

CFGS DAMi2 A/B

Mòdul 6 UF 1 Persistència en fitxers

02/10/2020

Conv. Ord.

EA3. Fitxers binaris

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 en els fitxers binaris.

Introducció

Els fitxers binaris emmagatzemen seqüències de dígit binari que no es poden llegir directament per l'usuari. Tenen l'avantatge que ocupen menys espai en disc. En Java, les dues classes que ens permeten treballar amb fitxers binaris són **FileInputStream** (per l'entrada) i **FileOutputStream** (per la sortida), aquestes classes treballen amb fluxes de bytes i creen un enllaç entre el flux de bytes i el fitxer.

Els mètodes que proporciona la classe **FileInputStream** per la lectura són similars als que hem vist per la classe **FileReader**, aquests mètodes retornen el número de bytes llegits o -1 si s'ha arribat al final del fitxer:

Mètode	Funció
<code>int read()</code>	Llegeix un byte i el retorna
<code>int read(byte[] b)</code>	Llegeix fins a <i>b.length</i> bytes de dades d'un array de bytes
<code>int read(byte[] b, int desplaçament, int n)</code>	Llegeix fins a <i>n</i> bytes de l'array <i>b</i> començant per <i>b[desplaçament]</i> i retorna el número llegit de bytes

Els mètodes que proporciona la classe **FileOutputStream** per l'escriptura són:

Mètode	Funció
<code>void write(int b)</code>	Escriu un byte
<code>void write(byte[] b)</code>	Escriu <i>b.length</i> bytes
<code>void write(byte[] b, int desplaçament, int n)</code>	Escriu <i>n</i> bytes a partir de l'array <i>b</i> bytes d'entrada començant per <i>b[desplaçament]</i>

El següent exemple escriu bytes en un fitxer i després el visualitza:

```
import java.io.*;

public class EscribirFichBytes {
    public static void main(String[] args) throws IOException {
        File fichero = new File("FichBytes.dat"); //declara fichero
        //crea flujo de salida hacia el fichero
        FileOutputStream fileout = new FileOutputStream(fichero, true);
        //crea flujo de entrada
        FileInputStream filein = new FileInputStream(fichero);
        int i;

        for (i = 1; i < 100; i++)
            fileout.write(i); //escribe datos en el flujo de salida
        fileout.close(); //cerrar stream de salida

        //visualizar los datos del fichero
        while ((i = filein.read()) != -1) //lee datos del flujo de entrada
            System.out.println(i);
    }
}
```

CFGS DAMi2 A/B

Mòdul 6 UF 1 Persistència en fitxers

02/10/2020

Conv. Ord.

EA3. Fitxers binaris

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**

```
filein.close(); //cerrar stream de entrada
}
}
```

Per afegir bytes al final del fitxer utilitzarem **FileOutputStream** de la següent manera, col·locant en el segon paràmetre del constructor el valor **true**:

```
FileOutputStream fileout = new FileOutputStream(fitxer, true);
```

Per llegir i escriure dades de tipus primitiu: *int*, *float*, *long*, etc. utilitzarem les classes **DataInputStream** i **DataOutputStream**. Aquestes classes defineixen diversos mètodes *readXXX* i *writeXXX* que són variacions dels mètodes *read()* i *write()* de la classe base per llegir i escriure dades del tipus primitiu. Alguns dels mètodes es mostren en la taula següent:

MÈTODES PER LECTURA	MÈTODES PER ESCRIPTURA
<pre>boolean readBoolean(); byte readByte(); int readUnsignedByte(); int readUnsignedShort(); short readShort(); char readChar(); int readInt(); long readLong(); float readFloat(); double readDouble(); string readUTF();</pre>	<pre>void writeBoolean(boolean v); void writeByte(int v); void writeBytes(String s); void writeShort(int v); void writeChars(String s); void writeChar(int v); void writeInt(int v); void writeLong(long v); void writeFloat(float v); void writeDouble(double v); void writeUTF(String str);</pre>

Per obrir un objecte **DataInputStream**, s'utilitzen els mateixos mètodes que per **FileInputStream**, exemple:

```
File fitxer = new File("FichData.dat");
FileInputStream filein = new FileInputStream(fitxer);
DataInputStream dataIS = new DataInputStream(filein);
```

O bé

```
File fitxer = new File("FichData.dat");
DataInputStream dataIS = new DataInputStream(new FileInputStream(fitxer));
```

Per obrir un objecte **DataOutputStream**, s'utilitzen els mateixos mètodes que per **FileOutputStream**, exemple:

```
File fitxer = new File("FichData.dat");
FileOutputStream fileout = new FileOutputStream(fitxer);
DataOutputStream dataOS = new DataOutputStream(fileout);
```

O bé

```
File fitxer = new File("FichData.dat");
DataOutputStream dataIS = new DataOutputStream(new FileOutputStream(fitxer));
```

CFGS DAMi2 A/B

Mòdul 6 UF 1 Persistència en fitxers

02/10/2020

Conv. Ord.

EA3. Fitxers binaris

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 següent exemple inserta dades en el fitxer *FichData.dat*, les dades els agafa de dos arrays, un conté els noms d'una sèrie de persones i l'altre les seves edats, recorrem els arrays i anem escrivint en el fitxer el nom (mitjançant el mètode **writeUTF(String)**) i l'edat (mitjançant el mètode **writeInt(int)**):

```
import java.io.*;
public class EscribirFichData {
    public static void main(String[] args) throws IOException {

        File fichero = new File("FichData.dat");
        DataOutputStream dataOS = new
            DataOutputStream(new FileOutputStream(fichero));
        String nombres[] = {"Ana", "Luis Miguel", "Alicia", "Pedro", "Manuel",
"Andrés", "Julio", "Antonio", "María Jesús"};

        int edades[] = {14, 15, 13, 15, 16, 12, 16, 14, 13};

        for (int i = 0; i < edades.length; i++) {
            dataOS.writeUTF(nombres[i]); //inserta nombre
            dataOS.writeInt(edades[i]); //inserta edad
        }
        dataOS.close(); //cerrar stream
    }
}
```

El següent exemple visualitza les dades gravades anteriorment en el fitxer, s'han de recuperar en el mateix ordre en el que es van escriure, és a dir, primer obtenim el nom i després l'edat.

```
import java.io.*;
public class LeerFichData {
    public static void main(String[] args) throws IOException {
        File fichero = new File("FichData.dat");
        DataInputStream dataIS = new DataInputStream(new FileInputStream(fichero));
        String n;
        int e;

        try {
            while (true) {
                n = dataIS.readUTF(); //recupera el nombre
                e = dataIS.readInt(); //recupera la edad
                System.out.println("Nombre: " + n +
                    ", edad: " + e);
            }
        } catch (EOFException eo) {
            eo.printStackTrace();
        }

        dataIS.close(); //cerrar stream
    }
}
```

S'obté la següent sortida a l'executar el programa:

```
Nombre: Ana, edad: 14
Nombre: Luis Miguel, edad: 15
Nombre: Alicia, edad: 13
Nombre: Pedro, edad: 15
```

CFGS DAMi2 A/B
Mòdul 6 UF 1 Persistència en fitxers
02/10/2020

Conv. Ord.

EA3. Fitxers binaris

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

Nombre: Manuel, edad: 16
Nombre: Andrés, edad: 12
Nombre: Julio, edad: 16
Nombre: Antonio, edad: 14
Nombre: María Jesús, edad: 13

Exercicis

Fes un projecte amb el IntelliJ IDEA que es digui "**cat.itb.EA3**" 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.- Crear un fitxer binari per guardar dades de departaments, el nom del fitxer és "**Departaments.dat**".

Introdueix com a mínim 10 departaments. Les dades per cada departament són: Número de departament: enter, Nom: String i Localitat:String

2.- Modificar les dades d'un departament. El mètode rep per teclat el número de departament a modificar, el nou nom de departament i la nova localitat. Si el departament no existeix, visualitza un missatge que ho indiqui. Visualitza també les dades antigues del departament i les noves dades.

3.- Eliminar físicament un departament. El mètode rep pel teclat el número de departament a eliminar. Si el departament no existeix, visualitza un missatge que ho indiqui. Visualitza també el número total de departaments que existeixen en el fitxer.

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.EA3_CognomNom**. Per exemple **cat.itb.EA3_VivesMarc**.