	IES LUIS VIVES - 1º DAW - EXAMEN Marzo-2º eval. Módulo "Programación"	Curso 2017/2018 Fecha:9/3/2018
Nombre y Apellidos		Nota:

Parte de Programación (8 ptos/10 ptos)

Ejercicio 1 (3 puntos)

Se dispone de un **fichero de texto (productos.txt)** con varias líneas en el que **cada línea** tiene un **código de producto** (número entero) seguido de cero, uno o **varios precios** de adquisición de ese producto (números reales siempre con punto decimal. Cada dato está separado del siguiente por un punto y coma (;). Realizar un programa que :

1.- Suponiendo que en el fichero sólo puede haber los siguientes **tipos de errores**:

ERROR A- Un código de **producto** no tiene **ningún precio** válido a continuación del mismo

ERROR B- Un código de **producto** o un **precio** no sean, respectivamente, un número **entero** o un **real**.

Muestre en pantalla las líneas con error con el formato.

Línea Número de la línea: tipos de errores en esa línea. -> contenido de la línea

Si hay varios errores del mismo tipo sólo se reflejará uno de ellos pero si hay errores de **distintos tipos** se **informará de todos** ellos.

2.- Muestre por pantalla los códigos de productos y el precio medio de adquisición de los mismos sin tener en cuenta los productos de líneas con error.

Se supondrá que nunca un precio de un producto está en una línea distinta a la del código del producto por tanto **las líneas siempre empiezan por un código** de producto (válido o no).

Ejemplo:

Para un fichero con el contenido siguiente	Se deberá mostrar :
34;3.4;3.6 23;3..33; 45;33.0 567 33; 23451;22.0;23.0;24.0 3.4aa;3.4	Producto:34 3.5 línea 2:Error B Error A ->23;3..33; Producto:45 33.0 línea 4:Error A ->567 línea 5:Error A ->33; Producto:23451 23.0 línea 7:Error B ->3.4aa;3.4

Ejercicio 2 (4 puntos)

Se dispone de un fichero **locales.dat** en el que están almacenados los **locales** comerciales de un **centro comercial**. El fichero se ha creado utilizando la siguiente clase y método escribir de la clase Local:

```
public class Local {
    private static final int TAMANODNI = 10;
    int id; //identificador
    double superficie;
    boolean horarioAmpliado; //true si está más tiempo abierto.
    String DNIDueño;
    byte [] pasarStringaArrayBytes(String nombre){
        byte nombreB[]=new byte[TAMANODNI];
        for (int i=0; i<nombre.length() && i<TAMANODNI; i++)
            nombreB[i]=(byte) nombre.charAt(i);
        for (int i=nombre.length(); i<TAMANODNI; i++)
            nombreB[i]=(byte) 0;
        return nombreB;
    }
    void escribir (RandomAccessFile f) throws IOException {
        byte dniB[];
        dniB=pasarStringaArrayBytes(DNIDueño);
        f.writeInt (this.id);
        f.writeDouble(superficie);
        f.writeBoolean (horarioAmpliado);
        f.write(dniB);
    }
}
```

Donde el dniB se graba en el fichero con un array de bytes de tamaño 10.

Suponiendo que el fichero anterior existe y tiene datos válidos y sin errores y sabiendo que el **local de identificador id está colocado en la posición id del fichero** (es decir los datos del local de id=0 está en la posición 0 y los datos del local de id=1 estarán justo después de los del local de id=0 y así sucesivamente). Realizar una aplicación que permita realizar las siguientes opciones:

1.-**Modificar el horario ampliado.** Se pedirá el id del local, se mostrará por pantalla si ese local tiene horario ampliado o no, se pedirá al usuario si desea cambiar el valor del horario ampliado y si responde afirmativamente se grabará en el fichero el nuevo valor de horarioAmpliado de ese local.

Si el número de local introducido por el usuario no existe en el fichero se mostrará el mensaje “Ese local no existe” y se volverá al menú.

2.- **Mostrar la superficie total y el número de locales de un determinado dueño.** Se pedirá el dni del dueño y se mostrará en pantalla en número total de locales que tienen ese dni como dueño y la superficie acumulada de todos ellos

3.-**Calcular cuota.** Pedirá la cantidad total de gastos del centro comercial por teclado y creará un fichero de texto *repartoGastos.txt* en el que se almacenará el id del local, un carácter de @ y la cuota a pagar por ese local para afrontar los gastos comunes. La cuota de cada local se calculará teniendo en cuenta que el 90% de los gastos comunes se paga en proporción a la superficie del local y el 10% restante lo pagan los locales, independientemente de su superficie, que tengan horario ampliado.

Es decir, si el total de gastos comunes son 200€ y en el centro comercial hay los siguientes locales (enumerados por id, superficie y valor del atributo horarioAmpliado

0 30 true ..

1 40 false ..

2 10 true

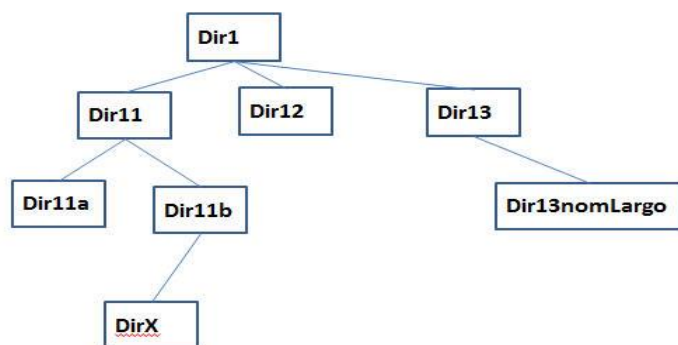
Como el 90% de los 200€ es $0,90 * 200 = 180€$ se paga en función de la superficie que es de $30 + 40 + 10 = 80m^2$ luego $180/80 = 2,25€$ por m^2

Además el 10% de 200 es $0,1 * 200 = 20€$ y hay 2 locales que tiene horario ampliado cada uno pagará $20€ / 2 = 10€$ por tanto el local 0 pagará $2,25€ * 30 m^2 + 10€ = 77,5€$, el local 1 pagará $2,25€ * 40 m^2 = 90€$ y el local 2 pagará $2,25€ * 10 m^2 + 10€ = 32,5€$

Todas las operaciones se realizarán directamente sobre el fichero. No se utilizarán arrays para almacenar los datos que hay en el fichero.

Ejercicio 3 (1 puntos)

Realizar un programa que reciba un nombre de directorio por teclado y muestre en pantalla la profundidad del subdirectorio que esté a más pasos del recibido como parámetro. Es decir, si se recibiera el siguiente directorio:



Mostrará: 3 que es el número de pasos desde Dir1 hasta DirX (Dir1 -> Dir11 -> Dir11b -> DirX) que es el directorio más alejado de Dir1 (el que está más profundo en la jerarquía).

Nota para todos los ejercicios:

Si fuera necesario leer de teclado o escribir en pantalla, sólo se leerán datos de teclado y se escribirán en pantalla en el programa principal, método main) o en métodos específicos para ello (métodos mostrar o pedir)

Se podrán realizar métodos y clases adicionales a los indicados si fuera necesario.

No se utilizarán métodos static excepto el main