

Archivos de texto II

1 Campos de datos

¿Cómo harías un programa que determine a partir del archivo "productos-precios.txt" el producto más barato? En este caso se presenta el problema de cómo separar el contenido de cada línea en nombre del producto y precio. Si se lee la primera línea mediante `con.readLine` se lee la línea completa y se guarda en una sola variable `String`, que habría que "quebrar" para trabajar con sus componentes.

La solución consiste en escribir el archivo delimitando los datos por un caracter especial como dos puntos (:). Para ello es necesario cambiar la línea de la guía anterior, donde se creaba el archivo, por

```
escr.print(prod+":"+precio);
```

De esta forma, el archivo en disco tendrá la siguiente estructura:

```
Productos-Precios.txt
jabón Lux:344
pasta de dientes Odontine:415
champú Sedal:507
...
```

En este tipo de archivos los datos que quedan delimitados por ":" se denominan campos (fields). Esta estrategia es muy útil para poder incluir varios datos en una misma línea y después poder separarlos cómodamente ubicando la posición de los ":".

A modo de ejemplo:

```
String linea= lect.readLine();
while (linea!=null) {
    int pos=0;
    while (linea.charAt(pos)!=':')
        pos++;
    String prod= linea.substring(0, pos);
    int precio=Integer.parseInt(linea.substring(pos+1));
    ...
}
```

2 La clase StringTokenizer

Para separar las líneas en sus múltiples campos, utilizaremos la clase `StringTokenizer`. Esta clase provee de objetos capaces de obtener uno a uno los campos de datos que hay en un `String`. Por ejemplo, supongamos

que el `String` `linea` contiene el nombre de un producto, su precio y las unidades disponibles, todos separados por ":". Para obtener estos campos se escribe el siguiente código:

```
StringTokenizer st=new StringTokenizer(linea,":");
String prod=st.nextToken();
int precio=Integer.parseInt(st.nextToken());
int unidades=Integer.parseInt(st.nextToken());
```

Con esta clase, el siguiente programa muestra en pantalla el contenido del archivo "Productos-Precios.txt":

```
import java.util.*;
...
BufferedReader lect= new BufferedReader(new FileReader("Productos-
String linea= lect.readLine();
while (linea!=null) {
    StringTokenizer st=new StringTokenizer(linea,":");
    String prod=st.nextToken();
    int precio=Integer.parseInt(st.nextToken());
    int unidades=Integer.parseInt(st.nextToken());
    //muestra en pantalla producto y precio
    con.println(prod+" "+precio);
    linea= lect.readLine();
}
lect.close();
```

2.1 Cuando no usar StringTokenizer

Algunas veces la información viene en campos de ancho fijo. Es decir, todas las líneas tienen el mismo ancho y el campo 1 comprende las columnas 1 a la `n`, el campo 2 las columnas `n+1` a `p`, el campo 3 las columnas `p+1` a `q`, etc. En este caso es más simple descomponer los datos usando `substring` directamente. Por ejemplo, si el archivo guarda además las unidades, el producto usará las 15 primeras columnas, el precio las 5 siguientes y las unidades las 4 últimas, como se ve a continuación

```
Productos-Precios.txt
Jabón Sussio    250  12
Cerveza Pin    1100 4311
Diario El Hoci120 8762
```

el programa sería

cuantas veces lo hace. Esta solución es bastante complicada... ¿cómo se puede simplificar?

```

1  BufferedReader lect= new BufferedReader(new FileReader("Mis Productos.txt"));
2  String linea= lect.readLine();
3  while (linea!=null) {
4      String prod= linea.substring(0,15);
5      int precio= Integer.parseInt(linea.substring(15,20).trim());
6      int unidades= Integer.parseInt(linea.substring(20).trim());
7      con.println(prod+" "+precio+" "+unidades);
8      linea= lect.readLine();
9  }
10 lect.close();

```

3 Problemas resueltos

En el archivo "nombres.txt" se encuentran todos los nombres de los alumnos de la sección 3.

3.1 Contar los alumnos

Escribe un programa que escriba en el archivo "cuantos.txt" los nombres de la sección y en la última línea la cantidad de alumnos.

3.1.1 Solución

```

1  BufferedReader lector=new BufferedReader(new FileReader("nombres.txt"));
2  con.printWriter escritor=new con.printWriter(new FileWriter("cuantos.txt"));
3  int total=0;
4  String nombre=lector.readLine();
5  while (nombre!=null) {
6      escritor.println(nombre);
7      nombre=lector.readLine();
8      total=total+1;
9  }
10 lector.close();
11 escritor.println(total);
12 escritor.close();

```

3.2 Contar nombre más común

Suponiendo que el archivo se encuentra en orden alfabético, escribe un programa que diga cuál es el nombre que más se repite en la sección y

3.2.1 Solución

```

1  int masRepetida=0;
2  BufferedReader lector=new BufferedReader(new FileReader("nombres.txt"));
3  String nombre=lector.readLine();
4  String nombreMasRepetido=nombre;
5  int repeticiones=0;
6  String exnombre=nombre;
7  while (nombre!=null) {
8      if (nombre.equals(exnombre)) {
9          repeticiones++;
10         if (repeticiones>masRepetida) {
11             masRepetida=repeticiones;
12             nombreMasRepetido=nombre;
13         }
14     } else
15         repeticiones=1;
16     exNombre=nombre;
17     nombre=lector.readLine();
18 }
19 con.print ("El nombre mas común es "+nombreMasRepetido);
20 con.println (" y aparece "+masRepetida+" veces");

```