

# Desarrollo de Aplicaciones Distribuidas

## Introducción

Para empezar miraremos algunos conceptos básicos de la programación orientada a objetos.

## Objetos

Un objeto es un elemento con diferentes atributos, un ejemplo sería el objeto Persona, que tendría los atributos DNI, Nombre y Apellidos.

## Métodos

Son funcionalidades aplicadas a cada objeto

## Clases

Son implementaciones totales o parciales de un tipo abstracto de datos. Son la representación de objetos del mundo real

## Aspectos a tener en cuenta

- El código debe ser reutilizable, de forma que un mismo código se pueda utilizar en diferentes programas.
- El código también debe ser extensible, de forma que podamos añadir funcionalidad a este sin que afecte a aplicaciones anteriores

## Objetos VS Invocación de Funciones

Un mensaje es una forma de comunicación entre objetos mientras que la invocación de una función o método será sobre un objeto.

## Herencia

Una clase puede heredar atributos y métodos de otra clase, esto nos permite:

- Reciclar código
- Nuevos Comportamientos
- Cambio de comportamiento
- Jerarquía de clases

## Aspectos a Tener en Cuenta en Herencia

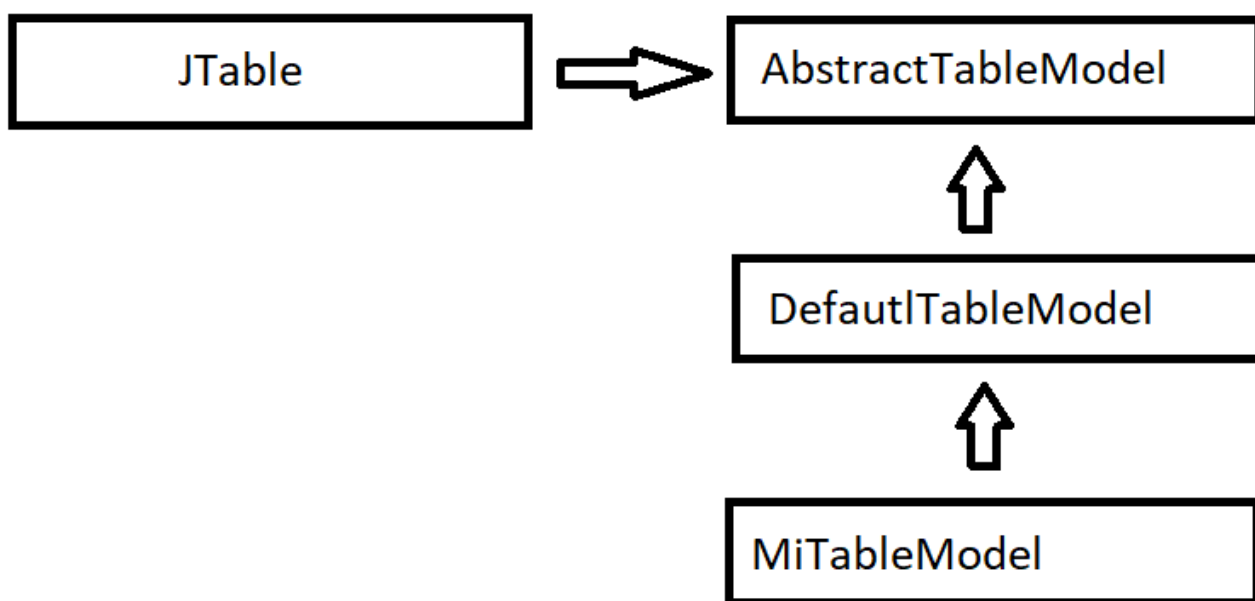
- No Hay herencia múltiple
- Se usa la palabra clave extends
- Toda clase hereda implícitamente de object
- Solo se heredan clases públicas y protegidas
- Para acceder a métodos del padre se usa Super

## Polimorfismos

### CONSTRUCCIÓN

## JTable

La JTable está asociada a AbstractTableModel. No podemos crear un objeto de esta clase ya que es una clase abstracta. AbstractTableModel tiene el DefaultTableModel, que es una clase ya instanciada. Para crear nuestra tabla debemos crear una clase a la que llamaremos MiTableModel que heredara de DefaultTableModel, donde modificaremos los métodos que necesitemos cambiar.



From:  
<https://knoppia.net/> - Knoppia

Permanent link:  
<https://knoppia.net/doku.php?id=dad&rev=1695641487>

Last update: **2023/09/25 11:31**

