

## **Lliurament 3**

### **1. Objectius**

Una vegada tenim organitzades les imatges dins la biblioteca i els àlbums, en aquest lliurament es tracta de definir les opcions necessàries per poder visualitzar tota la biblioteca o un àlbum determinat.

A més, es podran aplicar filtres a les imatges (sèpia o blanc i negre) així com visualitzar les imatges amb aquests filtres.

Per tal d'incloure aquestes noves funcionalitats a l'aplicació actual, caldrà modificar el menú principal. A continuació es mostra el menú del lliurament 2 i marcat amb negreta els canvis que s'han d'incorporar:

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. **Visualitzar Imatge:** Permet visualitzar una imatge de la biblioteca.
    - 1.4.1. **Visualització sense filtre:** Visualitza la imatge sense aplicar cap filtre.
    - 1.4.2. **Visualització amb filtre sèpia:** Visualitza la imatge aplicant un filtre sèpia. Serveix de pre-visualització abans de transformar la imatge aplicant el filtre definitivament (opcions en 1.5).
    - 1.4.3. **Visualització amb filtre blanc i negre:** Visualitza la imatge aplicant un filtre blanc i negre. Serveix de pre-visualització abans de transformar la imatge aplicant el filtre definitivament (opcions en 1.5).
  - 1.5. **Transformar imatge:** Aplica un filtre a una imatge de la biblioteca
    - 1.5.1. **Filtre sèpia:** Aplica un filtre per transformar la imatge a tons sèpia.
    - 1.5.2. **Filtre blanc i negre:** Aplica un filtre per transformar la imatge a blanc i negre.
    - 1.5.3. **Sense filtre:** Retorna a la imatge original.
  - 1.6. **Visualitzar Biblioteca:** Visualitzar totes les imatges de la biblioteca.

## Programació 2. Projecte de Pràctiques

---

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

- 1.7. Menú Anterior: Tornarà al menú principal
2. Gestió Àlbums: Dona 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. Mostar 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. **Visualitzar Imatge**: Permet visualitzar una imatge de l'àlbum.
      - 2.4.4.1. **Visualització sense filtre**: Visualitza la imatge sense aplicar cap filtre.
      - 2.4.4.2. **Visualització amb filtre sèpia**: Visualitza la imatge aplicant un filtre sèpia. Serveix de pre-visualització abans de transformar la imatge aplicant el filtre definitivament (opcions en 2.4.5).
      - 2.4.4.3. **Visualització amb filtre blanc i negre**: Visualitza la imatge aplicant un filtre blanc i negre. Serveix de pre-visualització abans de transformar la imatge aplicant el filtre definitivament (opcions en 2.4.5).
    - 2.4.5. **Transformar imatge**: Aplica un filtre a una imatge de l'àlbum.
      - 2.4.5.1. **Filtre sèpia**: Aplica un filtre per transformar la imatge a tons sèpia.
      - 2.4.5.2. **Filtre blanc i negre**: Aplica un filtre per transformar la imatge a blanc i negre.
      - 2.4.5.3. **Sense filtre**: Retorna a la imatge original.
    - 2.4.6. **Visualitzar Àlbum**: Visualitzar totes les imatges d'un àlbum.

## Programació 2. Projecte de Pràctiques

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

- 2.4.7. Modificar Dades: Permet modificar els atributs de l'àlbum (nom, autor i imatge de portada)
- 2.4.8. 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.

Durant la visualització de la biblioteca o d'un àlbum, es mostrarà a l'usuari el menú i aquest podrà seguir actuant amb ell.

Quan una imatge de la biblioteca es transforma aplicant un filtre, aquest filtre no es reflexa a la imatge afegida als àlbums prèviament a la transformació, però si es reflexa si s'afegeix posteriorment.

Recordeu que abans d'afegir una imatge a la biblioteca cal que comprovem que el fitxer existeix i que no està ja inclòs en la llista. A més, cal que gestioneu aquests errors mitjançant excepcions. També podeu fer servir excepcions per gestionar altres errors que vulgueu controlar.

Continuem amb la mateixa organització Model-Vista-Controlador (MVC).



## 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.

## 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. 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\_L3**. 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**.

#### 3.2 Classe GestioVisorUB

La classe de la vista **GestioVisorUB** tindrà una mètode estàtic main on s'ha de crear un objecte de tipus **VisorUB3** anomenat "vista" i fer la crida al mètode **gestioVisorUB** de la classe **VisorUB3**:

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

#### 3.3 Classe VisorUB3

Implementar el mètode d'objecte **gestioVisorUB** on s'implementarà el menú de l'aplicació utilitzant la classe **Menu** de la llibreria. A més, s'hauran d'utilitzar les noves classes del model: **ImatgeBN**, **ImatgeSepia** (següent apartat).

#### 3.4 Imatge amb filtres

Com si es tractés d'un Instagram, on podem escollir entre molts tipus de filtres diferents, el nostre visor podrà escollir entre dos filtres diferents: un que permet veure la imatge en blanc i negre i l'altre en tons sepia.

Tindrem l'opció de pre-visualitzar l'efecte dels filtres en imatges de la biblioteca (opcions 1.4.1, 1.4.2 i 1.4.3 del menú) i dels àlbums (opcions 2.4.4.1, 2.4.4.2 i 2.4.4.3 del menú), així com de filtrar de forma permanent la imatge guardada en la biblioteca (opcions 1.5.1, 1.5.2 del menú) i en un dels àlbums (opcions 2.4.5.1, 2.4.5.2 del menú).

L'operació de filtratge de la imatge, comportarà substituir la imatge que està inclosa en la biblioteca o àlbum per una de nova que ara serà de tipus **ImatgeBN** o **ImatgeSepia**.

Les classes **ImatgeBN** i **ImatgeSepia** són una especialització de la classe **Imatge**, ja que en elles s'implementarà una nova manera de visualitzar la imatge aplicant un filtre previ a la visualització.

## 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 implementar la nova manera de visualitzar la imatge haurem de sobreescrivre el mètode **show** de ImageFile:

```
public JDialog show(boolean modal) throws IOException, Exception;
```

Si poseu la variable **modal** a true aconseguireu que s'aturi l'execució mentre està obert el diàleg de visualització. Es a dir, que fins que no tanqueu la finestra on es mostra la imatge no continuarà executant-se el programa. Si poseu **modal** a false no es para l'execució.

Aquí teniu el mètode que haureu de fer servir per filtrar de color a blanc i negre:

```
private BufferedImage color2gray() {  
    BufferedImage inBufferedImage = (BufferedImage) getImage();  
    int width = inBufferedImage.getWidth();  
    int height = inBufferedImage.getHeight();  
    BufferedImage outImage = new BufferedImage( width, height,  
        BufferedImage.TYPE_3BYTE_BGR );  
    for(int i=0; i<height; i++){  
        for(int j=0; j<width; j++){  
            Color c = new Color(inBufferedImage.getRGB(j, i));  
            int red = (int)(c.getRed() * 0.2126);  
            int green = (int)(c.getGreen() * 0.7152);  
            int blue = (int)(c.getBlue() * 0.0722);  
            Color newColor = new  
            Color(red+green+blue,red+green+blue,red+green+blue);  
            outImage.setRGB(j,i,newColor.getRGB());  
        }  
    }  
    return( outImage );  
}
```

I aquí el mètode que haureu de fer servir per filtrar una imatge amb un filtre Sèpia:

## Programació 2. Projecte de Pràctiques

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

```
private BufferedImage color2sepia() {
    BufferedImage inBufferedImage = (BufferedImage) getImage();
    int width = inBufferedImage.getWidth();
    int height = inBufferedImage.getHeight();
    BufferedImage outImage = new BufferedImage( width, height,
    BufferedImage.TYPE_3BYTE_BGR );
    for(int i=0; i<height; i++){
        for(int j=0; j<width; j++){
            Color c = new Color(inBufferedImage.getRGB(j, i));
            int red = (int)(c.getRed());
            int green = (int)(c.getGreen());
            int blue = (int)(c.getBlue());
            int r = (int)((red* 0.393)+(green*0.769)+(blue* 0.189));
            int g = (int)((red* 0.349)+(green*0.686)+(blue* 0.168));
            int b = (int)((red* 0.272)+(green*0.534)+(blue* 0.131));
            if (r>255) r=255;
            if (g>255) g=255;
            if (b>255) b=255;
            Color newColor = new Color(r,g,b);
            outImage.setRGB(j,i,newColor.getRGB());
        }
    }
    return( outImage );
}
```

Reviseu els mètodes de la classe **ImageFile**. Aquesta classe utilitza les classes **Image** i **BufferedImage** implementades en Java. **Image** és una classe abstracta i **BufferedImage** una classe que hereta d'ella i que utilitzarem per filtrar i visualitzar les imatges.

A més, heu de tenir en compte que al mètode **show** de la classe **ImageFile** s'utilitza **getAbsolutePath**. Per tant, s'està llegint el fitxer de disc accedint al camí de l'objecte. Això vol dir que haurem de guardar a disc la imatge filtrada.

## Programació 2. Projecte de Pràctiques

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

El que haureu de fer és filtrar la imatge i guardar-la a disc amb un nou nom. Per exemple, la imatge "paisatge.jpg", una vegada filtrada passarà a ser "paisatge\_BN.jpg" o "paisatge\_Sepia.jpg".

Cada vegada que cridem a visualitzar un objecte de tipus **ImageBN** o **ImatgeSepia** la llegirem de disc com un objecte de tipus **Imatge** i cridarem al mètode **show** d'aquest nou objecte.

Podeu utilitzar el mètode **saveImage** per guardar la imatge filtrada a disc.

```
public static void saveImage(BufferedImage img, File file) throws IOException;
```

A més, recordeu que el mètode **getImage** et retorna l'objecte de tipus **Image** contingut en l'objecte:

```
public Image getImage();
```

### 3.5 Visualitzar imatges d'un àlbum o de la biblioteca

A partir de les indicacions de l'enunciat, definiu els mètodes de visualització de les imatges. Per fer això podeu crear un mètode anomenat **play** en el controlador. Aquest mètode hauria de rebre com a paràmetre d'entrada la llista d'imatges que s'ha de reproduir. Per controlar el temps de reproducció de les imatges utilitzeu el **timer** que hi ha definit al controlador que us proporcionem dins la llibreria **UtilsProg2**.

Una vegada està definida la freqüència de reproducció en milisegons mitjançant el mètode **setTimer**, s'engega la reproducció fent servir **startTimer**. Llavors, el mètode **onTimer** es va cridant automàticament quan passa el període definit amb **setTimer**. Heu de sobreescrivre el mètode **onTimer** per visualitzar una imatge cada cop que es crida, però no heu de fer cap invocació explícita d'aquest mètode.

Per tal d'entendre bé com funciona el mètode **onTimer**, us proposem el següent exercici: sobreescriviu el mètode **onTimer** a la classe **CtrlVisor** i afegiu dins d'aquest mètode aquesta línia:

```
System.out.println("Entra en onTimer \n");
```

Després, en el mètode **play**, que s'ha de cridar des de l'opció de visualitzar la biblioteca (podeu començar sense afegir-hi paràmetres) poseu:

## Programació 2. Projecte de Pràctiques

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

```
playFrequency=1000;  
setTimer(playFrequency);  
startTimer();
```

Executeu el programa i mireu que passa quan escolliu l'opció de visualitzar la biblioteca. Feu servir el javadoc de la llibreria si voleu saber més sobre els mètodes Timer.

Podeu utilitzar el mètode **stopTimer** per aturar la reproducció.

### 3 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.
- La classe **BasciCtrl** 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.

### 4 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\_L3

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.

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

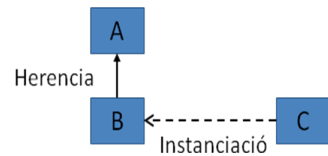


## Programació 2. Projecte de Pràctiques

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

- Introduïu el problema tractat en el lliurament (No s'acceptarà una còpia directa de l'enunciat).
- Expliqueu les classes implementades.
- Expliqueu quines classes heu pogut reutilitzar del lliurament anterior per a fer aquest. Quins canvis sobre les classes reutilitzades heu necessitat fer i perquè.
- Dibuixeu el diagrama de relacions entre les classes que heu utilitzat a la pràctica. Incloure tant les classes implementades per vosaltres com les que pertanyen a la llibreria UtilsProg2. No cal incloure la llista d'atributs i mètodes. Utilitzeu la notació de la següent figura, per indicar que B és una classe que hereta de A i C conté una instància de la classe B, com al següent codi:

```
public class B extends A{  
    //...  
}  
  
public class C{  
    B b =null;  
}
```



- Fes un esquema mostrant el recorregut que fa el teu programa quan s'executa l'opció de visualitzar imatges de la biblioteca. Especifica els mètodes que es criden en cadascuna de les classes. Fes servir números per indicar l'ordre de les crides.
- Proves realitzades per comprovar el correcte funcionament de la pràctica, resultats obtinguts i accions derivades.
- Observacions generals.