
Gestió de fitxers

— amb Java —

Introducció

- Un **fitxer** o arxiu és un **conjunt de bits emmagatzemat en un dispositiu**, com per exemple, un disc dur.
- Els fitxers tenen un **nom i una extensió**.
 - El nom ha de ser únic en el directori on es trobi el fitxer.
 - L'extensió no és més que una convenció que ens permet saber el tipus de fitxer.
- Un fitxer pot constar de **línies o registres o qualsevol cosa**; cada registre pot estar format per camps relacionats; cada línia pot tenir el text d'un imprès.
- La manera en què s'agrupen les dades en el fitxer depèn de la persona que el dissenye.

Classes associades a la gestió de fitxers

El paquet [java.io](#) conté les classes per a controlar entrada/sortida a Java.

- La primera classe a estudiar és [FILE](#).
- [FILE](#) encapsula tota la funcionalitat necessària per gestionar un sistema d'arxius:
 - Manipulació i consulta de l'estructura del sistema d'arxius.
 - Manipulació propietats dels elements del sistema d'arxius.
 - Gestió de permisos (depenent del SO).
- Cada instància de [FILE](#) representa una ruta. Pot ser ruta a un fitxer o a un directori.
- La ruta de la instància es manté sempre; no hi ha manera de modificar-la.
- Constructors:
 - [File](#) ([String](#) rutaAlFitxer)
 - [File](#) ([String](#) directori, [String](#) nomFitxer)
 - [File](#) ([File](#) directori, [String](#) nomFitxer)

Mètodes més importants de File

- **String[] list()** → Retorna un **array** de **String** amb els noms de fitxers i directoris associats a l'objecte **File**.
- **File[] listFiles** → Retorna un **array** d'objectes **File** contenint els fitxers que estiguin dins del directori representat per l'objecte **File**.
- **String getName()** → Nom del fitxer.
- **String getPath()** → Ruta relativa.
- **String getAbsolutePath()** → Ruta absoluta.
- **Boolean exists()** → Retorna **true** si el fitxer existeix.
- **Boolean canWrite()** → Retorna **true** si el fitxer es pot escriure.
- **Boolean canRead()** → Retorna **true** si el fitxer es pot llegir.
- **Boolean isFile()** → Retorna **true** si l'objecte **File** correspon a un fitxer normal.
- **Boolean isDirectory()** → Retorna **true** si l'objecte **File** correspon a un directori.

Mètodes més importants de File

- **long length()** → Retorna la grandària del fitxer en bytes.
- **Boolean mkdir()** → Crea un directori amb el nom indicat en la creació de l'objecte **File**. Només es crearà si no existeix.
- **Boolean renameTo (File nouNom)** → Canviar de nom fitxer.
- **Boolean delete()**.
- **Boolean createNewFile()** → Crea un nou fitxer, buit, associat a **File** si i només si no existeix un fitxer amb aquest nom.
- **String getParent()** → Retorna el nom del directori pare o **null** si no existeix.

Exemple

Mostrar la llista de fitxers al directori actual:

```
import java.io.*;

public class checkDir {
    public static void main (String[] args) {
        String dir = "."; // directori actual
        File f = new File(dir);
        String[] arxius = f.list();
        System.out.printf("Fitxers al directori actual: %d %n", arxius.length);
        for (int i = 0; i < arxius.length; i++){
            File f2 = new File(f, arxius[i]);
            System.out.printf("Nom: %s, és fitxer?: %b, és directori?:%b %n", arxius[i],
                f2.isFile(), f2.isDirectory());
        }
    }
}
```

Activitat

- **(A1) Canvia la ruta** de l'exemple anterior. Utilitza una ruta absoluta a la teva carpeta de descàrregues, per exemple.
- Ara fes els **canvis** necessaris perquè el programa anterior **mostri els fitxers del directori introduït** des de línia de comandes **com a argument** en executar el programa.
- **(A2)** Realitza un **programa** Java que utilitzi el **mètode** **listFiles()** per mostrar la llista de fitxers d'un directori que es passarà al programa des dels **arguments** del **main**.
- **Afegeix** al programa anterior les instruccions necessàries perquè envii un missatge d'error en cas que el **directori** passat com a argument **no existeixi**.
- **(A3)** Realitza un **programa Java** que **mostri** la següent informació d'un fitxer qualsevol: **Nom, ruta relativa, ruta absoluta, permisos i grandària**.

Exemple

Vegem ara un exemple de creació/eliminació de fitxers:

```
import java.io.*;

public class CrearDir{

    public static void main(String[] args){

        File d=new File("NouDir");

        File f1= new File(d,"Fitxer1.txt");

        File f2= new File (d,"Fitxer2.txt");

        d.mkdir();

        try {

            if (f1.createNewFile())

                System.out.println("Fitxer1 creat correctament");

            else

                System.out.println("No s'ha pogut crear Fitxer1");

            if (f2.createNewFile())

                System.out.println("Fitxer2 creat correctament");
```


Exemple

```
        else
            System.out.println("No s'ha pogut crear Fitxer2");
        } catch (IOException ioe) {
            ioe.printStackTrace();
        }

        f1.renameTo(new File(d, "Fitxer1Nou"));

        try {
            File f3 = new File ("NouDir/Fitxer3.txt");
            f3.createNewFile();
        } catch (IOException ioe) {
            ioe.printStackTrace();
        }
    }
}
```

El mètode `createNewFile()` pot llançar l'excepció `IOException`, per això hem utilitzat el bloc `try-catch`.

Activitat

- Còpia completament l'exemple anterior i executa'l.
- **(A4)** Modifica el programa per a que elimini el directori creat en el punt anterior. Per a això hauràs d'eliminar tots els arxius que es trobin dins del directori.