I.E.S. El Rincón
Departamento de Informática

EXAMEN FINAL DE PROGRAMACION

Lunes, 29 de mayo de 2017

INTRODUCCION: La prueba de evaluación consiste en resolver unos ejercicios usando el lenguaje Java. Es necesario disponer de:

1. Entorno de desarrollo Eclipse
2. Servidor de bases de datos mySQL
3. Las bases de datos del enunciado, cuyos backups se dan en el enunciado.

No está permitido el uso de dispositivos enchufables , el cable de red debe estar quitado y el móvil apagado.

Crea un proyecto con **tu nombre** y realiza en él los ejercicios.

El proyecto debe contener exclusivamente el código de los ejercicios. ¡NO subir cualquier otro código!.

Cada ejercicio debe incluir la llamada al método correspondiente.

A medida que se vayan haciendo los ejercicios y funcionen, se avisa al profesor para mostrarlo.

Leer atentamente las especificaciones de los ejercicios. Cada ejercicio tiene un párrafo “ORIENTACION” en el que se aconsejan pasos a seguir, pero no son obligatorios. Intentar primero el ejercicio que consideres más asequible.

EJERCICIO 1
3 puntos

Objetivo: Desarrollar un método que genere una matriz de números aleatorios, con el siguiente prototipo

public int[][] ejercicio1 (int nFilas, int nColumnas, int inferior, int superior)

Ejemplo de llamada (poner en contexto), sería

```
int[][] matriz examen.ejercicio1(9, 6, 10, 100);
```

Significa que la matriz obtenida tendría 9 x 6 (filas x columnas). Los números aleatorios generados deben estar comprendidos entre 10 y 100.

ORIENTACION:
Los números aleatorios se pueden obtener con Math.random() o con la clase Random y el método nextInt().

En clase se generaron números aleatorios para lanzar un dado.

EJERCICIO 2
4 puntos

Se dispone de dos ficheros de texto: cuentas.txt y movimientos.txt.
Ambos deben estar validados, es decir, no deben contener errores.

Los ficheros contienen la siguiente información:

Fichero "**cuentas.txt**":

número de cuenta (Integer)
saldo (float) , puede ser negativo, el saldo inicial de la cuenta

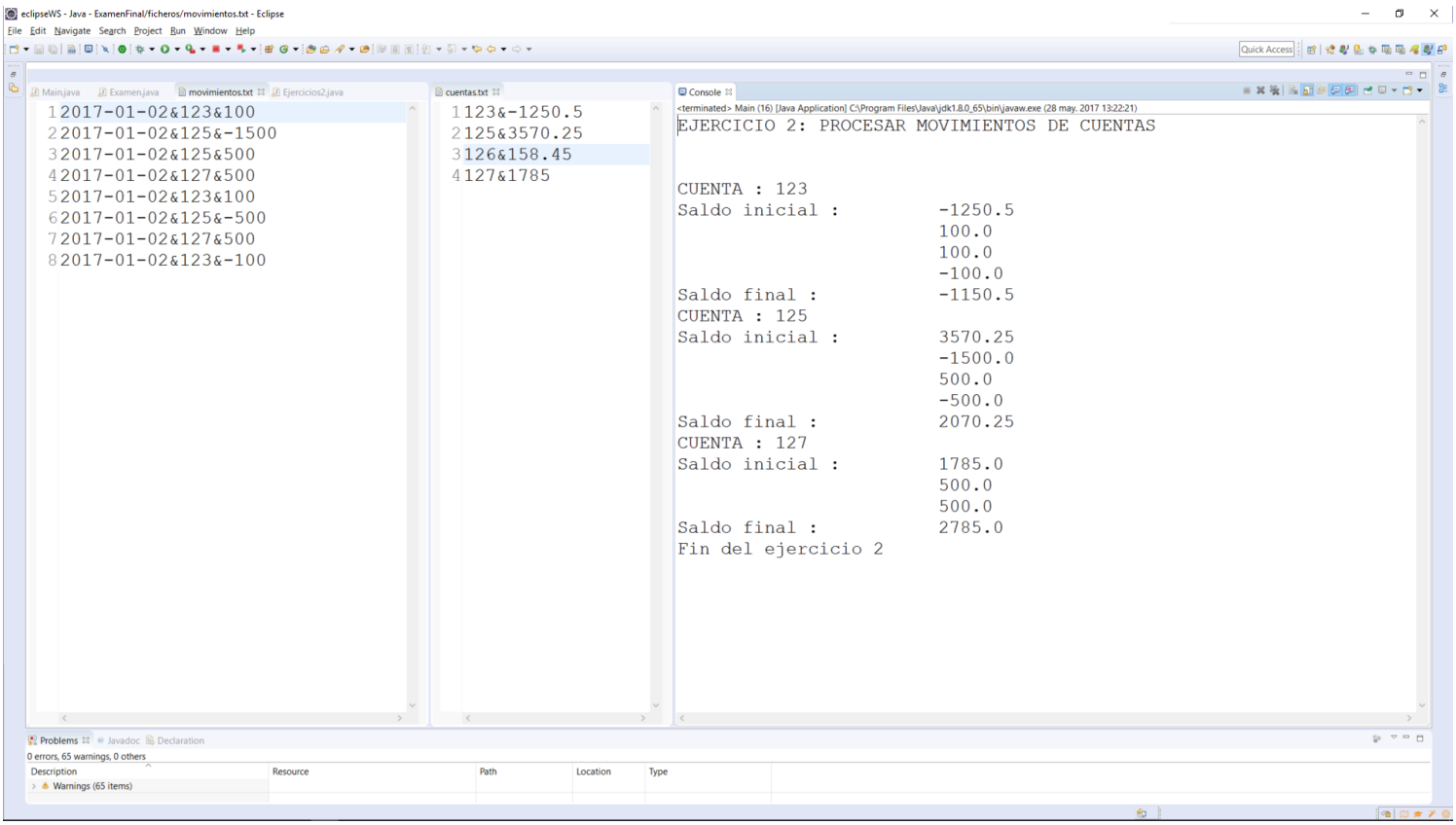
Fichero "**movimientos.txt**":

fecha movimiento (yyyy-MM-dd) , no hay que hacer nada con ella
número de cuenta,
importe (float) , puede ser negativo, valor que se suma al saldo

El fichero de movimientos viene ordenado cronológicamente. Con la fecha **no hay que hacer nada** en este proceso.

Hacer un listado por consola, ordenado por cuenta , que resulta de procesar los movimientos sobre las cuentas, obteniendo el saldo final de cada cuenta que haya tenido movimientos. No todas las cuentas tienen movimientos.

Ejemplo: Observar la imagen siguiente para una prueba con datos concretos en ambos ficheros



El separador de los datos es “&”.
Lo que se pretende es obtener el saldo final de las cuentas, una vez procesado el fichero de cuentas (“cuentas.txt”) contra el fichero de movimientos correspondiente (“movimientos.txt”).
Como se ve en la imagen, pueden haber cuentas sin movimientos.

ORIENTACION:

Para resolver el ejercicio es necesario leer los dos ficheros y almacenarlos en colecciones.

La solución del profesor usó dos colecciones:

```
HashMap<Integer, Float> cuentas
```

```
HashMap<Integer, ArrayList<Float>> movimientos
```

La clave de los dos HashMap , de tipo Integer, representa el código de la cuenta.

El valor en cuentas es el saldo inicial de la cuentas, cargado de “cuentas.txt”

El valor en movimientos representa la lista de movimientos de esa cuenta, procedente del fichero “movimientos.txt”.

Recuerden que la orientación no es obligatoria. Hay que conseguir que salga por consola lo que se ve en la imagen anterior, da igual como se haga.

EJERCICIO 3
3 puntos

A pàrtir de la tabla “libros”, de la base de datos “tunombreExamen” , que debes crear previamente, se pide un método que devuelva una lista de libros por cada tema.

Restaura la base de datos en tu servidor , usando el fichero que se encuentra en la carpeta “ficheros” con el nombre “examenfinal.sql”.

El prototipo del método debe ser:

public HashMap <Integer, ArrayList<Libro>> getLibrosPorTemas ()

La clave en el HashMap representa el código del tema.

La consulta sql es simple:

SELECT * FROM libros

Identifica la columna donde está el tema (la segunda). Esa es la clave en el HashMap. Recuerda que "al principio no hay nada". Por tanto , comprueba si existe la lista de libros para el tema leido. Si no existe, la creas (es un ArrayList<Libro> y la añades al HashMap. En clase se resolvió este problema con otra clase.

Un libro pertenece a uno y solo un tema , un tema puede tener muchos libros, es una relación 1:N.

Hay que crear la clase Libro con todas las propiedades que vienen en la tabla "libros". Poner los tipos adecuados (int, String, char, float,LocalDate).

Para asignar la propiedad fecha , usa lo siguientes

```
libro.setFecha(LocalDate.parse(rS.getString(10)));
```

Para comprobar que el método devuelve lo que se espera de él, usar el Debugger.

S U E R T E