

# Solución de la tarea para PROG09.

---

## Desarrollo de la aplicación "Gestión de Cuentas Bancarias".

Para el desarrollo de la aplicación he creado un proyecto **Netbeans** de tipo **Aplicación Java**. Este proyecto consta de un conjunto de **clases** e **interfaces** cada una de ellas en un archivo Java diferente:

- ✓ Clase **Persona**, que contiene los atributos y métodos necesarios para la gestión de los **datos personales básicos de una persona: nombre, apellidos y fecha de nacimiento**. Implementa la interfaz **Imprimible** para poder **formatear los datos** en una eventual salida por pantalla.
- ✓ Clase **CuentaBancaria**, que contiene los **atributos** y **métodos** necesarios para la gestión de una cuenta bancaria básica (archivo **CuentaBancaria.java**): **datos personales del titular, saldo**, etc. Se trata de una **clase abstracta** y no puede ser instanciada. Algunos de sus métodos son abstractos (**ingresar** y **retirar**), pues su implementación dependerá de las características específicas de cada cuenta (por ejemplo el método **retirar** puede ser diferente según se permitan o no **descubiertos**). También implementa el **interfaz Imprimible** para poder **formatear los datos de la cuenta**.
- ✓ Clase **CuentaAhorro** (archivo **CuentaAhorro.java**), que es **subclase** de **CuentaBancaria** y sí puede ser instanciada. Añade las características de una **cuenta de ahorro: un tipo de interés de remuneración**. También se definen los **métodos abstractos** de su **clase base** (**ingresar** y **retirar**). No permite **descubiertos**.
- ✓ Clase **CuentaCorriente** (archivo **CuentaCorriente.java**), **subclase** también de **CuentaBancaria**, aunque en este caso se trata nuevamente de otra **clase abstracta**. Incluye una **Hashtable** en la que se podrán ir insertando las distintas **entidades autorizadas a cobrar recibos** de esa cuenta.
- ✓ Clase **CuentaCorrientePersonal** (archivo **CuentaCorrientePersonal.java**), **subclase** de **CuentaCorriente** e **instanciable**. Incluye las **características específicas** de una **cuenta corriente personal: comisión de mantenimiento anual**. No permite **descubiertos**.
- ✓ Clase **CuentaCorrienteEmpresa** (archivo **CuentaCorrienteEmpresa.java**), **subclase** de **CuentaCorriente** e **instanciable**. No tiene **comisión de mantenimiento**. Permite **descubiertos**. Incluye las **características específicas** como **comisión por cada descubierto, tipo de interés por descubierto** y máximo **descubierto permitido**.
- ✓ Interfaz **Imprimible** (archivo **Imprimible**), que indica algunos métodos útiles para el **formateo de datos** para su salida por pantalla.
- ✓ Clase **AplicacionGestionCuentasBancarias**, que contiene un método **main** que se encarga de realizar las funciones de **programa principal** (creación de los objetos de **subclase** de **CuentaBancaria**, organización de todas las cuentas a través de una **Hashtable**, solicitud al usuario de los datos del titular y las cuentas, gestión de un menú de opciones para poder manipular las cuentas, etc.).

El método **main** de la clase principal **AplicacionGestionCuentasBancarias** trabajará con un menú de opciones como el que se ha solicitado en el enunciado de la tarea:

1. **Abrir una nueva cuenta.**
2. **Ver un listado de las cuentas disponibles.**
3. **Obtener los datos de una cuenta.**
4. **Realizar un ingreso en una cuenta.**
5. **Retirar efectivo de una cuenta.**
6. **Consultar saldo de una cuenta.**
7. **Salir de la aplicación.**

Según se soliciten algunas opciones u otras es posible que se nos pida información adicional (por ejemplo el número de cuenta y la cantidad de dinero que se quiere ingresar o retirar). En otros casos simplemente se proporcionará la información solicitada (mediante la llamada a algún método de la interfaz del objeto).

Si se produce alguna **situación inesperada** (códigos **CCC** erróneos, ingreso de cantidades negativas, nombre de titular excesivamente largos, intento de retirada de una cantidad mayor que el saldo, etc.), se producirá una **excepción** que la aplicación capturará adecuadamente para evitar un fallo en tiempo de ejecución.

El proyecto se encuentra el archivo .zip comprimido llamado [PROG09\\_SolucionTarea.zip](#).

Aquí tienes un esquema en el que se muestran las principales relaciones entre las distintas clases utilizadas:

