Báo cáo Thực hành lập trình hướng đối tượng

Table of Contents

[Use case diagram 2](#_Toc154768969)

[2](#_Toc154768970)

[Update Aims class diagram 2](#_Toc154768971)

[2](#_Toc154768972)

[1. Swing components 4](#_Toc154768973)

[1.1. AWTAccumulator 4](#_Toc154768974)

[1.1.1. Create class AWTAccumulator 4](#_Toc154768975)

[1.1.2. Explanation 6](#_Toc154768976)

[6](#_Toc154768977)

[6](#_Toc154768978)

[6](#_Toc154768979)

[6](#_Toc154768980)

[1.2. SwingAccumulator 6](#_Toc154768981)

[1.2.1. Create class SwingAccumulator 6](#_Toc154768982)

[1.2.2. Explanation 8](#_Toc154768983)

[1.3. Compare Swing and AWT elements 9](#_Toc154768984)

[2. Organizing Swing components with Layout Managers 13](#_Toc154768985)

[2.1. Swing top-level and secondary-level containers 13](#_Toc154768986)

[2.2. Using JPanel as secondary-level container to organize components 13](#_Toc154768987)

[2.2.1. Create class NumberGrid 13](#_Toc154768988)

[13](#_Toc154768989)

[2.2.2. Adding buttons 14](#_Toc154768990)

[14](#_Toc154768991)

[2.2.3. Complete inner class ButtonListener 15](#_Toc154768992)

[15](#_Toc154768993)

[2.2.4.Result 15](#_Toc154768994)

[3. Create a graphical user interface for AIMS with Swing 16](#_Toc154768995)

[3.1. View Store Screen 16](#_Toc154768996)

[3.1.1. Create the StoreScreen class 16](#_Toc154768997)

[16](#_Toc154768998)

[3.1.2. The NORTH component 16](#_Toc154768999)

[16](#_Toc154769000)

[3.1.3. The CENTER component 18](#_Toc154769001)

[3.1.4. The MediaStore class 19](#_Toc154769002)

[19](#_Toc154769003)

[3.1.5. Putting it all together 20](#_Toc154769004)

[20](#_Toc154769005)

[3.2. Adding more user interaction 20](#_Toc154769006)

[20](#_Toc154769007)

[4. JavaFX API 22](#_Toc154769008)

[4.1. Create the FXML file 22](#_Toc154769009)

[4.1.1. Create and open the FXML file 22](#_Toc154769010)

[4.1.2. Building the GUI 22](#_Toc154769011)

[22](#_Toc154769012)

[4.2. Create the controller class 25](#_Toc154769014)

[25](#_Toc154769015)

[4.3. Create the application 26](#_Toc154769016)

[26](#_Toc154769017)

[4.4. Practice exercise 27](#_Toc154769018)

[27](#_Toc154769019)

[5. Setting up the View Cart Screen with ScreenBuilder 27](#_Toc154769020)

[5.1. Setting up the BorderPane 27](#_Toc154769021)

[27](#_Toc154769022)

[5.2. Setting up the TOP area 28](#_Toc154769023)

[28](#_Toc154769024)

[5.3. Setting up the CENTER area 29](#_Toc154769025)

[5.4. Setting up the RIGHT area 30](#_Toc154769026)

[6. Integrating JavaFX into Swing application – The JFXPanel class 31](#_Toc154769027)

[7. View the items in cart – JavaFX’s data-driven UI 32](#_Toc154769028)

[32](#_Toc154769029)

[8. Updating buttons based on selected item in TableView – ChangeListener 33](#_Toc154769030)

[33](#_Toc154769031)

[9. Deleting a media 34](#_Toc154769032)

[34](#_Toc154769033)

[10. Filter items in cart – FilteredList 34](#_Toc154769034)

[34](#_Toc154769035)

[11. Complete the Aims GUI application 35](#_Toc154769036)

[35](#_Toc154769037)

[12. Check all the previous source codes to catch/handle/delegate runtime exceptions 37](#_Toc154769038)

[13. Create a class which inherits from Exception 38](#_Toc154769039)

[38](#_Toc154769040)

[14. Update the Aims class 39](#_Toc154769041)

[39](#_Toc154769042)

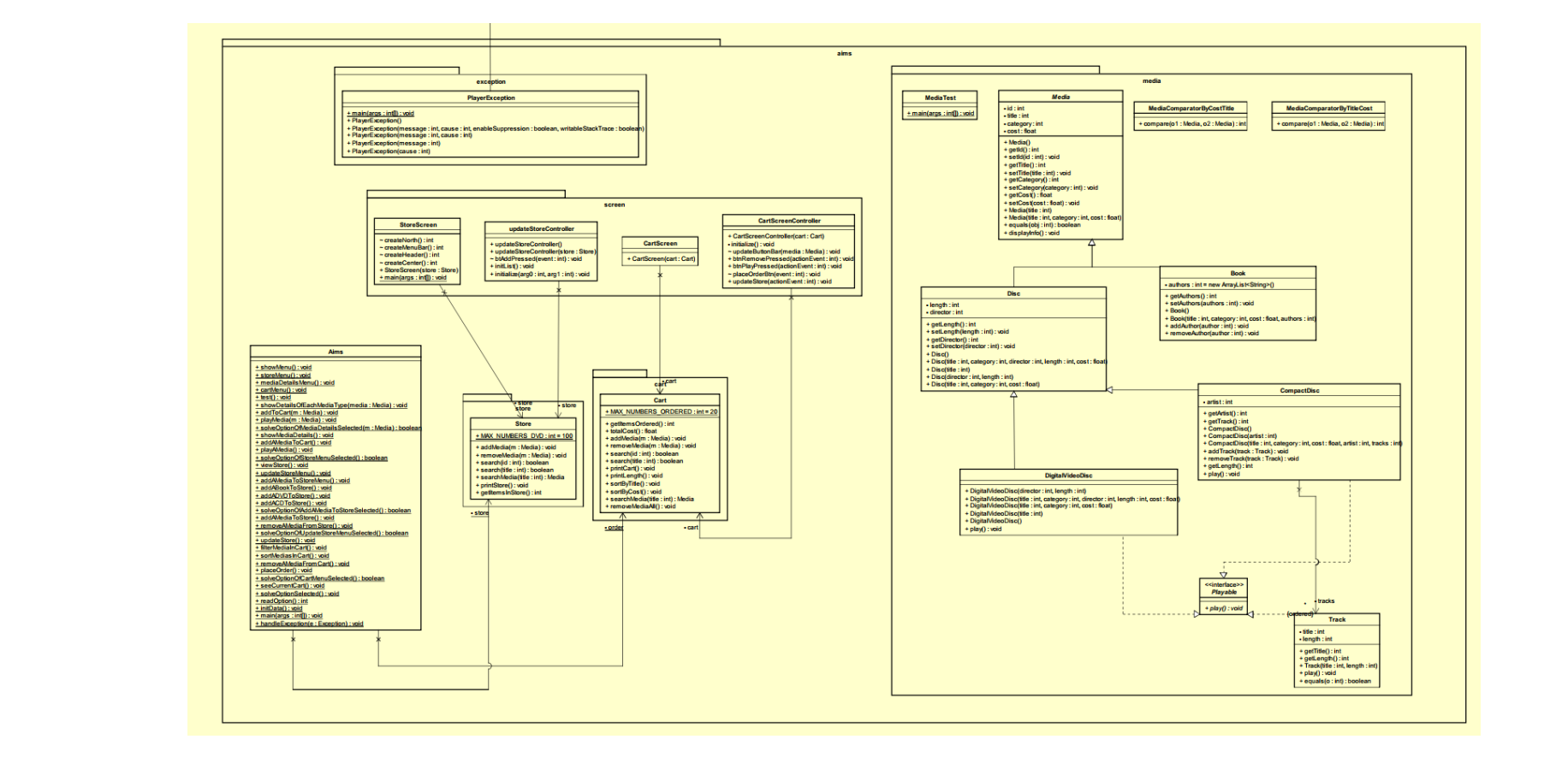
[**Sửa equals trong Media Class** 42](#_Toc154769043)

# Use case diagram

# 

# Update Aims class diagram

# 



# 1. Swing components

## 1.1. AWTAccumulator

### 1.1.1. Create class AWTAccumulator

A screen shot of a computer program

Description automatically generated

A computer screen with colorful text

Description automatically generated

### 1.1.2. Explanation

### A screenshot of a computer Description automatically generated

### A screenshot of a computer Description automatically generated

### A screenshot of a computer Description automatically generated

### A screenshot of a computer Description automatically generated

## 1.2. SwingAccumulator

### 1.2.1. Create class SwingAccumulator

A screen shot of a computer program

Description automatically generated

A computer screen shot of a program code

Description automatically generated

### 1.2.2. Explanation

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

## 1.3. Compare Swing and AWT elements

**The top-level containers in Swing and AWT**

|  |  |  |
| --- | --- | --- |
| Frame (java.awt.Frame): | Dialog (java.awt.Dialog): | Window (java.awt.Window): |
| * Represents a top-level window with a title and border. * Basic structure for creating standalone applications. | * Represents a pop-up window or a dialog box. * Used for getting user input or displaying information. * Can be modal or modeless. | * Base class for `Frame` and `Dialog`. * Represents a top-level window with no borders or title. |

**Swing Top-Level Containers:**

|  |  |  |  |
| --- | --- | --- | --- |
| JFrame (javax.swing.JFrame) | . JDialog (javax.swing.JDialog): | . JWindow (javax.swing.JWindow): | 4. JApplet (javax.swing.JApplet): |
| * Swing equivalent of AWT's `Frame`. * Represents a top-level window with decorations (title, border, etc.). * Supports Swing components and features. | * Swing equivalent of AWT's `Dialog`. * Represents a pop-up window or a dialog box. * Can be modal or modeless. | * Swing equivalent of AWT's `Window`. * Represents a top-level window without decorations. | * Used for creating Swing-based applets. * Extends AWT's `Applet` class and is designed for use in a web browser. |



**The class name of components in AWT and corresponding class’s name in Swing** In AWT and Swing, many components share similar functionalities, but they are represented by different classes. Below is a comparison of some common components in AWT and their counterparts in Swing:

1. Button:

* AWT: `java.awt.Button`
* Swing: `javax.swing.JButton`

2. Label:

* AWT: `java.awt.Label`
* Swing: `javax.swing.JLabel`

3. TextField:

* AWT: `java.awt.TextField`
* Swing: `javax.swing.JTextField`

4. TextArea:

* AWT: `java.awt.TextArea`
* Swing: `javax.swing.JTextArea`

5. Checkbox:

* AWT: `java.awt.Checkbox`
* Swing: `javax.swing.JCheckBox`

6. CheckboxGroup:

* AWT: `java.awt.CheckboxGroup`
* Swing: `javax.swing.ButtonGroup` (used with `JRadioButton`)

7. Choice:

* AWT: `java.awt.Choice`
* Swing: `javax.swing.JComboBox`

8. List:

* AWT: `java.awt.List`
* Swing: `javax.swing.JList`

9. Scrollbar:

* AWT: `java.awt.Scrollbar`
* Swing: `javax.swing.JScrollBar`

1. \*\*Panel:\*\*
   * AWT: `java.awt.Panel`
   * Swing: `javax.swing.JPanel`

1. Frame:
   * AWT: `java.awt.Frame`
   * Swing: `javax.swing.JFrame`

1. Dialog:
   * AWT: `java.awt.Dialog`
   * Swing: `javax.swing.JDialog`

1. Window:
   * AWT: `java.awt.Window`
   * Swing: `javax.swing.JWindow`

1. Applet:
   * AWT: `java.applet.Applet`
   * Swing: `javax.swing.JApplet`

1. Canvas:
   * AWT: `java.awt.Canvas`
   * Swing: `javax.swing.JPanel` (used for custom painting)

1. Label:
   * AWT: `java.awt.Label`
   * Swing: `javax.swing.JLabel`

1. MenuBar:
   * AWT: `java.awt.MenuBar`
   * Swing: `javax.swing.JMenuBar`

1. Menu:
   * AWT: `java.awt.Menu`
   * Swing: `javax.swing.JMenu`

1. MenuItem:
   * AWT: `java.awt.MenuItem`
   * Swing: `javax.swing.JMenuItem`

1. CheckboxMenuItem:
   * AWT: `java.awt.CheckboxMenuItem`

Swing: `javax.swing.JCheckBoxMenuItem`

# 2. Organizing Swing components with Layout Managers

## 2.1. Swing top-level and secondary-level containers

## 2.2. Using JPanel as secondary-level container to organize components

### 2.2.1. Create class NumberGrid

### A computer screen shot of a program Description automatically generated

### 2.2.2. Adding buttons

### A screen shot of a computer program Description automatically generated

### 2.2.3. Complete inner class ButtonListener

### A computer screen shot of a program code Description automatically generated

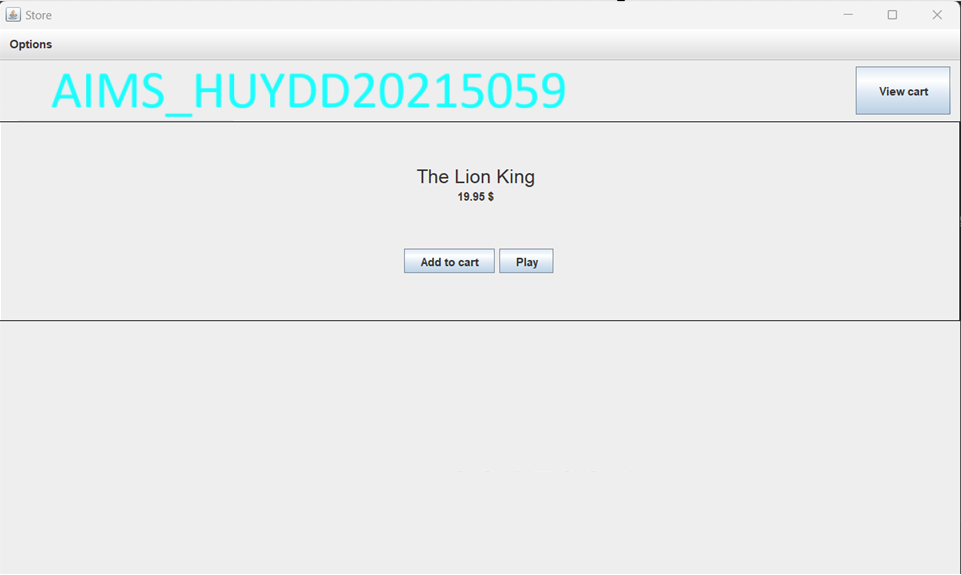
### 2.2.4.Result

A screenshot of a number grid

Description automatically generated

# 3. Create a graphical user interface for AIMS with Swing

## 3.1. View Store Screen



### 3.1.1. Create the StoreScreen class

### A screen shot of a computer screen Description automatically generated

### 3.1.2. The NORTH component

Create the method createNorth(), which will create our NORTH component:

### A screen shot of a computer code Description automatically generated

Create the method createMenuBar():

A screen shot of a computer program

Description automatically generated

Create the method createHeader():

A computer screen shot of a program

Description automatically generated

### 3.1.3. The CENTER component

A computer screen shot of a program code

Description automatically generated

### 3.1.4. The MediaStore class

### A screen shot of a computer Description automatically generated

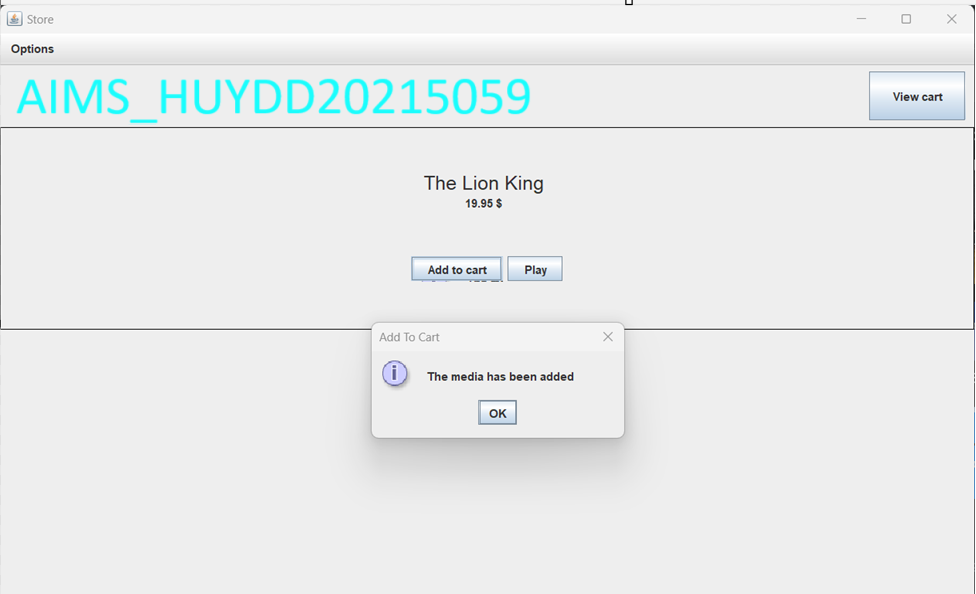
### 3.1.5. Putting it all together

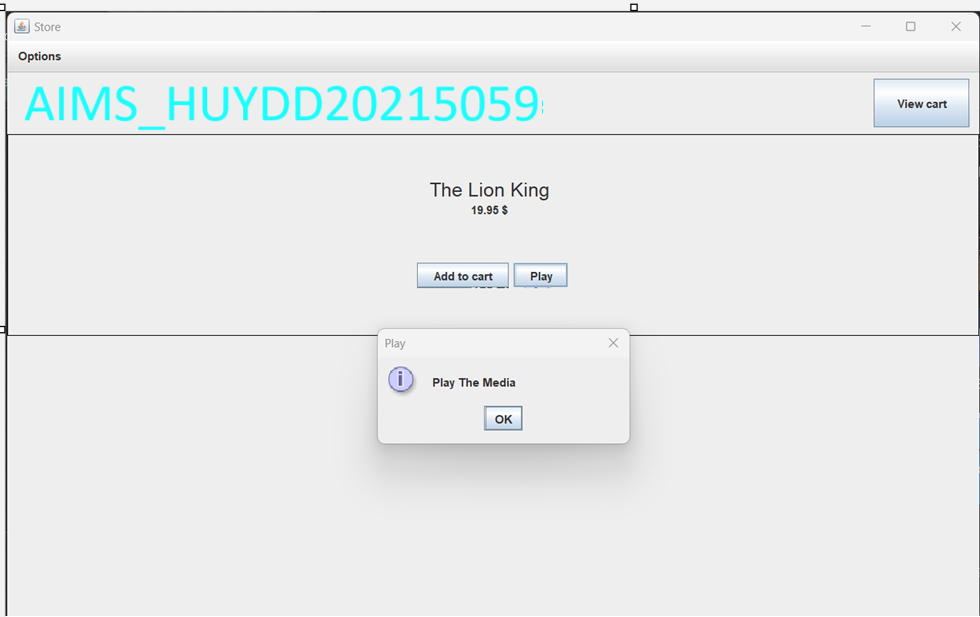
### A screen shot of a computer code Description automatically generated

## 3.2. Adding more user interaction

## 

Result:





# 4. JavaFX API

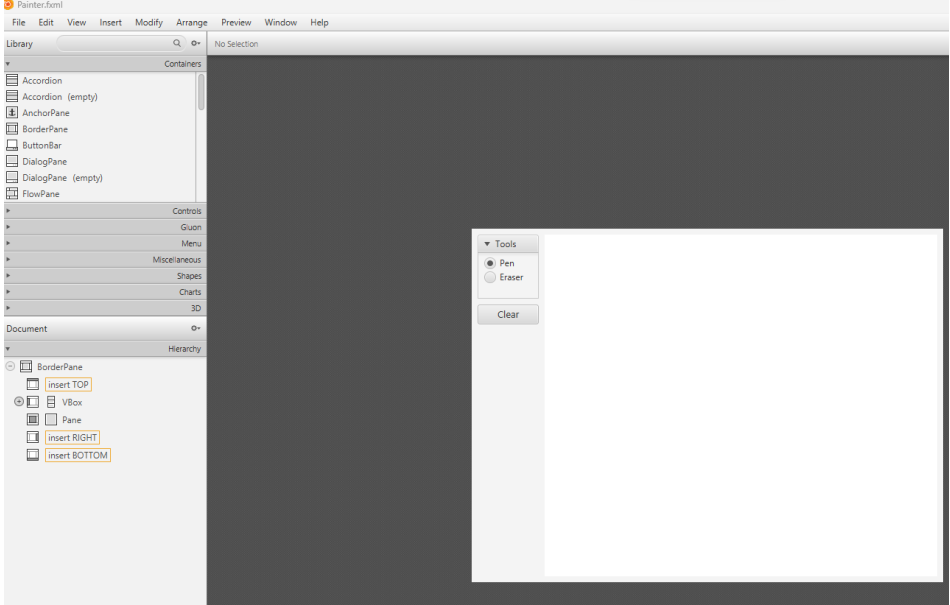
## 4.1. Create the FXML file

### 4.1.1. Create and open the FXML file

### 4.1.2. Building the GUI

### A screenshot of a computer Description automatically generated

### A screenshot of a computer Description automatically generatedA screenshot of a computer Description automatically generated



## 4.2. Create the controller class

## 

A computer screen shot of a program

Description automatically generated

## 4.3. Create the application

## A screen shot of a computer Description automatically generated

## 4.4. Practice exercise

## 

# 5. Setting up the View Cart Screen with ScreenBuilder

## 5.1. Setting up the BorderPane

## A screenshot of a phone Description automatically generated

## 5.2. Setting up the TOP area

## 

## 5.3. Setting up the CENTER area

A screenshot of a computer

Description automatically generatedA screenshot of a computer

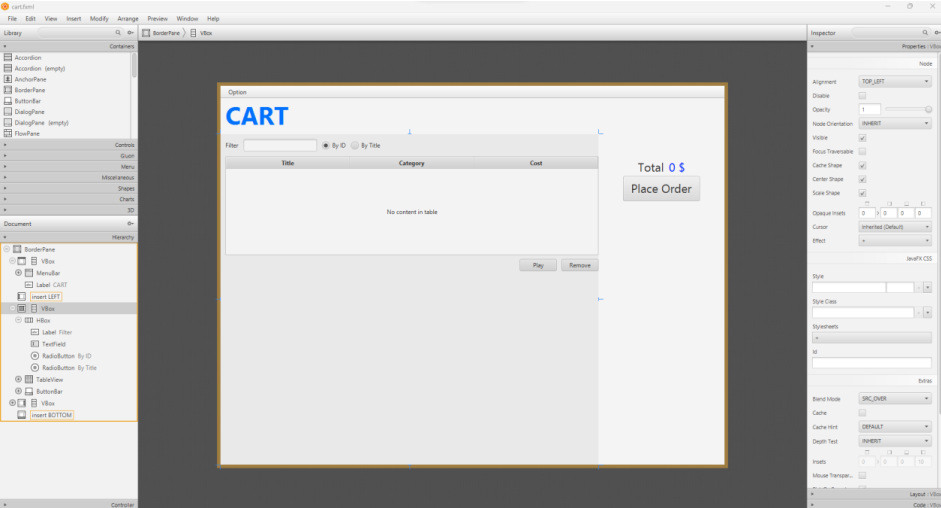
Description automatically generated

## 5.4. Setting up the RIGHT area

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated



# 6. Integrating JavaFX into Swing application – The JFXPanel class

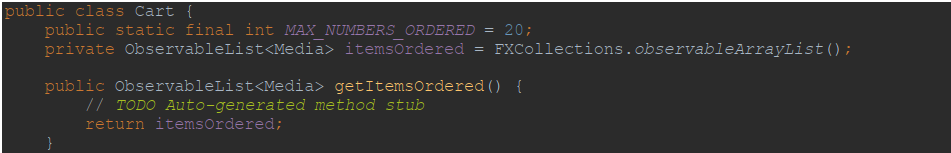
A computer screen shot of a program code

Description automatically generated

# 7. View the items in cart – JavaFX’s data-driven UI

# A screenshot of a computer program Description automatically generated

SỬA ĐỔI CLASS



A screenshot of a computer program

Description automatically generated

# 8. Updating buttons based on selected item in TableView – ChangeListener

# A screen shot of a computer code Description automatically generated

# 9. Deleting a media

# A screen shot of a computer program Description automatically generated

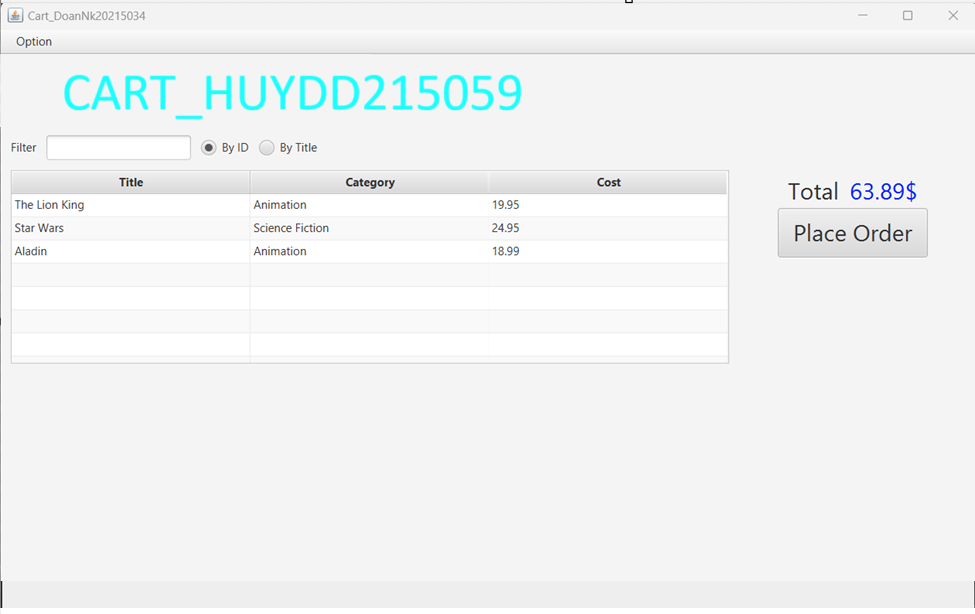
# 10. Filter items in cart – FilteredList

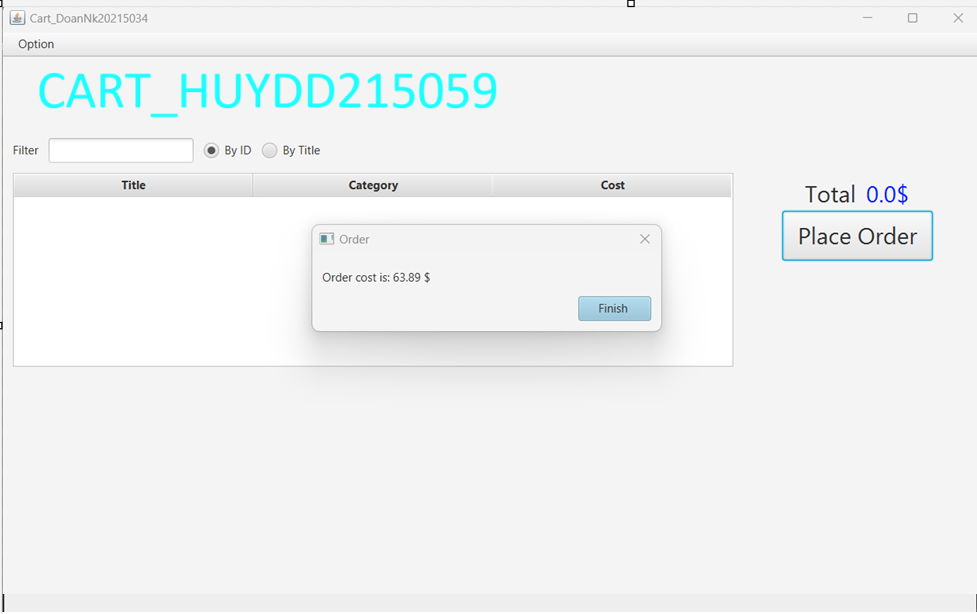
# A computer screen with text on it Description automatically generated

# 11. Complete the Aims GUI application

# A screenshot of a computer program Description automatically generated

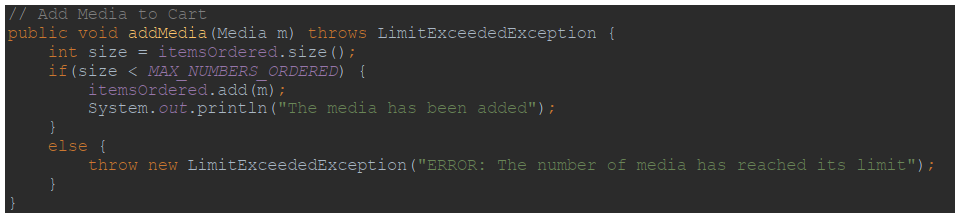
Result:







# 12. Check all the previous source codes to catch/handle/delegate runtime exceptions



# 13. Create a class which inherits from Exception

# A screenshot of a computer program Description automatically generated

A computer screen with white text

Description automatically generated

# 14. Update the Aims class

# A computer screen shot of a program code Description automatically generated

A screen shot of a computer

Description automatically generated

A computer screen with white and blue text

Description automatically generated

A screen shot of a computer program

Description automatically generated

A computer screen shot of a program

Description automatically generated

A screen shot of a computer program

Description automatically generated

15. Update the Aims class

## Sửa equals trong Media Class

