

# **Lliurament 2**

## **1. Objectius**

Organitzar imatges en una biblioteca i poder crear diferents àlbums amb les imatges de la biblioteca. En el lliurament 1 heu implementat les classes **TaulaImatges** i **LlistaImatges** que permeten gestionar un conjunt d'imatges. La classe **TaulaImatges** era un exercici introductori, que permetia aprendre a utilitzar l'entorn de programació i repassar els coneixements de Programació 1. A partir d'aquest lliurament, utilitzarem només la classe **LlistaImatges**, que ara no tindrà límit de capacitat. A partir d'aquesta classe, definirem una classe **BibliotecaImatges** que serà una llista d'imatges amb les següents propietats:

1. No permetrà que hi hagin dues imatges iguals. Entenem que dues imatges són iguals si el camí, nom i extensió del fitxers són iguals.
2. Quan s'afegeix una imatge a la biblioteca, es verificarà que el fitxer associat a la imatge existeix.
3. No hi haurà cap límit en el nombre d'imatges

A més a més, definirem una classe **AlbumImatges** que serà una llista d'imatges amb les següents propietats:

1. Permet imatges duplicades
2. Estarà limitat a un màxim d' $N$  imatges, on  $N$  serà un valor que per defecte serà 10, però que es podrà canviar.
3. Cada àlbum portarà associat:
  - Un títol
  - Un autor
  - La imatge a mostrar com a representació de l'àlbum.
    - En cas que no hi hagi cap imatge a l'àlbum, serà null.
    - Per defecte serà la primera imatge afegida a l'àlbum.
    - Es podrà seleccionar la imatge que es desitgi, però ha d'estar a l'àlbum.

L'aplicació podrà tenir una i només una biblioteca però un nombre indeterminat d'àlbums. Els àlbums quedaran identificats pel seu títol, i per tant, no podrem tenir-ne de repetits.

## **Programació 2. Projecte de Pràctiques**

---

Grau d'Enginyeria Informàtica. Facultat de Matemàtiques. UB  
Curs 2014-2015. Professorat: Laura Igual, Santi Seguí i Carles Franquesa

Per tal d'interactuar amb la llista, caldrà definir un menú amb les opcions següents, aprofitant tot el possible del lliurament 1:

1. Gestió Biblioteca: Dona accés a un menú per a la gestió de la biblioteca.
  - 1.1. Afegir Imatge: Demanarà les dades d'un fitxer d'imatge i l'afegirà a la biblioteca.
  - 1.2. Mostrar Biblioteca: Mostra el contingut de la biblioteca, mostrant davant de cada fitxer, el número de la seva posició a la llista començant per 1.
  - 1.3. Eliminar Imatge: Elimina de la biblioteca la imatge corresponent a una posició donada.
  - 1.4. Menú Anterior: Tornarà al menú principal
2. Gestió Àlbums: Dóna accés a un menú per a la gestió dels àlbums, que conté les següents opcions:
  - 2.1. Afegir Àlbum: Demana les dades d'un nou àlbum i l'afegeix a l'aplicació.
  - 2.2. Mostrar Àlbums: Mostra els àlbums disponibles a l'aplicació, mostrant davant de cada àlbum, el número de la seva posició a la llista començant per 1.
  - 2.3. Eliminar Àlbum: Elimina de l'aplicació l'àlbum corresponent a una posició donada.
  - 2.4. Gestionar Àlbum: Permet seleccionar un àlbum dels disponibles a l'aplicació, i obre un menú per a gestionar-lo. Aquest menú tindrà les següents opcions:
    - 2.4.1. Afegir Imatge: Mostra les imatges de la biblioteca i permet seleccionar-ne una, que passarà a formar part de l'àlbum.
    - 2.4.2. Mostrar àlbum: Mostra el contingut de l'àlbum, mostrant davant de cada imatge, el número de la seva posició a la llista començant per 1.
    - 2.4.3. Eliminar Imatge: Mostra les imatges de l'àlbum i permet seleccionar-ne una, que serà eliminada de l'àlbum.
    - 2.4.4. Modificar Dades: Permet modificar els atributs de l'àlbum (nom, posició i imatge de portada)
    - 2.4.5. Menú Anterior: Tornarà al menú anterior.
  - 2.5. Menú Anterior: Tornarà al menú principal.
3. Guardar Dades: Guarda les dades de l'aplicació a un fitxer.
4. Recuperar Dades: Carrega les dades de l'aplicació prèviament guardades d'un fitxer.
5. Sortir: Surt de l'aplicació.

## Programació 2. Projecte de Pràctiques

Grau d'Enginyeria Informàtica. Facultat de Matemàtiques. UB  
Curs 2014-2015. Professorat: Laura Igual, Santi Seguí i Carles Franquesa

Quan una imatge s'elimina de la biblioteca, també cal eliminar-la dels àlbums que la contenen.

Seguint la organització Model-Vista-Controlador (MVC), definirem una classe **DadesVisor** que contingui totes les dades de l'aplicació. Totes les accions que afectin a les dades s'hauran de fer a través d'aquesta classe. També crearem la classe **CtrlVisor** que implementi el controlador de la nostra aplicació.

A més a més, volem evitar que la vista utilitzi directament classes del model, per delegar aquesta tasca al controlador. Per tant, la vista només podrà utilitzar el controlador, i aquest s'encarregarà de fer les crides oportunes a la classe del model, la qual utilitzarà les classes que guarden les dades.



## 2. Material pel lliurament

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

- **ImageFile** (abstract class)
- **InImageList** (interface)
- **ImageList** (abstract class)
- **Menu**
- **BasicCtrl**
- **VisorException**

Haureu d'utilitzar aquestes classes 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 seguïu aquests passos.

### 3.1 Creació del projecte

El primer pas serà crear un nou projecte, al qual li podeu posar com a nom **Cognom1Cognom2Nom\_L2**. De la mateixa manera que en el lliurament 1, la classe principal s'ha de dir **GestioVisorUB**, i el paquet per defecte **edu.ub.prog2.Congom1Cognom2Nom.vista**. En el Netbeans s'ha d'indicar la classe principal com **edu.ub.prog2.Congom1Cognom2Nom.vista.GestioVisorUB**.

## Programació 2. Projecte de Pràctiques

Grau d'Enginyeria Informàtica. Facultat de Matemàtiques. UB  
Curs 2014-2015. Professorat: Laura Igual, Santi Seguí i Carles Franquesa

### 3.2 Classe GestioVisorUB

La classe de la vista **GestioVisorUB** tindrà una mètode estàtic main on s'ha de crear un objecte de tipus **VisorUB2** anomenat "vista" i fer la crida al mètode gestioVisorUB de la classe **VisorUB2** :

```
vista.gestioVisorUB();
```

### 3.3 Classe VisorUB2

#### 3.3.1 Mètodes principals

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

```
public void gestioVisorUB();
```

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

Seguint l'exemple del lliurament 1, creeu la lògica del programa, utilitzant la classe **Menu** de la llibreria i mostrant un missatge per a cada opció. Comproveu especialment que podeu passar d'un menú a un altre correctament i que l'aplicació finalitza correctament.

### 3.3 Creació de les classes principals de l'aplicació

Tal com s'indica al apartat d'objectius de l'enunciat, en aquest lliurament començarem a utilitzar el disseny MVC. Per facilitar la separació dels diferents components, utilitzarem una classe principal per a cada component. Pel que fa a la Vista, ja heu definit la classe **GestioVisorUB** i **VisorUB2**. Ara crearem les classes per al model i per al controlador.

Creeu la classe **DadesVisor** que contindrà totes les dades de l'aplicació. De moment podeu deixar la classe totalment en blanc. Quan comencem a implementar les opcions del menú ja anireu afegint els mètodes que necessitis. Igual que totes les classes relacionades amb el model, aquesta classe haurà d'estar dins el [paquet Model] (**edu.ub.prog2.Congom1Cognom2Nom.model**).

Creeu la classe **CtrlVisor**, que exercirà de controlador de l'aplicació, controlant els diferents processos que es duguin a terme en el visor. Aquesta classe rebrà les indicacions de l'usuari per mitjà de la vista i farà les accions pertinents, comunicant-se amb el model quan sigui necessari. Igual que totes les classes que tinguin a veure amb el control de l'aplicació, aquesta classe anirà dins del [paquet Controlador] (**edu.ub.prog2.Congom1Cognom2Nom.controlador**). Tot i que per

## Programació 2. Projecte de Pràctiques

Grau d'Enginyeria Informàtica. Facultat de Matemàtiques. UB  
Curs 2014-2015. Professorat: Laura Igual, Santi Seguí i Carles Franquesa

aquest lliurament encara no ho necessiteu, per poder resoldre el següent lliurament, caldrà que el vostre controlador hereti les funcionalitats d'un controlador bàsic que us hem proporcionat a la llibreria. Per fer això, definiu aquesta classe com:

```
public class CtrlVisor extends BasicCtrl {  
  
}
```

Un cop definides les classes principals, caldrà que definiu els atributs necessaris per connectar aquestes classes, tenint en compte que la vista només pot accedir al model a través del controlador.

### 3.4 Gestió de la Biblioteca

A partir de les indicacions de l'enunciat, definiu la classe **Bibliotecalmatges** que permet gestionar les imatges de la biblioteca.

Creeu tots els mètodes necessaris per fer la gestió de les opcions 1.1, 1.2 i 1.3, tenint en compte que equivalen a les opcions 1, 2 i 3 del lliurament 1, i per tant, tot i que cal reestructurar el projecte seguint el model MVC, heu de poder reutilitzar gran part del codi. Recordeu que disposeu de la classe **Llistalmatges** del lliurament 1.

Per comprovar si un fitxer existeix o no, teniu el mètode **exists** de la classe **File**.

### 3.5 Gestió dels àlbums

A partir de les indicacions de l'enunciat, definiu la classe **Albumimatges** que permet gestionar les imatges d'un àlbum. Prenent com a exemple la gestió de la biblioteca, implementeu tots els mètodes necessaris per fer la gestió dels àlbums.

### 3.6 Excepcions

En aquest lliurament, utilitzarem excepcions per fer la gestió d'alguns errors, com per exemple, l'error d'afegir una imatge a la biblioteca quan aquesta ja ha estat inclosa prèviament o l'error corresponent a afegir una imatge que no existeix al disc.

S'utilitzarà la classe **VisorException** per representar aquests errors particulars.

A la part del codi on es produeixen aquests problemes, s'hauran de crear aquestes excepcions i llançar-les, fent servir [throw](#).

## Programació 2. Projecte de Pràctiques

---

Grau d'Enginyeria Informàtica. Facultat de Matemàtiques. UB  
Curs 2014-2015. Professorat: Laura Igual, Santi Seguí i Carles Franquesa

### 3.7 Persistència de dades

Igual que al lliurament 1, volem poder guardar totes les dades de l'aplicació. En aquest cas, ja no tenim una llista d'imatges, sinó que tenim una biblioteca i uns àlbums. Seguint la persistència del lliurament 1, implementa les opcions 3 i 4 del menú principal. Recorda que les dades de l'aplicació es troben al model.

## 4. Ajuda pel lliurament

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**, **LlistatImatges**.
- La classe **BasicCtrl** és una classe capaç de respondre davant de certs events. Us servirà per realitzar tasques automàtiques al controlador, i per tant cal que la classe **CtrlVisor** hereti d'aquesta classe.

## 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\_L2.tar.gz

Que conté:

- el projecte sencer de Netbeans
- la memòria del lliurament

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

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

1. Introduïu el problema tractat en el lliurament (No s'acceptarà una còpia directa de l'enunciat).
2. Expliqueu les classes implementades.
3. Expliqueu quines classes heu pogut reutilitzar del primer lliurament per a fer aquest. Quins canvis sobre les classes reutilitzades heu necessitat fer i perquè.

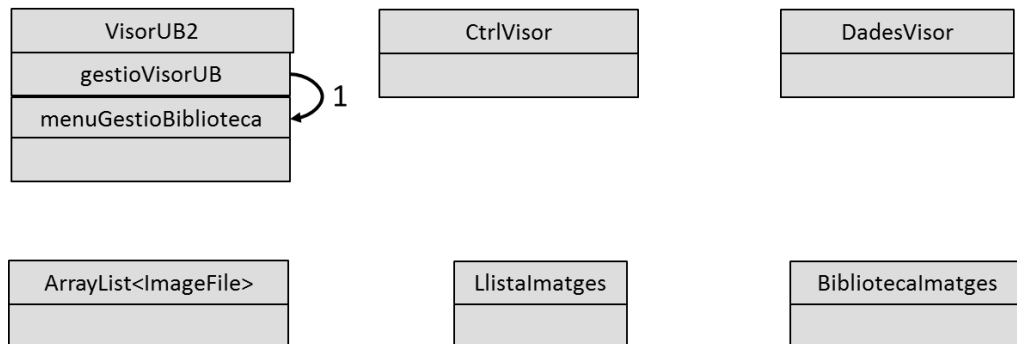
## Programació 2. Projecte de Pràctiques

Grau d'Enginyeria Informàtica. Facultat de Matemàtiques. UB  
Curs 2014-2015. Professorat: Laura Igual, Santi Seguí i Carles Franquesa

4. Expliqueu quants objectes s'han creat en l'execució d'aquest mètode main si estem a l'última línia:

```
public static void main(String[] args) {  
    // Creem un objecte de la vista  
    VisorUB2 vista=new VisorUB2();  
  
    // Inicialitza l'execució de la vista  
    vista.gestioVisor();  
}
```

5. Expliqueu com heu implementat i on heu utilitzat el mètode equals heretat de la classe Object.
6. Expliqueu com heu implementat la comprovació de que una imatge no està inclosa en la biblioteca.
7. Expliqueu com heu utilitzat la classe **VisorException** al vostre codi.
8. Expliqueu si heu fet servir la sobrecàrrega de mètodes a la classe CtrlVisor i detallar com.
9. Completeu aquest dibuix per tal de mostrar el recorregut que fa el teu programa quan s'executa l'opció de afegir una imatge a la biblioteca. Especificar els mètodes que es criden en cadascuna de les classes. Fer servir fletxes i números per indicar l'ordre de les crides.



10. Expliqueu les proves realitzades per comprovar el correcte funcionament de la pràctica, resultats obtinguts i accions derivades.
11. Observacions generals.

## 6. Data límit del lliurament

El lliurament s'ha de realitzar abans del dia 12 d'abril a les 23:55h.