
Fitxers d'accés aleatòri

Fluxes a fitxers

Introducció

- Java disposa de la classe RandomAccessFile que conté mètodes per accedir al contingut d'un fitxer binari de manera aleatòria (no seqüencial) i per a posicionar-nos en una posició concreta del mateix.
- Dos constructors per a RandomAccessFile, tots dos poden llançar l'excepció FileNotFoundException:
 - RandomAccessFile (String nomFitxer, String modeAcces): escrivint ruta al fitxer
 - RandomAccessFile (File objecteFile, String modeAcces): amb un objecte File associat al fitxer
 - L'argument modeAcces pot tenir dos valors:
 - r → manera de només lectura. Si tractem d'escriure es llançarà una IOException
 - rw → manera lectura i escriptura. Si el fitxer no existeix es crea

Introducció

- Una vegada obert el fitxer poden usar-se els mètodes `readXXX` i `writeXXX` de les classes `DataInputStream` i `DataOutputStream`
- La classe `RandomAccessFile` controla un punter que indica la posició actual en el fitxer. Quan el fitxer es crea el punter es col·loca a l'inici d'aquest. Les successives crides a `read()` i `write()` desplaçaran el punter segons la quantitat de bytes llegits o escrits.
- Els mètodes més importants són:
 - `long getFilePointer()` → Retorna la posició actual del punter.
 - `void seek (long posicion)` → Col·loca el punter en una `posicion` determinada des del principi del mateix.
 - `void length ()` → Retorna la grandària del fitxer en bytes. La posició `length()` marca el final del fitxer.
 - `int skipBytes (int desplaçament)` → Desplaça el punter des de la posició actual el nombre de bytes indicats en desplaçament.

Exemple complet

Al següent exemple, "EscriuFitxerAleatori", inserirem dades d'empleats en un fitxer. Les dades a inserir s'obtenen de diversos **arrays** que s'omplen en el programa. Les dades s'inseriran de manera seqüencial pel que no serà necessari utilitzar el mètode **seek()**. Per cada empleat també s'inserirà un identificador que coincidirà amb l'índex+1 amb el qual es recorren els **arrays**. La longitud del registre de cada empleat és la mateixa (36 bytes) i els tipus que s'insereixen i la seva grandària és el següent:

- L'identificador que és un sencer → **4 bytes**.
- El cognom que és una cadena de 10 caràcters → Com que Java utilitza UNICODE16, seran **20 bytes**.
- El departament que és un altre sencer → **4 bytes**.
- Un tipus **Double** per a emmagatzemar el salari → **8 bytes**.

```
import java.io.*;
```

```
public class EscriuFitxerAleatori {
```

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

```
        File fitxer = new File ("AleatoriTreballador.dat");
```

```
        RandomAccessFile file = new RandomAccessFile (fitxer , "rw");
```

```
        String cognom[] = {"Fernández", "Gil", "López", "Ramos"};
```

```
        int dep[] = {10,20,10,10};
```

```
        Double salari[] = {1000.45,2400.60,3000.0,1500.56};
```

```
        StringBuffer buffer = null; //Buffer per emmagatzemar cognom
```

```
        int n = cognom.length; //Número d'elements en l'arrai
```

```
        for (int i = 0; i<n; i++) { //Recorrem els arrais
```

```
            file.writeInt (i+1);
```

```
            buffer = new StringBuffer (cognom[i]);
```

```
            buffer.setLength(10); // Fixem en 10 caràcters la longitud del cognom
```

```
            file.writeChars (buffer.toString());
```

```
            file.writeInt(dep[i]);
```

```
            file.writeDouble (salari[i]);
```

```
        }
```

```
        file.close(); // Tanquem el fitxer
```

```
    }
```

```
}
```

Declaració del fitxer
d'accés aleatori



Dades dels
treballadors

i+1 com identificador del registre

Insercions
seqüencials al fitxer

Exemple complet

El següent exemple pren el fitxer anterior i visualitza tots els registres. El posicionament per a començar a recórrer els registres **comença en 0**, per a recuperar els següents registres cal **sumar 36** (grandària del registre, com hem vist abans) a la variable utilitzada pel posicionament.

```
import java.io.*;

public class LlegirFitxerAleatori {

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

        File fitxer = new File ("AleatoriTreballador.dat");

        RandomAccessFile file = new RandomAccessFile (fitxer, "r");

        int id, dep, posicio;

        Double salari;

        char cognom[]= new char[10], aux;

        posicio = 0;

        for ( ; ; ){

            file.seek (posicio); // Ens posicionem a "posició"

            id = file.readInt(); // Obtenim l'identificador del treballador

            for (int i=0; i<cognom.length; i++) {

                aux = file.readChar(); // Llegim caràcter a caràcter el cognom i...

                cognom[i]=aux; //... el desem a l'array cognom

            }

        }

    }

}
```

Ens situem a l'inici del fitxer

```
String cognoms = new String (cognom);
```

```
dep = file.readInt(); //Lectura de departament i salari
```

```
salari = file.readDouble();
```

```
if (id >0)
```

```
    System.out.printf("ID: %s, cognom: %s, Departament: %d, Salari:
```

```
%.2f %n", id, cognoms.trim(), dep, salari);
```

```
    posicio = posicio + 36;
```

Em posicione pel següent treballador.
Cada treballador ocupa 36 bytes.

```
    if (file.getFilePointer() == file.length()) break; // Si he
```

```
recorregut tot el fitxer, acabem
```

```
    } // Tancament del for
```

```
    file.close();
```

```
}
```

```
}
```


Activitats

Implementa els exemples anteriors i comprova el seu correcte funcionament.

A1.- CONSULTA. Crea un programa Java que **consulti les dades d'un empleat del fitxer aleatori**. El programa s'executarà des de la línia de comandes i ha de rebre un identificador d'empleat. Si l'empleat existeix es visualitzaran les seves dades, si no existeix es visualitzarà un missatge indicant-ho.

A2.- INSERCIÓ. Crea un programa Java que **insereixi dades** en el fitxer aleatori. El programa s'executarà des de la línia de comandes i ha de rebre 4 paràmetres: identificador d'empleat, cognom, departament i salari. **Abans d'inserir** es comprovarà si l'identificador existeix, en aquest cas s'ha de visualitzar un missatge indicant-ho i s'aturarà l'execució del programa; si no existeix s'haurà d'inserir.

Activitats

A3.- MODIFICACIÓ. Crea un programa Java que rebi des de la línia de comandes un identificador d'empleat i un import. S'ha de realitzar la **modificació del salari**. La modificació consistirà a sumar al salari de l'empleat l'import introduït. El programa ha de visualitzar el cognom, el salari antic i el nou. Si l'identificador no existeix es visualitzarà missatge indicant-ho i s'aturarà l'execució.

A4.- ESBORRAT. Crea un programa Java que en executar-lo des de la línia de comandes rebi un identificador d'empleat i l'esborri. Es farà un **esborrat lògic** marcant el registre amb la següent informació: l'identificador serà igual a -1, el cognom serà igual a l'identificador que s'esborra, i el departament i salari seran 0.