

BÁO CÁO THỰC HÀNH LAB 05 LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG

Mục lục nội dung

1. Swing components	3
1.1. AWTAccumulator	3
1.1.1. Create class <code>AWTAccumulator</code> with the source code as below	3
1.2. SwingAccumulator	5
1.2.1.	5
1.3 Compare Swing and AWT elements	7
2. Organizing Swing components with Layout Managers	9
2.2 Using JPanel as secondary-level container to organize components	9
2.2.1 Create class <code>NumberGrid</code>	9
2.2.2 Adding buttons	10
2.2.3 complete inner class <code>ButtonListener</code> :	11
3. Create a graphical user interface for AIMS with Swing	11
3.3 Demo	16
4. JavaFX API	17
4.1 Create class <code>Painter</code>	17
4.2 Create and open the FXML file in Scene Builder from Eclipse	18
4.3. Tạo controller class	19
4.4 Practice Exercise	21
5. View Cart Screen	21
5.1 Create <code>cart.fxml</code>	21
5.2. Create <code>CartScreen</code>	24
6. Integrating JavaFX into Swing application – The JFXPanel class	28
7. View the items in cart – JavaFX's data-driven UI	31
8. Deleting a media	31
8.1 Code	31
9. Complete	33
10. Update the Aims class diagram	35

Mục lục hình ảnh

Figure 1 AWTAccumulator class	5
Figure 2 Result 1.1.....	5
Figure 3 result 1.1	5
Figure 4 Result SwingAccumulator.....	6
Figure 5 Result SwingAccumulator.....	7
Figure 6 Compare AWT and Swing.....	8
Figure 7 class NumberGrid.....	9
Figure 8 add buttons	10
Figure 9 complete ButtonListener.....	11
Figure 10 Demo.....	16
Figure 11 Add to cart button	16
Figure 12 Demo play button.....	17
Figure 13 View cart button.....	17
Figure 14 Demo View Cart Screen	28
Figure 15 initialize() method	31
Figure 16 Code of btnRemovePressed Method	31
Figure 17 Demo remove.....	32
Figure 18 Demo remove.....	32
Figure 19: Demo store before add book	33
Figure 20: Demo Store after add book.....	34
Figure 21 Demo Store add DVD	34
Figure 22 Demo Store after add DVD.....	35
Figure 23 Catch Exception.....	35
Figure 24 media class diagrams	36
Figure 25 Screen class diagram	37
Figure 26 Store Class diagram	38
Figure 27 Cart class diagram	39

1. Swing components

1.1. AWTAccumulator

1.1.1. Create class `AWTAccumulator` with the source code as below

```
package hust.soict.dsai.swing;

import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
// Pham Duc Dung 20215265
public class AWTAccumulator extends Frame {
    5 usages
    private TextField tfInput;
    4 usages
    private TextField tfOutput;
    2 usages
    private int sum = 0; //Accumulated sum, init to 0

    // Constructor to setup the GUI components and event handlers
    1 usage
    public AWTAccumulator() {

        setLayout(new GridLayout( rows: 2, cols: 2));

        add(new Label( text: "Enter an Integer: "));

        tfInput = new TextField( columns: 10);
        add(tfInput);
        tfInput.addActionListener(new TFInputListener());

        add(new Label( text: "The Accumulated Sum is: "));

        tfOutput = new TextField( columns: 10);
        tfOutput.setEditable(false);
    }
}
```

Figure 1 AWTAccumulator class

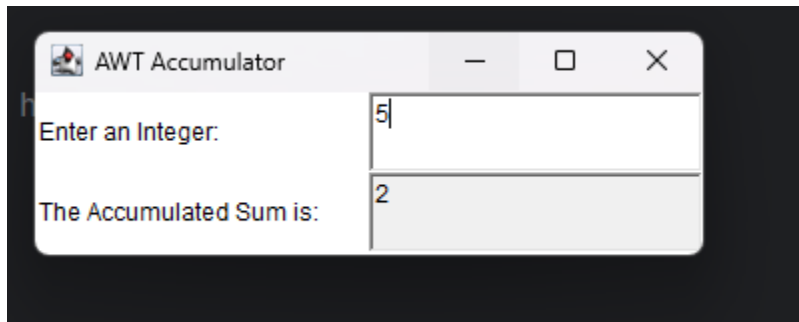


Figure 2 Result 1.1

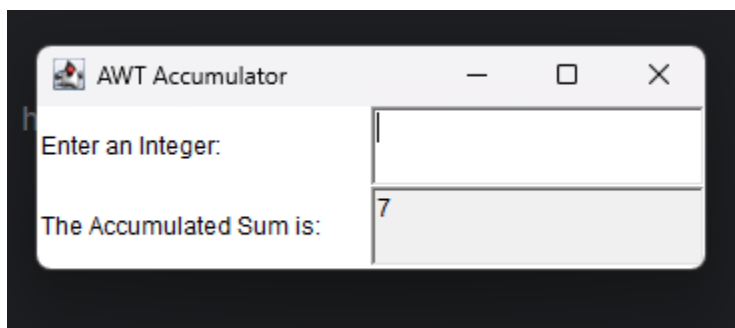


Figure 3 result 1.1

1.2. SwingAccumulator

1.2.1.

```
2. package hust.soict.dsai.swing;

import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

public class SwingAccumulator extends JFrame {
    // Pham Duc Dung 20215265
    private JTextField tfInput;
    private JTextField tfOutput;
    private int sum = 0 ; //Accumulated sum, init to 0

    // Constructor to setup the GUI components and event handlers
    public SwingAccumulator(){
        Container cp = getContentPane();
        setLayout(new GridLayout(2,2));

        add(new Label("Enter an Integer: "));
```

```
tfInput = new JTextField(10);
cp.add(tfInput);
tfInput.addActionListener(new TFInputListener());

add(new Label("The Accumulated Sum is: "));

tfOutput = new JTextField(10);
tfOutput.setEditable(false);
cp.add(tfOutput);

setTitle("Swing Accumulator");
setSize(350,120);
setVisible(true);
}

public static void main(String[] args) {
    new SwingAccumulator();
}

private class TFInputListener implements ActionListener {
    @Override
    public void actionPerformed(ActionEvent event){
        int numberIn = Integer.parseInt(tfInput.getText());
        sum += numberIn;
        tfInput.setText("");
        tfOutput.setText(sum + "");
    }
}
}
```

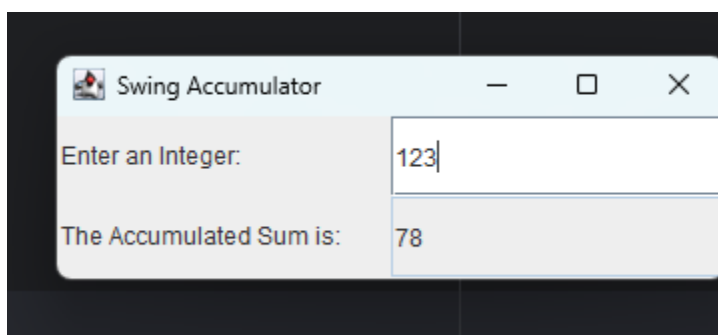


Figure 4 Result SwingAccumulator

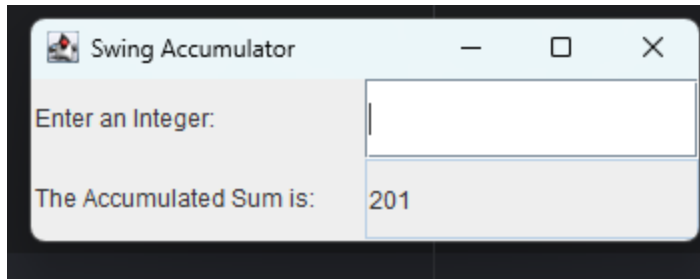


Figure 5 Result SwingAccumulator

1.3 Compare Swing and AWT elements

Top-Level Containers in Swing and AWT

1. AWT Top-Level Containers:

- **Frame:** This is the basic window with a title bar and a border. It's used to house other components.
- **Dialog:** A pop-up window that takes focus away from the main application until dismissed.
- **Applet:** A small application that can be embedded in a web page (now largely obsolete).

2. Swing Top-Level Containers:

- **JFrame:** Equivalent to AWT's Frame but with more features and functionalities. It's part of Swing's lightweight component architecture.
- **JDialog:** Similar to AWT's Dialog but offers more options and is more flexible.
- **JApplet:** The Swing version of Applet, which has now been deprecated and is not commonly used in modern Java applications.
- **JWindow:** A top-level window with no borders or title bar. Swing provides this additional option.

Class Names of Components in AWT and Corresponding Classes in Swing

AWT Component	Swing Component
Button	JButton
Checkbox	JCheckBox
Choice	JComboBox
Label	JLabel
List	JList
MenuBar	JMenuBar
Panel	JPanel
Scrollbar	JScrollBar
TextField	JTextField
TextArea	JTextArea

Figure 6 Compare AWT and Swing

2.Organizing Swing components with Layout Managers

2.2 Using JPanel as secondary-level container to organize components

2.2.1 Create class NumberGrid

```
public class NumberGrid extends JFrame {  
    6 usages  
    private JButton[] btnNumbers = new JButton[10];  
    3 usages  
    private JButton btnDelete;  
    3 usages  
    private JButton btnReset;  
    8 usages  
    private JTextField tfDisplay;  
  
    1 usage  
    public NumberGrid() {  
  
        tfDisplay = new JTextField();  
        tfDisplay.setComponentOrientation(ComponentOrientation.RIGHT_TO_LEFT);  
  
        JPanel panelButtons = new JPanel(new GridLayout( rows: 4, cols: 3));  
        addButtons(panelButtons);  
  
        Container cp = getContentPane();  
        cp.setLayout(new BorderLayout());  
        cp.add(tfDisplay, BorderLayout.NORTH);  
        cp.add(panelButtons, BorderLayout.CENTER);  
  
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
        setTitle("Number Grid");  
        setSize( width: 200, height: 200);  
        setVisible(true);  
    }  
}
```

Figure 7 class NumberGrid

2.2.2 Adding buttons

```
void addButtons(JPanel panelButtons){  
    ButtonListener btnListener = new ButtonListener();  
    for (int i=1 ;i<10; i++){  
        btnNumbers[i] = new JButton( text: ""+i);  
        panelButtons.add(btnNumbers[i]);  
        btnNumbers[i].addActionListener(btnListener);  
    }  
  
    btnDelete = new JButton( text: "DEL");  
    panelButtons.add(btnDelete);  
    btnDelete.addActionListener(btnListener);  
  
    btnNumbers[0] = new JButton( text: "0");  
    panelButtons.add(btnNumbers[0]);  
    btnNumbers[0].addActionListener(btnListener);  
  
    btnReset = new JButton( text: "C");  
    panelButtons.add(btnReset);  
    btnReset.addActionListener(btnListener);  
}
```

Figure 8 add buttons

2.2.3 complete inner class ButtonListener :

```

2 usages
class ButtonListener implements ActionListener {
    public void actionPerformed(ActionEvent e) {
        // Handle button clicks
        String button = e.getActionCommand();

        // Check which button was clicked and perform the necessary action
        if (button.charAt(0) >= '0' && button.charAt(0) <= '9') {
            tfDisplay.setText(tfDisplay.getText() + button);
        } else if (button.equals("DEL")) {
            // Delete the last digit
            String currentText = tfDisplay.getText();
            if (!currentText.isEmpty()) {
                tfDisplay.setText(currentText.substring(0, currentText.length() - 1));
            }
        } else {
            // Clear the entire text
            tfDisplay.setText("");
        }
    }
}

public static void main(String[] args) { SwingUtilities.invokeLater(() -> new NumberGrid()); }
}

```

Figure 9 complete ButtonListener

3. Create a graphical user interface for AIMS with Swing

```

package hust.soict.dsai.aims.screen;
import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.util.List;

import hust.soict.dsai.aims.cart.Cart;
import hust.soict.dsai.aims.store.Store;
import hust.soict.dsai.aims.media.*;

public class StoreScreen extends JFrame {
    // Attribute
    private Store store;
    private Cart cart;
}

```

```
// Constructor
public StoreScreen(Store store, Cart cart) {
    this.store = store;
    this.cart = cart;
    Container cp = getContentPane();
    cp.setLayout(new BorderLayout());

    cp.add(createNorth(), BorderLayout.NORTH);
    cp.add(createCenter(), BorderLayout.CENTER);

    setVisible(true);
    setTitle("Store");
    setSize(1024, 768);
}

// Method to create NORTH component
JPanel createNorth() {
    JPanel north = new JPanel();
    north.setLayout(new BoxLayout(north, BoxLayout.Y_AXIS));
    north.add(createMenuBar());
    north.add(createHeader());
    return north;
}

// Method to create Menu Bar
JMenuBar createMenuBar() {
    JMenu menu = new JMenu("Options");

    JMenu smUpdateStore = new JMenu("Update Store");
    MenuListener menuListener = new MenuListener(store, cart);
    JMenuItem addBook = new JMenuItem("Add Book");
    addBook.addActionListener(menuListener);
    smUpdateStore.add(addBook);
    JMenuItem addCD = new JMenuItem("Add CD");
    addCD.addActionListener(menuListener);
    smUpdateStore.add(addCD);
    JMenuItem addDVD = new JMenuItem("Add DVD");
    addDVD.addActionListener(menuListener);
    smUpdateStore.add(addDVD);

    menu.add(smUpdateStore);
    JMenuItem viewStore = new JMenuItem("View store");
    viewStore.addActionListener(menuListener);
    menu.add(viewStore);
    JMenuItem viewCart = new JMenuItem("View cart");
    viewCart.addActionListener(menuListener);
}
```

```
        menu.add(viewCart);

        JMenuBar menuBar = new JMenuBar();
        menuBar.setLayout(new FlowLayout(FlowLayout.LEFT));
        menuBar.add(menu);

        return menuBar;
    }

    // Method to create Header
    JPanel createHeader() {
        JPanel header = new JPanel();
        header.setLayout(new BoxLayout(header, BoxLayout.X_AXIS));

        JLabel title = new JLabel("AIMS");
        title.setFont(new Font(title.getFont().getName(), Font.PLAIN,
50));
        title.setForeground(Color.CYAN);

        ButtonListener buttonListener = new ButtonListener(cart);
        JButton cart = new JButton("View cart");
        cart.setPreferredSize(new Dimension(100, 50));
        cart.setMaximumSize(new Dimension(100, 50));
        cart.addActionListener(buttonListener);

        header.add(Box.createRigidArea(new Dimension(10, 10)));
        header.add(title);
        header.add(Box.createHorizontalGlue());
        header.add(cart);
        header.add(Box.createRigidArea(new Dimension(10, 10)));

        return header;
    }

    // Method to create Center
    JPanel createCenter() {
        JPanel center = new JPanel();
        center.setLayout(new GridLayout(3, 3, 2, 2));

        List<Media> mediaInStore = store.getItemsInStore();
        MediaStore.setCart(cart);
        for (int i = 0; i < 9; i++) {
            if (i >= mediaInStore.size()) {
                break;
            }
            MediaStore cell = new MediaStore(mediaInStore.get(i));
            center.add(cell);
        }
    }
}
```

```
    }

    return center;
}

public static void main(String[] args) throws Exception {
    Store store = new Store();
    Cart cart = new Cart();
    // Create new media and add them to the store
    // Adding DVDs
    Media dvd1 = new DigitalVideoDisc(1, "Inception",
        "Science Fiction", "Christopher Nolan", 148, 19.99f);
    store.addMedia(dvd1);

    // Adding CDs
    Media cd1 = new CompactDisc(3, "Random Access Memories",
        "Electronic", 15.99f, "Daft Punk", 13, "Daft Punk");
    store.addMedia(cd1);

    // Adding Books
    Media book1 = new Book(5, "The Silent Patient",
        "Thriller", 14.95f);
    store.addMedia(book1);
    new StoreScreen(store, cart);
}

// Button Listener for View Cart
private class ButtonListener implements ActionListener {
    // Attribute
    private Cart cart;

    // Constructor
    public ButtonListener(Cart cart) {
        super();
        this.cart = cart;
    }

    @Override
    public void actionPerformed(ActionEvent e) {
        String button = e.getActionCommand();
        if (button.equals("View cart")) {
            new CartScreen(cart);
            setVisible(false);
            dispose();
        }
    }
}
```

```
// Menu Listener for Menu Option
private class MenuListener implements ActionListener {
    // Attribute
    private Store store;
    private Cart cart;

    // Constructor
    public MenuListener(Store store, Cart cart) {
        super();
        this.store = store;
        this.cart = cart;
    }

    @Override
    public void actionPerformed(ActionEvent e) {
        String menu = e.getActionCommand();
        if (menu.equals("Add Book")) {
            new AddBookToStore(store, cart);
            setVisible(false);
            dispose();
        } else if (menu.equals("Add CD")) {
            new AddCDToStore(store, cart);
            setVisible(false);
            dispose();
        } else if (menu.equals("Add DVD")) {
            new AddDVDToStoreScreen(store, cart);
            setVisible(false);
            dispose();
        } else if (menu.equals("View store")) {
            new StoreScreen(store, cart);
            setVisible(false);
            dispose();
        } else if (menu.equals("View cart")) {
            new CartScreen(cart);
            setVisible(false);
            dispose();
        }
    }
}
```

3.3 Demo

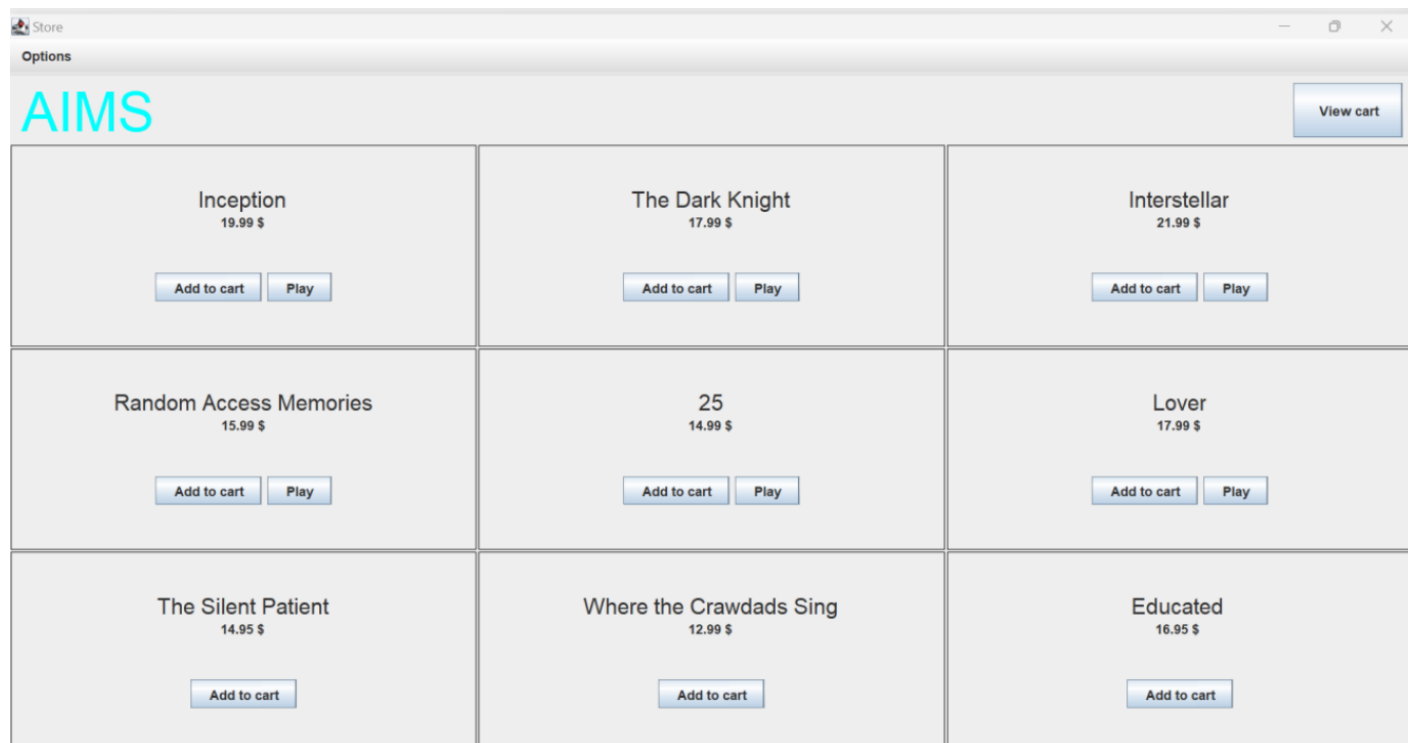


Figure 10 Demo

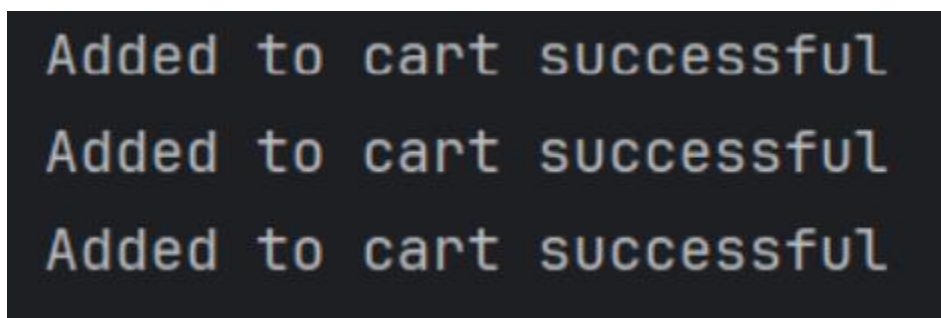


Figure 11 Add to cart button


```

Playing DVD: Inception
DVD length: 148
Playing CD: 25
CD artist: Adele
CD length: 0
Playing DVD: Interstellar
DVD length: 169

```

Figure 12 Demo play button

```

*****CART*****
Ordered Items:
1. DVD - Inception - Science Fiction - Christopher Nolan - 148: 19.99$
2. DVD - The Dark Knight - Action - Christopher Nolan - 152: 17.99$
7. DVD - Interstellar - Science Fiction - Christopher Nolan - 169: 21.99$
Total cost: 59.97
*****

```

Figure 13 View cart button

4. JavaFX API

Tạo package `hust.soict.dsai.javaafx` (for DS & AI). **You might need to add the JavaFX library to this project if you are using JDK version after 1.8.**

4.1 Create class Painter

```

package hust.soict.dsai.javaafx;

import javafx.application.Application;
import javafx.fxml.FXMLLoader;
import javafx.stage.Stage;
import javafx.scene.Scene;
import javafx.scene.Parent;

```

```

public class Painter extends Application {

    @Override
    public void start(Stage stage) throws Exception{
        try {
            Parent root = FXMLLoader.load(getClass().getResource("/hust/soict/dsai/javafx/Painter.fxml"));

            stage.setTitle("Painter");
            stage.setScene(new Scene(root));
            stage.show();
        }
        catch (Exception e) {
            e.printStackTrace();
        }
    }
    public static void main(String[] args) {
        launch(args);
    }
}

```

4.2 Create and open the FXML file in Scene Builder from Eclipse

```
<?xml version="1.0" encoding="UTF-8"?>
```

```

<?import javafx.geometry.Insets?>
<?import javafx.scene.control.Button?>
<?import javafx.scene.control.RadioButton?>
<?import javafx.scene.control.TitledPane?>
<?import javafx.scene.control.ToggleGroup?>
<?import javafx.scene.layout.AnchorPane?>
<?import javafx.scene.layout.BorderPane?>
<?import javafx.scene.layout.Pane?>
<?import javafx.scene.layout.VBox?>

```

```

<BorderPane minHeight="-Infinity" minWidth="-Infinity" prefHeight="480.0" prefWidth="640.0"
xmlns:fx="http://javafx.com/fxml/1" xmlns="http://javafx.com/javafx/21.0.0"
fx:controller="hust.soict.dsai.javafx.PainterController">
    <padding>
        <Insets bottom="8.0" left="8.0" right="8.0" top="8.0" />
    </padding>
    <left>
        <VBox maxHeight="1.7976931348623157E308" prefHeight="464.0" prefWidth="87.0" spacing="8.0"

```

```

BorderPane.alignment="CENTER">
    <BorderPane.margin>
        <Insets right="8.0" />
    </BorderPane.margin>
    <children>
        <TitledPane animated="false" prefHeight="78.0" prefWidth="200.0" text="Tool">
            <content>
                <AnchorPane maxHeight="-Infinity" maxWidth="-Infinity" minHeight="0.0" minWidth="0.0"
prefHeight="79.0" prefWidth="86.0">
                    <children>
                        <RadioButton layoutX="7.0" layoutY="7.0" mnemonicParsing="false" onAction="#toolButtonPressed"
text="Pen" AnchorPane.bottomAnchor="20.0" AnchorPane.leftAnchor="-7.0" AnchorPane.rightAnchor="30.0"
AnchorPane.topAnchor="-7.0">
                            <toggleGroup>
                                <ToggleGroup fx:id="tools" />
                            </toggleGroup>
                        </RadioButton>
                        <RadioButton layoutY="37.0" mnemonicParsing="false" onAction="#toolButtonPressed"
prefHeight="15.0" prefWidth="88.0" text="Eraser" toggleGroup="$tools" AnchorPane.bottomAnchor="-10.0"
AnchorPane.leftAnchor="-7.0" AnchorPane.rightAnchor="5.0" AnchorPane.topAnchor="15.0" />
                    </children>
                </AnchorPane>
            </content>
            <VBox.margin>
                <Insets />
            </VBox.margin>
        </TitledPane>
        <Button mnemonicParsing="false" onAction="#clearButtonPressed" prefHeight="26.0" prefWidth="67.0"
text="Clear" />
    </children>
</VBox>
</left>
<center>
    <Pane fx:id="drawingAreaPane" onMouseDragged="#drawingAreaMouseDragged" prefHeight="200.0"
prefWidth="200.0" style="-fx-background-color: white;" BorderPane.alignment="CENTER" />
</center>
</BorderPane>

```

4.3. Tạo controller class

```

package hust.soict.dsai.javafx;

import javafx.event.ActionEvent;

```

```
import javafx.fxml.FXML;
import javafx.scene.layout.Pane;
import javafx.scene.input.MouseEvent;
import javafx.scene.shape.Circle;
import javafx.scene.paint.Color;
import javafx.scene.control.RadioButton;

public class PainterController {

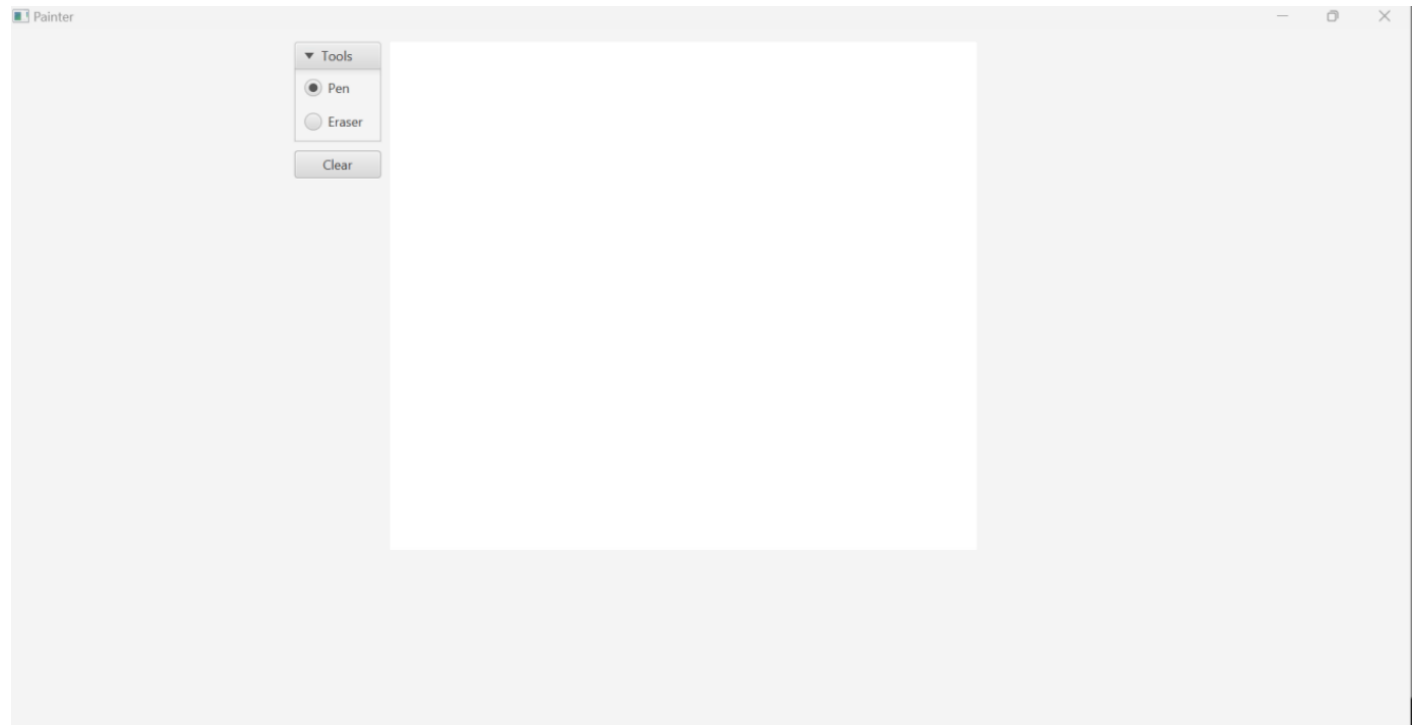
    @FXML
    private Pane drawingAreaPane;
    private String tool = "";

    @FXML
    void clearButtonPressed(ActionEvent e) {
        drawingAreaPane.getChildren().clear();
    }

    @FXML
    void drawingAreaMouseDragged(MouseEvent e) {
        Circle newCircle = new Circle();
        if(tool.equals("Pen")) {
            newCircle = new Circle(e.getX(),
                                   e.getY(), 4, Color.BLACK);
        }
        else if (tool.equals("Eraser")) {
            newCircle = new Circle(e.getX(),
                                   e.getY(), 4, Color.WHITE);
        }
        if (e.getX() >= 0 && e.getX() <= drawingAreaPane.getWidth()
            && e.getY() >= 0 && e.getY() <= drawingAreaPane.getHeight()) {
            drawingAreaPane.getChildren().add(newCircle);
        }
    }

    @FXML
    void toolButtonPressed(ActionEvent e) {
        tool = ((RadioButton) e.getSource()).getText();
    }
}
```

4.4 Praticce Exercise



5. View Cart Screen

5.1 Create cart.fxml

```
<?xml version="1.0" encoding="UTF-8"?>

<?import javafx.geometry.Insets?>
<?import javafx.scene.control.Button?>
<?import javafx.scene.control.ButtonBar?>
<?import javafx.scene.control.Label?>
<?import javafx.scene.control.Menu?>
<?import javafx.scene.control.MenuBar?>
<?import javafx.scene.control.MenuItem?>
<?import javafx.scene.control.RadioButton?>
<?import javafx.scene.control.TableColumn?>
<?import javafx.scene.control.TableView?>
<?import javafx.scene.control.TextField?>
<?import javafx.scene.control.ToggleGroup?>
<?import javafx.scene.layout.AnchorPane?>
<?import javafx.scene.layout.BorderPane?>
<?import javafx.scene.layout.HBox?>
<?import javafx.scene.layout.VBox?>
<?import javafx.scene.text.Font?>
```

```

<AnchorPane prefHeight="400.0" prefWidth="600.0"
xmlns="http://javafx.com/javafx/21" xmlns:fx="http://javafx.com/fxml/1">
    <children>
        <BorderPane layoutX="79.0" layoutY="14.0" prefHeight="768.0"
prefWidth="1024.0">
            <top>
                <VBox prefWidth="100.0" BorderPane.alignment="CENTER">
                    <children>
                        <MenuBar>
                            <menus>
                                <Menu mnemonicParsing="false" text="Options">
                                    <items>
                                        <Menu mnemonicParsing="false" text="Update
Store">
                                            <items>
                                                <MenuItem mnemonicParsing="false"
text="Add Book" />
                                                <MenuItem mnemonicParsing="false"
text="Add CD" />
                                                <MenuItem mnemonicParsing="false"
text="Add DVD" />
                                            </items>
                                        </Menu>
                                        <MenuItem mnemonicParsing="false"
text="View Store" />
                                        <MenuItem mnemonicParsing="false"
text="View Cart" />
                                    </items>
                                </Menu>
                            </menus>
                        </MenuBar>
                        <Label text="CART" textFill="AQUA">
                            <font>
                                <Font size="50.0" />
                            </font>
                            <VBox.margin>
                                <Insets />
                            </VBox.margin>
                            <padding>
                                <Insets left="10.0" />
                            </padding>
                        </Label>
                    </children>
                </VBox>
            </top>
            <center>

```

```

        <VBox prefHeight="200.0" prefWidth="100.0"
BorderPane.alignment="CENTER">
    <padding>
        <Insets left="10.0" />
    </padding>
    <children>
        <HBox alignment="CENTER_LEFT" prefWidth="200.0"
spacing="10.0">
            <padding>
                <Insets bottom="10.0" top="10.0" />
            </padding>
            <children>
                <Label text="Filter" />
                <TextField fx:id="tfFilter" />
                <RadioButton fx:id="radioBtnFilterId"
mnemonicParsing="false" selected="true" text="By ID">
                    <toggleGroup>
                        <ToggleGroup fx:id="filterCategory" />
                    </toggleGroup>
                </RadioButton>
                <RadioButton fx:id="radioBtnFilterTitle"
mnemonicParsing="false" text="By Title" toggleGroup="$filterCategory" />
            </children>
        </HBox>
        <TableView fx:id="tblMedia">
            <columns>
                <TableColumn fx:id="colMediaTitle"
prefWidth="75.0" text="Title" />
                <TableColumn fx:id="colMediaCategory"
prefWidth="75.0" text="Category" />
                <TableColumn fx:id="colMediaCost"
prefWidth="75.0" text="Cost" />
            </columns>
            <columnResizePolicy>
                <TableView
fx:constant="CONSTRAINED_RESIZE_POLICY" />
            </columnResizePolicy>
        </TableView>
        <ButtonBar prefHeight="40.0" prefWidth="200.0">
            <buttons>
                <Button fx:id="btnPlay" mnemonicParsing="false"
onAction="#btnPlayPressed" text="Play" />
                <Button fx:id="btnRemove"
mnemonicParsing="false" onAction="#btnRemovePressed" text="Remove" />
            </buttons>
        </ButtonBar>
    </children>

```

```

        </VBox>
    </center>
    <right>
        <VBox alignment="TOP_CENTER" prefHeight="200.0"
BorderPane.alignment="CENTER">
            <padding>
                <Insets top="50.0" />
            </padding>
            <children>
                <HBox alignment="CENTER">
                    <children>
                        <Label lineSpacing="10.0" text="Total:">
                            <font>
                                <Font size="24.0" />
                            </font>
                        </Label>
                        <Label fx:id="lbTotalCost" text="0 $"
textFill="AQUA">
                            <font>
                                <Font size="24.0" />
                            </font>
                        </Label>
                    </children>
                </HBox>
                <Button mnemonicParsing="false"
onAction="#btnPlaceOrderPressed" style="-fx-background-color: red;"
text="Place Order" textFill="WHITE">
                    <font>
                        <Font size="24.0" />
                    </font>
                </Button>
            </children>
        </VBox>
    </right>
</BorderPane>
</children>
</AnchorPane>

```

5.2. Create CartScreen

```
package hust.soict.dsai.aims.screen;
```

```
import javafx.application.Platform;
import javafx.embed.swing.JFXPanel;
import javafx.fxml.FXMLLoader;
import javafx.scene.Parent;
```



```

import javafx.scene.Scene;
import hust.soict.dsai.aims.cart.Cart;

import javax.swing.*;
import java.io.IOException;
import java.net.URL;

public class CartScreen extends JFrame {
    private Cart cart;

    public CartScreen(Cart cart) {
        super();

        this.cart = cart;

        JFXPanel fxPanel = new JFXPanel();
        this.add(fxPanel);

        this.setTitle("Cart");
        this.setVisible(true);
        Platform.runLater(new Runnable() {
            @Override
            public void run() {
                try {
                    FXMLLoader loader = new FXMLLoader(getClass().getResource("cart.fxml"));
                    CartScreenController controller =
                        new CartScreenController(cart);
                    loader.setController(controller);
                    Parent root = loader.load();
                    fxPanel.setScene(new Scene(root));
                } catch (IOException e) {
                    e.printStackTrace();
                }
            }
        });
    }
}

```

5.3. Class CartScreenController

```

package hust.soict.dsai.aims.screen;
import javafx.beans.value.ChangeListener;
import javafx.beans.value.ObservableValue;
import javafx.event.ActionEvent;
import javafx.fxml.FXML;

```

```
import javafx.fxml.FXMLLoader;
import javafx.scene.control.*;
import hust.soict.dsai.aims.cart.Cart;
import hust.soict.dsai.aims.media.Media;
import javafx.scene.control.cell.PropertyValueFactory;
import hust.soict.dsai.aims.media.Playable;

public class CartScreenController {
    private Cart cart;

    @FXML
    private TableView<Media> tblMedia;

    @FXML
    private TableColumn<Media, String> colMediaTitle;

    @FXML
    private TableColumn<Media, String> colMediaCategory;

    @FXML
    private TableColumn<Media, Float> colMediaCost;

    @FXML
    private Button btnPlay;

    @FXML
    private Button btnRemove;

    @FXML
    private Label lbTotalCost;

    public CartScreenController(Cart cart) {
        super();
        this.cart = cart;
        // loader.load();
    }

    @FXML
    private void initialize() {
        colMediaTitle.setCellValueFactory(
            new PropertyValueFactory<Media, String>("title"));
        colMediaCategory.setCellValueFactory(
            new PropertyValueFactory<Media, String>("category"));
        colMediaCost.setCellValueFactory(
```

```

        new PropertyValueFactory<Media, Float>("cost"));
tblMedia.setItems(this.cart.getItemsOrdered());
lbTotalCost.setText(Double.toString(cart.totalCost()) + " $");

btnPlay.setVisible(false);
btnRemove.setVisible(false);

tblMedia.getSelectionModel().selectedItemProperty().addListener(
    new ChangeListener<Media>() {
        @Override
        public void changed(ObservableValue<? extends Media> observable, Media oldValue,
            Media newValue) {
            if (newValue != null) {
                updateButtonBar(newValue);
            }
        }
    });
}

void updateButtonBar(Media media){
    btnRemove.setVisible(true);
    btnPlay.setVisible(media instanceof Playable);
}

@FXML
void btnRemovePressed (ActionEvent event) throws Exception {
    Media media = tblMedia.getSelectionModel().getSelectedItem();
    cart.removeMedia(media);
    lbTotalCost.setText(Double.toString(cart.totalCost()) + " $");
}

@FXML
void btnPlaceOrderPressed(ActionEvent event) {

}

@FXML
void btnPlayPressed(ActionEvent event) {

}
}

```

5.4. Demo

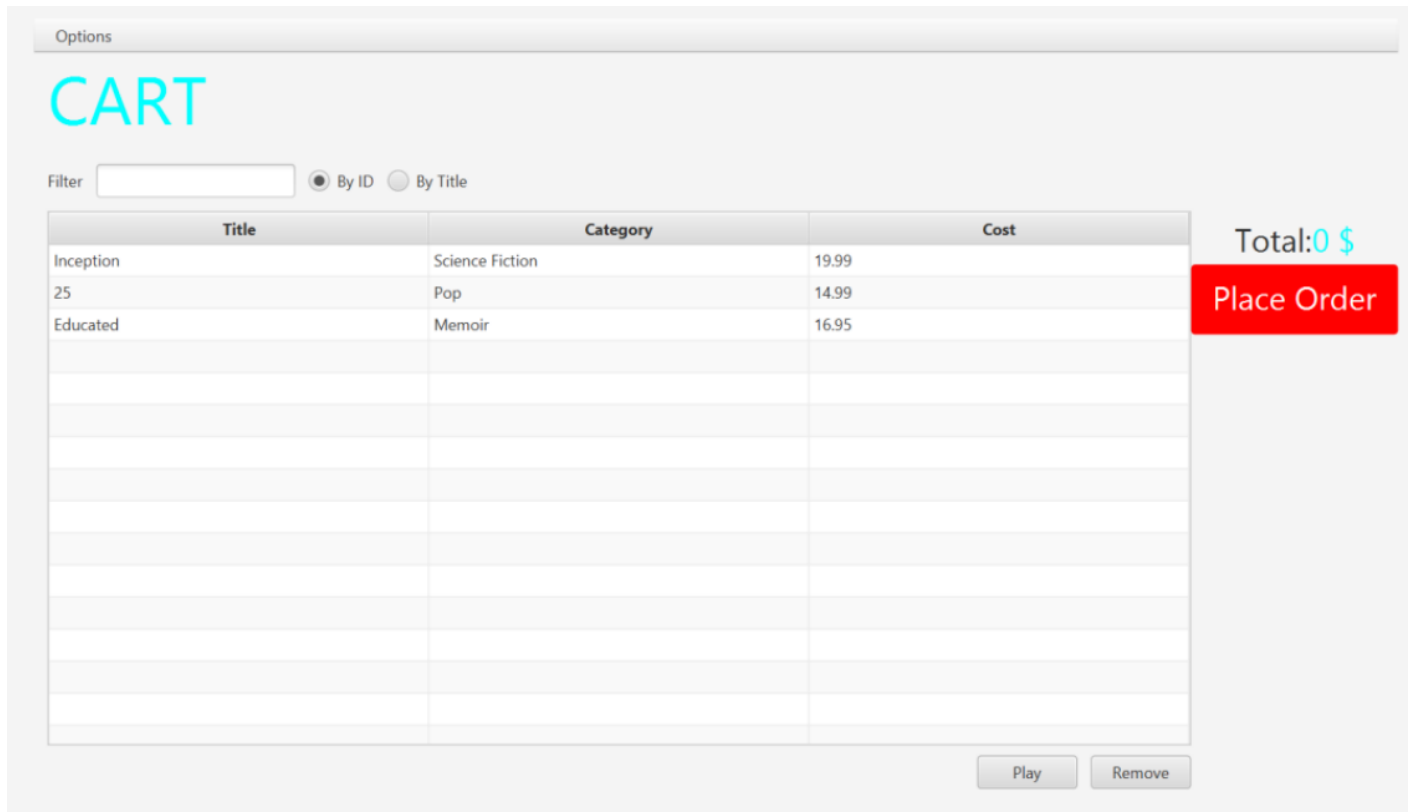


Figure 14 Demo View Cart Screen

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

6.1 Edit class CartScreenController

```
package hust.soict.dsai.aims.screen;
import javafx.beans.value.ChangeListener;
import javafx.beans.value.ObservableValue;
import javafx.event.ActionEvent;
import javafx.fxml.FXML;
import javafx.fxml.FXMLLoader;
import javafx.scene.control.*;
import hust.soict.dsai.aims.cart.Cart;
import hust.soict.dsai.aims.media.Media;
import javafx.scene.control.cell.PropertyValueFactory;
import hust.soict.dsai.aims.media.Playable;

public class CartScreenController {
    private Cart cart;
```

```
@FXML
private TableView<Media> tblMedia;

@FXML
private TableColumn<Media, String> colMediaTitle;

@FXML
private TableColumn<Media, String> colMediaCategory;

@FXML
private TableColumn<Media, Float> colMediaCost;

@FXML
private Button btnPlay;

@FXML
private Button btnRemove;

@FXML
private Label lbTotalCost;

public CartScreenController(Cart cart) {
    super();
    this.cart = cart;
    // loader.load();
}

@FXML
private void initialize() {
    colMediaTitle.setCellValueFactory(
        new PropertyValueFactory<Media, String>("title"));
    colMediaCategory.setCellValueFactory(
        new PropertyValueFactory<Media, String>("category"));
    colMediaCost.setCellValueFactory(
        new PropertyValueFactory<Media, Float>("cost"));
    tblMedia.setItems(this.cart.getItemsOrdered());
    lbTotalCost.setText(Double.toString(cart.totalCost()) + " $");

    btnPlay.setVisible(false);
    btnRemove.setVisible(false);

    tblMedia.getSelectionModel().selectedItemProperty().addListener(
        new ChangeListener<Media>() {
```

```
@Override
public void changed(ObservableValue<? extends Media> observable, Media oldValue,
    Media newValue) {
    if (newValue != null) {
        updateButtonBar(newValue);
    }
}

});

}

void updateButtonBar(Media media){
    btnRemove.setVisible(true);
    btnPlay.setVisible(media instanceof Playable);
}

@FXML
void btnRemovePressed (ActionEvent event) throws Exception {
    Media media = tblMedia.getSelectionModel().getSelectedItem();
    cart.removeMedia(media);
    lbTotalCost.setText(Double.toString(cart.totalCost()) + " $");
}

@FXML
void btnPlaceOrderPressed(ActionEvent event) {

}

@FXML
void btnPlayPressed(ActionEvent event) {

}
}
```

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

```
public void initialize() {  
    colMediaTitle.setCellValueFactory(  
        new PropertyValueFactory<Media, String>(s: "title"));  
    colMediaCategory.setCellValueFactory(  
        new PropertyValueFactory<Media, String>(s: "category"));  
    colMediaCost.setCellValueFactory(  
        new PropertyValueFactory<Media, Float>(s: "cost"));  
  
    btnPlay.setVisible(false);  
    btnRemove.setVisible(false);  
}
```

Figure 15 initialize() method

8. Deleting a media

8.1 Code

```
@FXML  
void btnRemovePressed (ActionEvent event) {  
    Media media = tblMedia.getSelectionModel().getSelectedItem();  
    cart.removeMedia(media);  
}
```

Figure 16 Code of btnRemovePressed Method

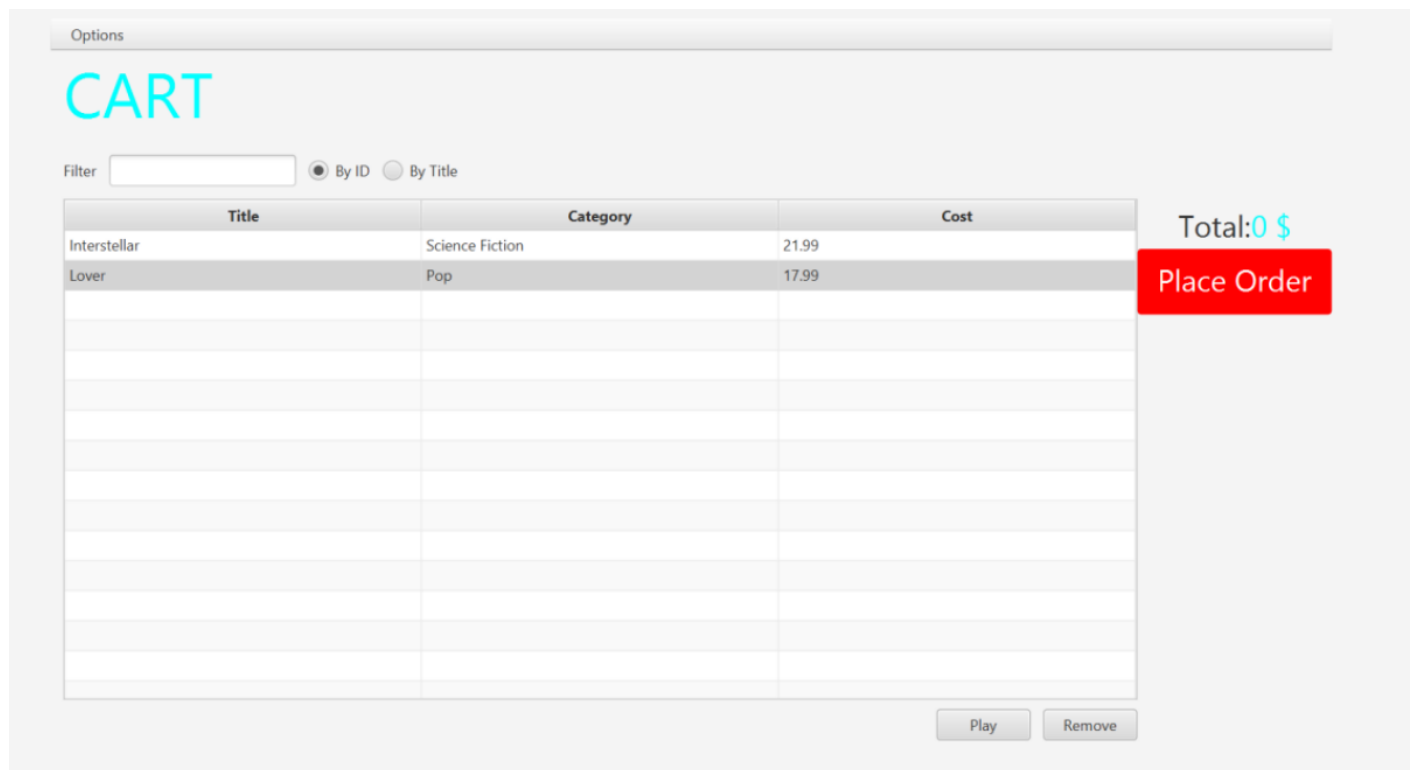


Figure 17 Demo remove

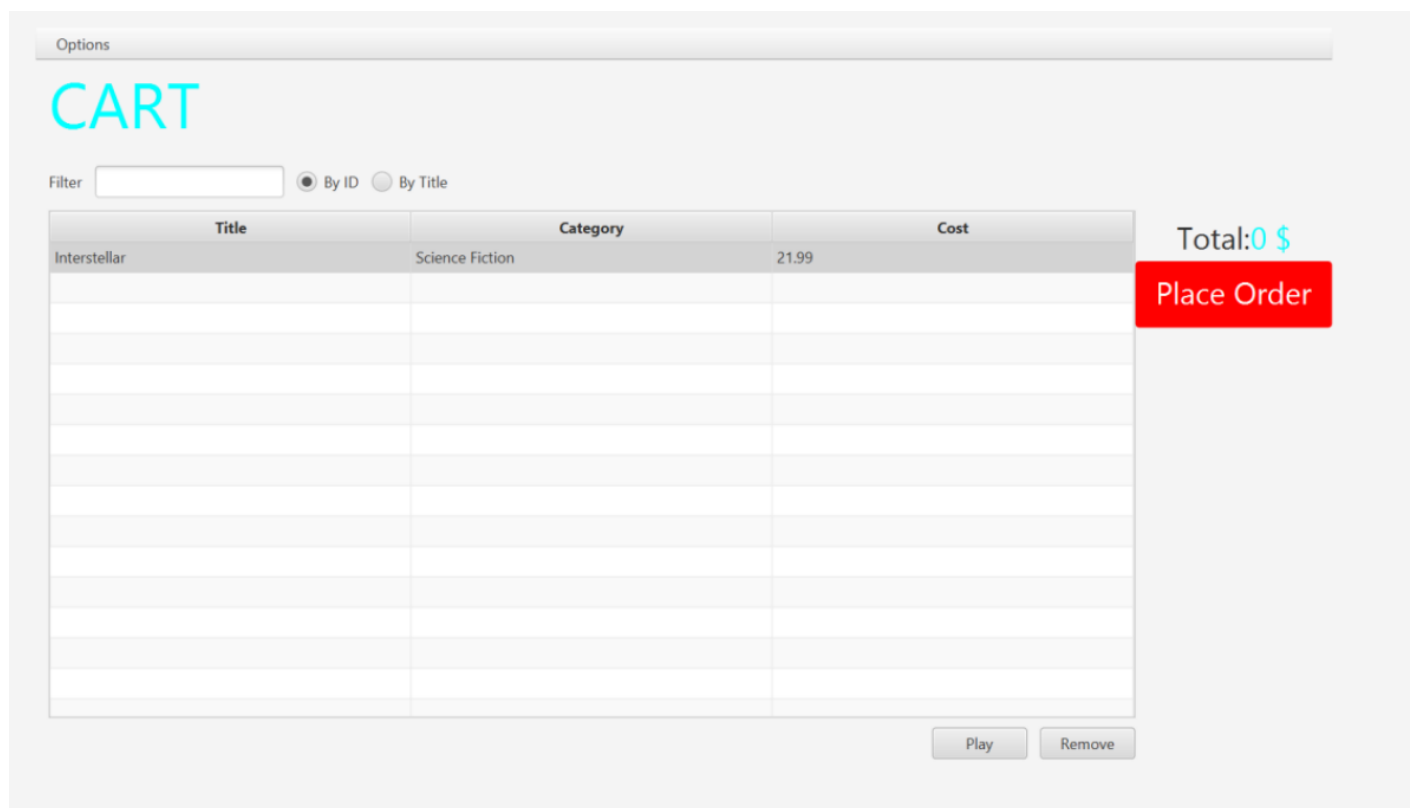


Figure 18 Demo remove

9. Complete

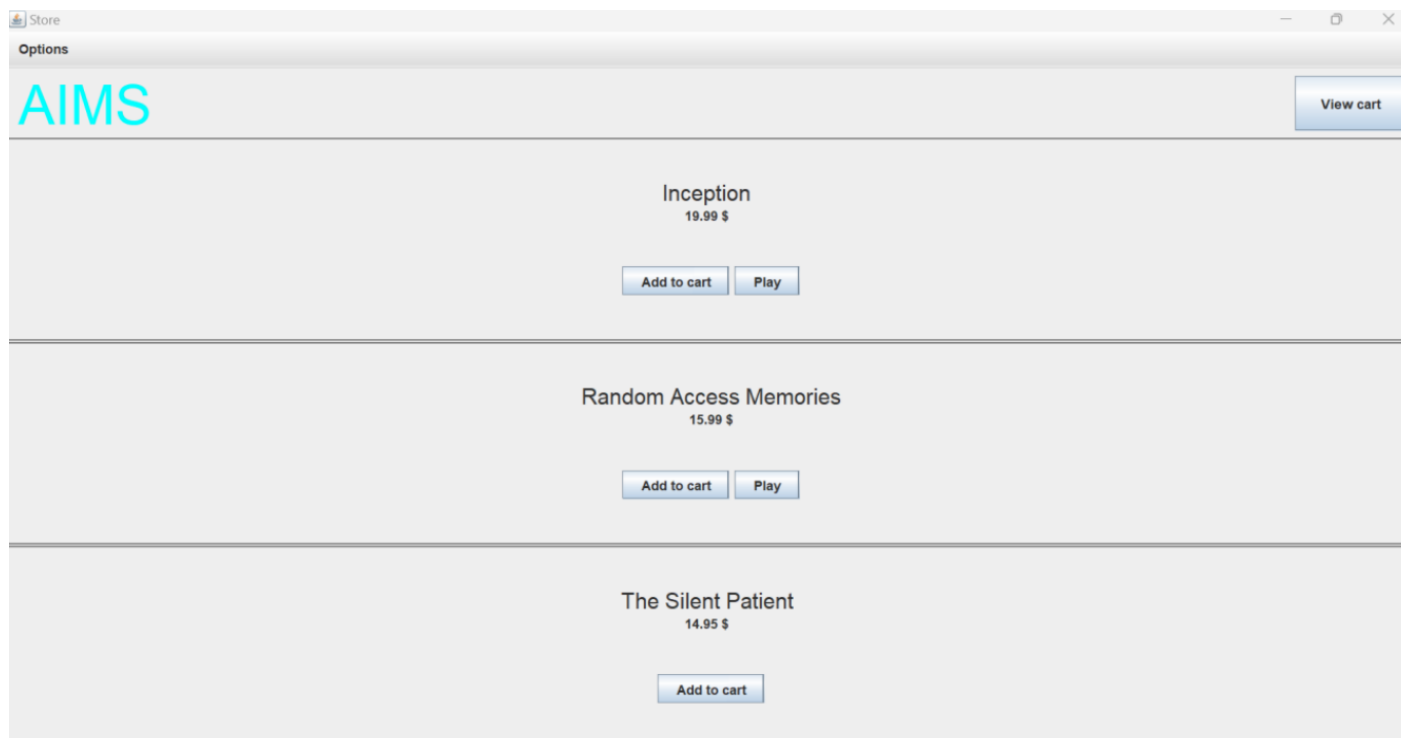


Figure 19: Demo store before add book

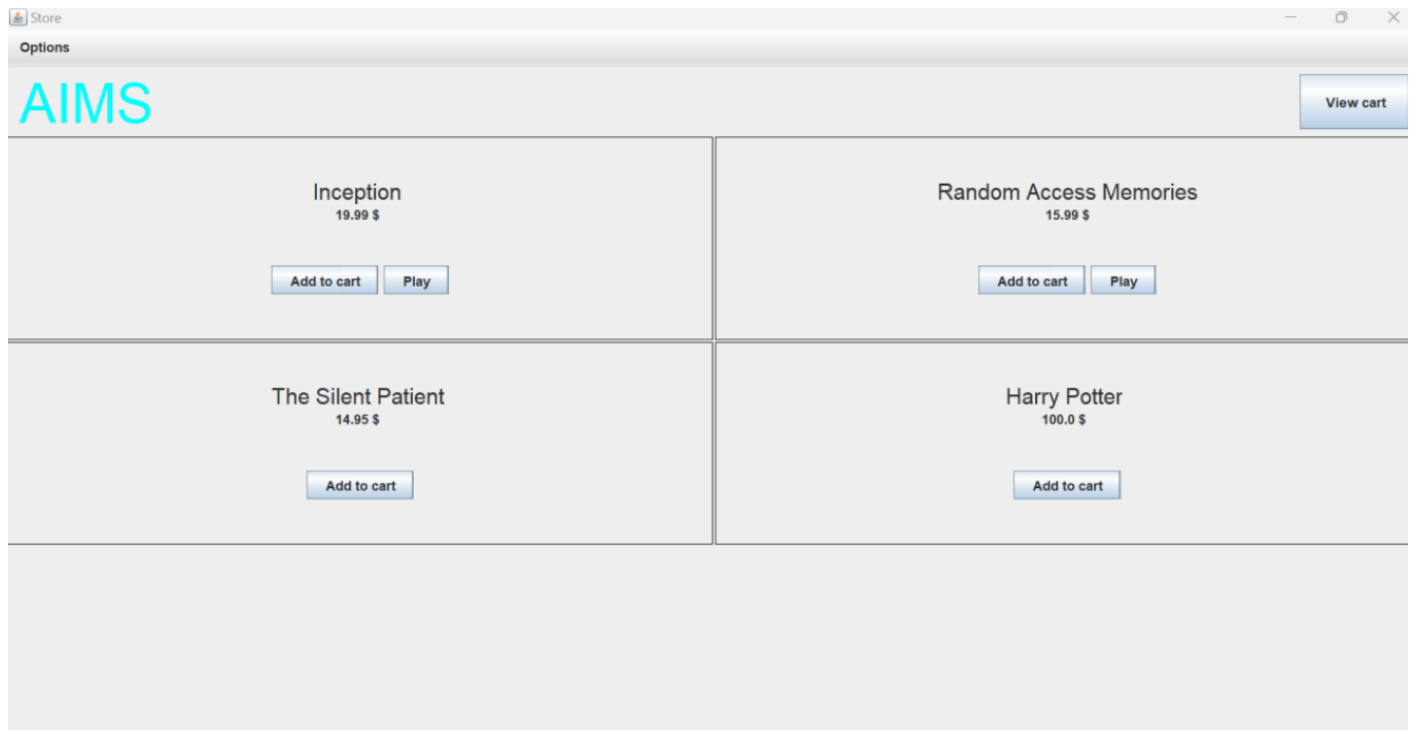


Figure 20: Demo Store after add book

The screenshot shows a web application window titled "Add DVD to store". The header area contains the text "ADD DVD TO STORE" in large blue letters and an "Add" button in the top right corner. Below the header, the form is divided into two main sections. The left section contains six input fields for adding a new DVD, each with a label to its right. The right section is a large, empty area for additional information or notes.

Field	Label
6	Id
The Pacific	Title
Documentation	Category
80	Cost
HBO	Director
1000	Length

Figure 21 Demo Store add DVD

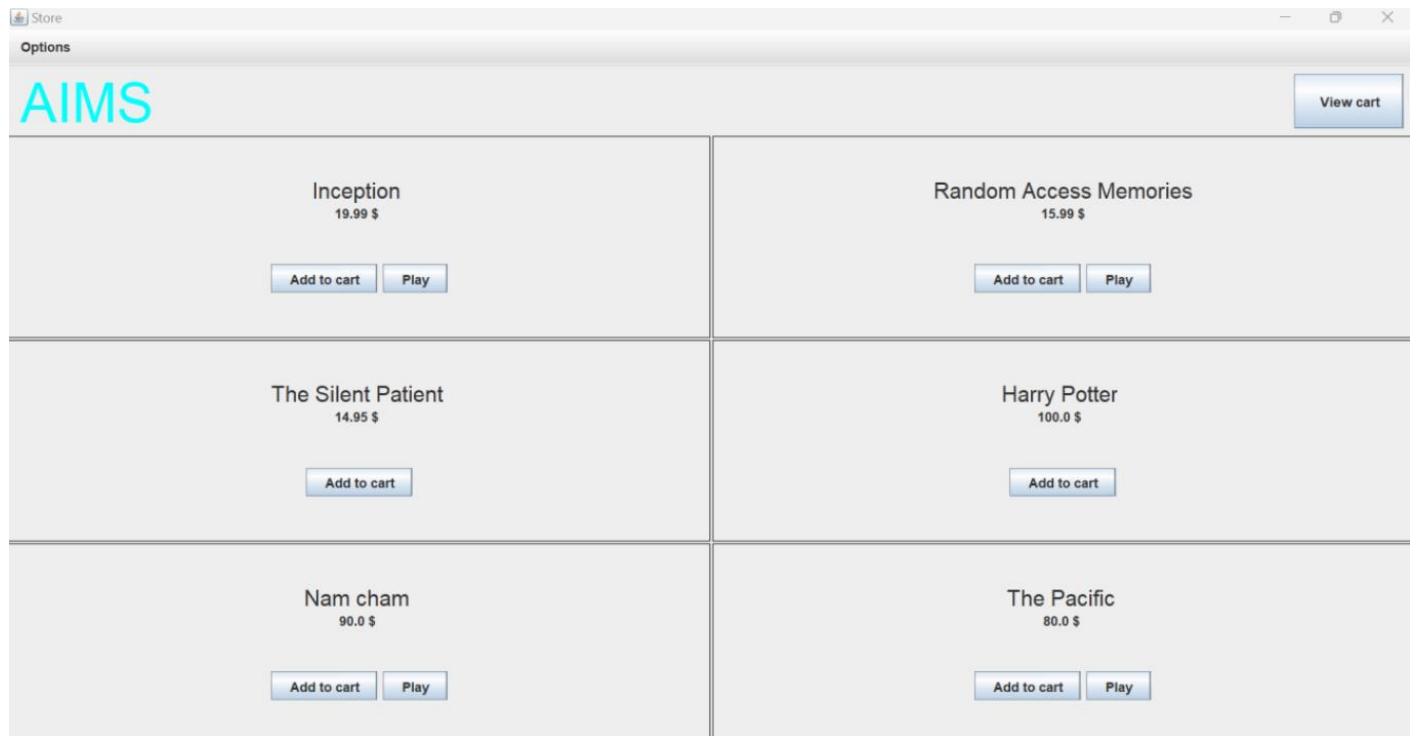


Figure 22 Demo Store after add DVD

```
"AWT-EventQueue-0" java.lang.RuntimeException Create breakpoint : lab05.AimsProject.exception.PlayerException: ERROR: CD length is non-positive!  
object.screen.MediaStore$ButtonListener.actionPerformed(MediaStore.java:87) <4 internal lines>  
/javax.swing.plaf.basic.BasicButtonListener.mouseReleased(BasicButtonListener.java:279) <30 internal lines>  
msProject.exception.PlayerException Create breakpoint : ERROR: CD length is non-positive!  
object.media.CompactDisc.play(CompactDisc.java:73)  
object.screen.MediaStore$ButtonListener.actionPerformed(MediaStore.java:85)
```

Figure 23 Catch Exception

10. Update the Aims class diagram

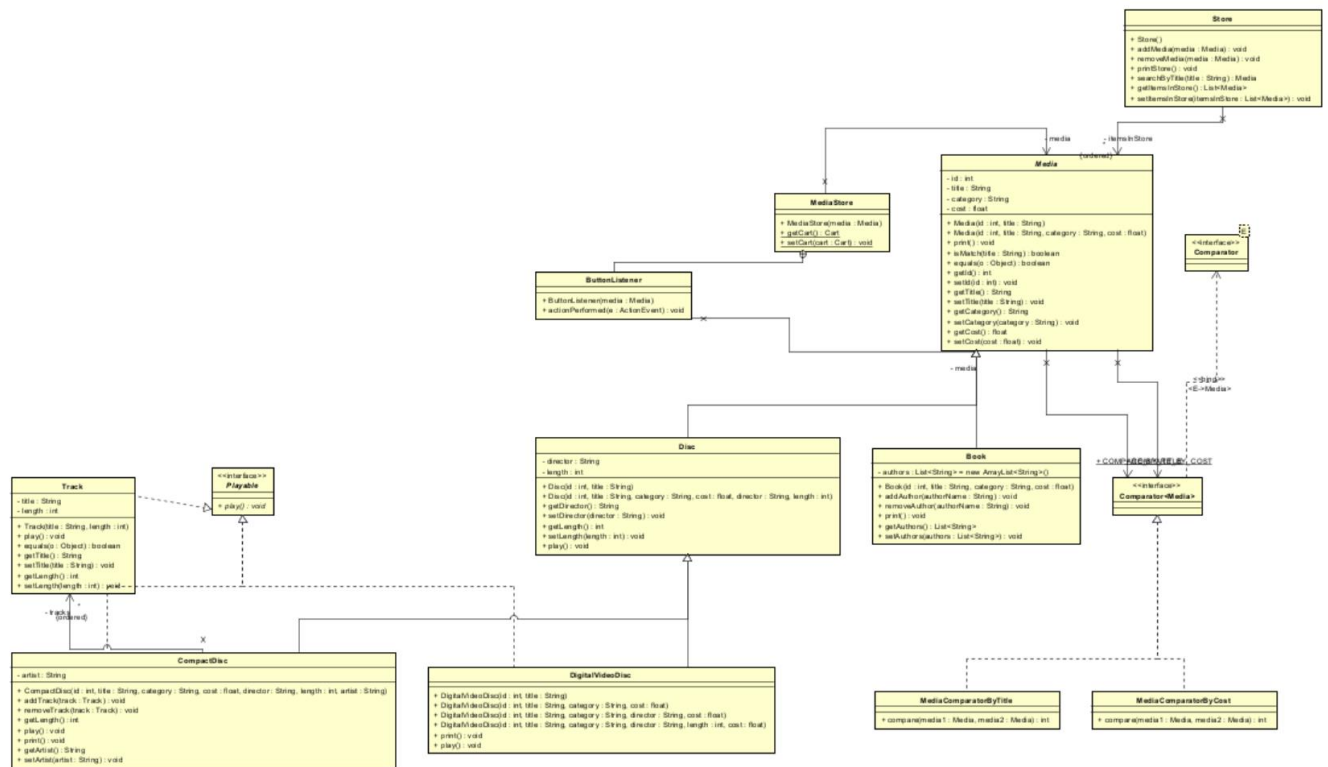


Figure 24 media class diagrams

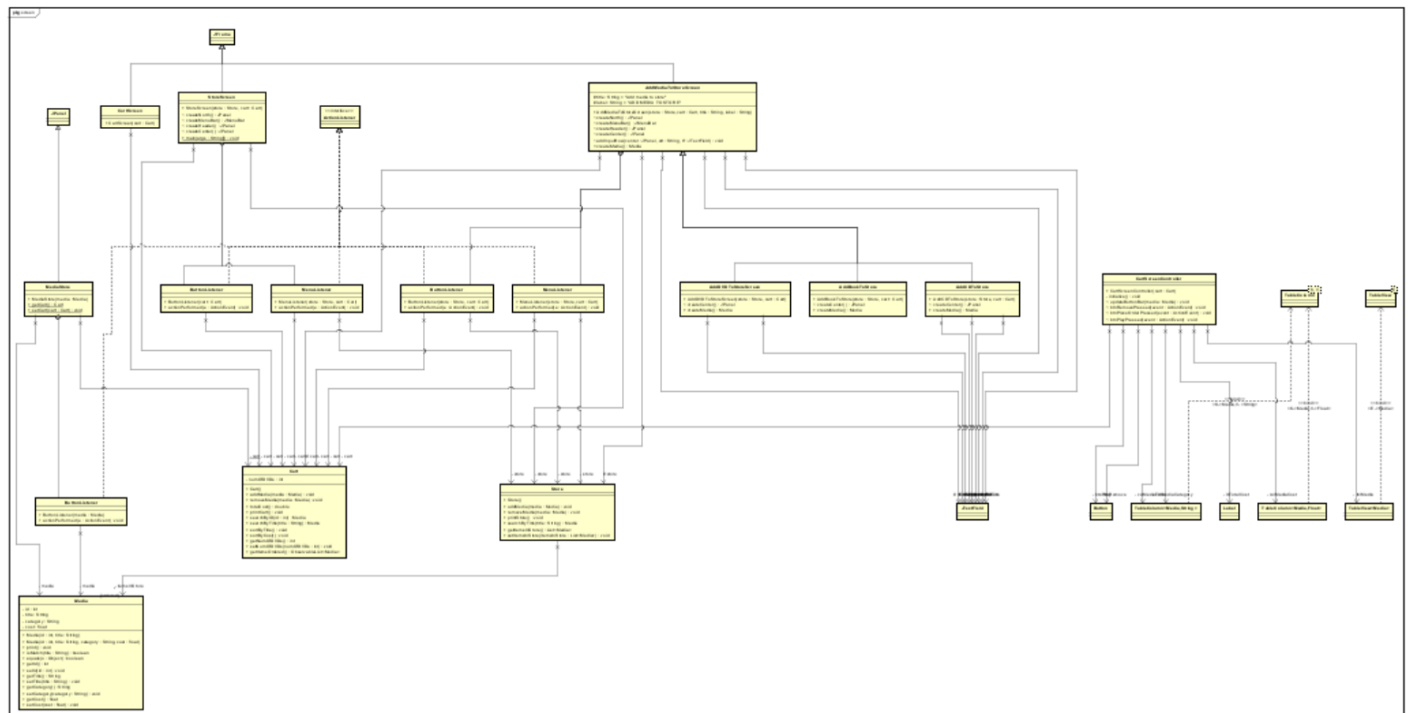


Figure 25 Screen class diagram

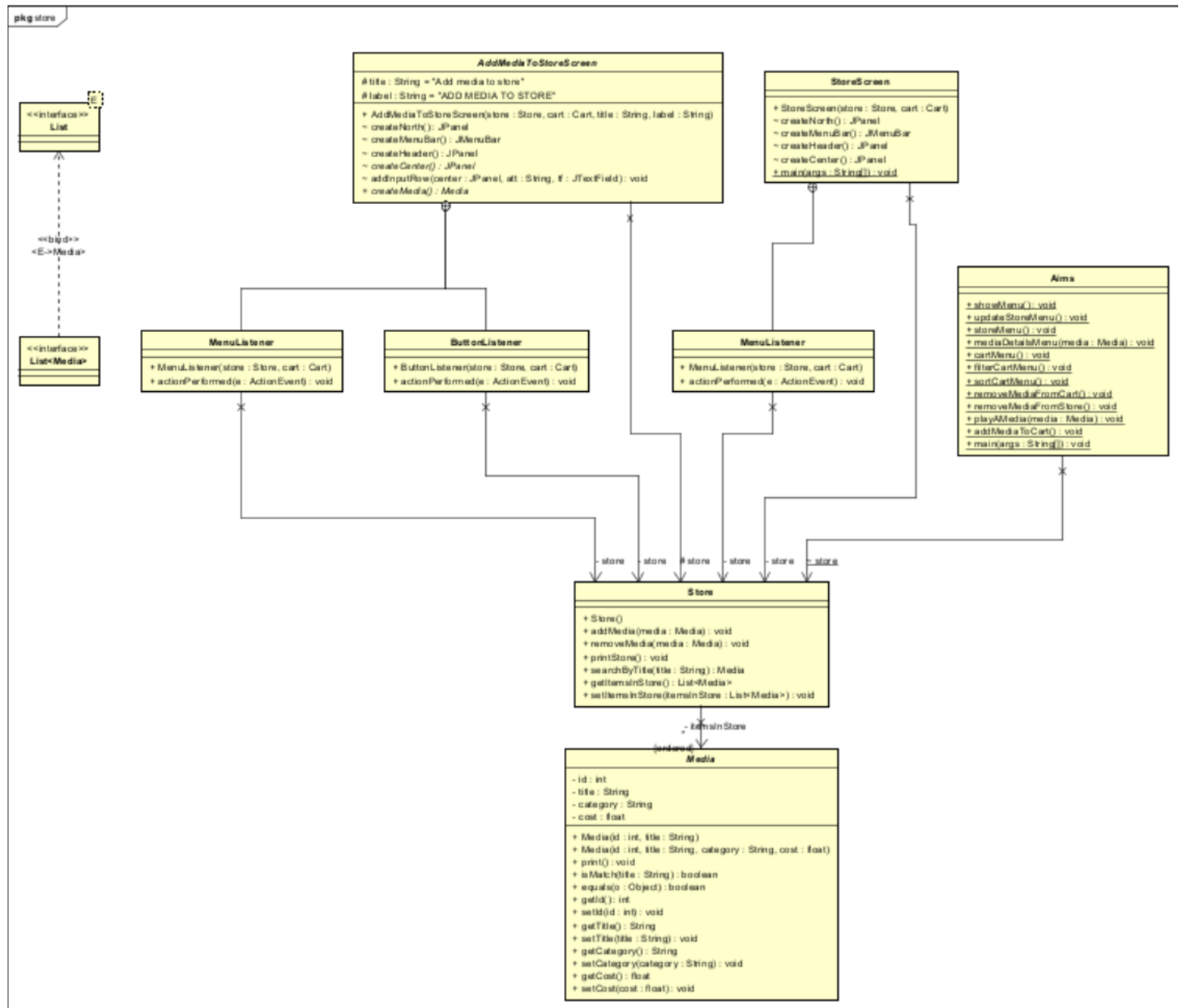


Figure 26 Store Class diagram

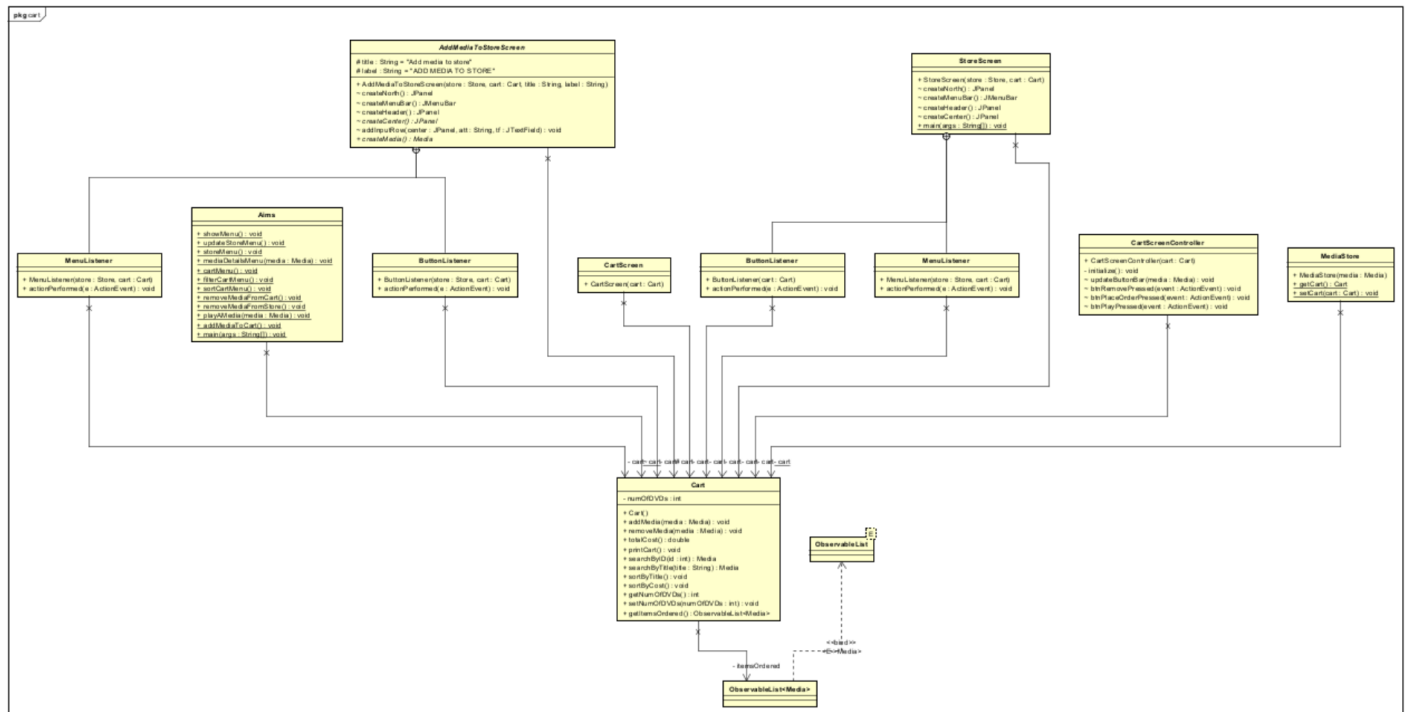


Figure 27 Cart class diagram