

Lliurament 1

Índex

1. Objectius.....	2
2. Material pel lliurament.....	2
3. Descripció del lliurament.....	2
3.1. Creació del projecte.....	2
3.2. Classe InicialadorAplicacioUB.....	3
3.3. Classe AplicacioUB1.....	3
3.3.1. Mètodes principals	3
3.3.2. Implementació del menú d'opcions i la lògica del programa	3
3.4. Creació de la classe per emmagatzemar les dades dels fitxers multimèdia.....	4
3.4.1. Classe FitxerMultimedia	4
3.4.2. Classe CarpetaFitxers.....	5
3.5. Gestió de les dades d'entrada	7
4. Ajuda pel lliurament	7
5. Format del lliurament.....	7
6. Data límit del lliurament.....	8

Programació 2. Projecte de Pràctiques

Grau d'Enginyeria Informàtica. Facultat de Matemàtiques i Informàtica. UB
Curs 2018-2019.

1. Objectius

Organitzar fitxers multimèdia dins una carpeta de fitxers. Cada fitxer multimèdia quedarà representat pel seu fitxer, guardant la següent informació:

- Camí on es troba el fitxer (ex. /home/prog2/fitxers/)
- Nom del fitxer (sense extensió en cas que la tingui)
- Extensió (ex. mp3, mp4, jpg, ...)
- Data última modificació
- Descripció del fitxer multimèdia (ex. "Foto esquuada a la Molina")

Per tant, el dipòsit de fitxers multimèdia en realitat serà una carpeta de fitxers. Per tal d'interactuar amb la carpeta, caldrà definir un menú amb les opcions següents:

1. Afegir fitxer multimèdia: Demana les dades d'un fitxer i afegeix el fitxer a la carpeta de fitxers.
2. Eliminar fitxer multimèdia: Elimina un fitxer de la carpeta.
3. Mostrar carpeta: Mostra el contingut de la carpeta de fitxers, mostrant davant de cada fitxer, el nombre de la seva posició a la carpeta començant per 1.
4. Sortir: Surt de l'aplicació.

Per aquesta pràctica, els fitxers no han d'existir físicament a disc. Per tant, les dades del fitxer en el moment de la creació no cal que es corresponguin amb les de fitxers a disc. Nota: La data de modificació la podeu entendre com la data en la que s'executi l'aplicació.

2. Material pel lliurament

Per aquest lliurament us proporcionem una llibreria **UtilsProg2.jar** que conté la classe:

- **Menu**

Haureu d'utilitzar aquesta classe en el desenvolupament del lliurament. Podeu trobar aquesta llibreria al Campus Virtual i afegir-la al vostre projecte.

3. Descripció del lliurament

A continuació us anirem plantejant els diferents passos per resoldre la pràctica proposada. Us recomanem que seguiu aquests passos.

3.1. Creació del projecte

Programació 2. Projecte de Pràctiques

Grau d'Enginyeria Informàtica. Facultat de Matemàtiques i Informàtica. UB
Curs 2018-2019.

El primer pas serà crear un projecte al Netbeans, al qual li heu de posar com a nom **Cognom1Nom1Cognom2Nom2**, on 1 i 2 fa referència als dos membres de la parella, i tenint en compte les següents consideracions:

- La primera lletra de cada part en majúscula i la resta en minúscula.
- Eviteu utilitzar accents i caràcters del tipus ñ o ç.
- La classe principal s'ha de dir **IniciadorAplicacioUB**, i el paquet per defecte **edu.ub.prog2.Congom1Nom1Cognom2Nom2.vista**. En el nom del paquet utilitzeu els mateixos criteris anteriors.
- En el NetBeans s'ha d'indicar la classe principal com **edu.ub.prog2.Congom1Nom1Cognom2Nom2.vista.IniciadorAplicacioUB**

Per exemple, una estudiant amb nom Dolça Martínez Castaña, hauria de crear un projecte amb el nom *MartinezCastanaDolca*, i la seva classe principal hauria de ser **edu.ub.prog2.MartinezCastanaDolca.vista. IniciadorAplicacioUB**.

3.2. Classe IniciadorAplicacioUB

La classe de la vista **IniciadorAplicacioUB** tindrà una mètode estàtic **main** on s'ha de crear un objecte de tipus **AplicacioUB1** anomenat "aplicacio".

3.3. Classe AplicacioUB1

3.3.1.Mètodes principals

Creeu un mètode d'objecte **gestioAplicacioUB** on s'implementarà el menú de l'aplicació. Aquest mètode ha de tenir la següent signatura:

```
public void gestioAplicacioUB();
```

Un cop creat, aquest mètode és cridarà des del mètode **main** de la classe **IniciadorAplicacioUB**:

```
aplicacio.gestioAplicacioUB ();
```

3.3.2.Implementació del menú d'opcions i la lògica del programa

Un cop tenim el projecte i la classe principal, definiu la lògica del programa, o sigui, com es comportarà el programa dins del mètode **gestioAplicacioUB**. Donat que encara no hem definit res, de moment només es mostrarà un missatge per cada opció del menú, indicant quina opció s'ha triat, excepte en el cas de l'opció de sortir, que finalitzarà l'aplicació.

Programació 2. Projecte de Pràctiques

Grau d'Enginyeria Informàtica. Facultat de Matemàtiques i Informàtica. UB
Curs 2018-2019.

Per fer aquest punt, disposeu de la classe **Menu** dins la llibreria. Aquesta classe gestiona un menú genèric, permetent mostrar la llista d'opcions i controlar la selecció d'una opció per part de l'usuari.

3.4. Creació de la classe per emmagatzemar les dades dels fitxers multimèdia

Per fer aquest lliurament, utilitzeu el mecanisme d'herència, que veureu a fons a les classes de teoria. De moment no us heu de preocupar del què implica aquest fet, us donarem els detalls per a que el pugueu utilitzar de forma pràctica. Les classes que utilitzeu per guardar les dades han d'anar dins el [paquet Model] (***edu.ub.prog2.Congom1Nom1Cognom2Nom2.model***).

3.4.1. Classe FitxerMultimedia

El primer pas serà implementar una classe **FitxerMultimedia** on guardarem la informació d'un fitxer (amb la informació indicada en l'apartat d'objectius de l'enunciat). Aquesta classe heretarà de la classe **File** de Java, que conté la definició dels atributs i mètodes bàsics que també tindrà un fitxer multimèdia. Per fer l'herència, utilitzareu el següent codi:

```
public class FitxerMultimedia extends File {  
  
}
```

El mateix NetBeans us informarà de que heu d'afegir un constructor i ell mateix us proposarà les diferents opcions per crear-lo. Podeu definir el constructor de la classe que rep un String corresponent al camí al fitxer. La signatura d'aquest mètode ha de ser:

```
public FitxerMultimedia(String cami);
```

A més, caldrà que afegiu tots els atributs demanats en l'enunciat, els mètodes d'accés (*getters*) i modificació (*setters*) corresponents, un mètode per tal de demanar les dades del fitxer per teclat i "setejar" els valors dels atributs, un mètode per comprovar si dos fitxers són iguals i, finalment, un mètode per mostrar la informació del fitxer en format cadena de text.

Implementeu primer els *getters* amb les següent signatures:

Programació 2. Projecte de Pràctiques

Grau d'Enginyeria Informàtica. Facultat de Matemàtiques i Informàtica. UB
Curs 2018-2019.

```
public Date getUltimaModificacio();  
  
public String getCamiAbsolut(); // sense nom del fitxer  
  
public String getNomFitxer();  
  
public String getExtensio();  
  
public String getDescripció();
```

Aquests mètodes retornen, respectivament, la data de l'última modificació del fitxer (que serà la data en la que s'executi la aplicació), el camí absolut al fitxer (sense la part del nom del fitxer), el nom del fitxer, la seva extensió i una descripció. A continuació, creeu els *setters* corresponents als mateixos atributs.

A continuació, caldrà definir un mètode per comparar si dos fitxers són iguals (el passat per paràmetre i l'objecte que rep la crida en qüestió). La signatura d'aquest mètode ha de ser:

```
public boolean equals(Object fitxerMultimedia);
```

i sobreescriurà el mètode ***equals*** existent en la classe *File*. Direm que dos fitxers són iguals si el seu estat (atributs) és el mateix.

Per últim, no oblideu implementar el mètode ***toString*** – tot sobreescrivint el de ***File*** – per mostrar el resum dels atributs del fitxer multimèdia en forma de cadena de text:

```
public String toString();
```

Penseu que és també bona pràctica que aquest mètode utilitzi *getters*, enlloc d'accedir al valor dels atributs directament. Aquesta funció s'utilitzarà més endavant per mostrar la informació de cada fitxer d'una carpeta de fitxers.

3.4.2. Classe CarpetaFitxers

Un cop tenim la classe necessària per guardar la informació d'un fitxer multimèdia, ara definiu la classe que representarà una carpeta de fitxers. Igual que en el cas del fitxer multimèdia, caldrà que implementeu una sèrie de mètodes que us permetran manipular aquesta classe. Els mètodes són:

Programació 2. Projecte de Pràctiques

Grau d'Enginyeria Informàtica. Facultat de Matemàtiques i Informàtica. UB
Curs 2018-2019.

```
public int getSize();  
  
public void addFitxer(File fitxer) ;  
  
public void removeFitxer(File fitxer) ;  
  
public File getAt(int position) ;  
  
public void clear();  
  
public boolean isFull();
```

Aquests mètodes serviran per:

- **getSize:** Retornar el nombre d'elements que hi ha a la carpeta.
- **addFitxer:** Afegir un nou fitxer a la carpeta. La carpeta permet fitxers duplicats.
- **removeFitxer:** Eliminar un fitxer de la carpeta si coincideix amb el fitxer passat.
- **getAt:** Retornar el fitxer a la posició indicada de la carpeta.
- **clear:** Eliminar tots els elements de la carpeta.
- **isFull:** Indicar si la carpeta està plena o per el contrari podem afegir més elements.

Per implementar la carpeta de fitxers heu d'utilitzar la classe **ArrayList** de Java (java.util.ArrayList). Aquesta carpeta ha de tenir una capacitat màxima de 100 fitxers multimèdia, o bé se li pot donar la mida màxima com a paràmetre en la construcció de la taula.

Per últim, no oblideu implementar els mètodes **toString** per mostrar el resum de la carpeta de fitxers i que alhora faci servir el mètode **toString** dels objectes fitxer multimedia. Aquest resum s'haurà de mostrar de la següent manera:

Carpeta Fitxers:

=====

[1] Descripció= Nadal, data=Thu Jan 08 12:34:02 CET 2015, nom fitxer=carmen, ext=mp4, camí complet=F:\carmen.mp4

[2] Descripció=Vacances a la platja, name, data=Thu Jan 08 12:35:02 CET 2015, nom fitxer=video100, ext=mp4, camí complet=F:\video100.mp4

3.5. Gestió de les dades d'entrada

Un cop definides les classes que permetran guardar els fitxers multimèdia i les llistes de fitxers, ja podeu començar a donar funcionalitat a les **opcions 1, 2 i 3 del menú**. En tot moment tingueu en compte que es valorarà la modularitat del vostre codi, la reutilització i el seguiment de les bones pràctiques de programació. A més, sempre que pugueu utilitzeu els mètodes estàndards dels objectes (***toString***, ***equals***, ...).

4. Ajuda pel lliurament

Utilitzeu la classe [File](#) de Java per implementar la classe **FitxerMultimedia**. Com podeu consultar a l'ajuda de Java la classe **File** disposa dels següents mètodes:

```
public String getName();
```

Retorna el nom de l'arxiu o el nom del directori del camí. Això és només l'últim nom de la seqüència del camí del fitxer. Si la seqüència de nom és buida, llavors es retorna un String buit. Es podrà utilitzar per obtenir el nom del fitxer multimèdia.

```
public String getAbsolutePath();
```

Retorna un String amb el camí absolut d'aquest fitxer. Es podrà utilitzar per obtenir el camí complet (directori, nom i extensió) del fitxer multimèdia.

Per més informació sobre aquestes classes consulteu [l'ajuda de Java](#).

Utilització de la llibreria **UtilsProg2.jar**:

- Podeu instanciar objectes de la classe **Menu** a la vostra classe vista per implementar la gestió del menú de la aplicació. Teniu un exemple de com utilitzar la classe **Menu** al Campus Virtual.

5. Format del lliurament

El lliurament consistirà en tot el codi generat en els diferents punts de l'enunciat, juntament amb la documentació especificada en aquest apartat.

En concret, cal generar un fitxer comprimit (ZIP) amb el nom dels dos membres de la parella: **Cognom1Nom1Cognom2Nom2_L1**, que contingui:

1. El projecte sencer de NetBeans

Tot el codi generat ha d'estar correctament comentat per a poder executar el JavaDoc, generant automàticament la documentació en línia del codi.

Programació 2. Projecte de Pràctiques

Grau d'Enginyeria Informàtica. Facultat de Matemàtiques i Informàtica. UB
Curs 2018-2019.

2. La memòria del lliurament.

La memòria ha de contenir els punts descrits en la normativa de pràctiques i els punts següents:

1. Explicar breument les classes implementades.
2. 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. No cal incloure la llista d'atributs i mètodes.
3. Explicar com heu implementat i on heu utilitzat el mètode **isFull**.
4. 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?

6. Data límit del lliurament

Consultar el calendari de lliuraments al Campus Virtual.