Práctica 9:

POO, Ficheros, mezcla, …

Objetivo de la práctica

El objetivo de esta práctica es la implementación de un programa en Java en el que se repase todo lo aprendido, ficheros, arrays, métodos, etc. La práctica planteada fue un ejercicio de examen.

Ejercicio 1

El banco BFI ha almacenado en un fichero de texto, llamado "ingresosBFI.txt", todos los movimientos monetarios de sus clientes desde su fundación. Cada línea del fichero corresponde con la información de un único cliente. Más concretamente, cada línea consta de: un primer entero que determina el número de movimientos realizados por el cliente, una secuencia de enteros formada por el importe de cada movimiento (los enteros positivos representan ingresos y los negativos retiradas de dinero); y, finalmente, el nombre del cliente. Por ejemplo, la siguiente línea representa que Luis Alberto Pérez Martín ha realizado 4 movimientos siendo su saldo final de 4250 euros.

4 3500 1250 -3000 2500 Luis Alberto Pérez Martín

El fichero está libre de errores y el banco conoce su número de clientes, es decir, el número de líneas del fichero.

El departamento de informática del banco BFI ha programado en JAVA la clase `CuentaCorriente`. Un objeto de esta clase almacena la información de una cuenta corriente, concretamente, el titular de la cuenta y su saldo. Para almacenar esta información, la clase consta de dos atributos privados: `titular` y `saldo`. Además, la clase ofrece un constructor y los correspondientes métodos SET y GET.

```
public class CuentaCorriente {
      private String _titular;
      private int _saldo;

      public CuentaCorriente(String t, int s){}
      public void setTitular(String t){}
      public void setSaldo(int s){}
      public String getTitular(){}
      public int getSaldo(){}
}
```

El banco quiere procesar la información del fichero de texto y almacenarla en una estructura arreglo de tipo `CuentaCorriente[]`. Se creará un objeto `CuentaCorriente` con la información de cada cliente almacenada en el fichero. El nombre del titular corresponderá con el nombre del cliente y el saldo será calculado sumando todos sus movimientos. Por tanto, **SE PIDE** programar en Java el siguiente método:

```
public static void CuentaCorriente[] crearCC (String fileName, int max_cc)
```

Los parámetros de entrada del método son el nombre de un fichero de texto con el formato anterior (`fileName`) y el número de clientes (o filas) del fichero (`max_cc`). El método debe devolver como resultado la estructura de tipo arreglo `CuentaCorriente[]` con la información de los clientes del banco.

Ejercicio 2

La crisis económica actual ha provocado que el banco BFI del ejercicio anterior se haya fusionado con el banco BGT. Utilizado el método implementado en el apartado anterior, cada banco ha almacenado en una estructura de tipo `CuentaCorriente[]` la información de sus respectivos clientes. Dado que BFI y BGT se han fusionado, les interesa tener la información de los dos bancos en una única estructura de tipo `CuentaCorriente[]` que contenga los clientes de ambos bancos.

SE PIDE programar en Java el siguiente método:

```
public static void CuentaCorriente[] mezclarCC
                                    (CuentaCorriente[] cb1, CuentaCorriente[] cb2)
```

Los parámetros de entrada del método son las estructuras de tipo arreglo con las cuentas corrientes de los dos bancos (`cb1` y `cb2`, respectivamente). Por simplicidad, un cliente no puede tener una cuenta en ambos bancos. Además, las cuentas corrientes contenidas en cada estructura están ordenadas de menor a mayor según el saldo. Por tanto, el método `mezclarCC` debe crear una nueva estructura de tipo `CuentaCorriente[]` que contenga las cuentas de `cb1` y `cb2` y esté también ordenada de menor a mayor según el saldo.

IMPORTANTE: No puede utilizarse ningún algoritmo predefinido de ordenamiento de arreglos para ordenar por saldo las cuentas de la estructura resultado del método.

Debes entregar la clase `CuentaCorriente` con los métodos especificados en los dos ejercicios y lo puedes probar con el siguiente programa Java y los siguientes ficheros de texto:

```
pruebaCuentaCorriente {
 public static void main (String [] args){
       CuentaCorriente[] Banco1=CuentaCorriente.crearCC("banco1.txt",5);
       CuentaCorriente[] Banco2=CuentaCorriente.crearCC("banco2.txt",6);

       CuentaCorriente.ordenaLista(Banco1);
       CuentaCorriente.ordenaLista(Banco2);

       System.out.println();
       System.out.println("Datos Banco1");
       for (int i=0; i<Banco1.length;i++){
             System.out.println(Banco1[i].getTitular()+"Saldo:"+Banco1[i].getSaldo());
       }

       System.out.println();
       System.out.println("Datos Banco2");
       for (int i=0; i<Banco2.length;i++){
             System.out.println(Banco2[i].getTitular()+"Saldo:"+Banco2[i].getSaldo());
       }

       System.out.println();
       System.out.println("Prueba fusion y ordenación");
       CuentaCorriente[] BancoFusion=CuentaCorriente.meclarCC(Banco1, Banco2);
       System.out.println("Datos BancoFusion");
       for (int i=0; i<BancoFusion.length;i++){
       System.out.println(BancoFusion[i].getTitular()+"Saldo:"+BancoFusion[i].getSaldo());
       }

  }
}
```

banco1.txt

```
2 3500 200 Luis Alberto Diez
3 100 2300   550 Carlos Perez
1  450 Pedro Ferrer
4 100 100 250 3000 Maria Pascual
3 800 250 10000 Laura Jimenez
```

banco2.txt

```
3 1900 3500 200 Mario Lacruz
2 100 550 Laura Ruiz
5 100 100 200 300 450 Carmen Calvo
2 100 100 Mar Alvarez
3 1800 2000 1000 Raul Antunez
1 3500 Juan Perez
```