**Tarea para PROG07**

El proyecto está dividido en tres paquetes:

* **Clases**: que contiene las siguientes clases:
* **Persona**

Clase que se utilizará en CuentaBancaria como titular de la cuenta.

* **CuentaBancaria**

Clase abstracta que implemente Imprimible. Puede ser de dos tipos, CuentaAhorro y CuentaCorriente, que a su vez puede ser CuentaCorrienteEmpresa o CuentaCorrientePersonal.

* **CuentaAhorro**

Es un tipo de CuentaBancaria.

* **CuentaCorriente**

Es un tipo de CuentaBancaria.

* **CuentaCorrienteEmpresa**

Es un tipo de CuentaCorriente que a su vez es un tipo de CuentaBancaria.

* **CuentaCorrientePersonal**

Es un tipo de CuentaCorriente que a su vez es un tipo de CuentaBancaria.

* **Banco**

Contiene como atributo, un array de **CuentaBancaria** donde se van a ir

almacenando las distintas cuentas.

Contiene varios métodos que se usarán en la clase Principal para el desarrollo de la aplicación.

* **abrirCuenta**: recibe por parámetro un objeto CuentaBancaria y lo almacena en la estructura. Devuelve true o false indicando si la operación se realizó con éxito.
* **listadoCuentas**: no recibe parámetro y devuelve un array donde cada elemento es una cadena que representa la información de una cuenta.
* **informacionCuenta**: recibe un iban por parámetro y devuelve una cadena con la información de la cuenta o null si la cuenta no existe.
* **ingresoCuenta**: recibe un iban por parámetro y una cantidad e ingresa la cantidad en la cuenta. Devuelve true o false indicando si la operación se realizó con éxito.
* **retiradaCuenta**: recibe un iban por parámetro y una cantidad y trata de retirar la cantidad de la cuenta. Devuelve true o false indicando si la operación se realizó con éxito.
* **obtenerSaldo**: Recibe un iban por parámetro y devuelve el saldo de la cuenta si existe. En caso contrario devuelve -1.
* **Útil**: que contiene:
* Interfaz **Imprimible**

Se implementará en las clases Persona y Cuenta Bancaria.

Solo consta de un método, **devolverInfoString**, que devolverá la información de una cuenta como una cadena de caracteres.

* Clase **Validar**

Contiene dos métodos estáticos para validar el formato del DNI y del IBAN.

* **Tarea**: donde se encuentra la clase **principal**

Usando **switch** dentro de la estructura de repetición **while**, creo un menú que aparece por pantalla con diferentes opciones que funcionará hasta que elijamos salir.

* 1. Abrir una nueva cuenta.

Se van solicitando todos los datos para crear la nueva cuenta.

Se hace distinción de los datos que se solicitan dependiendo del tipo de cuenta, de ahorros, corriente de empresa o corriente personal.

Una vez que se tienen los datos y se hacen las validaciones necesarias en el DNI y en el IBAN, se crea un nuevo objeto **CuentaAhorros**, **CuentaCorrienteEmpresa** o **CuentaCorrientePersonal** dependiendo del tipo elegido.

Se añade la nueva cuenta creada al banco, por orden de creación.

* 2. Ver un listado de las cuentas disponibles.

Muestra un listado con el titular, el saldo y el IBAN de las cuentas introducidas.

Solo con esos datos, que son comunes a todas las cuentas, por lo que utilizamos el método **toString** de **CuentaBancaria**.

* 3. Obtener los datos de una cuenta. Realizar un ingreso en una cuenta.

En este apartado hace dos cosas, primero nos muestra todos los datos de una cuenta, para ello se usa el método **devolverInfoString** de la interfaz **Imprimible.** Y en segundo lugar ingresamos dinero en una cuenta utilizando el método **ingresoCuenta** de la clase **Banco**.

* 4. Retirar efectivo de una cuenta.

Utilizando el método **retiradaCuenta** de la clase **Banco**, se resta una cantidad de dinero del saldo de una cuenta, siempre que haya suficiente saldo.

* 5. Consultar el saldo actual de una cuenta.

Con el método **obtenerSaldo** de la clase **Banco**, insertando un IBAN, localiza la cuenta, si existe, y muestra su saldo por pantalla.

* 6. Salir de la aplicación.

Para abandonar la aplicación.