

TITULO

PROGRAMACION ORIENTADA A OBJETOS

Semana 12

HERENCIA

OBJETIVOS

- ◆ Entender el concepto de Herencia.
- ◆ Aplicar la programación haciendo uso de Herencia.



AGENDA

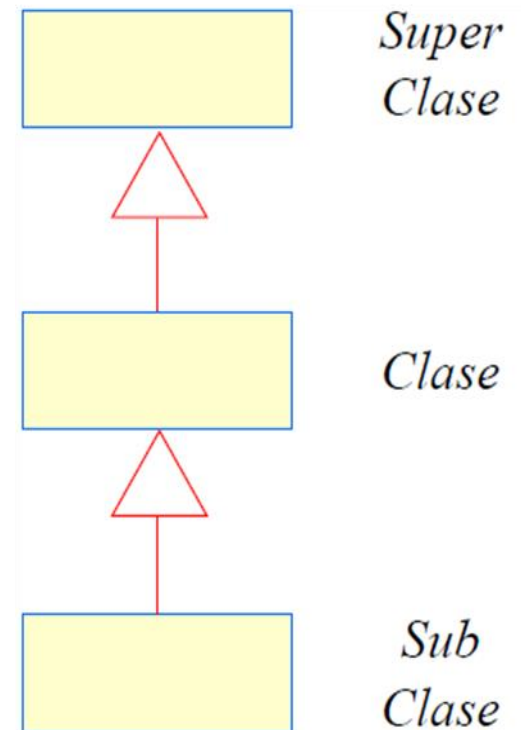
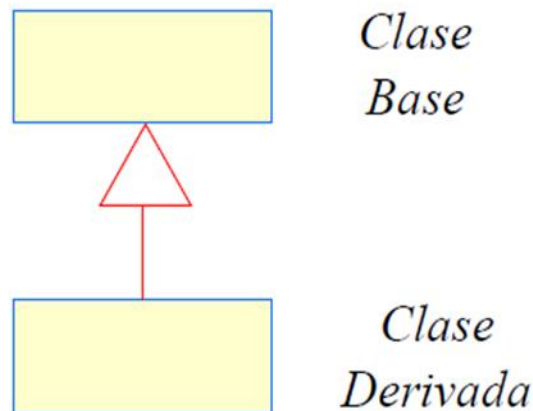
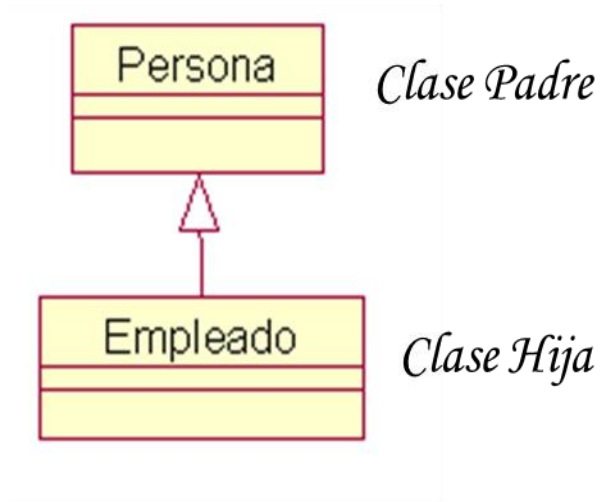
- Definición de Herencia.
- Tipos de Herencia.
 - Herencia Simple.
 - Herencia Múltiple.
- Implementación en Android.
- Ejemplos Generales.



HERENCIA

- Propiedad de la POO a través de la cual se transmiten los atributos y métodos de una Clase hacia sus instancias o hacia sus subclases.

Nomenclatura



HERENCIA

Tipos

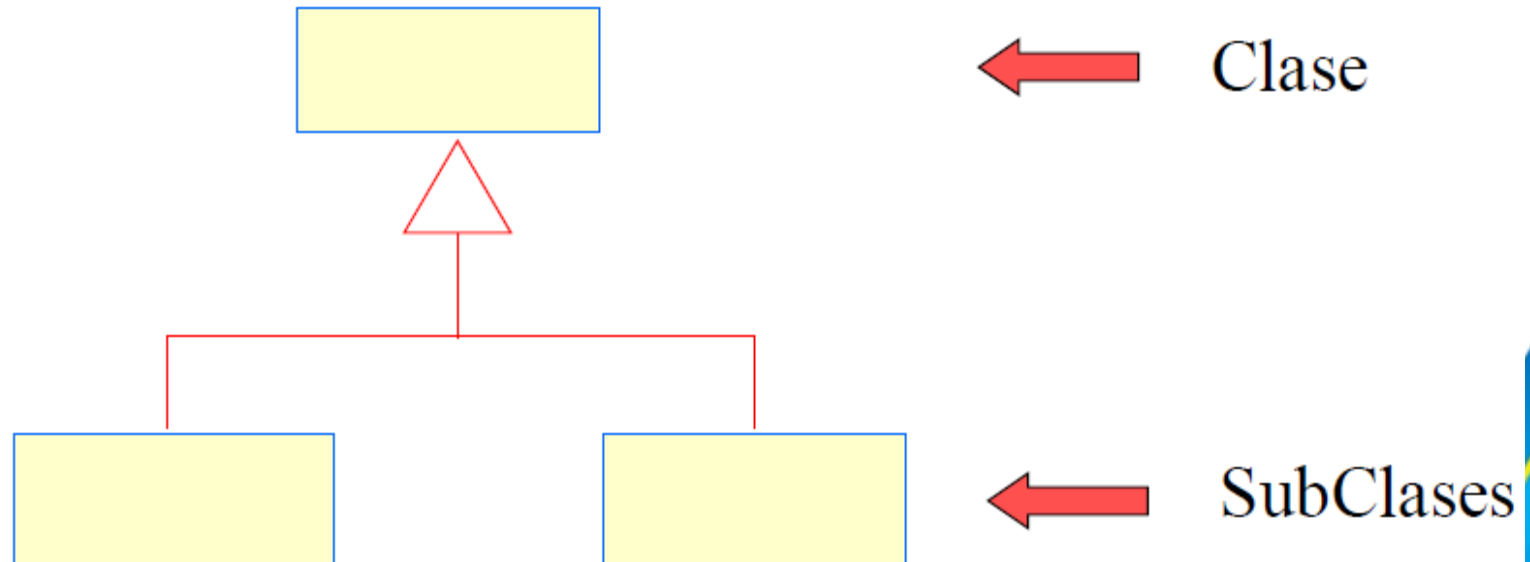
- Herencia desde una clase hacia sus instancias.
- Herencia desde una clase hacia sus subclases.
 - Simple
 - Múltiple

HERENCIA

Herencia simple

Cuando una SubClase hereda atributos y funciones sólo de una Clase.

Representación gráfica

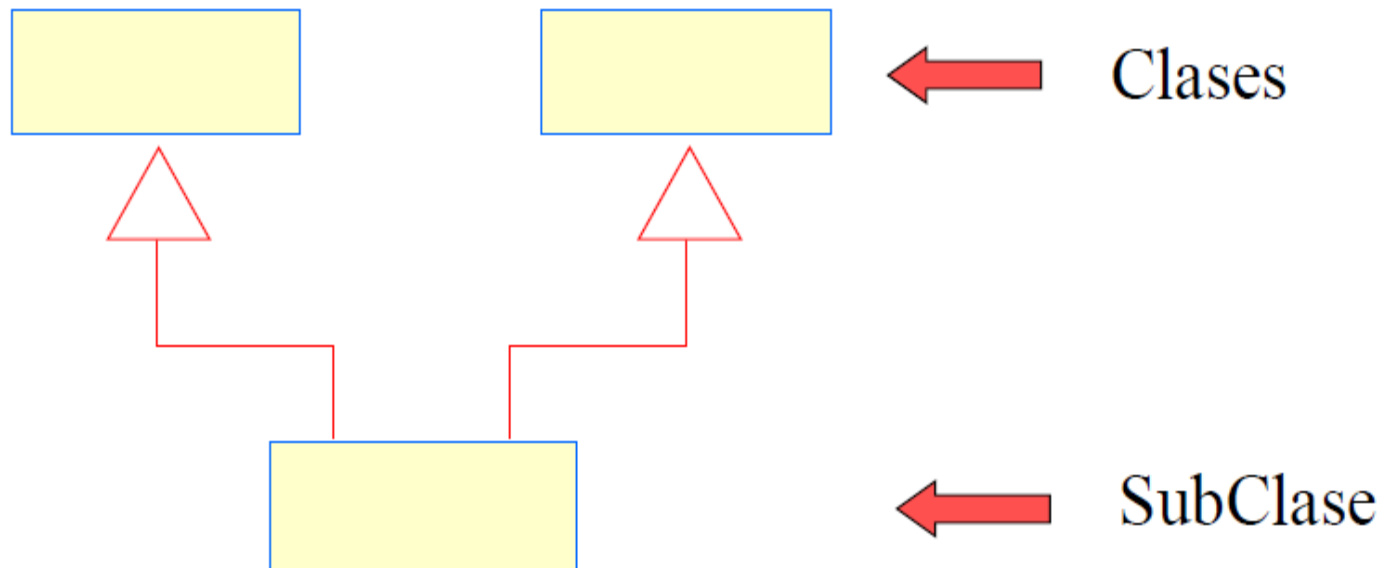


HERENCIA

Herencia múltiple

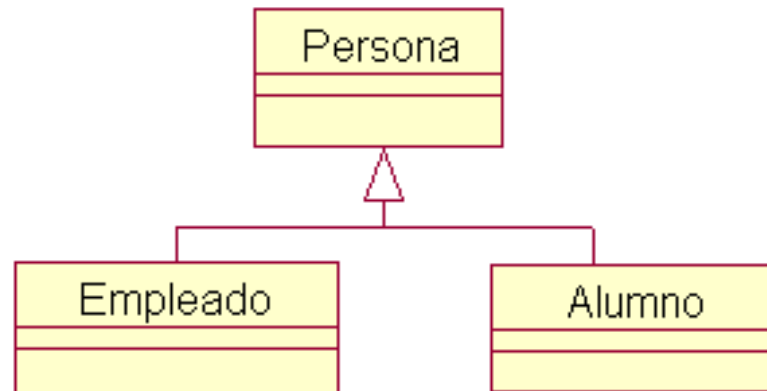
Cuando una SubClase hereda atributos y funciones de 2 ó más Clases.

Representación gráfica



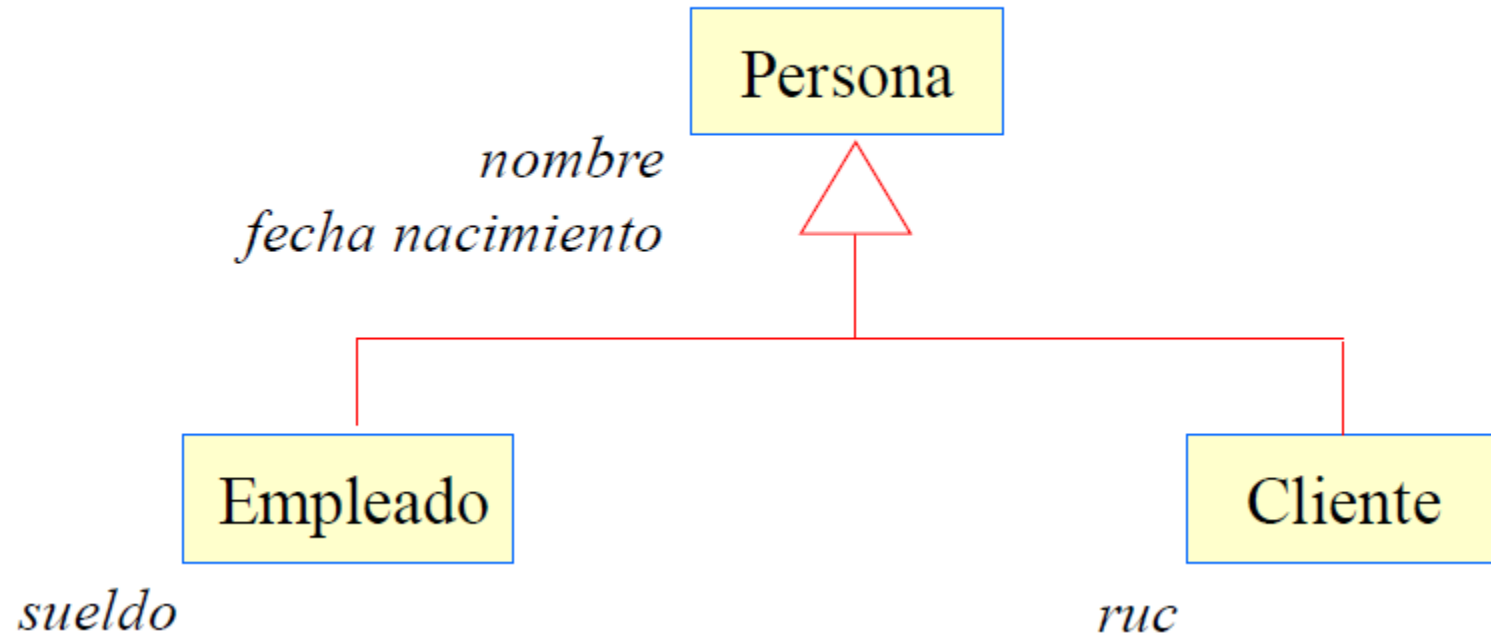
HERENCIA

EJEMPLO



HERENCIA

Propuesto



IMPLEMENTACIÓN en ANDROID

Forma general de la declaración de una clase que hereda los miembros de otra clase es :

```
class nombreDeSubclase extends nombreDeSuperclase{  
    // Cuerpo de la clase  
}
```

La sentencia reservada en Java para denotar que una clase era la funcionalidad de otra clase es la palabra *extends*.

Es importante indicar que el constructor de la clase Derivada puede invocar al constructor de la clase Base utilizando la palabra reservada *super*.

super (lista de argumentos);

Esta instrucción tiene que ser la primera sentencia a ejecutarse dentro del constructor de la clase Derivada.

HERENCIA

EJEMPLOS GENERALES

Un Mamífero es un Animal

Un Ave es un Animal

Una Vaca es un Mamífero

Un Pato es un Ave

Lo que puede expresarse como:

```
class Animal { ... }
```

```
class Mamifero extends Animal { ... }
```

```
class Vaca extends Mamifero { ... }
```

```
class Ave extends Animal { ... }
```

```
class Pato extends Ave { ... }
```