

CFGS DAMv2A

Mòdul 6 UF 1 Persistència en fitxers

05/10/2020

Conv Ord.

EA.2 Accés aleatori

Aquesta EA correspon al RA1

S'avalua com apte/no apte i contribueix junt amb les altres EAs al 15% de la nota final del RA1

Formador/a: **marc.vives@itb.cat**

Objectiu

L'objectiu d'aquest exercici és introduir a l'alumne a l'accés aleatori a fitxers.

Introducció

Revisa l'apartat de fitxers aleatoris de la presentació "**Persistència en fitxers Iñaki Martin**" publicat a l'apartat "**Teoria**" del Classroom.

Les operacions realitzades fins ara sobre els fitxers es realitzaven de forma seqüencial. Es començava la lectura en el primer caràcter o byte i seguidament es llegien els següents un darrere l'altre fins a arribar al final del fitxer. Igualment quan escrivíem les dades en el fitxer s'anaven escrivint a continuació de l'última informació escrita.

Java disposa de la classe **RandomAccessFile** que disposa de mètodes per accedir al contingut d'un fitxer binari de forma aleatòria (no seqüencial) i per posicionar-nos en una posició concreta. Aquesta classe no és part de la jerarquia InputStream/OutputStream, ja que el seu comportament és totalment diferent, ja que es pot avançar i retrocedir dins d'un fitxer.

Disposem de dos constructors per crear el fitxer d'accés aleatori, aquest poden llançar l'excepció

FileNotFoundException:

- **RandomAccessFile(String nomFitxer, String modeAcces):** escrivint el nom del fitxer incluint el path.
- **RandomAccessFile(File objecteFile, String modeAcces):** amb u objecte File associat a un fitxer.

El paràmetre o argument *modeAcces* pot tenir dos valors:

Mode de accés	Significat
r	Obre el fitxer en mode de només lectura. El fitxer ha d'existir. Una operació d'escriptura en aquest fitxer llançarà l'excepció IOException
rw	Obre el fitxer en mode lectura i escriptura. Si el fitxer no existeix es crea.

Un cop obert el fitxer es poden utilitzar els mètodes `readXXX` i `writeXXX` de les classes **DataInputStream** i **DataOutputStream**. La classe **RandomAccessFile** gestiona un punter que indica la posició actual en el fitxer. Quant el punter al fitxer es col·loca a 0, apuntant al principi del fitxer. Les successives crides als mètodes **read()** i **write()** posicionen el punter segons la quantitat de bytes llegits o escrits.

CFGS DAMv2A

Mòdul 6 UF 1 Persistència en fitxers

05/10/2020

Conv Ord.

EA.2 Accés aleatori

Aquesta EA correspon al RA1

S'avalua com apte/no apte i contribueix junt amb les altres EAs al 15% de la nota final del RA1

Formador/a: **marc.vives@itb.cat**

Els mètodes més importants són:

Mètode	Funció
long getFilePointer()	Retorna la posició actual del punter del fitxer
void seek(long posicio)	Col·loca el punter del fitxer en una posició determinada des de l'inici del fitxer
long length()	Retorna la mida del fitxer en bytes. La posició <i>length()</i> marca el final del fitxer
int skipBytes(int desplaçament)	Desplaça el punter des de la posició actual el número de bytes indicat en <i>desplaçament</i>

Preparació dels exercicis

Prova amb el IntelliJ IDEA les següents classes Java d'exemple que trobaràs a la carpeta "**ClasesEdGarceta**" que proporciona el llibre "Acceso a datos de la Editorial Garceta". I i revisa el codi.

EscribirFichAleatorio.java

EscribirFichAleatorioUnReg.java

LeerFichAleatorio.java

LeerFichAleatorioUnReg.java

ModifFichAleatorio.java

En executar la classe **EscribirFichAleatorio.java** es crea el fitxer "**AleatorioEmple.dat**". El fitxer que es crea és un fitxer binari, i inserta dades d'empleats en un fitxer aleatori.

Les dades a insertar: **apellido, departamento y salario**, s'obtenen de varis arrays que s'omplen. Les dades es van introduint de forma seqüencial i per tant no es necessita utilitzar el mètode **seek()**. Per cada empleat també s'insertarà un identificador (més gran que 0) que coincidirà amb el índex + 1 amb que es recorren els arrays. La longitud del registre de cada empleat és la mateixa (36 bytes) i els tipus que s'inserten i la seva mida són elee següents:

- S'inserta un identificador que és enter i ocupa 4 bytes.
- S'inserta després el cognom, un string de 10 caràcters. Com que Java utilitza caràcters UNICODE, cada caràcter d'una cadena de caràcters ocupa 16 bits (2 bytes), per tant, el cognom ocupa 20 bytes.
- El departament és un enter i ocupa 4 bytes.
- El salari és Double i ocupa 8 bytes.

Mida d'altres tipus: short (2 bytes), byte (1 byte), long (8 bytes), boolean (1 bit), float (4 bytes).

CFGS DAMv2A

Mòdul 6 UF 1 Persistència en fitxers

05/10/2020

Conv Ord.

EA.2 Accés aleatori

Aquesta EA correspon al RA1

S'avalua com apte/no apte i contribueix junt amb les altres EAs al 15% de la nota final del RA1

Formador/a: **marc.vives@itb.cat**

El fitxer s'obre en mode "rw" per lectura i escriptura. El codi és el següent:

```
import java.io.*;

public class EscribirFichAleatorio {

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

        File fichero = new File("AleatorioEmple.dat");

        //declara el fichero de acceso aleatorio

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

        //arrays con los datos

        String apellido[] = {"FERNANDEZ", "GIL", "LOPEZ", "RAMOS", "SEVILLA", "CASILLA", "REY"}; //apellidos

        int dep[] = {10, 20, 10, 10, 30, 30, 20}; //departamentos

        Double salario[] = {1000.45, 2400.60, 3000.0, 1500.56,
                             2200.0, 1435.87, 2000.0}; //salarios

        StringBuffer buffer = null; //buffer para almacenar apellido

        int n = apellido.length; //numero de elementos del array

        for (int i=0; i<n; i++){ //recorro los arrays

            file.writeInt(i+1); //uso i+1 para identificar empleado

            buffer = new StringBuffer( apellido[i] );

            buffer.setLength(10); //10 caracteres para el apellido

            file.writeChars(buffer.toString()); //insertar apellido

            file.writeInt(dep[i]); //insertar departamento

            file.writeDouble(salario[i]); //insertar salario

        }

        file.close(); //cerrar fichero

    }

}
```

Com que el fitxer creat és binari per consultar-lo ho hem de fer amb el codi de la classe
"LeerFichAleatorio.java"

Aquesta classe agafa el fitxer **"AleatorioEmple.dat"** i visualitza tots els registres. el posicionament per començar a recórrer els registres comença en 0, per recuperar els següents registres s'ha de sumar 36 (mida del registre) a la variable utilitzada pel posicionament.

CFGS DAMv2A

Mòdul 6 UF 1 Persistència en fitxers

05/10/2020

Conv Ord.

EA.2 Accés aleatori

Aquesta EA correspon al RA1

S'avalua com apte/no apte i contribueix junt amb les altres EAs al 15% de la nota final del RA1

Formador/a: **marc.vives@itb.cat**

El codi de la classe:

```
import java.io.*;

public class LeerFichAleatorio {

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

        File fichero = new File("AleatorioEmple.dat");

        //declara el fichero de acceso aleatorio

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

        //

        int id, dep, posicion;

        Double salario;

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

        posicion = 0; //para situarnos al principio

        for(;;){ //recorro el fichero

            file.seek(posicion); //nos posicionamos en posicion

            id = file.readInt(); // obtengo id de empleado

            //recorro uno a uno los caracteres del apellido

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

                aux = file.readChar();

                apellido[i] = aux; //los voy guardando en el array

            }

            //convierto a String el array

            String apellidos = new String(apellido);

            dep = file.readInt(); //obtengo dep

            salario = file.readDouble(); //obtengo salario

            if(id > 0)

                System.out.printf("ID: %s, Apellido: %s, Departamento: %d, Salario: %.2f %n",

                    id, apellidos.trim(), dep, salario);

            //me posiciono para el sig empleado, cada empleado ocupa 36 bytes

            posicion= posicion + 36;

            //Si he recorrido todos los bytes salgo del for

            if (file.getFilePointer() == file.length())break;

        }

        file.close(); //cerrar fichero

    }

}
```

CFGS DAMv2A

Mòdul 6 UF 1 Persistència en fitxers

05/10/2020

Conv Ord.

EA.2 Accés aleatori

Aquesta EA correspon al RA1

S'avalua com apte/no apte i contribueix junt amb les altres EAs al 15% de la nota final del RA1

Formador/a: **marc.vives@itb.cat**

Si executes el codi ens mostra el contingut del fitxer:

ID: 1, Apellido: FERNANDEZ, Departamento: 10, Salario: 1000,45

ID: 2, Apellido: GIL, Departamento: 20, Salario: 2400,60

ID: 3, Apellido: LOPEZ, Departamento: 10, Salario: 3000,00

ID: 4, Apellido: RAMOS, Departamento: 10, Salario: 1500,56

ID: 5, Apellido: SEVILLA, Departamento: 40, Salario: 4000,87

ID: 6, Apellido: CASILLA, Departamento: 30, Salario: 1435,87

ID: 7, Apellido: REY, Departamento: 20, Salario: 2000,00

Per consultar un empleat determinat no és necessari recórrer tots els registres del fitxer, coneixent el seu identificador podem accedir a la posició que ocupa dins del fitxer i obtenir les seves dades.

Revisa el codi de la classe "**LeerFichAleatorioUnReg.java**" per veure com es consulta un empleat.

Per afegir registres a partir de l'últim insertat hem de posicionar el punter al final del fitxer. Revisa el codi de la classe "**EscribirFichAleatorioUnReg.java**" per veure com s'inserta un nou empleat al fitxer.

Per modificar un registre determinat, accedim a la seva posició i efectuem les modificacions. el fitxer s'ha d'obrir en mode "rw". Revisa el codi de la classe "**ModifFichAleatorio.java**" per veure com es modifiquen les dades d'un empleat.

CFGS DAMv2A

Mòdul 6 UF 1 Persistència en fitxers

05/10/2020

Conv Ord.

EA.2 Accés aleatori

Aquesta EA correspon al RA1

S'avalua com apte/no apte i contribueix junt amb les altres EAs al 15% de la nota final del RA1

Formador/a: **marc.vives@itb.cat**

Exercicis

Fes un projecte amb el IntelliJ IDEA que es digui "**cat.itb.EA2**" amb una classe "Main" que cridi els mètodes per realitzar les següents tasques:

(Recorda que has de controlar les excepcions i no deixar fitxers oberts. Anomena els mètodes que crida el main amb el número de l'exercici. Per exemple "**ex1**")

1.- Consultar les dades d'un empleat del fitxer "**AleatorioEmple.dat**". El mètode ha de rebre un identificador d'empleat que li passem per paràmetre o bé l'introduïm per teclat. Si l'empleat existeix, es visualitzaran les seves dades, si no existeix es visualitzarà un missatge que ho indiqui.

2.- Insertar dades en el fitxer "**AleatorioEmple.dat**". El mètode ha de rebre 4 paràmetres: identificador d'empleat, cognom, departament i salari, que li passem per paràmetre o bé introduïm per teclat. Abans d'insertar es comprovarà si l'identificador existeix, si existeix s'ha de visualitzar un missatge que ho indiqui, si no existeix s'ha d'insertar.

3.- Modificar dades en el fitxer "**AleatorioEmple.dat**". El mètode rep pel teclat o li passem com a paràmetres un identificador d'empleat i un import. S'ha de realitzar la modificació del salari. La modificació consistirà en sumar al salari de l'empleat l'import introduït. El mètode ha de visualitzar el cognom, el salari antic i el nou. Si l'identificador no existeix, es visualitzarà un missatge que ho indiqui.

4.- Esborrar dades en el fitxer "**AleatorioEmple.dat**". Crea un mètode que rebi pel teclat o li passem com a paràmetre un identificador d'empleat i l'esborri. Serà un esborrat lògic marcant el registre amb la següent informació: l'identificador serà "-1", el cognom serà l'identificador que s'elimina, i el departament i el salari seran 0.

5.- Crea un mètode que mostri els identificadors dels empleats eliminats.

Lliurament de l'activitat

Envia el teu projecte en un fitxer comprimit al classroom del **M06UF1** a la data indicada. El nom del fitxer comprimit ha de ser **cat.itb.EA2_CognomNom**. Per exemple **cat.itb.EA2_ColomerJoan**.