

Programación II - 2010/2011 - Universitat Jaume I

Evaluación continua - Sesiones 3 a 5 de prácticas - Grupo A

22 de marzo de 2011

La duración máxima de esta prueba es de 45 minutos. No puedes consultar libros ni apuntes.

Ejercicio

Supongamos que tenemos dos ficheros de texto con información de una sucursal bancaria:

- El primer fichero contiene información de los clientes con saldo negativo en su cuenta corriente. Cada línea contiene, en este orden, el DNI de un cliente, el saldo y su nombre. La información del fichero está **ordenada** de menor a mayor DNI sin que existan DNIs repetidos. Se muestran a continuación algunas líneas de ejemplo:

```
10107580B    -2523,43    Perez Cacho, Germán
10221259I    -3066,36    Ortega San, Adrian
10233736B    -2448,90    Mrkonjic Lopez-Malla, Lorenzo
```

Como puedes ver, este formato coincide con el de las sesiones de prácticas.

- El segundo fichero contiene los DNIs de todos los clientes que tienen domiciliada su nómina en esa sucursal, tanto los que tienen saldo negativo como los que no. Cada línea contiene únicamente un DNI. Esta información también está **ordenada** de menor a mayor DNI, sin repetidos.

Queremos generar un nuevo fichero de texto que sea un subconjunto del primero, con las líneas de aquellos clientes que tienen saldo negativo y, además, tienen domiciliada su nómina en esa sucursal. La información en este nuevo fichero también debe quedar **ordenada** de menor a mayor DNI.

Escribe en lenguaje Java un método que reciba como parámetros los nombres de los tres ficheros y genere el nuevo fichero descrito.

Por simplicidad, puedes suponer que los ficheros de datos existen, puedes leerlos y tienen el formato esperado y tienen al menos una línea

NOTA. En la evaluación tendrán una penalización importante las soluciones cuyo tiempo de ejecución no sea lineal, puesto que los ficheros de datos están **ordenados**. También se penalizará el uso de vectores innecesarios.

Recordatorio sobre la gestión de ficheros de texto

Abrir fichero para lectura	<code>Scanner miEntrada = new Scanner(new File(miNombreFichero));</code>
Abrir fichero para escritura	<code>Formatter miSalida = new Formatter(new File(miNombreFichero));</code>
Averiguar si quedan datos	<code>boolean quedanDatos = miEntrada.hasNext();</code>
Leer palabra	<code>String miCadena1 = miEntrada.next();</code>
Leer flotante	<code>double miFlotante = miEntrada.nextDouble();</code>
Leer hasta fin de línea	<code>String miCadena2 = miEntrada.nextLine();</code>
Escribir datos	<code>miSalida.format("%s %f %s%n", miCadena1, miFlotante, miCadena2);</code>
Cerrar fichero	<code>miEntrada.close(); miSalida.close();</code>