Práctica 6 3T
1º DAM PROGRAMACIÓN
Ficheros

1. Como sabemos java dispone de la clase genérica *java.util.Vector* que sirve para almacenar elementos del mismo tipo. Queremos escribir una clase *VectorStrings*, parte de un paquete *utilidades*, específica para almacenar cadenas de caracteres (Strings). La clase tendrá todos los métodos de la clase vector, además de:

    a. Una constructora por defecto que cree el vector vacío.
    b. Una constructora que reciba como parámetro un *String* conteniendo el nombre de un fichero de texto, e inicialice el vector con las líneas de texto del fichero (cada línea será un elemento del vector). Si el fichero no existe la constructora creará el vector vacío (igual que la constructora por defecto).
    Nota: Para el manejo de ficheros de texto en Java ver los ejemplos al final de la práctica.
    c. Un método *escribe* que grabará el contenido del vector en un fichero de texto, cuyo nombre recibe el método como parámetro. Cada elemento del vector se convertirá en una línea del fichero.
    d. Un método *inserta* que reciba como parámetro un String y lo añada al vector en el lugar que le corresponda por orden alfabético.
    Pista: El método *compareTo* de la clase *String* recibe como parámetro un *String* y lo compara alfabéticamente con el *String* actual. Una llamada de la forma *s1.compareTo(s2)* devuelve 0 si s1 y s2 son iguales, un número
    <0 si s1 es menor que s2 alfabéticamente, y un número >0 si s1 es mayor alfabéticamente que s2.

2. Escribir un programa *Principal* para probar la clase anterior. El programa:
    a. Declarará un objeto de tipo *VectorStrings* pasándole en la constructora el nombre de un fichero de texto que habremos creado previamente (por ejemplo "C:\\datos.txt").
    b. Añadirá una frase cualquiera al vector mediante el método *inserta*.
    c. Mostrará el contenido del vector por pantalla (un elemento por línea).
    d. Grabará (usando el método *escribe*) el vector en un fichero con otro nombre (por ejemplo "*c:\\datos2.txt*"). Si los métodos están correctamente realizados el nuevo fichero debe ser una copia del original con la frase añadida en su lugar correcto.

Recordatorio de algunos métodos de la clase Vector: Si v es un vector...

- v.size(): número de elementos del vector.
- v.addElement(e): Añade e al final de v
- v.elementAt(n): Devuelve el elemento que ocupa la pos. n.
- v.insertElementAt(e,n): Inserta e en la pos. n.

Manejo de ficheros de texto en Java

Java ofrece muchas clases y variantes para leer/escribir en un fichero. En este apartado nos limitaremos al caso más sencillos: los ficheros de texto de acceso secuencial (se dice secuencial porque tenemos que ir leyendo o escribiendo en secuencia, comenzando desde el primer elemento del fichero en adelante)

Lectura línea a línea

El procesamiento más sencillo de un fichero de texto es línea a línea. Es decir se lee una línea, se trata, y se repite el proceso hasta que no queden más líneas por leer. He aquí un fragmento de código que hace esto para un fichero de nombre ''datos.txt'':

```
import java.io.*; // para utilizar las clases y excepciones de fichero

.....

tr
y  // así se declara el fichero para leer de él
{  BufferedReader fichero = new BufferedReader(new

   FileReader("c:\\datos.txt")); String línea =

   fichero.readLine();

   // cuando ya no se puede leer del fichero readLine devuelve null
   while (línea != null) {
     // procesar como se quiera la línea. Aquí me
     limito a escribirla por pantalla
     System.out.println(línea);
     línea = fichero.readLine();
```

```
   }
   // al acabar siempre hay que cerrar el
   fichero fichero.close();
  } catch (IOException e) {
   // aquí se pondrá el tratamiento de errores por si no se puede
   leer
   // (por ejemplo si el fichero no
   existe) System.out.println("No puedo
   abrir el fichero!!");
  }
```

Escritura

Para escribir en un fichero podemos usar un código como el siguiente:

```
import java.io.*;
…..
  try{
    PrintWriter fichero = new PrintWriter(new FileWriter("datos.txt"));

    fichero.println ("Apenas él le amalaba
    el noema"); fichero.println ("a ella
    se le agolpaba el clémiso");
    fichero.println ("y caían en hidromurias, en
    salvajes ambonios,"); fichero.println ("en sustalos
    exasperantes.  ");
    // no olvidad cerrar el fichero al acabar!!
    fichero.close();
  } catch (IOException e) {
   //.      tratamiento de los errores
  }
```

En este código se abre para escritura el fichero *datos.txt* y se escribe con el método *println* de la clase *PrintWriter*. También se puede usar *print* si no se quiere que se ponga un fin de línea al final de lo escrito). Además de strings se pueden escribir con *println* enteros, números reales, etc. Es muy importante observar que si el fichero ya existe se borra y se crea de nuevo, perdiéndose su antiguo contenido.