**UNIDAD DIDÁCTICA 04: PROGRAMACIÓN ORIENTADA A OBJETOS CON JAVA II**

# Modo de entrega

***Creareis un proyecto nuevo llamado UD0401-Ejercicios, creareis un paquete por ejercicio.***

***Una vez acabados los ejercicios, iréis donde se guardan vuestros proyectos Java ( Si seguisteis la guía de instalación, se encuentra en C:\ProyectosEclipse). Seleccionareis las carpetas donde se guardan los proyectos que contienen los ejercicios y los añadiréis a un archivo zip o rar con el siguiente formato de nombre:***

***DAM-UD04-Apellidos\_Nombre.zip***

***DAM-UD04-Garcia\_Garcia-Antonio.zip***

# Ejercicio 1:

Define una clase abstracta Cuenta con los siguientes atributos:

* numerocuenta : entero largo
* saldo : double
* cliente : atributo de la clase Persona (que tiene nombre y apellidos, y NIF).

Y con los siguientes métodos:

* Constructor parametrizado que recibe un cliente y un número de cuenta
* Accedentes para los tres atributos
* ingresar(double): permitirá ingresar una cantidad en la cuenta.
* abstract retirar(double): permitirá sacar una cantidad de la cuenta (si hay saldo). No se implementa, depende del tipo de cuenta
* actualizarSaldo(): actualizará el saldo de la cuenta, pero cada cuenta lo hace de una forma diferente.

Define las subclases de Cuenta que se describen a continuación:

* CuentaCorriente: Cuenta normal con un interés fijo del 1.5%. Incluir constructor parametrizado y método toString().
* CuentaAhorro: Esta cuenta tiene como atributos el interés variable a lo largo del año y un saldo mínimo necesario. Al retirar dinero hay que tener en cuenta que no se sobrepase el saldo mínimo. Incluir constructor parametrizado, método toString() y método para cambiar el interés.

Crea un programa que cree varias cuentas y pruebe sus características.

# Ejercicio 2:

Se trata de crear una pequeña base de datos de personas de una universidad. De momento definiremos y probaremos las siguientes clases:

* **Dirección**:
  + atributos: calle, ciudad, código postal, país
  + Constructores predeterminado y parametrizado.
* **Persona**: Clase ya creada (con nombre, apellidos y NIF, ver ejercicio anterior) a la que añadiremos el atributo dirección y sus métodos accedentes y mutadores. Esta clase implementa la interface Humano, con un método indentificate(), que muestra el tipo de la clase que lo implementa (el tipo de persona, en este caso).
* **Estudiante**: Subclase de Persona.
  + Atributos: ID de estudiante
  + Constructores : predeterminado y constructor parametrizado que admita el ID.
  + Métodos accedentes y mutadores y toString().
* **Profesor**: Subclase de Persona.
  + Atributos : despacho
  + Constructores: predeterminado y constructor parametrizado que admita el despacho.
  + Métodos accedentes y mutadores y toString()

Crea una clase principal donde probar las diferentes clases creadas.