INFORME SOBRE LA APLICACIÓN DE CUENTA BANCARIA

El proyecto consta de 2 clases:

* El programa principal o clase main al que llamamos **AplicacionCuentaBancaria**
* Otra clase llamada **CuentaBancaria** que contiene los métodos utilizados por la clase principal

La clase **AplicacionCuentaBancaria**comienza con la declaración de una serie de atributos generales, generamos un objeto de la clase Scanner al que llamamos teclado y nos servirá para recoger los valores que introduzca el usuario por teclado.

Esta clase se compone de 3 bloques bien diferenciados:

1. En el primer bucle condicional do-while pedimos al usuario que introduzca un nombre de titular, este lo pasamos como parámetro al método *validarTitular()* de la clase **CuentaBancaria**, comprobando así que no tenga más de 40 caracteres, si fuera así el bloque se repetiría hasta que sea válido.
2. El segundo bloque condicional do-while pide al usuario que introduzca un número de cuenta, este lo pasamos como parámetro a 2 métodos de la clase **CuentaBancaria:** *validarFormatoCCC y validarNumeroCCC*. Estos métodos realizan comprobación de que el formato sea el correcto, es decir que tenga 20 números y que el número sea válido utilizando el algoritmo de los dígitos de control. Si alguno de los 2 no es correcto nos dará un mensaje de error y se volverá a repetir el bloque.

Una vez que nos aseguramos que el titular y el número CCC son correctos, generamos un objeto *datosCuenta* de la clase **CuentaBancaria** utilizando el constructor de la misma y pasandole como parámetros el titular y el número ccc.

1. El tercer y último bloque imprime el menú por pantalla, solicita al usuario una opción y evalúa dicha opción mediante un switch con todas las opciones disponibles más un opción default por si el número introducido fuera mayor que 10 y menor a 128 (tipo byte).

Cada opción del switch hace una llamada al método correspondiente de la clase **CuentaBancaria**.

En 3 puntos del código hemos metido excepciones, nos sirve para controlar que hacer en caso de que el usuario introduzca caracteres o números no válidos por teclado, generando un mensaje de error y volviéndonos a pedir el dato hasta que sea correcto. De otro modo el programa daría fallo y se interrumpiría inmediatamente.

En la clase **CuentaBancaria** declaramos varios atributos al inicio con distinto métodos de acceso para hacer una buena encapsulación del programa:

* *protected String nombreTitular, numeroCCC* son del tipo protected porque los vamos a usar en las 2 clases dentro del mismo paquete.
* El resto son del tipo private porque solo los usaremos en esta clase.

Hemos creado un método constructor con los 2 parámetros más importantes de la aplicación que son el titular y el número de ccc. Considerando que para cualquier objeto que generemos de esta clase vamos a necesitar siempre estos datos para trabajar con ellos.

El resto del código contiene varios métodos:

* Del tipo *public* tenemos varios *public String informacionEntidad(), public String informacionOficina(), public String informacionNumCuenta(), public String informacionDigitosControl(), public int consultaSaldo()*, todos ellos nos dan información mediente el retorno del valor correspondiente, no reciben parámetros y pueden ser usados por cualquier otra clase o método de cualquier proyecto o paquete.
* Otros 2 métodos *public void realizarIngreso(int ingreso) y public void realizarRetirada(int retirada)* del tipo public también, pero que reciben parámetros, evalúan si no son valores negativos o si el saldo es suficiente en caso de petición de retirada, modifican el atributo saldo si fuera necesario y no retornan ningún valor.
* Los métodos *protected static boolean validarFormatoCCC(String numeroCCC) y protected static boolean validarTitular(String nombreTitular)* como su propios nombres indican, comprueban que tanto el formato del número de cuenta sea de 20 digitos (sino devuelve no válido) como que la longitud del String que recoge el nombre del titular no sea mayor a 40 caracteres (sino retorna no válido).

Son declarados como protected porque solo queremos usarlos dentro del mismo paquete y son del tipo static porque nos interesa que sean métodos de la clase, es decir que no tengamos que instanciar un objeto para poder usarlo dentro de nuestro programa. Los vamos a llamar dentro de la aplicación principal antes de generar un objeto de dicha clase.

* Los 2 últimos métodos *protected static boolean validarNumeroCCC(String numeroCCC) y private static byte bucleValidacionCCC(String numeroCCC)* están relacionados entre si, el primero llama por 2 veces al segundo y este le devuelve un valor del tipo byte. Explicándolo mejor, el primer método que recibe como parámetro el número de ccc lo descompone en 2 partes, los primeros 8 digitos (numero entidad y oficina) y llama al segundo método pasándole dicho número como parámetro para que este realice las operaciones con el array de PESOS y nos devuelve un número de tipo byte que comparándolo con el primer digito de control incluido en el número ccc nos dirá si esta primera parte del número es correcta o no. En caso que sea incorrecto retornará un valor true y no continua el programa, si fuera correcto pasa a realizar la misma operación pero esta vez con los 10 últimos números del ccc (numero de cuenta) y el resultado lo comparamos esta vez con el segundo digito de control. Si los 2 códigos son correctos retornará un valor false.