PAC 4. UF1. - Desarrollo

Acceso a datos.

Aníbal Santos Gómez

1. Ejercicio 1.

Responde a este test:

1. ¿Cuál de estas opciones consideras una gran ventaja de los ficheros?

a. Es volátil.

b. Permite control sobre los datos

c. Mayor disponibilidad de los datos que en una base de datos

d. Ninguna de las opciones anteriores es correcta

2. El método list() de la clase File devuelve:

a. Boolean

b. String[]

c. String

d. File[]

3. ¿Con qué método de la clase File podemos conocer el número de líneas de un fichero de texto?

a. isFile()

b. length()

c. exists()

d. No existe un método para esto, es necesario recorrer el fichero.

4. ¿Qué operación no podremos realizar en un fichero? Selecciona la opción d si consideras que todas las opciones son correctas.

a. Alta

b. Modificación

c. Buscar

d. Se pueden realizar todas las operaciones anteriores

5. ¿Cuál de estos métodos no pertenece a la clase FileReader?

a. int read (char[] buf)

b. int read (char[] buf, int desplazamiento, int numCaracteres)

c. int read (byte[] b)

d. Todas las opciones anteriores son métodos de la clase FileReader

6. ¿El lenguaje XML tiene la misma función que HTML?

a. Sí

b. No

c. En un 90%

d. Ninguna de las respuestas anteriores es cierta

7. ¿Qué son DOM y SAX?

a. Variantes de XML

b. Metalenguajes

c. Lenguajes de marcas

d. Analizadores de documentos XML

8. ¿Cuál de estas afirmaciones sobre DOM no es cierta?

a. DOM consume más memoria que SAX

b. DOM almacena la estructura en forma de árbol

c. El mismo procesador DOM define métodos para generar un archivo XML a partir de un árbol DOM

d. Tiene origen en el W3C

9. ¿Cuál de estas clases o interfaces no pertenece a SAX?

a. InputSource

b. DocumentBuilderFactory

c. XMLReader

d. Ninguna de las 3 anteriores

10. ¿A qué hace referencia XSL?

a. Recomendaciones para expresar hojas de estilo en lenguaje XML

b. eXtensible Stylish Language

c. Una variante de XML

d. Un parser de documentos XML

RESPUESTAS:

1 -D

2 - B

3 - D

4 - D

5 - C

6 - B

7 - D

8 - C

9 - B

10 - A

2. Ejercicio. Realiza una comparación de los archivos según su método de acceso.

Los archivos con acceso secuencial tienen la particularidad de que los datos y el fichero se leen y se escriben de forma ordenada. Si queremos acceder a un dato localizado en una parta concreta del fichero, tendremos que leer todos los datos del primero hasta el punto de dicho dato. La clase que utilizaremos para realizar la busqyeda dependerá del tipo de archivo, si es binario utilizaremos FileInputStream y FileOutputStream. Si el fichero es de texto entonces tendremos Filer Reader y FileWriter.

Por el contrario en los archivos cuyo acceso es directo o aleatorio no tendremos la necesidad de leer todo el contenido, sino que accederemos directamente al registro buscado. La clase que utilizaremos para la búsqueda es RandomAccessFile.

3. Ejercicio. Diferencia entre los ficheros de texto y los ficheros binarios.

En los archivos de texto se suele utilizar de forma genérica un formato como ASCII, UTF8, o UNICODE, que se utilizarán para almacenar los caracteres. Utilizaremos las clases FileReader para leer y FileWriter para escribir estos caracteres.

En los archivos binarios, almacenaremos secuencias de dígitos que son ilegibles, mediante este tipo de archivos. Utilizaremos las clases FileInputStream y FileOutputStream.

4. Ejercicio. Enumera las diferencias más importantes entre DOM y SAX.

DOM almacena la estructura del documento en memoria en forma de árbol y recorre los diferentes nodos analizando a que tipo particular pertencen. Tiene un mayor consumo de memoria que SAX y permite tener una visión global del documento.

SAX por el contrario, lee el fichero de forma secuencial produciendo una secuncia de eventos, en función del resultado de la lectura. Tiene un menor consumo de memoria que DOM y no permite ver de forma global el documento.

5. Ejercicio. Describe las interfaces más habituales que se utilizan en DOM .

Las interfaces más habituales en DOM son:

- Document, que crea nuevos nodos.
- Element, que expone propiedades y métodos con los que se pueden manipular los elementos del documento y sus atributos.
- Node, que representa cualquier nodo.
- Nodelist que lista con nodos hijos de un nodo.
- Attr, que accede a atributos de un nodo.

- Text, que proporciona los datos de carácter de un elemento.
- CharacterData, que proporciona atributos o métodos para manipular datos de caracteres.
- DocumentType, que proporciona la información contendia en la etiqueta <!DOCTYPE>.

6. Ejercicio.

Realiza un programa en Java que cree un fichero binario para guardar la información sobre los alumnos de Ilerna Online. Es necesario conocer el identificador, el nombre del alumno y el número de asignaturas escogidas. El programa debe tener un menú que permita:

- **Crear el fichero con nombre alumnos.dat**
- **Introducir un alumno**
- **Modificar el número de asignaturas matriculadas**
- **Eliminar un alumno**
- **Mostrar datos**
- **Salir Utiliza la parte de excepciones para realizar todas las validaciones de datos que consideres oportunas.**

```
import java.io.File;
import java.io.FileNotFoundException;
import java.io.IOException;
import java.io.RandomAccessFile;
import java.util.Scanner;


public class Alumni {
```

```java
    public static void main(String[] args) throws IOException {

        RandomAccessFile file=null;
        int option = 0;


        /* Inicializamos la lectura para el menú y creamos el menú con el
condicional do while */

        Scanner typing = new Scanner(System.in);
        do {
            do {
                System.out.println("Elige una opción para continuar:");
                System.out.println("1. Crear fichero");
                System.out.println("2. Introduce alumno");
                System.out.println("3. Modificar asignaturas matriculadas");
                System.out.println("4. Eliminar alumno");
                System.out.println("5. Mostrar datos");
                System.out.println("6. Salir");
                option = typing.nextInt();

            }while(option<1 || option>6);

        /* Utlizaremos switch para abordar los casos del menú que dirigiremos a
las funciones correspondientes, listadas a continuación*/

            switch (option) {
            case 1:
                file = createFile(".\\alumnos.dat");
                break;
            case 2:
                createAlumn(file);
                break;
            case 3:
                alterData(file);
                break;
            case 4:
                deleteData(file);
                break;
            case 5:
                showData(file);
                break;
            case 6:
                file.close();
                break;
            }
```

```
        }while(option!=6);
        typing.close();
    }


    /* 1 - Creamos el fichero con el nombre de "alumnos.dat". */

    public static RandomAccessFile createFile(String file) throws
FileNotFoundException {
        File fichero = new File(file);
        return new RandomAccessFile(fichero,"rw" );
    }

    /* 2 - Introducimos un alumno. */

    public static boolean createAlumn(RandomAccessFile file) throws IOException {
        Scanner scan = new Scanner(System.in);
        int id = 0, course= 0, position;
        String name;

        position = (id - 1) * 48;

        System.out.println("Introduce un nombre (Máximo 25 caracteres): ");
        name = scan.next();

        if (name.length()>25) name = name.substring(0, 25);
        System.out.println("Introduce un número (entero) de asignaturas: ");
        course = scan.nextInt();

        try {
            file.seek(position);
            file.writeInt(id);

            StringBuffer buffer = new StringBuffer(name);
            buffer.setLength(25);
            file.writeChars(buffer.toString());
            file.writeInt(course);
            return true;

        } catch (IOException exception) {
            return false;
        }

    }

    /* 3 - Modificamos el número de asignaturas matriculadas. */

    public static void alterData(RandomAccessFile file) throws IOException {
```

```
        Scanner scan = new Scanner(System.in);
        int id, course;

        System.out.println("Introduce el id del alumno a modificar: ");
        id = scan.nextInt();

        System.out.println("Introduce el nuevo número (entero) de alumnos: ");
        course = scan.nextInt();

        int position = 0;
        position = (id - 1) * 48;

        if(position >= file.length()) {
            System.out.println("La asignatura introducida no existe.");
        } else {
            file.seek(position + 24);
            file.writeInt(course);
            file.close();
            System.out.println("Cambios realizados con éxito");
        }
    }

    /* 4 - Eliminamos un alumno. */

    public static void deleteData(RandomAccessFile file) throws IOException {
        Scanner scan = new Scanner(System.in);
        int id ;

        System.out.println("Introduce el id del alumno a modificar: ");
        id = scan.nextInt();

        int position = 0;
        position = (id - 1) * 48;

        if (position > file.length()) {
            System.out.println("La asignatura introducida no existe.");
        } else {
            file.seek(position);
            file.writeInt(0);
            file.writeChars("");
            file.writeInt(0);
            System.out.println("Eliminación realizada");
        }
    }

    /* 5 - Mostramos los datos. */

    public static void showData(RandomAccessFile randomFile) throws IOException{
```

```
        String name = "";
        int position=0, id , alumns;
        char ch;

        if(randomFile.length()>0)
        {
            while(true){
                randomFile.seek(position);

                id = randomFile.readInt();

                if(id == ((position/48)+1))
                {
                    //Leemos la cadena
                    for(int i=0;i<20;i++){
                        ch = randomFile.readChar();
                        if((int)ch>0 && (int)ch<128)
                            name +=ch;
                    }
                    alumns = randomFile.readInt();
                    System.out.println("id: " + id + " Nombre: "+ name +"
Alumnos: "+ alumns);
                    name = "";
                }
                position += 48;
                if(position == randomFile.length()) break;
            }
        }

    }

    /* 6 - Salimos del programa. */
}
```