

MEMORIA DEL LLIURAMENT 2

PROGRAMACIO 2

DATA DEL LLIURAMENT: 4-4-2013

NOM: VICTOR GOMEZ FARRUS

GRUP: B, DNI: 49184353 A

INDEX:

Introducció

Anàlisi(mòduls i funcionalitat)

Desenvolupament

Resultats

INTRODUCCIO:

L'objectiu principal és aconseguir organitzar les imatges que tinguem en una biblioteca i albums. Farem una classe per la biblioteca, `Bibliotecalmatges` i una altra per cada album, `AlbumImatges`. Les dues classes heretan les propietats de `Llistalmatges` de la practica 1. La biblioteca i l'album tindran propietats diferents que explicaré més endavant. Utilitzarem "super" per poder accedir als mètodes que no necessitem sobrecarregar de `Llistalmatges`.

A conseqüència de tot això, el menú i els mètodes del visor també canviaran.

També se'ns ha demanat d'implementar una classe que faci de controlador del visor, `CtrlVisor`, que heretarà una classe del .jar que se'ns ha proporcionat i una altra que s'encarregui de gestionar les dades (la qual estarà dins model), `DadesVisor`.

ANALISI:

El programa segueix dividit en aquestes 3 parts:

VISTA:

Conté les classes principals i les que permetran una relació interactiva amb l'usuari. És la manera que tenim de veure les funcionalitats i dades de la nostra aplicació.

MODEL:

Conté les classes que serviran per manejar les dades, és a dir, les que contindran les estructures necessàries per poder gestionar la biblioteca i els albums d'imatges.

CONTROLADOR:

Conté les classes que controlen l'execució i les encarregades de visualitzar les imatges, etc.

DESENVOLUPAMENT:

La universitat em proporciona un arxiu .jar que conté unes classes que em seran d'utilitat. Les aniré mencionant a mesura que les utilitzi.

Dividiré el meu desenvolupament en els mòduls dels que he parlat a l'anàlisi. Ressaltaré els elements de la manera següent:

Mètodes Tipus de variable Atribut de classe Classe

VISTA:

#VisorUB2:

En el lliurament 2 de pràctiques, serà la classe principal(conté el main). Aquesta classe proporciona un menú via consola, el qual permet les 6 opcions demanades a l'enunciat.

Els atributs d'aquesta classe seran tots privats:

-*enum OpcionsMenuVisorUB2*: És el **enum** corresponent al menú principal. És un atribut estàtic.

-*enum OpcionsGestioBiblio*: És el **enum** corresponent a un menú secundari per gestionar la biblioteca. És un atribut estàtic.

-*enum OpcionsGestioAlbums*: És el **enum** corresponent a un menú secundari per gestionar els àlbums. És un atribut estàtic.

-*enum OpcionsGestioAlbum*: És el **enum** corresponent a un menú secundari per gestionar un àlbum. És un atribut estàtic.

-*CtrlVisor controlador*: Serà el controlador que servirà per fer totes les accions que relacionin l'usuari amb les dades.

- ***String[] descMenuVisorUB***: Cada part de l'array correspon a una opció del menú principal. És un atribut estàtic.
- ***String[] descGestioBiblio***: Cada part de l'array correspon a una opció del menú secundari de la biblioteca. És un atribut estàtic.
- ***String[] descGestioAlbums***: Cada part de l'array correspon a una opció del menú secundari dels àlbums. És un atribut estàtic.
- ***String[] descGestioAlbum***: Cada part de l'array correspon a una opció del menú secundari d'un àlbum. És un atribut estàtic.
- ***Menu<OpcionsMenuVisorUB2> menuPrinc***: Utilitzant ***Menu*** proporcionada al .jar, serveix per crear el menú principal.
- ***Menu<OpcionsGestioBiblio> menBiblio***: Utilitzant ***Menu*** proporcionada al .jar, serveix per crear un menú secundari per gestionar la biblioteca.
- ***Menu<OpcionsGestioAlbums> menAlbum***: Utilitzant ***Menu*** proporcionada al .jar, serveix per crear un menú secundari per gestionar els àlbums.
- ***Menu<OpcionsGestioAlbum> menAlbum***: Utilitzant ***Menu*** proporcionada al .jar, serveix per crear un menú secundari per gestionar un àlbum.

Els mètodes d'aquesta classe seran tots privats:

- ***void gestioVisorUB()***: Permetrà gestionar per consola el programa començant pel menú principal. Ho farà de manera interactiva amb l'usuari gràcies a la inicialització de un ***Scanner***. Mentre no triem l'opció sortir, el programa es mantindrà en funcionament. Cada opció cridarà els següents mètodes segons convingui:

- **void gestioMenuBiblio(Scanner sc)** : Crea un menú per gestionar la biblioteca d'imatges i permet interactuar amb l'usuari de la mateixa manera que el menu principal.
- **void gestioMenuAlbums(Scanner sc)** : Crea un menú per gestionar els àlbums d'imatges i permet interactuar amb l'usuari de la mateixa manera que el menú principal.
- **void gestioMenuAlbum(Scanner sc, int idxAlb)** : Crea un menú per gestionar un àlbum d'imatges i permet interactuar amb l'usuari de la mateixa manera que el menú principal.
- **void createImg(Scanner sc)** : Serveix per crear una imatge a partir d'un fitxer ubicat en un lloc introduït per **Scanner** de la llista.
- **void addImgBibl(Scanner sc)** : Utilitza **createImg(sc)** i afegeix la imatge a la biblioteca.
- **void addImgAlb(Scanner sc, int idxAlb)** : Utilitza **createImg(sc)** i afegeix la imatge a l'àlbum situat a idxAlb.
- **void showBibl()**: Serveix per mostrar per pantalla les dades de la biblioteca.
- **void showAlb(int idxAlb)**: Serveix per mostrar per pantalla les dades de l'àlbum situat a idxAlb.
- **void delImgBibl (Scanner sc)**: Serveix per eliminar una imatge de la biblioteca a partir del seu índex.
- **void delImgAlb (Scanner sc, int idxAlb)** :Serveix per eliminar una imatge a partir del seu índex de l'àlbum situat a idxAlb.
- **void showImgBibl (Scanner sc)**: Serveix per mostrar una imatge de la biblioteca a partir del seu índex.

- **void showImgAlb (Scanner sc, int idxAlb)** :Serveix per mostrar una imatge a partir del seu índex de l'àlbum situat a idxAlb.
 - **void modDataAlb (Scanner sc, int idxAlb)** :Serveix per modificar les dades de l'àlbum situat a idxAlb.
 - **void addAlb(Scanner sc)** : Crea i afegeix un àlbum a la llista d'àlbums.
 - **void showAlbs ()**: Serveix per identificar per pantalla cada àlbum pel seu títol.
 - **void delAlb(Scanner sc)** : Elimina un àlbum de la llista d'àlbums.
 - **int chooseAlb(Scanner sc)** : Retorna l'índex de l'àlbum que volem gestionar.
 - **void saveData (Scanner sc)**: Serveix per guardar les nostres dades en una ubicació introduïda.
 - **void retrData (Scanner sc)**: Serveix per recuperar les nostres dades a partir d' una ubicació introduïda.
 - **void clear()**: Serveix per fer un clear de les dades actuals.
- (·) El **main** inicialitza un **VisorUB2**, i utilitza el mètode **gestioVisorUB** amb els paràmetres inicialitzats.
- (·) El **constructor** d'aquesta classe inicialitza **controlador**.

MODEL:

#Imatge:

Representarà una imatge de l'àlbum o la biblioteca.

Heretarà la classe abstracta **ImageFile**. S'ha d'aclarir que s'entén per ruta completa d'un arxiu com: (ex:

"C:\Users\VICTOR\Pictures\Sample Pictures\Koala.jpg").

Els atributs d'aquesta classe seran tots privats:

-**File arxImatge**: Correspondrà a l'arxiu que manipulem.

A continuació tenim els mètodes abstractes i també públics de **ImageFile**:

-**String getFullPath()**: Serveix per obtenir la ruta completa de l'imatge.

-**String getExtension()**: Serveix per obtenir l'extensió(ex: ".jpg").

-**String toString()**: Serveix per obtenir totes les dades de la imatge(ruta completa i data d'última modificació).

(•) El **constructor** d'aquesta classe té com a paràmetre un **String**, on trobem la ruta completa de l'imatge en concret i la guardem dins **arxImatge**.

#LlistaImatges:

Representarà la llista d'imatges que tenim.

És la classe “progenitora” **Bibliotecalmatges** i de **AlbumImatges** i “filla” de **ImageList** del arxiu .jar.

Els atributs d'aquesta classe seran tots privats:

-**ArrayList<Imatge> llista**: Correspondrà a la llista d'imatges que tenim.

-**int maxtamany**: Correspondrà al màxim tamany de la llista d'imatges.

A continuació tenim els mètodes abstractes i també públics de **ImageList**:

-**int getSize()**: Serveix per obtenir el tamany de **llista**.

-**void addImage(ImageFile i)**: Serveix per afegir una imatge a **llista** sempre i quan el nombre d'imatges dins la llista sigui menor que **maxtamany**.

-**void removeImage(ImageFile i)**: Serveix per eliminar una imatge de **llista**.

-**void getAt(int i)**: Serveix per obtenir una imatge de **llista** a l'índex entrat com a paràmetre.

-**boolean isFull()**: Serveix per saber si el nombre d'elements de **llista** es igual que **maxtamany**.

A continuació tenim altres mètodes útils:

-**boolean contains(Imatge i)**: Serveix per comprovar si **llista** conté el paràmetre. S'ha fet manualment perquè el **contains** de **ArrayList** no retornava el què es desitjava.

-**boolean isFull()**: Serveix per saber si **llista** no conté cap element.

-**void setMaxTamany(int i)**: Serveix per canviar el valor de **maxtamany**.

-**String toString()**: Serveix per obtenir les dades de tots els elements de **llista**, passades a un sol String.

(·) El **constructor** d'aquesta classe inicialitza **llista** i posa a **maxtamany** el valor de 100.

#Bibliotecalmatges:

Representarà la biblioteca d'imatges.

Heretarà les propietats de **Llistalmatges**. Utilitzarem "super" per utilitzar mètodes de **Llistalmatges**.

A continuació tenim els mètodes públics sobrecarregats:

-**void addImage(ImageFile i)**: Serveix per afegir una imatge mitjançant "super". La diferència amb la classe progenitor és que implica més condicions(explicitades a l'enunciat).

-**String toString()**: Fa el mateix però està adaptat a la classe.

(·) El **constructor** d'aquesta classe inicialitza **portada** a null i posa a **maxtamany** ("super") el valor de 1. Aquest valor anirà augmentant cada cop que afegim una imatge. Això s'ha fet de manera preventiva per si de cas el mètode **isFull** de **Llistalmatges** no ens permet executar algunes accions. D'aquesta manera sempre retornarà false.

#AlbumImatges:

Representarà la l'album d'imatges.

Heretarà les propietats de **Llistalimatges** . Utilitzarem “super” per utilitzar mètodes de **Llistalimatges**.

Els atributs d'aquesta classe seran tots privats

-**imatge portada**: Correspondrà a la portada de l'àlbum.

-**double[] long_lat**: Correspondrà a la posició GPS(longitud,latitud) de l'àlbum.

-**String titol**: Correspondrà al títol de l'àlbum.

A continuació tenim els mètodes públics sobrecarregats:

-**void addImage(ImageFile i)**: Serveix per afegir una imatge mitjançant “super”. La diferència amb la classe progenitor és que si és la primera imatge que afegim, llavors serà **portada**.

-**String toString()**: Fa el mateix però està adaptat a la classe.

A continuació tenim altres mètodes útils:

-**void setPortada(imatge i)**: Serveix per posar a **portada** el valor del paràmetre.

-**void setTitle(String s)**: Serveix per posar a **titol** el valor del paràmetre.

-**void setLongLat(double[] long_lat)**: Serveix per posar a **long_lat** el valor del paràmetre.

(•) El **constructor** d'aquesta classe inicialitza **portada** i **titol** a null, crea un nou **long_lat** i posa a **maxtamany** (“super”) el valor de 10(definit a l'enunciat).

#DadesVisor:

Representarà totes les dades del visor(biblioteca i àlbums). Serà **Serializable** per poder fer Streams.

Els atributs d'aquesta classe seran tots privats:

-**Biblioteca**matges **biblioteca**: Correspondrà a la biblioteca d'imatges del programa.

-**ArrayList**<**Album**imatges> **albums**: Correspondrà al conjunt d'àlbums del programa.

A continuació tenim els mètodes públics:

-**Biblioteca**matges **getBibl()**: Retorna **biblioteca**.

-**ArrayList**<**Album**imatges> **getAlbs()**: Retorna **albums**.

A continuació tenim altres mètodes útils:

-**String** **albString()**: Serveix per obtenir un **String** a partir de les dades de **albums**.

-**String** **toString()**: Serveix per obtenir un **String** amb totes les dades.

(·) El **constructor** d'aquesta classe crea **biblioteca** i **albums**.

CONTROLADOR:

#CtrlVisor:

Representarà el controlador del visor.

Heretarà les propietats de **BasicCtrl** del fitxer.jar.

Els atributs d'aquesta classe seran tots privats:

-**DadesVisor** **dadesVisor**: Correspondrà a les dades del visor.

A continuació tenim els mètodes públics:

-**AlbumImatges** **getAlb(int index)**: Retorna l' **AlbumImatges** situat a l'índex de **getAlbs** d'àlbums de **dadesVisor**.

-**Bibliotecalmatges** **getBibl()**: Retorna la **getBibl** de **dadesVisor**.

-**ArrayList<AlbumImatges>** **getAlbs()**: Retorna **getAlbs** de **dadesVisor**.

-**void** **addAlb()**: Afegeix un àlbum nou a la llista d' **AlbumImatges** de **dadesVisor**.

-**String** **getBiblString()**: Retorna el **toString** de la **Bibliotecalmatges** de **dadesVisor**.

-**String** **getAlbsString()**: Retorna **albString** de **dadesVisor**.

-**String** **getAlbString(int idxAlb)**: Retorna **toString** de l' **AlbumImatges** situat a idxAlb dels àlbums de **dadesVisor**.

-**void** **addImgBibl(Imatge i)**: Serveix per afegir la imatge paràmetre a la **Bibliotecalmatges** de **dadesVisor**.

-**void** **addImgAlb(Imatge i, int idxAlb)**: Serveix per afegir la imatge paràmetre a l' **AlbumImatges** situat a idxAlb de **getAlbs** de **dadesVisor**.

- void delImgBibl(Imatge i)**: Serveix per eliminar la imatge paràmetre de la **Bibliotecalmatges** de **dadesVisor**.
- void delImgBibl(Imatge i, int idxAlb)**: Serveix per eliminar la imatge paràmetre de l' **AlbumImatges** situat a idxAlb de **getAlbs** de **dadesVisor**.
- void clearDades()**: Serveix per posar **dadesVisor** a null.
- void showImg(Imatge i)**: Serveix per mostrar una imatge paràmetre.
- Imatge getImgBibl(int index)**: Serveix per obtenir una imatge de la **Bibliotecalmatges** de **dadesVisor** a partir del parametre index.
- Imatge getImgAlb(int index, int idxAlb)**: Serveix per obtenir una imatge a partir del parametre index de l' **AlbumImatges** situat a idxAlb de **getAlbs** de **dadesVisor**.
- void emptyBibl()**: Retorna true si la **Bibliotecalmatges** de **dadesVisor** esta buida.
- void emptyAlb(int idxAlb)**: Retorna true si l' **AlbumImatges** situat a idxAlb de **getAlbs** de **dadesVisor** està buit.
- void fullAlb(int idxAlb)**: Retorna true si l' **AlbumImatges** situat a idxAlb de **getAlbs** de **dadesVisor** està ple.
- void setTitleAlb(int idxAlb, String title)**: Amb el paràmetre title fa **setTitle** a l' **AlbumImatges** situat a idxAlb de **getAlbs** de **dadesVisor**.
- void setLongLatAlb(int idxAlb, double[] longLat)**: Amb el paràmetre longLat fa **setLongLat** a l' **AlbumImatges** situat a idxAlb de **getAlbs** de **dadesVisor**.

-**void setPortadaAlb(int idxAlb, Imatge imgPortada)**: Amb el paràmetre imgPortada fa **setPortada** a l' **AlbumImatges** situat a idxAlb de **getAlbs** de **dadesVisor**.

-**void setTamanyAlb(int idxAlb, int tamany)**: Amb el paràmetre tamany fa **setMaxTamany** a l' **AlbumImatges** situat a idxAlb de **getAlbs** de **dadesVisor**.

-**void emptyAlbs()**: Retorna true si **ArrayList<AlbumImatges>** de **dadesVisor** esta buit.

-**void containsAlb(String titol)**: Retorna true si **ArrayList<AlbumImatges>** de **dadesVisor** esta te un àlbum amb el mateix títol que el paràmetre.

-**void saveDades(String ubicacio)**: Serveix per guardar **dadesVisor** en un fitxer existent sobreescrivint-lo a la ubicació paràmetre. Es fa mitjançant Streams.

-**void saveNewDades(String ubicacio)**: Serveix per guardar **dadesVisor** en un fitxer nou a la ubicació paràmetre. Es fa mitjançant Streams.

-**void retrDades(String ubicacio)**: Serveix per canviar **dadesVisor** donant-li els valors que té el fitxer situat a la ubicació paràmetre. Es fa mitjançant Streams.

(·) El **constructor** d'aquesta classe crea **dadesVisor**.

RESULTATS:

En aquest lliurament l'aplicació ja té una gestió d'àlbums i de biblioteca. He hagut de fer un "repàs" d'algunes classes per adaptar-les a aquesta nova pràctica (simplement he afegit algun mètode útil).

Pel que fa a la classe principal, l'he canviat totalment i he corregit el que se m'ha suggerit en la retroacció del lliurament 1.

En un principi havia posat un **AlbumImatges** com a atribut de **DadesVisor**, ja que em resultava molt més còmode a l'hora d'implementar el codi. Aquell **AlbumImatges** era "l'àlbum actual". Ara he canviat la implementació i es passa tot per índexs de coses, de manera que només hi ha els atributs justos i necessaris.