

Lliurament 1

1. Objectius

Organitzar imatges dins una biblioteca. Cada imatge quedarà representada 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. jpg, bmp, png, ...)
- Data última modificació

Per tant, la biblioteca d'imatges en realitat serà una llista de fitxers. Per tal d'interactuar amb la llista, caldrà definir un menú amb les opcions següents:

1. Afegir Fitxer: Demanarà les dades d'un fitxer i l'afegirà a la llista de fitxers.
2. Mostrar llista: Mostra el contingut de la llista de fitxers, mostrant davant de cada fitxer, el nombre de la seva posició a la llista començant per 1.
3. Eliminar imatge: Elimina de la llista el fitxer corresponent a una posició donada.
4. Guardar llista: Guarda el contingut de la llista en un fitxer.
5. Recuperar llista: Carrega una llista prèviament guardada d'un fitxer.
6. Sortir: Surt de l'aplicació.

2. Descripció del lliurament

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

2.1 Creació del projecte

El primer pas serà crear un projecte, al qual li heu de posar com a nom **Cognom1Cognom2Nom**, tenint en compte les següents consideracions:

- La primera lletra de cada part en majúscula i la resta en minúscula.
- Eviteu utilitzar accents i caràcters estranys (ñ o ç).
- La classe principal s'ha de dir **VisorUB1**, i el paquet per defecte **edu.ub.prog2.Cognom1Cognom2Nom.vista**. En el nom del paquet utilitzeu els mateixos criteris anteriors.

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.VisorUB1*.

Programació 2. Projecte de Pràctiques

Grau d'Enginyeria Informàtica. Facultat de Matemàtiques. UB
Curs 2012-2013. Professorat: Laura Igual, Xavier Baró i Santi Seguí

2.2 Implementació del menú d'opcions i la lògica del programa

Un cop tenim el projecte principal, definirem la lògica del programa, o sigui, com es comportarà el programa. 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 la opció de sortir, que finalitzarà l'aplicació.

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. Teniu un exemple de com utilitzar aquesta classe al campus virtual. És un exemple simple d'utilització de la classe, **no us val utilitzar-lo tal qual**, cal adaptar-lo a les necessitats i tenint en compte els criteris de modularitat de la pràctica.

2.3 Creació de les classes per emmagatzemar les dades dels fitxers d'imatge

Per fer aquest lliurament, utilitzarem el mecanisme d'herència, que veureu a fons a les classes de teoria. De moment no us heu de preocupar pel què implica aquest fet, us donarem els detalls per a que el pugueu utilitzar de forma pràctica.

A la llibreria que us hem proporcionat trobareu les classes **ImageList** i **ImageFile**, que contenen la definició dels mètodes bàsics que tindran una llista d'imatges i un fitxer d'imatge respectivament. Les classes que utilitzeu per guardar les dades han d'anar dins el [paquet Model] (**edu.ub.prog2.Congom1Cognom2Nom.model**).

2.3.1 Classe Imatge

El primer pas serà implementar una classe **Imatge** on guardarem la informació d'un fitxer (amb la informació indicada a l'enunciat). Aquesta classe heretarà de la classe **ImageFile**, i haureu de definir-ne els mètodes d'accés a la informació. Per fer la herència, utilitzareu el següent codi:

```
public class Imatge extends ImageFile {  
  
}
```

El mateix Netbeans us informarà de que hi ha mètodes abstractes sense definir, i us proposarà de crear-los ell mateix. Ho podeu fer així o definint-los vosaltres mateixos. Els mètodes de la classe que cal definir són **getFullPath** i **getExtension**, que retornen el camí al fitxer i la seva extensió, respectivament. La signatura d'aquests mètodes ha de ser:

```
public String getFullPath();  
  
public String getExtension() ;
```

Programació 2. Projecte de Pràctiques

Grau d'Enginyeria Informàtica. Facultat de Matemàtiques. UB
Curs 2012-2013. Professorat: Laura Igual, Xavier Baró i Santi Seguí

A més a més, caldrà que afegiu tots els atributs i mètodes que necessiteu per tal de guardar els atributs demanats en l'enunciat. Us pot ser útil utilitzar la classe `File` de Java.

2.3.2 Classe `TaulaImatges`

Un cop tenim la classe necessària per guardar la informació d'una imatge, ara és el torn de definir la classe que representarà una llista d'imatges. De la mateixa manera que en el cas anterior, us donem una classe base, de la qual haureu d'estendre la vostra implementació. A continuació es mostra com heu de declarar la classe:

```
public class TaulaImatges extends ImageList {  
  
}
```

Igual que en el cas del fitxer, caldrà que implementeu una sèrie de mètodes que ja venen definits a la classe base:

```
public int getSize();  
  
public void addImage(ImageFile imageFile) ;  
  
public void removeImage(ImageFile imageFile) ;  
  
public ImageFile getAt(int position) ;  
  
public void clear();  
  
protected boolean isFull();
```

Aquests mètodes serviran per:

- **getSize:** Retorna el nombre d'elements que hi ha a la llista.
- **addImage:** Afegeix una nova imatge a la llista.
- **removeImage:** Elimina una imatge de la llista si coincideix amb la imatge passada. Caldrà definir el mètode ***equals*** per a la vostra classe ***Imatge***.
- **getAt:** Retorna la imatge a la posició indicada de la llista.
- **clear:** Elimina tots els elements de la llista.
- **isFull:** Indica si la llista està plena o per el contrari podem afegir més elements. És un mètode intern a la classe.

Programació 2. Projecte de Pràctiques

Grau d'Enginyeria Informàtica. Facultat de Matemàtiques. UB
Curs 2012-2013. Professorat: Laura Igual, Xavier Baró i Santi Seguí

Per implementar la llista de fitxers heu d'utilitzar les classes vistes a **Programació 1** (Taules, tuples,). En aquesta part podeu suposar que la llista té una capacitat màxima de 100 imatges, o se li pot donar la mida màxima com a paràmetre en la construcció de la taula.

2.4 Gestió de les dades d'entrada

Un cop definides les classes que permetran guardar les imatges i les llistes d'imatges, 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 a més, sempre que pugueu utilitzeu els mètodes estàndards dels objectes (toString, equals, ...).

2.5 Persistència de dades

Arribats a aquest punt ja hauríeu de poder demanar les dades d'una imatge a l'usuari i guardar-les a una llista. També poder mostrar i eliminar elements de la llista.

Ara el què volem és poder guardar les dades a un fitxer i poder-les carregar posteriorment. Implementeu les opcions 4 i 5 del menú.

Per fer aquesta opció, necessiteu seguir els següents passos:

1. Obtenir la ruta al fitxer on voleu guardar les dades o des del qual voleu carregar-les. Necessitareu guardar aquesta informació en un objecte de tipus File. Per exemple, si volem utilitzar el fitxer "dades.dat", farem:

```
File fitxer=new File("dades.dat");
```

2. L'accés de lectura i escriptura a un fitxer es fa mitjançant Streams. Per llegir d'un fitxer utilitzarem un objecte de tipus FileInputStream, i per escriure a un fitxer utilitzarem un objecte FileOutputStream:

```
FileInputStream fin=new FileInputStream(fitxer);
```

```
FileOutputStream fout= new FileOutputStream(fitxer);
```

3. Finalment, existeixen objectes per gestionar la lectura i escriptura d'un objecte a un Stream. Per escriure un objecte utilitzarem un objecte de tipus ObjectOutputStream, mentre que per llegir-lo utilitzarem un objecte de tipus ObjectInputStream:

```
ObjectOutputStream oos = new ObjectOutputStream(fout);
```

```
ObjectInputStream ois = new ObjectInputStream(fin);
```

Programació 2. Projecte de Pràctiques

Grau d'Enginyeria Informàtica. Facultat de Matemàtiques. UB
Curs 2012-2013. Professorat: Laura Igual, Xavier Baró i Santi Seguí

Teniu més informació sobre Streams en el document d'ajuda al lliurament 1. No oblideu tancar correctament els objectes d'accés als fitxers. Tenint en compte que en aquest lliurament només tenim una llista de fitxers, podeu guardar directament aquesta llista d'imatges.

2.6 Implementar una llista d'imatges utilitzant una implementació estàndard

En els passos anteriors heu implementat tota la funcionalitat d'aquest lliurament, utilitzant, bàsicament, els coneixements adquirits a Programació 1. En aquest punt, es demana utilitzar una implementació estàndard de Java de taules per a la llista d'imatges. Implementa la classe *Llistalmatges* que al igual que la classe *Taulalmatges* hereti de la classe *ImageList*, utilitzant com a implementació de la llista la classe *ArrayList* de Java.

Tingueu en compte que caldrà que es pugui utilitzar una llista o l'altra. A la memòria haureu d'explicar quins canvis heu hagut de fer al vostre programa per a que funcioni amb la nova implementació i què cal fer per triar una implementació o l'altra.

3. Material pel lliurament

Per aquest lliurament us proporcionem una llibreria **UtilsProg2.jar** que conté les classes:

- **ImageFile**
- **ImageList**
- **Menu**

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

4. Ajuda pel lliurament

Utilitzeu la classe **File** de Java per implementar la classe **Imatge**.

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

- Podeu instanciar objectes de la classe **Menu** a la vostra classe vista per implementar la gestió del menú del visor. Teniu un exemple de com utilitzar la classe **Menu** al Campus Virtual.
- Les classes **ImageFile** i **ImageList** són classes abstractes que heu d'utilitzar per implementar les vostres classes **Imatge**, **Llistalmatges** i **Taulalmatges**.

Programació 2. Projecte de Pràctiques

Grau d'Enginyeria Informàtica. Facultat de Matemàtiques. UB
Curs 2012-2013. Professorat: Laura Igual, Xavier Baró i Santi Seguí

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 amb el nom:

Cognom1Cognom2Nom_L1.tar.gz

Que conté:

- el directori "src"
- el fitxer build.xml
- i un fitxer "Cognom1Cognom2Nom_L1.jar" amb els .class

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

També cal lliurar una **memòria** amb els següent punts (a més dels descrits en la normativa de pràctiques)

- Introducció al problema tractat en el lliurament (No s'acceptarà una còpia directa de l'enunciat).
- Explicació de les classes implementades.
- Explicar què has pogut reutilitzar del punt 2.3.2 per al punt 2.6, i què cal fer per canviar el tipus d'implementació de la llista d'imatges.
- Proves realitzades per comprovar el correcte funcionament de la pràctica, resultats obtinguts i accions derivades.
- Observacions generals.