



PROJECTE DE PRÀCTIQUES

PROGRAMACIÓ II — LLIURAMENT 1

JOSEP NÚÑEZ RIBA

RODRIGO CABEZAS QUIRÓS

11 DE MARÇ DE 2018

Índex

1. Introducció	3
2. Anàlisi	4
3. Desenvolupament	5
4. Qüestions.....	9
5. Resultats.....	11

1. Introducció

L'objectiu del projecte és desenvolupar un reproductor multimèdia, tot passant per totes les fases de desenvolupament d'un projecte de software i familiaritzar-se amb l'ús d'eines informàtiques de suport a la programació. Està enfocat cap a la programació orientada a objectes, programació orientada a esdeveniments i programació guiada per una especificació.

Els objectius a assolir en aquest lliurament són els següents:

- Organitzar arxius multimèdia dins d'una carpeta, guardant la seva informació essencial (path on es troba el fitxer, nom del fitxer, extensió del fitxer, data d'última modificació i una breu descripció del contingut).
- Realitzar un menú per al reproductor que permeti realitzar les següents operacions:
 1. Afegir fitxer multimèdia: Donades les dades del fitxer, el guarda a la carpeta corresponent.
 2. Eliminar fitxer multimèdia: Suprimir un fitxer donat.
 3. Mostrar carpeta: Mostrar per pantalla el contingut de una carpeta donada amb els fitxers ordenats.
 4. Sortir: Sortir del reproductor.

2. Anàlisi

Per fer el menú utilitzarem un switch i la classe Scanner de Java. La classe Scanner recollirà la entrada del usuari, aquesta en funció de quina sigui i gràcies al switch realitzarà la operació pertinent.

Per implementar la classe FitxerMultimedia heretarem de la classe File, així podrem utilitzar els atributs i mètodes bàsics que també ha de tenir un fitxer multimèdia. Alguns d'aquests mètodes seran setters i getters per garantir que l'usuari no té accés directe als atributs del fitxer. El mètode equals de la classe File caldrà sobreescriure'l, de tal forma que comprovi que tots els atributs dels fitxers són iguals.

Pel que respecta a la classe CarpetaFitxers, per guardar els arxius utilitzarem la classe ArrayList, posant per defecte la capacitat de la carpeta a 100 fitxers. També farem que el constructor per paràmetres permeti crear carpetes de mida variable.

3. Desenvolupament

- **Classe CarpetaFitxers:**

- public CarpetaFitxers(): Constructor per defecte. Inicialitza per defecte, la mida màxima a 100 arxius i crea un Array amb 100 slots.
- public CarpetaFitxers(int mida): Constructor de carpeta amb mida variable. Inicialitza mida màxima en funció de la mida que se li passi i crea l'Array corresponent.
- public int getSize(): Retorna el nombre de fitxers a la carpeta.
- public void addFitxer(File fitxer): Si hi ha espai a la carpeta i l'arxiu no hi és en aquesta, afegeix el fitxer. En els altres cassos llença una excepció.
- public removeFitxer(File fitxer): Suprimeix un fitxer donat de la carpeta. Amb un bucle i un comptador, comprova que l'element iterat sigui equivalent al que es vol eliminar. Si l'arxiu és a la carpeta el borra i incrementa l'espai disponible. Cas contrari llença una excepció.

```
1. public boolean removeFitxer(FitxerMultimedia fitxer){
2.     boolean removed = false;
3.     int i =0;
4.     while((i<this.getSize())&&(!(removed))){
5.         if (fitxer.equals(this.getAt(i))){
6.             this.carpeta.remove(i);
7.             removed = true;
8.         }
9.         i++;
10.    }
11.    if(!removed){
12.        System.out.print("\n\033[31mError!
L'arxiu no existeix o no es troba en aquesta
carpeta.\033[0m\n");
13.        return false;
14.    }
15.    System.out.print("\n\033[32mFitxer eliminat
amb èxit.\033[0m\n");
16.    return true;
17. }
```

- public File getAt(int position): Comprova que la posició de la carpeta no és buida. En cas de no ser-ho retorna l'element corresponent en aquesta posició a la carpeta.
- public void clear(): Buida la carpeta de fitxers.
- public boolean isFull(): Comprova si la mida màxima i la utilitzada són iguals, en cas de ser-ho la carpeta és plena i retorna true. Cas contrari retorna false.
- public boolean isEmpty(): Si la mida de la carpeta és zero, retorna true.
- public String toString(): Retorna un resum amb tota la informació dels fitxers de la carpeta(nom, descripció,...). Utilitza el mètode toString de FitxerMultimedia.

```

1.    public String toString(){
2.        String resum = "Carpeta
fitxers:\n===== \n\n";
3.        for (int i=0;i<carpeta.size();i++){
4.            resum=resum+"["+(i+1)+"]
"+this.getAt(i).toString()+"\n";
5.        }
6.        return resum;
7.    }

```

- **Classe FitxerMultimedia:**

- public FitxerMultimedia(String camí): Constructor per defecte de FitxerMultimedia. Comprova que l'arxiu indicat existeix i que es troba al path indicat (si no és compleix avisa del error). Per defecte inicialitza la descripció com a un String buit. El nom del fitxer i la seva extensió la agafa fent un substring del nom del fitxer.

```
1.    public FitxerMultimedia(String camí) {
2.        super(camí);
3.        if((this.exists()) && (!(this.isDirectory()))){
4.            this.setUltimaModificacio(new Date(this.lastMo
dified()));
5.            this.setCamiAbsolut(this.getAbsolutePath());
6.            String name=this.getName();
7.            int punt=name.lastIndexOf(".");
8.            this.setNomFitxer(name.substring(0,punt));

9.            this.setExtensio(name.substring(punt));
10.        }else{
11.            //System.out.println("\nEl fitxer no
existeix.\n");
12.            this.setUltimaModificacio(new Date());
13.            this.setCamiAbsolut(camí);
14.            this.setNomFitxer("");
15.            this.setExtensio("");
16.        }
17.        this.setDescripcio("");
18.    }
```

- public FitxerMultimedia(String camí, String desc): Inicialitza tots els atributs del fitxer amb els seus valors corresponents i afegeix la descripció del fitxer.
- public void setUltimaModificacio(Date lastModified): Estableix la data d'última modificació del fitxer.
- public void setCamiAbsolut(String camí): Estableix la ruta on es troba el fitxer.
- public void setNomFitxer(String nom): Dóna nom a un fitxer donat.
- public void setExtensio(String extensió): Assigna una extensió a un fitxer.
- public void setDescripció(String desc): Dóna descripció al fitxer.
- public Date getUltimaModificacio(): Retorna la data de la última modificació.

- public String getCamiAbsolut(): Retorna la path on es troba el fitxer.
- public String getNomFitxer(): Retorna el nom del fitxer.
- public String getExtensio(): Retorna l'extensió de l'arxiu.
- public String getDescripcio(): Retorna la descripció del fitxer.
- public Boolean equals(FitxerMultimedia fitxerMultimedia): Compara els atributs d'un fitxer amb els d'un altre, si són equivalents retorna true, cas contrari retorna false.

```

1.    public boolean equals(/*Object*/FitxerMultimedia
    fitxerMultimedia){
2.        boolean return=true;
3.        if (!(fitxerMultimedia.getUltimaModificacio().equals(this
    s.getUltimaModificacio()))){
4.            return false;
5.        }else if (!(fitxerMultimedia.getCamiAbsolut().equals(this
    s.getCamiAbsolut()))){
6.            return false;
7.        }else if (!(fitxerMultimedia.getNomFitxer().equals(this.
    getNomFitxer()))){
8.            return false;
9.        }else if (!(fitxerMultimedia.getExtensio().equals(this.g
    etExtensio()))){
10.            return false;
11.        }else if (!(fitxerMultimedia.getDescripcio().equals
    (this.getDescripcio()))){
12.            return false;
13.        }else{
14.            return true;
15.        }
16.    }

```

- public String toString(): Retorna un resum amb tots els atributs del fitxer (nom, extensió, path, ...).
- **Classe AplicacioUB1:**
 - public void gestioAplicacioUB(Scanner sc): S'encarrega de gestionar les diferents opcions i accions a realitzar. Amb la classe Scanner, recull la opció seleccionada per l'usuari.

4. Qüestions

- Explicar breument les classes implementades:

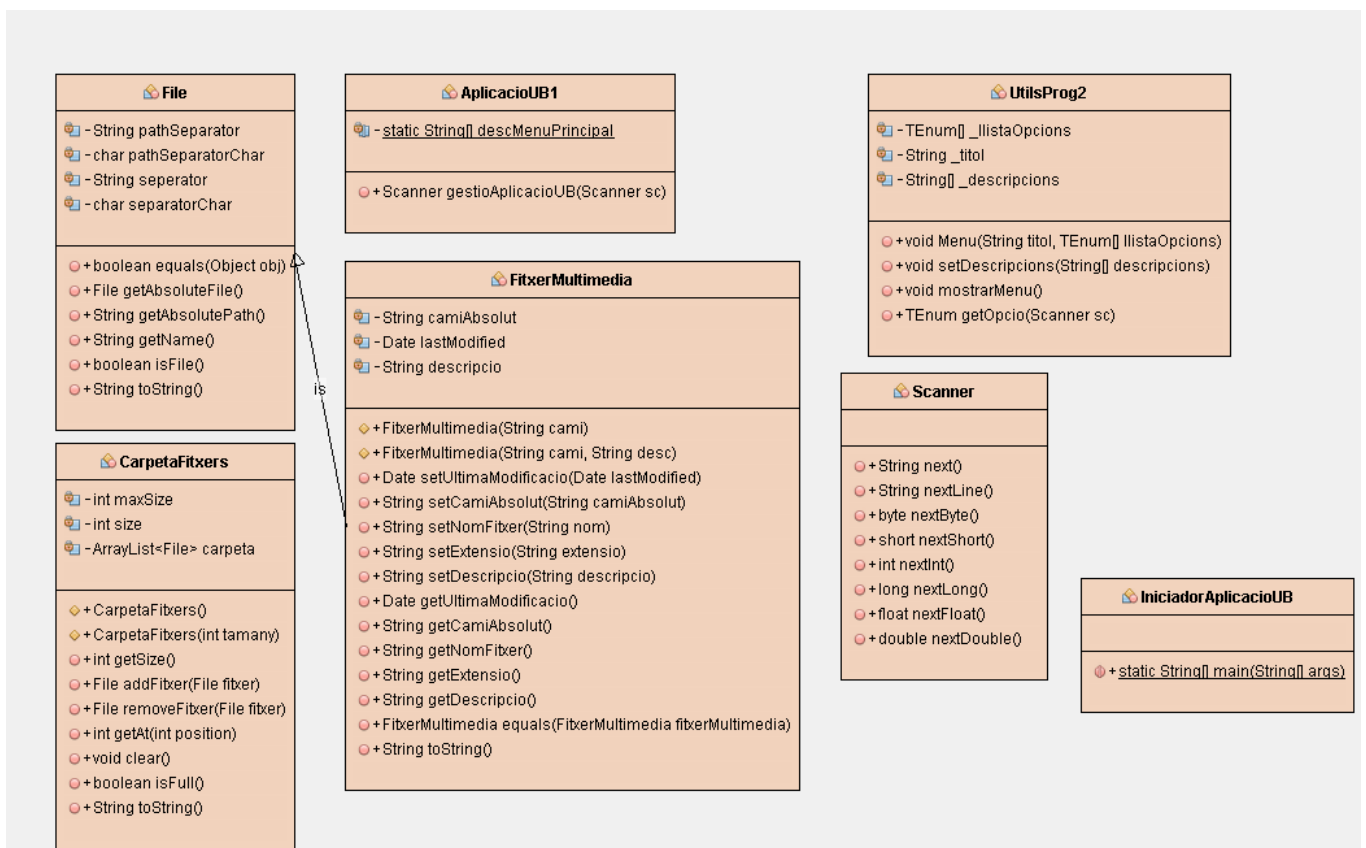
Classe AplicacioUB1: S'encarrega de la gestió de les accions disponibles que l'usuari pot escollir.

Classe IniciadorUB: Classe principal que s'encarrega d'executar l'aplicació.

Classe FitxerMultimedia: Hereta de la classe File. Gestiona els atributs dels fitxers.

Classe CarpetaFitxers: S'encarrega del emmagatzematge d'arxius.

- Dibuixar el diagrama de relacions entre les classes que heu utilitzat a la vostra pràctica. Incloure tant les classes implementades com les que pertanyen a la llibreria de Java i UtilsProg2.



- Explicar com heu implementat i on heu utilitzat el mètode **isFull**.

Hem creat una variable `maxSize` que correspon a la mida màxima de la carpeta. Si la mida actual de la carpeta és igual a la mida màxima de la carpeta, la carpeta és plena i el mètode retorna `true`.

- Segons la implementació de la classe **CarpetaFitxers**, si tenim dos fitxers multimèdia corresponents al mateix fitxer, quan cridem al mètode per eliminar un d'aquests fitxers eliminarà l'altre també o no?

Tal i com nosaltres l'hem implementada no. L'hem implementat amb un bucle que s'executarà fins que l'element s'hagi suprimit. D'aquesta forma, en cas de fitxers repetits només eliminarà el primer.

5. Resultats

- **Proves AplicacióUB1:**

- Objectiu: Comprovar el correcte funcionament del mètode gestioAplicacioUB.

Descripció	Resultat esperat	Resultat obtingut / Correcció
Des del menú principal es vol afegir un nou fitxer	Apareix opció d'afegir fitxer.	Correcte: Demana path del arxiu i si es vol afegir descripció.
Des del menú principal es vol eliminar un fitxer.	S'elimina el fitxer desitjat.	Correcte: Elimina el fitxer de la carpeta.
S'introdueix una opció incorrecta al menú principal.	Error, la opció no és correcta.	Correcte: Torna a demanar una opció vàlida.

- **Proves CarpetaFitxers:**

- Objectiu: Verificar que les accions que demanades és realitzen correctament i que els fitxers s'eliminen o guarden correctament.

Descripció	Resultat esperat	Resultat obtingut / Correcció
Afegir un fitxer a una carpeta buida.	El fitxer es guarda correctament.	Correcte: Fitxer afegit amb èxit.
Afegir un fitxer a una carpeta plena.	Error, la carpeta és plena.	Correcte: La carpeta no admet més fitxers.
Afegir un fitxer inexistent a una carpeta.	Error, el fitxer no existeix.	Correcte: Afegeix el fitxer sense nom ni descripció (com indica el guió de la pràctica).
Eliminar un fitxer d'una carpeta.	El fitxer s'elimina.	Correcte: Elimina fitxer i ho indica per pantalla.
Eliminar un fitxer que no és troba en la carpeta.	Error, el fitxer no hi és a la carpeta.	Correcte: Error, no pot eliminar el fitxer.

Eliminar un fitxer duplicat en una carpeta.	S'elimina el fitxer i totes còpies com hi hagi.	Correcte: Només elimina un dels fitxers duplicats.
Recuperar un arxiu d'una posició de la carpeta.	Retorna el fitxer en aquesta posició.	Correcte: Mostra per pantalla el fitxer corresponent a la posició de la carpeta.
Buidar una carpeta sencera.	La carpeta és buida.	Correcte: No hi han fitxers a la carpeta.
Visualitzar un resum dels elements de la carpeta.	Mètode toString imprimeix per pantalla el resum.	Correcte: Imprimeix per pantalla un resum amb els elements de la carpeta.

○ **Classe FitxerMultimedia:**

- Objectiu: Comprovar que els setters i getters funcionen correctament i que els atributs dels fitxers es guarden correctament.

Descripció	Resultat esperat	Resultat obtingut / Correcció
Crear nou fitxer existent en un directori determinat.	El fitxer es crea amb una data de última modificació, agafant la resta d'atributs.	Correcte: Crea fitxer amb la data d'última modificació corresponent, i tant el nom com la extensió d'aquest de forma satisfactòria.
Crear nou fitxer inexistent en un directori.	El fitxer es crea però sense nom ni descripció.	Correcte: Com el fitxer no existeix en el directori donat, només guarda data d'última modificació i el path. Tant el nom com la extensió són Strings buits.