Tarea para PROG05.

Detalles de la tarea de esta unidad.

Enunciado.

A lo largo de esta unidad has ido aprendiendo a crear tus propias clases así como sus distintos miembros (atributos y métodos). Has experimentando con la encapsulación y accesibilidad (modificadores de acceso a miembros), has creado miembros estáticos (de clase) y de instancia (de objeto), has escrito constructores para tus clases, has sobrecargado métodos y los has utilizado en pequeñas aplicaciones. También has tenido tu primer encuentro el concepto de herencia, que ya desarrollarás en unidades más avanzadas junto con otros conceptos avanzados de la Programación Orientada a Objetos.

Una vez finalizada la unidad se puede decir que tienes un dominio adecuado del lenguaje Java como para desarrollar tus propias clases y utilizarlas en una aplicación final que sea capaz de manipular un conjunto de datos simple. Dada esa premisa, esta tarea tendrá como objetivo escribir una pequeña aplicación en Java empleando algunos de los elementos que has aprendido a utilizar.

Se trata de desarrollar una aplicación Java en consola que permita gestionar una cuenta bancaria. Mediante un **menú** se podrán realizar determinas operaciones:

1. Ver el número de cuenta completo (CCC – Código Cuenta Cliente).
2. Ver el titular de la cuenta.
3. Ver el código de la entidad.
4. Ver el código de la oficina.
5. Ver el número de la cuenta (solamente el número de cuenta, sin entidad, oficina ni dígitos de control).
6. Ver los dígitos de control de la cuenta.
7. Realizar un ingreso. Habrá que solicitar por teclado la cantidad que se desea ingresar.
8. Retirar efectivo. Habrá que solicitar por teclado la cantidad que se desea retirar.
9. Consultar saldo.
10. Salir de la aplicación.

Para ello se creará un proyecto Netbeans con dos archivos: **CuentaBancaria.java** y **AplicacionCuentaBancaria.java**. El primero de ellos contendrá la clase CuentaBancaria con los atributos y métodos necesarios para trabajar con un objeto CuentaBancaria. El segundo contendrá la función principal main que gestionará la aplicación. A continuación se detalla el diseño de estas dos clases:

La clase CuentaBancaria

Una cuenta bancaria se compondrá de una persona titular de la cuenta, el saldo en euros, y un código de cuenta (el CCC o Código Cuenta Cliente). El CCC consta de 20 dígitos: 4 dígitos que representan el código de la entidad, 4 dígitos que representan el código de la oficina, 2 dígitos de control y 10 dígitos con el número de la cuenta. Los dos dígitos de control se calculan con los valores de la entidad, la oficina y el número de cuenta por lo que no será necesario almacenarlos. Así mismo deseamos que el nombre del titular tenga una longitud máxima de 100 caracteres y una longitud mínima de 10 caracteres. Por tanto, el diseño de la clase se ajustará al siguiente modelo:

• **Atributos privados:** titular, saldo, entidad, oficina y numCuenta.
• **Atributos de clase (static) públicos:** constantes con el tamaño máximo y mínimo del nombre

del titular.
• **Métodos públicos:**
◦ Constructor con los parámetros básicos de la cuenta corriente. Si alguno de los parámetros y/o el CCC no es válido se lanzará la excepción IllegalArgumentException. **(1 punto)**
public CuentaBancaria(String titular, String entidad, String oficina, String DC, String numCuenta);

◦ Constructor con los parámetros titular y CCC sin descomponer. Este constructor descompondrá el CCC y llamará al constructor básico mostrado anteriormente. **(1 punto)**
public CuentaBancaria(String titular, String CCC);

◦ Métodos obtener para los atributos: titular, saldo, entidad, oficina y numCuenta.**(0.5 puntos)**

◦ Métodos establecer para los atributos: titular. Se debe comprobar la validez del nuevo titular y lanzar la excepción IllegalArgumentException si no se verifica. **(0.5 puntos)**

◦ Método ingresar. Ingresará el dinero correspondiente al saldo de la cuenta. Este valor debe ser positivo, en caso contrario se lanzará la excepción IllegalArgumentException. **(0.5 puntos)**
public void ingresar(double cantidad);

◦ Método retirar. Se restará la cantidad indicada del saldo de la cuenta. Este valor debe ser positivo y nunca superior al saldo de la cuenta. De no ser así se lanzará la excepción IllegalArgumentException. **(0.5 puntos)**
public void retirar(double cantidad);

◦ Comprobar la validez de un CCC. El método devolverá true o false si el CCC indicando si el CCC es válido o no. **(1 punto)**
public static boolean comprobarCCC(String CCC);

◦ Calcular los dígitos de control de un CCC. El método devuelve los dos dígitos de control dado los códigos de entidad, oficina y número de cuenta. **(1 punto)**
public static String obtenerDigitosControl(String entidad, String oficina, String numCuenta)

◦ Método String toString(). Devolverá una cadena con los datos del titular, CCC y saldo de la cuenta. **(0.5 puntos)**

Puedes implementar los métodos privados que creas necesarios para mejorar la encapsulación y evitar la duplicación de código. Debes documentar la clase adecuadamente añadiendo comentarios javadoc para la clase y los métodos y otros comentarios descriptivos dentro de los métodos si lo consideras útil.

Cálculo de los dígitos de control de un CCC

El primer dígito de control se calcula con los datos de la entidad y la oficina mientras que el segundo dígito de control se calcula con el número de cuenta. El procedimiento es el mismo en ambos casos. Supongamos que vamos a calcular el CCC para la cuenta 1234 5678 - - 1234567890 Debemos multiplicar los dígitos por unos pesos en función de la posición que ocupan. Los pesos son los siguiente: 1, 2, 4, 8, 5, 10, 9, 7, 3, 6 y sumar los productos obtenidos.

Para el caso del primer dígito, como la entidad y la oficina tienen 8 dígitos, se rellena con ceros por la izquierda. Calculemos los productos y la suma de los mismos:

0 0 1 2 3 4 5 6 7 8
1 2 4 8 5 10 9 7 3 6
0 0 4 16 15 40 45 42 21 48

4+16+15+40+45+42+21+48 = 231

La suma obtenida la dividimos por 11 y nos quedamos con el resto. Posteriormente, a 11 le restamos el resto anterior y ese será el dígito de control. En caso de que el resultado fuera 10, el dígito sería 1 y si fuera 11, tomaríamos como dígito el 0.

231 % 11 = 0
11 – 0 = 11

Por tanto el dígito de control sería 0.
De igual forma calcularíamos el segundo dígito de control que debe ser 6.

El programa principal

El programa principal se encarga de gestionar la solicitud de datos al usuario y la creación de una cuenta bancaria. Las opciones disponibles se mostrarán en un menú como el mostrado al principio de esta tarea. Se deben comprobar y manejar los errores de forma adecuada para evitar que el programa termine de forma inesperada. Por tanto se comprobará que el usuario introduce valores correctos en todo momento. Se valorará el diseño modular y estructurado, la claridad y simplicidad del código y su documentación.

Recursos:

http://www.cuadernosbancarios.com/2009/04/ccc/
http://es.wikipedia.org/wiki/C
%C3%B3digo_cuenta_cliente#C.C3.A1lculo_de_los_d.C3.ADgitos_de_control

Criterios de puntuación. Total 10 puntos.

• Clase CuentaBancaria: 6.5 puntos.
• Clase AplicacionCuentaBancaria: 3.5 puntos.
Están disponibles todos los menús de la aplicación: 1.5 puntos
Se manejan adecuadamente los errores introducidos por el usuario: 1.5 puntos
El código está bien documentado y organizado: 0.5 puntos

• Total: 10 puntos.