INS MANUEL VÁZQUEZ MONTALBÁN

M03. Programació bàsica

UF3. Fonament de gestió de fitxers

1. Introducció

Fitxers

Què és un fitxer?

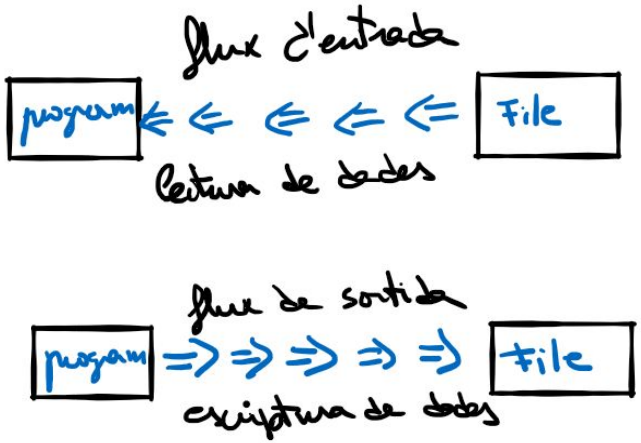
És un recurs informàtic que s'utilitza per a registrar dades discretament en un dispositiu d'emmagatzematge informàtic. Pot ser editat i transferit a través d'Internet en aquest sistema informàtic en particular.
Disposem de diferents tipus d'arxius, identificats mitjançant l'extensió.

Les operacions més bàsiques que els programes poden realitzar en un arxiu són:

- Crear un nou arxiu
- Canviar els permisos d'accés i els atributs d'un arxiu
- Obrir un arxiu, la qual cosa fa que el contingut de l'arxiu estigui disponible per al programa
- Llegir les dades d'un arxiu
- Escriure dades en un arxiu
- Modificar/afegir dades en un arxiu
- Eliminar un arxiu
- Tancar un arxiu, acabant l'associació entre ell i el programa

Fitxers

Comunicació bàsica



La comunicació bàsica entre el programa i l'origen/destí de la informació es realitza mitjançant un flux d'informació (stream): una estructura de dades que fa d'intermediari entre el programa i l'origen o destí de la informació.

Fitxers

Comunicació bàsica

Quan un programa comença la seva execució, s'obren automàticament tres fluxos, vinculats amb altres tres dispositius.

Flux	Dispositiu al que està vinculat
stdin	Dispositiu d'entrada estàndard (teclat)
stdout	Dispositiu de sortida estàndard (pantalla)
stderr	Dispositiu d'error estàndard (pantalla)

Fitxers

Els algorismes per a llegir i escriure dades segueixen la següent estructura:

Llegir	**Escriure**

Open (stream)
While (information)
    Read (information)
Close (stream)
	Open (stream)
	While (information)
	    Write (information)
	Close (stream)

Fitxers

Els modes principals d'obertura de fitxers són els següents:

Mode	Descripció del mode
r	Obre el fitxer per a lectura. Si no existeix el fitxer, retorna NULL
w	Obre el fitxer per a escriptura. Si no existeix, el crea. Si existeix, el sobreescriu
a	Obre un fitxer per afegir-hi contingut. Si el fitxer existeix, el punter de fitxer es troba al final del fitxer. És a dir, el fitxer es troba en el mode *append*. Si el fitxer no existeix, crea un fitxer nou per escriure.

Fitxers. Lectura de fitxers

```java
package cat.institutmvm;

import java.io.File;
import java.io.FileNotFoundException;
import java.util.Scanner;

public class Main {
    public static void main(String[] args) {
        try {
            File file = new File( pathname: "files/document.txt");
            Scanner sc = new Scanner( source: file);
            while (sc.hasNextLine()) {
                String data = sc.nextLine();
                System.out.println( x: data);
            }
            sc.close();
        } catch (FileNotFoundException e) {
            System.out.println( x: "An error occurred: ");
            e.printStackTrace();
        }
    }
}
```

Fitxers. exception Handling

Una **excepció** és un esdeveniment que succeeix durant l'execució d'un programa, que interromp el flux normal de les instruccions del programa.

En aquest cas, cal tractar aquesta excepció (exception handling) de manera que pugui continuar l'execució del programa o s'informi (de manera amigable) a l'usuari final.

Un dels mètodes més habituals per a treballar les excepcions és mitjançant el bloc try-catch, on dins del bloc *try* s'inclouen les instruccions que poden generar l'excepció. Al bloc *catch*, el que farem serà mostrar el detall de l'excepció o incloure unes instruccions alternatives.

Fixers. exception Handling

```
    public void writeList() throws IOException {
        PrintWriter out = null;
        try {
            System.out.println("Entering" + " try statement");

            out = new PrintWriter(new FileWriter( fileName:"OutFile.txt"));
            for (int i = 0; i < 100; i++) {
                out.println("Value at: " + i);
            }
        } catch (IndexOutOfBoundsException e) {
            System.err.println("Caught IndexOutOfBoundsException: "
                    + e.getMessage());

        } catch (IOException e) {
            System.err.println("Caught IOException: " + e.getMessage());

        } finally {
            if (out != null) {
                System.out.println( x:"Closing PrintWriter");
                out.close();
            } else {
                System.out.println( x:"PrintWriter not open");
            }
        }
    }
}
```