**Ficheros de Acceso Aleatorio en Java. Clase RandomAccessFile**

La clase Java RandomAccessFile se utiliza para acceder a un fichero de forma aleatoria.

Los constructores de la clase son:

RandomAccessFile(String path, String modo);

RandomAccessFile(File objetoFile, String modo);

Lanzan una excepción **FileNotFoundException**.

El argumento ***modo*** indica el modo de acceso en el que se abre el fichero.

Los valores permitidos para este parámetro son:

|  |  |
| --- | --- |
| **modo** | **Significado** |
| "r" | Abre el fichero en **modo solo lectura**.  El fichero debe existir.  Una operación de escritura en este fichero lanzará una excepción IOException. |
| "rw" | Abre el fichero en **modo lectura y escritura**. Si el fichero no existe se crea. |

**Ejemplo**: abrir un fichero aleatorio para lectura

Se abre el fichero clientes.dat para lectura usando el primer constructor.

RandomAccessFile fichero = new RandomAccessFile("/ficheros/clientes.dat", "r");

**Ejemplo**: abrir un fichero aleatorio para lectura/escritura

Se abre el fichero personas.dat para lectura/escritura usando el segundo constructor. Si el fichero no existe se crea.

File f = new File ("/ficheros/personas.dat");

RandomAccessFile fichero = new RandomAccessFile(f, "rw");

**ACCESO A LOS DATOS EN FICHEROS ALEATORIOS**

Para acceder de forma aleatoria a los datos contenidos en el fichero, la clase RandomAccessFile dispone de varios métodos. Entre ellos:

**long getFilePointer();**

Devuelve la posición actual del puntero del fichero. Indica la posición (en bytes) donde se va a leer o escribir.

**long length();**

Devuelve la longitud del fichero en bytes.

**void seek(long pos);**

Coloca el puntero del fichero en una posición *pos* determinada. La posición se da como un desplazamiento en bytes desde el comienzo del fichero. La posición 0 indica el principio del fichero. La posición *length()* indica el final del fichero.

Además, dispone de métodos de lectura/escritura:

**public int read()**

Devuelve el byte leído en la posición marcada por el puntero. Devuelve -1 si alcanza el final del fichero. Se debe utilizar este método para leer los caracteres de un fichero de texto.

**public final String readLine()**

Devuelve la cadena de caracteres que se lee, desde la posición marcada por el puntero, hasta el siguiente salto de línea que se encuentre.

**public xxx readXxx()**

Hay un método read para cada tipo de dato básico: **readChar**, **readInt**, **readDouble**, **readBoolean, etc**.

**public void write(int b)**

Escribe en el fichero el byte indicado por parámetro. Se debe utilizar este método para escribir caracteres en un fichero de texto.

**public final void writeBytes(String s)**

Escribe en el fichero la cadena de caracteres indicada por parámetro.

**public final void writeXxx(*argumento*)**

También existe un método write para cada tipo de dato básico: **writeChar**, **writeInt**, **writeDouble**, **writeBoolean, etc**.

**Ejemplos de operaciones con ficheros de acceso aleatorio**

**Ejemplo 1:**Programa Java que pide un número entero por teclado y lo añade al final de un fichero binario *enteros.dat* que contiene números enteros. El programa utiliza un método mostrarFichero() que se llama dos veces. La primera muestra el contenido del fichero antes de añadir el nuevo número y la segunda llamada muestra el fichero después de añadirlo.

package random1;

import java.io.EOFException;

import java.io.FileNotFoundException;

import java.io.IOException;

import java.io.RandomAccessFile;

import java.util.Scanner;

public class Random1 {

static Scanner sc = new Scanner(System.in);

static RandomAccessFile fichero = null;

public static void main(String[] args) {

int numero;

try {

//se abre el fichero para lectura y escritura

fichero = new RandomAccessFile("/ficheros/enteros.dat", "rw");

mostrarFichero(); //muestra el contenido original del fichero

System.out.print("Introduce un número entero para añadir al final del fichero: ");

numero = sc.nextInt(); //se lee el entero a añadir en el fichero

fichero.seek(fichero.length()); //nos situamos al final del fichero

fichero.writeInt(numero); //se escribe el entero

mostrarFichero();//muestra el contenido del fichero después de añadir el número

} catch (FileNotFoundException ex) {

System.out.println(ex.getMessage());

} catch (IOException ex) {

System.out.println(ex.getMessage());

} finally {

try {

if (fichero != null) {

fichero.close();

}

} catch (IOException e) {

System.out.println(e.getMessage());

}

}

}

public static void mostrarFichero() {

int n;

try {

fichero.seek(0); //nos situamos al principio

while (true) {

n = fichero.readInt(); //se lee un entero del fichero

System.out.println(n); //se muestra en pantalla

}

} catch (EOFException e) {

System.out.println("Fin de fichero");

} catch (IOException ex) {

System.out.println(ex.getMessage());

}

}

}

**Ejemplo 2:**Programa Java para modificar un entero dentro del fichero enteros.dat con acceso aleatorio.

Para ello se pide la posición que ocupa el entero a modificar dentro del fichero, a continuación, se lee y muestra el valor actual, se pide el nuevo valor y finalmente se escribe el nuevo valor en la posición indicada, modificando de esta forma el valor antiguo por el nuevo.

La posición deberá estar comprendida entre 1 y el número de enteros que contiene el fichero.

La cantidad de enteros que contiene el fichero se calcula asi:

//se asigna a size el tamaño en bytes del fichero

size = fichero.length();

//cada int en Java ocupa 4 bytes. Si dividimos el total de bytes entre 4 obtenemos el número de //enteros que contiene el fichero.

size = size / 4;

package random2;

import java.io.FileNotFoundException;

import java.io.IOException;

import java.io.RandomAccessFile;

import java.util.Scanner;

public class Random2{

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

RandomAccessFile fichero = null;

int pos, numero;

long size;

try {

fichero = new RandomAccessFile("/ficheros/enteros.dat", "rw");

//calcular cuántos enteros tiene el fichero

size = fichero.length();

size = size / 4;

System.out.println("El fichero tiene " + size + " enteros");

//Modificar el entero que se encuentra en una posición determinada

do {

System.out.println("Introduce una posición (>=1 y <= " + size + "): ");

pos = sc.nextInt();

} while (pos < 1 || pos > size);

pos--; //la posición 1 realmente es la 0

//nos situamos en la posición (byte de inicio) del entero a modificar

//en Java un entero ocupa 4 bytes

fichero.seek(pos\*4);

//leemos y mostramos el valor actual

System.out.println("Valor actual: " + fichero.readInt());

//pedimos que se introduzca el nuevo valor

System.out.println("Introduce nuevo valor: ");

numero = sc.nextInt();

//nos situamos de nuevo en la posición del entero a modificar

//esto es necesario porque después de la lectura que hemos realizado para mostrar

//el valor el puntero de lectura/escritura ha avanzado al siguiente entero del fichero.

//si no hacemos esto escribiremos sobre el siguiente entero

fichero.seek(pos\*4);

//escribimos el entero

fichero.writeInt(numero);

} catch (FileNotFoundException ex) {

System.out.println(ex.getMessage());

} catch (IOException ex) {

System.out.println(ex.getMessage());

}finally {

try {

if (fichero != null) {

fichero.close();

}

} catch (IOException e) {

System.out.println(e.getMessage());

}

}

}

}

**Ejemplo 3:**Ejemplo de uso de un fichero de caracteres con acceso aleatorio.

Se pide por teclado una palabra, la busca el fichero de texto texto.txt y la modifica escribiéndola en mayúsculas cada vez que aparece en el fichero.

Para hacer el cambio de la palabra por su equivalente en mayúsculas, el programa lee el fichero por líneas. Para cada línea leída se comprueba si contiene la palabra buscada y si es así se modifica y se sobrescribe la línea completa modificada.

(Lo hacemos como ejercicio)