Tarea 7

## Informe de la PROG09

## **Ejercicio 1**

Se trata de modificar la aplicación desarrollado en la Unidad de Trabajo 8, Ejercicio 1 para dar persistencia a los datos de cuentas bancarias. El nombre será PROG09\_Ejerc1 Para ello:

- Cuando la aplicación finalice, es decir, el usuario seleccione la opción Salir, la aplicación volcará el contenido de la estructura de datos con las cuentas bancarias a un fichero binario denominado datoscuentasbancarias.dat.
- Cuando la aplicación inicie la ejecución, antes de mostrar el menú, deberá cargar en la estructura de datos el contenido del fichero datoscuentasbancarias.dat.

Como ya sabes, para poder realizar estas tareas es necesarios que nuestros objetos que representan cuentas bancarias sean serializables. Habrá que realizar las convenientes modificaciones a la clase CuentaBancaria.

Método guardarDatos Guardar el cuentasBancarias en el fichero.

CuentasBancarias es un conjunto de tipo HashSet en el cual están todas las cuentas del Banco.

El método en si lo que hace es crear un objeto de tipo FileOutputStream para manipular el fichero y otro Objeto de tipo ObjectOutputStream para escribir en el fichero

Si todo se ha creado correctamente guarda con writeObject el hashSet.

José Ramón Blanco Gutiérrez 1/6

```
Clase Persona
Para que poder guardar el objeto
cuentaBancaria he tenido que
añadir a las diferentes clase la
                                   public class Persona implements Imprimible, Serializable {
interfaz Serializable.
 Clase Abstracta CuentaBancaria
                                    public abstract class CuentaBancaria implements Imprimible, Serializable {
      Clase Cuenta Ahorro
Clase Abstracta Cuenta Corrient
 Clase CuentaCorrienteEmpresa
```

José Ramón Blanco Gutiérrez 2/6

Clase CuentaCorrientePersonal

```
public class CuentaCorrientePersonal extends CuentaCorriente implements Serializable{
    /**
    * Atributos de la clase
    */
    private float comisionMantenimiento;

/**
```

Y en la opción de salir llama al método y si todo ha ido correctamente crea el fichero con los datos.

```
case 8: //"8. Salir
  if (banco.guardarDatos()) {
     System.out.println("Datos guardados con éxito.");
} else {
     System.out.println("Error guardando datos.");
}
salir = true;
System.out.println("Saliendo ...");
break;
```

Método cargarDatos Este método carga los datos del disco y los guarda en el atributo cuentasBancarias.

Para leer el fichero usamos readObject que devuelve un objeto de tipo Object por eso he tenido que castearlo con el HashSet.

```
public boolean cargarDatos() {
    /**
    * Definimos variables para usar ficheros
    */
FileInputStream fichero = null;
ObjectInputStream salida = null;
boolean valorRetorno = true;

try {
    fichero = new FileInputStream("datoscuentasbancarias.dat");
    salida = new ObjectInputStream(fichero);
    try {
        this.cuentasBancarias = (HashSet) salida.readObject(); //Lee todo el
    } catch (ClassNotFoundException ex) {
        System.out.println(ex.getMessage());
    }
} catch (FileNotFoundException ex) {
    System.out.println(ex.getMessage());
    valorRetorno = false;
} catch (IOException ex) {
    System.out.println(ex.getMessage());
    valorRetorno = false;
} finally {
    if (fichero != null) { //SI el fichero tiene datos es que esta abierto
        try {
            fichero.close(); //Cierras el fichero abierto
        } catch (IOException ex) {
            System.out.println(ex.getMessage());
            valorRetorno = false;
        }
    }
} return valorRetorno;
}
```

José Ramón Blanco Gutiérrez 3/6

Antes de que se inicie el bucle principal se llama al método para que cargue los datos si les hubiera.

José Ramón Blanco Gutiérrez 4/6

## **Ejercicio 2**

Añade una nueva opción al menú de la aplicación denominado "Listado clientes" de modo que al seleccionarla, se genere un fichero de texto denominado ListadoClientesCCC.txt que contenga una línea de texto por cada cuenta bancaria almacenada, donde se visualice nombre del propietario y CCC por cada una de ellas.

La última línea del fichero contendrá el número total de cuentas existente.

Método listadoClientes Este método genera un fichero con el número de cuenta y el propietario de dicha cuenta.

El funcionamiento básicamente lo primero que hace es comprobar que hay datos en el banco y si es así entra y recorre la colección con un foreach y cada cuenta y propietario lo guarda.

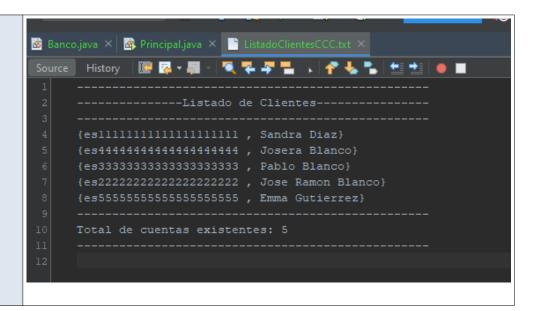
Al final de recorrer todas las cuentas, almacena el numero de cuentas en el fichero de texto.

Llamo al método y si ha ido bien muestro mensaje de todo correcto y sino ha ido bien pongo un mensaje de error.

```
case 7: //7. Listado clientes
  if (banco.listadoClientes()) {
      System.out.println("Listado de clientes guardado con éxito en: ListadoClientesCCC.txt");
} else {
      System.out.println("Error guardando listado de clientes");
}
break;
```

José Ramón Blanco Gutiérrez 5/6

Contenido del fichero de texto creado.



José Ramón Blanco Gutiérrez 6/6