Ficheros de acceso aleatorio

Clase RandomAccessFile

- Java dispone de la clase **RandomAccessFile** que dispone de métodos para acceder al contenido de un fichero binario de forma aleatoria (no secuencial) y para posicionarnos en una posición concreta del mismo
- Dos constructores para **RandomAccessFile**, ambos pueden lanzar la excepción **FileNotFoundException**:
  - **RandomAccessFile (String nombrefichero, String modoAcceso):** escribiendo ruta al fichero
  - **RandomAccessFile (File objetoFile, String modoAcceso):** con un objeto **File** asociado al fichero
  - El argumento **modoAcceso** puede tener dos valores:
    - r → modo de solo lectura. Si tratamos de escribir se lanzará una IOException
    - rw → modo lectura y escritura. Si el fichero no existe se crea

- Una vez abierto el fichero pueden usarse los métodos readXXX y writeXXX de las clases **DataInputStream** y **DataOutputStream**
- La clase RandomAccessFile maneja un puntero que indica la posición actual en el fichero. Cuando el fichero se crea el puntero se coloca al inicio del mismo. Las sucesivas llamadas a read() y write() desplazarán el puntero según la cantidad de bytes leídos o escritos.
- Los métodos más importantes son:
  - **long getFilePointer()** → Devuelve la posición actual del puntero
  - **void seek (long posicion)** → Coloca el puntero en una *posicion* determinada desde el principio del mismo
  - **void length ()** → Devuelve el tamaño del fichero en bytes. La posición *length()* marca el final del fichero
  - **int skipBytes (int desplazamiento)** → Desplaza el puntero desde la posición actual el número de bytes indicados en *desplazamiento*.

Ejemplo completo

En el siguiente ejemplo, “EscribirFicheroAleatorio”, insertaremos datos de empleados en un fichero. Los datos a insertar se obtienen de varios arrays que se llenan en el programa. Los datos se insertarán de forma secuencial por lo que no será necesario utilizar el método **seek()**. Por cada empleado también se insertará un identificador que coincidirá con el índice+1 con el que se recorren los arrays. La longitud del registro de cada empleado es la misma (36 bytes) y los tipos que se insertan y su tamaño es el siguiente:

- El identificador que es un entero → 4 bytes
- El apellido que es una cadena de 10 caracteres → Puesto que Java utiliza UNICODE16, serán 20 bytes.
- El departamento que es otro entero → 4 bytes
- Un tipo Double para almacenar el salario → 8 bytes

```
import java.io.*;
public class EscribirFicheroAleatorio {
        public static void main (String [] args) throws IOException {
                File fichero = new File ("AleatorioEmpleado.dat");
                RandomAccessFile file = new RandomAccessFile (fichero , "rw");
                String apellido[] = {"Fernández", "Gil", "López", "Ramos"};
                int dep[] =  {10,20,10,10};
                Double salario[] = {1000.45,2400.60,3000.0,1500.56};
                StringBuffer buffer = null; //Buffer para almacenar apellido
                int n = apellido.length; //Número de elementos en el array
                for (int i = 0; i<n; i++) {   //Recorro los arrays
                        file.writeInt (i+1);
                        buffer = new StringBuffer (apellido[i]);
                        buffer.setLength(10); // Fijo en 10 caracteres la longitud del apellido
                        file.writeChars (buffer.toString());
                        file.writeInt(dep[i]);
                        file.writeDouble (salario[i]);
                }
        file.close();  // No olvidarse de cerrar el fichero
        }
}
```

Declaro el fichero de acceso aleatorio

Datos Empleados

i+1 como identificador del registro

Inserciones secuenciales en fichero

El siguiente ejemplo toma el fichero anterior y visualiza todos los registros. El posicionamiento para empezar a recorrer los registros empieza en 0, para recuperar los siguientes registros hay que sumar 36 (tamaño del registro, como hemos visto antes) a la variable utilizada para el posicionamiento.

```
import java.io.*;
public class LeerFicheroAleatorio {
        public static void main(String[] args) throws IOException {
                File fichero = new File ("AleatorioEmpleado.dat");
                RandomAccessFile file = new RandomAccessFile (fichero, "r");
                int id, dep ,posicion;
                Double salario;
                char apellido[]= new char[10], aux;
                posicion =0;
                for ( ; ; ){
                        file.seek (posicion); // Nos posicionamos en posicion
                        id = file.readInt();     // Obtengo identificar de Empleado
                        for ( int i =0; i<apellido.length; i++) {
                                aux = file.readChar(); // Voy leyendo carácter a carácter el apellido y lo guardo
                                apellido[i]=aux;           // en el array apellido
                        }
                        String apellidos = new String (apellido);
                        dep = file.readInt();              //Lectura de departamento y salario
                        salario = file.readDouble();
                        if (id >0)
                                System.out.printf("ID: %s, Apellido: %s, Departamento: %d, Salario: %.2f %n", id,
                         apellidos.trim(), dep, salario);
```

Nos situamos al principio del fichero

```
                posicion = posicion + 36;
                if (file.getFilePointer() ==file.length()) break;  // Si he recorrido todo el fichero, salgo
            }                                                      // del for
            file.close();
        }
}
```

Me posiciono para el siguiente empleado. Cada empleado ocupa 36 bytes.

Problemas FicherosAleatorios

1. Implementa los ejemplos anteriores y comprueba su correcto funcionamiento
2. **CONSULTA.** Crea un programa Java que consulte los datos de un empleado del fichero aleatorio. El programa se ejecutará desde la línea de comandos y debe recibir un identificador de empleado. Si el empleado existe se visualizarán sus datos, si no existe se visualizará un mensaje indicándolo.
3. **INSERCION.** Crea un programa Java que inserte datos en el fichero aleatorio. El programa ejecutará desde la línea de comandos y debe recibir 4 parámetros: identificador de empleado, apellido, departamento y salario. Antes de insertar se comprobará si el identificador existe, en ese caso se debe visualizar un mensaje indicándolo; si no existe se deberá insertar.

4. **MODIFICACION.** Crea un programa Java que reciba desde la línea de comandos un identificador de empleado y un importe. Se debe realizar la modificación del salario. La modificación consistirá en sumar al salario del empleado el importe introducido. El programa debe visualizar el apellido, el salario antiguo y el nuevo. Si el identificador no existe se visualizará mensaje indicándolo.
5. **BORRADO.** Crea un programa Java que al ejecutarlo desde la línea de comandos reciba un identificador de empleado y lo borre. Se hará un borrado lógico marcando el registro con la siguiente información: el identificador será igual a -1, el apellido será igual al identificador que se borra, y el departamento y salario serán 0.