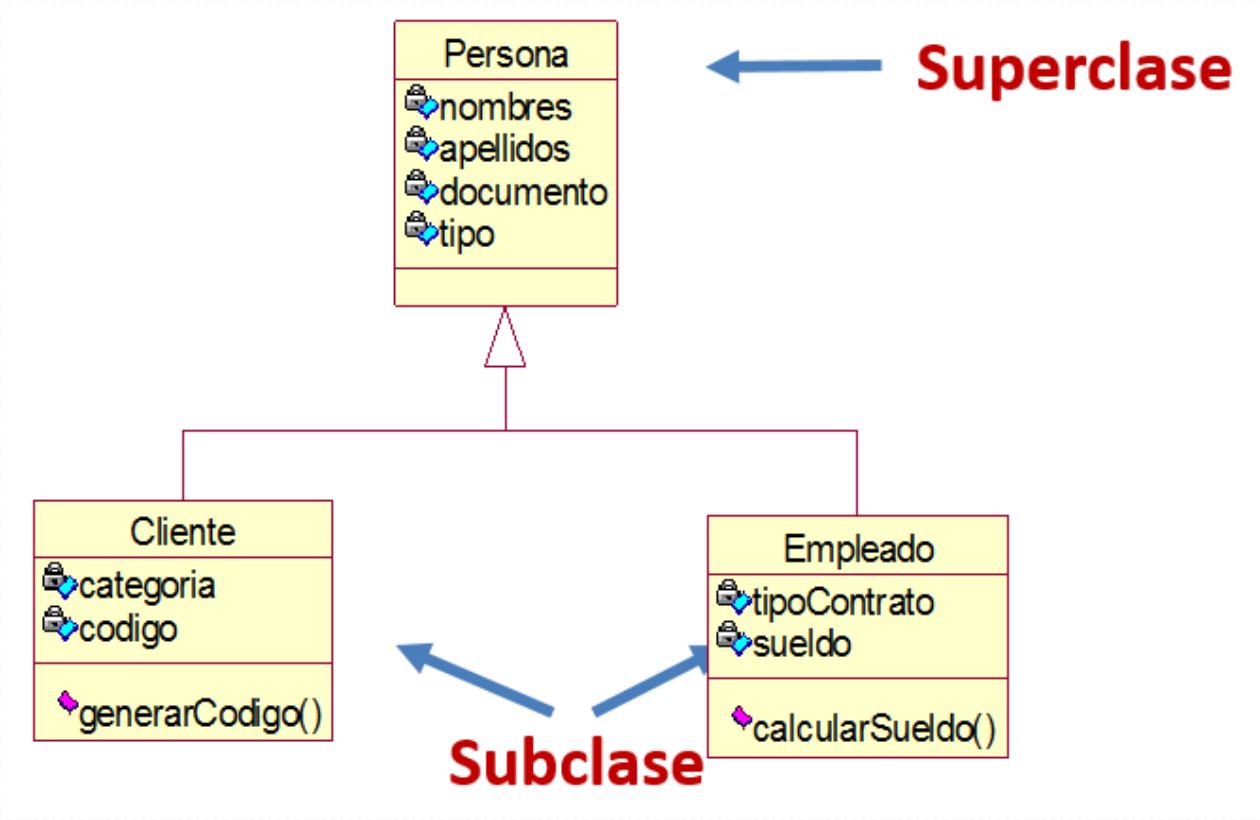
Herencia (Generalización o Especialización)

- La herencia es una relación entre dos clases donde una denominada derivada o subclase (hija) y la otra clase base o general (padre)
- Una generalización se conoce como una relación:
  - es-un, es-un-tipo-de.
- Es una relación entre clases especializadas de una clase general.
  - Un Vendedor es-un Empleado.
  - Un Rectángulo es-un-tipo-de Forma.

Herencia (Generalización o Especialización)

- La clase general agrupa los atributos y los métodos comunes a las clases especializadas.
- Una subclase o clase hija puede ser a su vez clase base de otra lo que produce jerarquías de clases.
- La implementación de la generalización en un lenguaje de programación se conoce como herencia.

Herencia


Persona
nombres
apellidos
documento
tipo
Superclase
Cliente
categoria
codigo
generarCodigo()
Empleado
tipoContrato
sueldo
calcularSueldo()
Subclase

Herencia en Java

- En java la herencia se usa la palabra “extend”

Clase derivada o hija

Clase base o padre

```
public class Automovils extends Vehiculo
{
}
```

Tipos de herencia

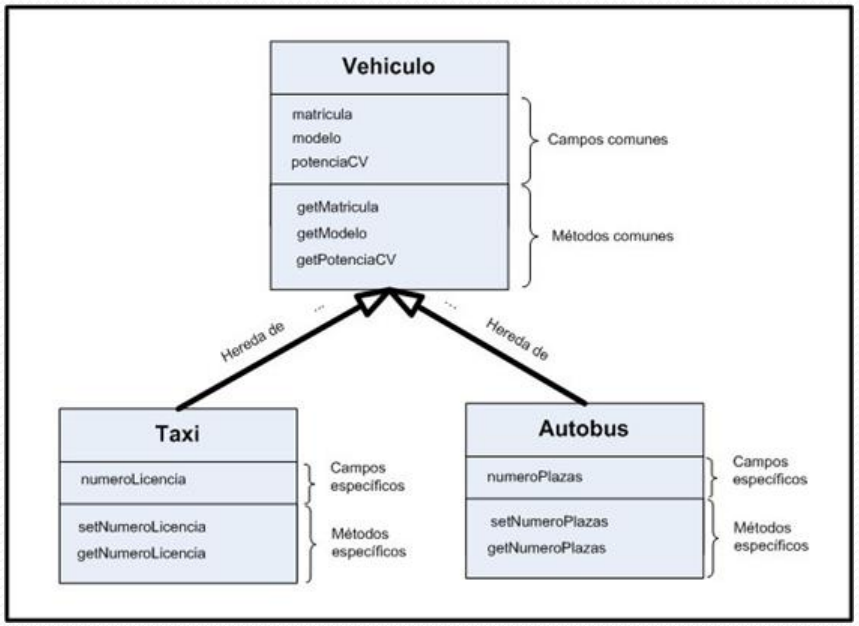
La **herencia simple** es aquella en la que cada clase hereda de una única clase.

La **herencia múltiple** es la transmisión de métodos y datos de más de una clase base a la clase derivada.

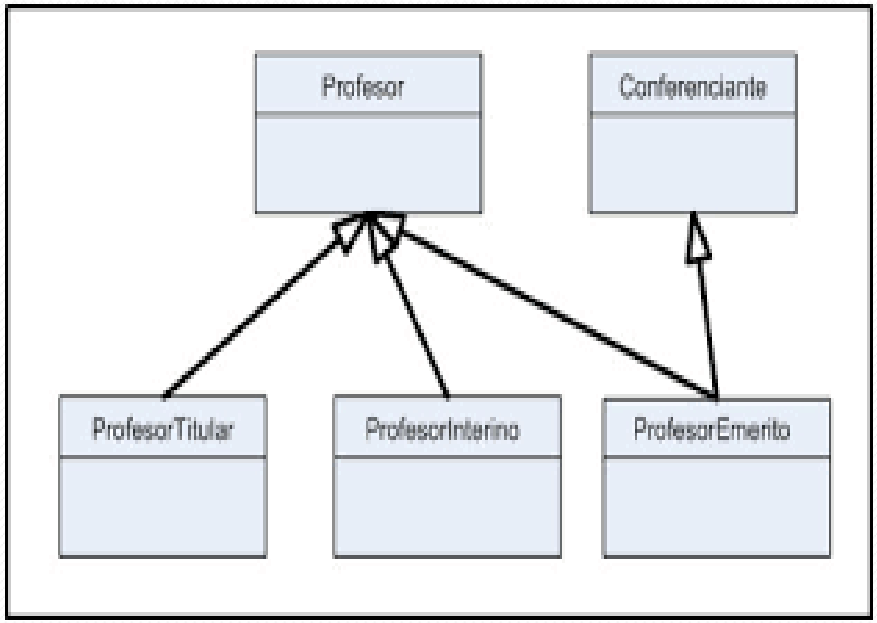
Java, C# y Smalltalk **no implementa la herencia múltiple**

Tipos de Herencia

HERENCIA SIMPLE



HERENCIA MULTIPLE



IMPLEMENTACIÓN DE HERENCIA

Formato

```
class Base
{ Atributos generales
   Metodos generales o comunes…

}

 class Derivada extends Base
   {
      // atributos nuevos  de
Derivada
      // métodos nuevos de Derivada
   }
```

Los atributos deben ser protegidos de la clase base para ser visibles en la clase derivada

Si se declaran privados la clase derivada no los puede usar.

En la clase derivada se pueden invocar atributos y métodos utilizando super.

En general, super hace **referencia a la porción del objeto Padre** que tiene el objeto Hijo.

```
class Cartilla
{
  private String tit;
  protected double saldo;
  public void ingresar(double q)
  {
    saldo += q;
  }


class Ahorrro extends Cartilla
{
  private int duracion;
   public void informe()
   {
     if (saldo > 0 ...// heredado
     System.out.println(tit);   //
}
```

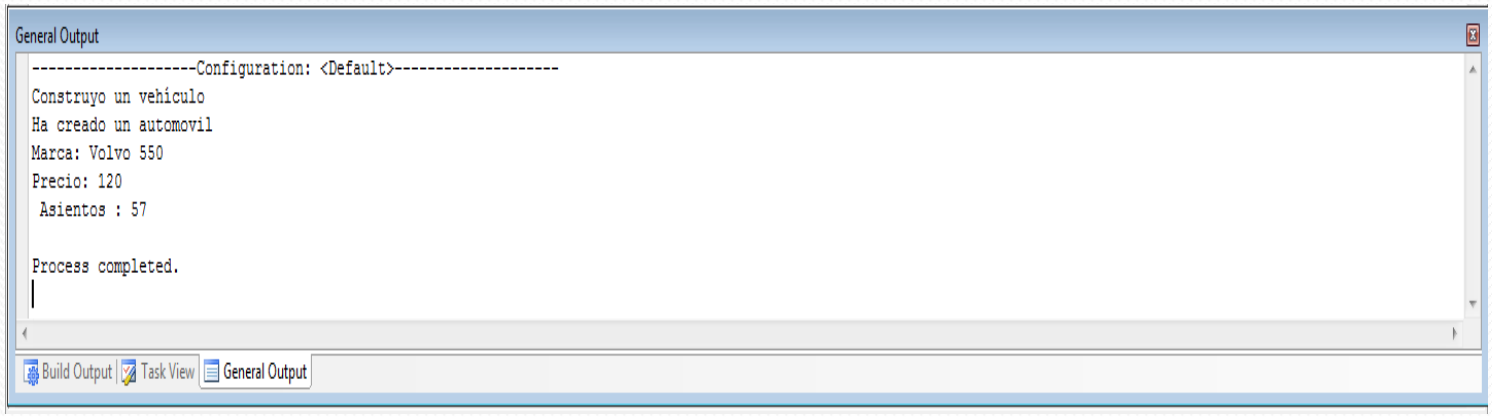
ERROR tit es privada en Padre no se puede usar en clase hija

¿Se puede utilizar todas las variables en la clase hija?

```
public class Vehiculo
{
   protected int precioDia;

   protected String marcaModelo ;

//constructor
   public Vehiculo( String marcaModelo1, int precioDia1 )
 {
     marcaModelo= marcaModelo1;
     precioDia=precioDia1;
     System.out.println( "Construyo un vehículo");
  }

//metodo
  public void MostrarCaracteristicas()

{
     System.out.println ( “Marca: “ + marcaModelo );
     System.out.println (“ Precio: “ + precioDia);

   }
}
```

```
public class Automovil extends Vehiculo
{
  private int numasientos = 5;

 public  Automovil( String marcaModelo, int precioDia, int asientos1)
 {
    super( marcaModelo, precioDia );
    numasientos = asientos1;
    System.out.println( "Ha creado un automovil" );
  }
  public Automovil( String marcaModelo, int precioDia ) {
    super( marcaModelo, precioDia );
    System.out.println( "Ha creado un automovil" );
  }
  public void mostrarCaracteristicas() {
    super.MostrarCaracteristicas();
    System.out.println( " Asientos : " + numasientos );
  }
}
```

```
public class  EjecutaVehiculos
{
public static void main(String args[])
  {
    Automovil v1 = new Automovil( "Volvo 550", 120, 57);
     v1.mostrarCaracteristicas();
   }
}// fin clase
```



Ejercicio: Relaciona cada concepto

SIMPLE

100 PUNTOS

PROTEGIDOS

DERIVADA

MÉTODOS

BASE

HERENCIA

extends

super

ATRIBUTOS

- Es la propiedad de pasar sus propiedades entre clase base e hija _____
- La clase padre o ascendiente se denomina _____
- La clase hija se denomina _____
- Los atributos de la clase padre para accederse de la clase hija deben declararse como _____
- Se heredan _____ y _____
- La palabra _____ sirve para invocar_____ y _____
- En Java la herencia entre clases se usa la palabra _____
- Tipo de herencia que Java usa _____

Reflexionamos y contesta

- ¿Qué es una clase base?
- ¿Qué es una subclase?
- ¿Cómo se le conoce también a la herencia?
- ¿Cómo se representa la herencia en UML?

Actividad en Equipo

Realiza los siguientes diagramas de clases:

- Modela la Clase Persona que tiene tres atributos: Nombre, Edad y Genero
- Modela la Clase Cliente que tiene tres atributos: No-cliente, dirección y teléfono