ĐẠI HỌC BÁCH KHOA HÀ NỘI
TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

BÁO CÁO THỰC HÀNH
IT3103-744529-2024.1

BÀI THỰC HÀNH 5

Họ và tên sv: Nguyễn Thành Vinh
MSSV: 20225779
Lớp: Việt Nhật 02 – K67
GVHD: Lê Thị Hoa
HTGD: Bùi Trọng Dũng

Hà Nội 12/2024

BÁO CÁO THỰC HÀNH LAB 5
LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG

Contents

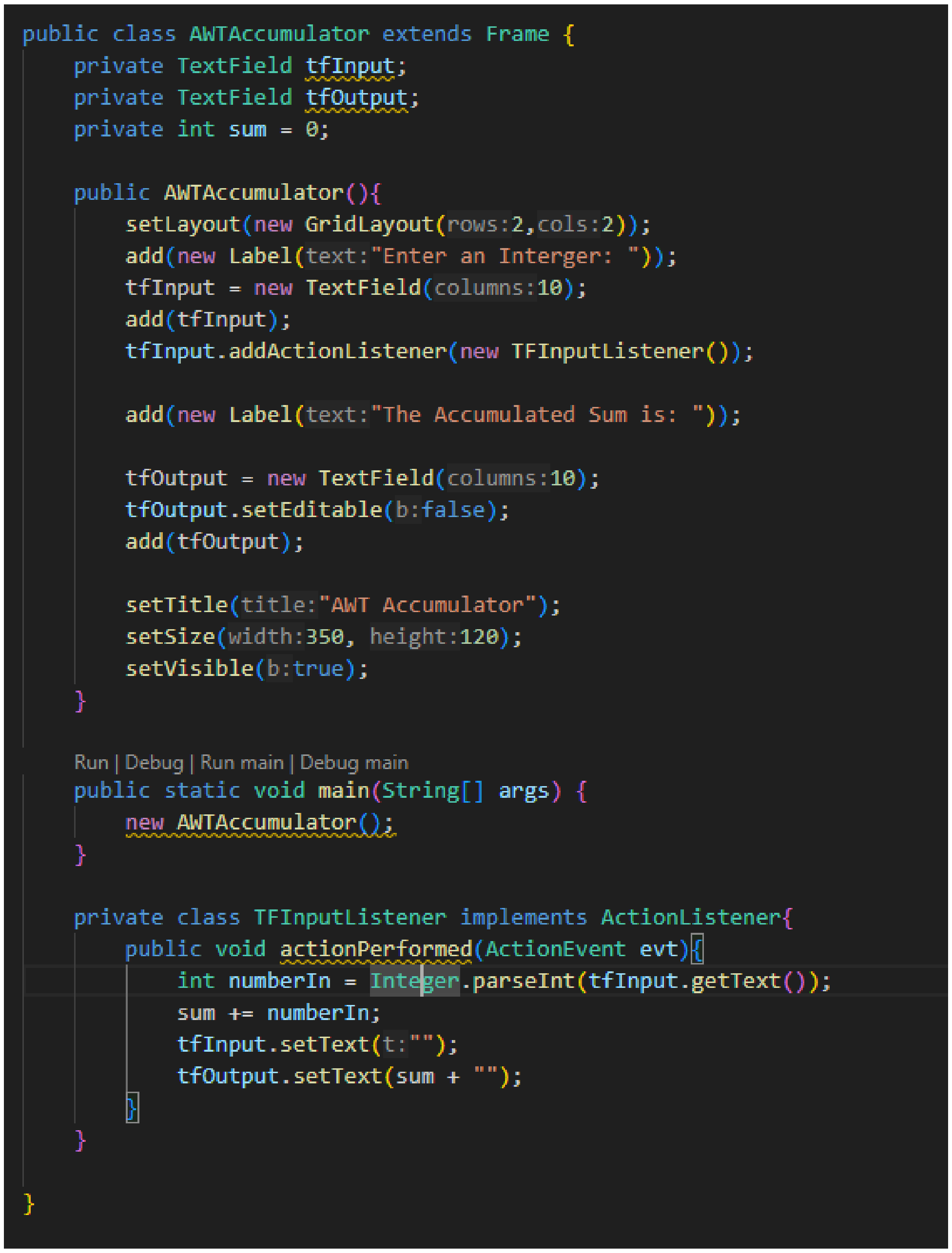
1. Swing components 4
1.1 AWTAccumulator 4
1.2 SwingAccumulator 5
2 Organizing Swing components with Layout Managers 6
2.1 Code 6
2.2 Demo 8
3 Create a graphical user interface for AIMS with Swing 9
3.1 Create class StoreScreen 9
3.2 Create class MediaStore 13
3.3 Demo 14
4 JavaFX API 16
4.1 Create class Painter 16
4.2 Create Painter.fxml 16
4.3 Create class PainterController 17
5 View Cart Screen 19
5.1 Create cart.fxml 19
5.2 Create class CartScreen 20
5.3 Create class CartScreenController 21
5.4 Demo 22
6 Updating buttons based on selected item in TableView – ChangeListener 22
6.1 Edit class CartScreenController 22
6.2 Demo 23
7 Deleting a media 24
7.1 Code 24
7.2 Demo 25
8 Complete the Aims GUI application 26
9 Use case Diagram 30
10 Class Diagram 31

Figure 1.1: Source code of AWTAccumulator 4
Figure 1.2: Demo of AWTAccumulator 5
Figure 1.3: Source code of SwingAccumulator 5
Figure 1.4: Demo of SwingAccumulator 6
Figure 2.1: Source code of NumberGrid 1 6
Figure 2.2: Source code of NumberGrid 2 7
Figure 2.3: Demo buttons 0-9........ 8
Figure 2.4: Demo DEL button 8
Figure 2.5: Demo C button 8
Figure 3.1: Class StoreScreen 1 9
Figure 3.2: Class StoreScreen 2 10
Figure 3.3: Class StoreScreen 3 10
Figure 3.4: Class StoreScreen 4 11
Figure 3.5: Class StoreScreen 5 11
Figure 3.6: Class StoreScreen 6 12
Figure 3.7: Class MediaStore 1 13
Figure 3.8: Class MediaStore 2 13
Figure 3.9: Class MediaStore 3 14
Figure 3.10: StoreScreen 14
Figure 3.11 Demo Add to cart button 15
Figure 3.12 Demo Play button........ 15
Figure 3.13 Demo View cart button 15
Figure 4.1: Class Painter 16
Figure 4.2: Painter.fxml 1 16
Figure 4.3: Painter.fxml 2 17
Figure 4.4: PainterController 17
Figure 4.5: Use Pen........ 18
Figure 4.6: Use Eraser........ 18
Figure 4.7: Clear button........ 18
Figure 5.1: Cart.fxml 1 19
Figure 5.2: Cart.fxml 2 19
Figure 5.3: Cart.fxml 3 20
Figure 5.4: CartScreen class 20
Figure 5.5: CartScreenController 1 21
Figure 5.6: CartScreenController 2 21
Figure 5.7: Demo CartScreen........ 22
Figure 6.1: CartScreenController 1 22
Figure 6.2: CartScreenController 2 23
Figure 6.3: Demo media playable........ 23
Figure 6.4: Demo media unplayable........ 24
Figure 7.1: btnRemovePressed Method........ 24
Figure 7.2: button Remove........ 25
Figure 7.3: button Remove........ 25
Figure 8.1: Store before add book........ 26

Figure 8.2: Add book 26
Figure 8.3: Store after add book........ 27
Figure 8.4: Add CD........ 27
Figure 8.5: Store after add CD 28
Figure 8.6 Add DVD 28
Figure 8.7: Store after add DVD........ 29
Figure 8.8: Cart 29
Figure 8.9: Exception 30

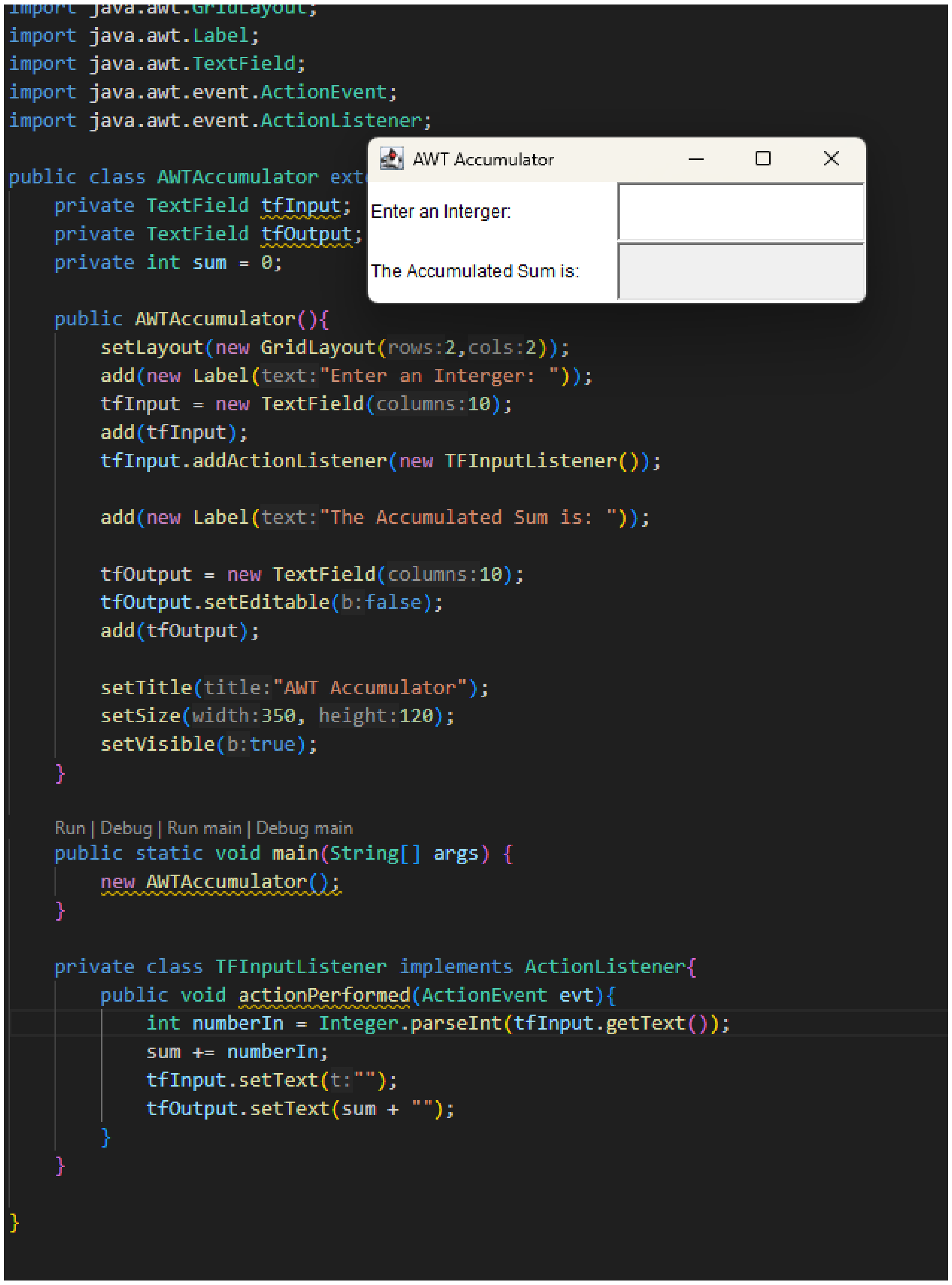
1. Swing components

1.1 AWTAccumulator



```
public class AWTAccumulator extends Frame {
    private TextField tfInput;
    private TextField tfOutput;
    private int sum = 0;

    public AWTAccumulator(){
        setLayout(new GridLayout(rows:2,cols:2));
        add(new Label(text:"Enter an Interger: "));
        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(b:false);
        add(tfOutput);

        setTitle(title:"AWT Accumulator");
        setSize(width:350, height:120);
        setVisible(b:true);
    }

    Run | Debug | Run main | Debug main
    public static void main(String[] args) {
        new AWTAccumulator();
    }

    private class TFInputListener implements ActionListener{
        public void actionPerformed(ActionEvent evt){
            int numberIn = Integer.parseInt(tfInput.getText());
            sum += numberIn;
            tfInput.setText(t:"");
            tfOutput.setText(sum + "");
        }
    }

}
```

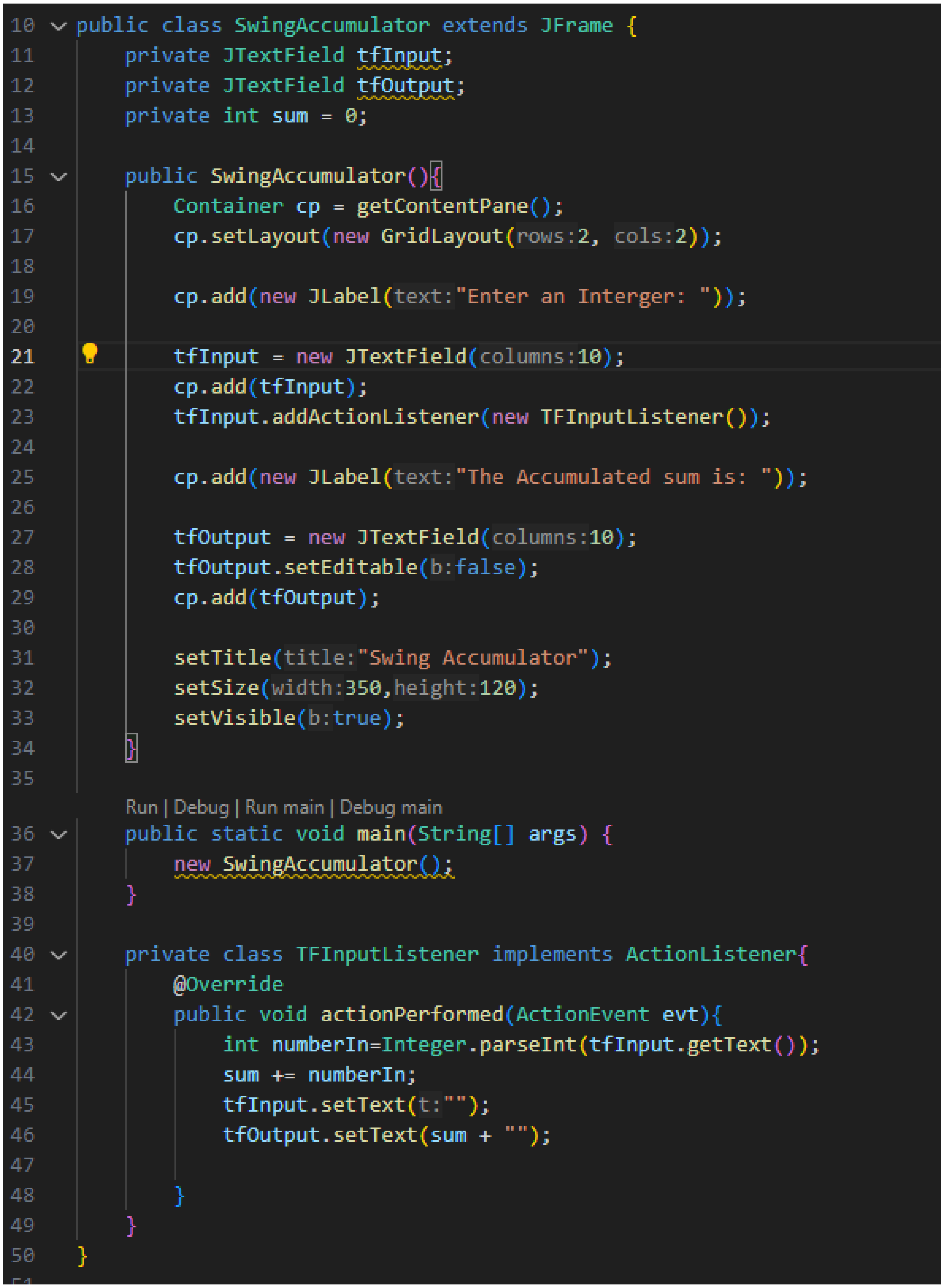
Figure 1.1: Source code of AWTAccumulator



```
import java.awt.GridLayout;
import java.awt.Label;
import java.awt.TextField;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

public class AWTAccumulator ext
    private TextField tfInput;
    private TextField tfOutput;
    private int sum = 0;

    public AWTAccumulator(){
        setLayout(new GridLayout(rows:2,cols:2));
        add(new Label(text:"Enter an Interger: "));
        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(b:false);
        add(tfOutput);

        setTitle(title:"AWT Accumulator");
        setSize(width:350, height:120);
        setVisible(b:true);
    }

    Run | Debug | Run main | Debug main
    public static void main(String[] args) {
        new AWTAccumulator();
    }

    private class TFInputListener implements ActionListener{
        public void actionPerformed(ActionEvent evt){
            int numberIn = Integer.parseInt(tfInput.getText());
            sum += numberIn;
            tfInput.setText(t:"");
            tfOutput.setText(sum + "");
        }
    }

}
```

Figure 1.2: Demo of AWTAccumulator

1.2 SwingAccumulator



```
public class SwingAccumulator extends JFrame {
    private JTextField tfInput;
    private JTextField tfOutput;
    private int sum = 0;

    public SwingAccumulator(){
        Container cp = getContentPane();
        cp.setLayout(new GridLayout(rows:2, cols:2));

        cp.add(new JLabel(text:"Enter an Interger: "));

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

        cp.add(new JLabel(text:"The Accumulated sum is: "));

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

        setTitle(title:"Swing Accumulator");
        setSize(width:350,height:120);
        setVisible(b:true);
    }

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

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

        }
    }
}
```

Figure 1.3: Source code of SwingAccumulator

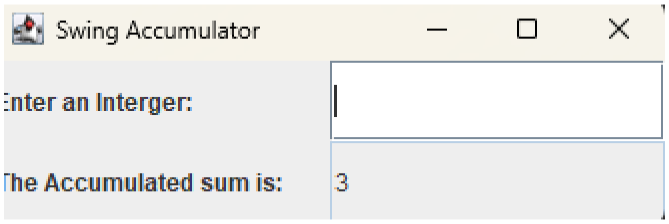
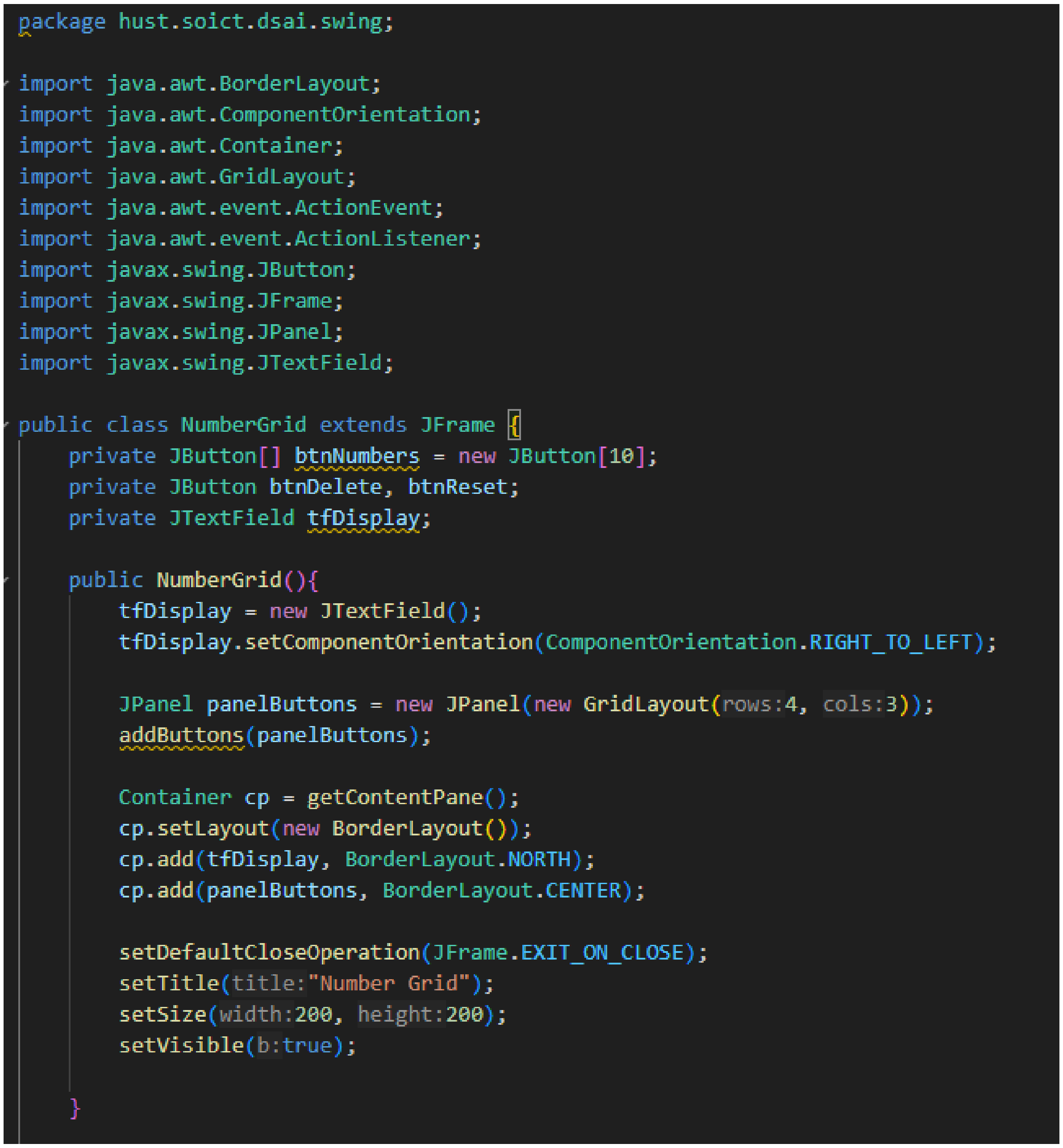


Figure 1.4: Demo of SwingAccumulator

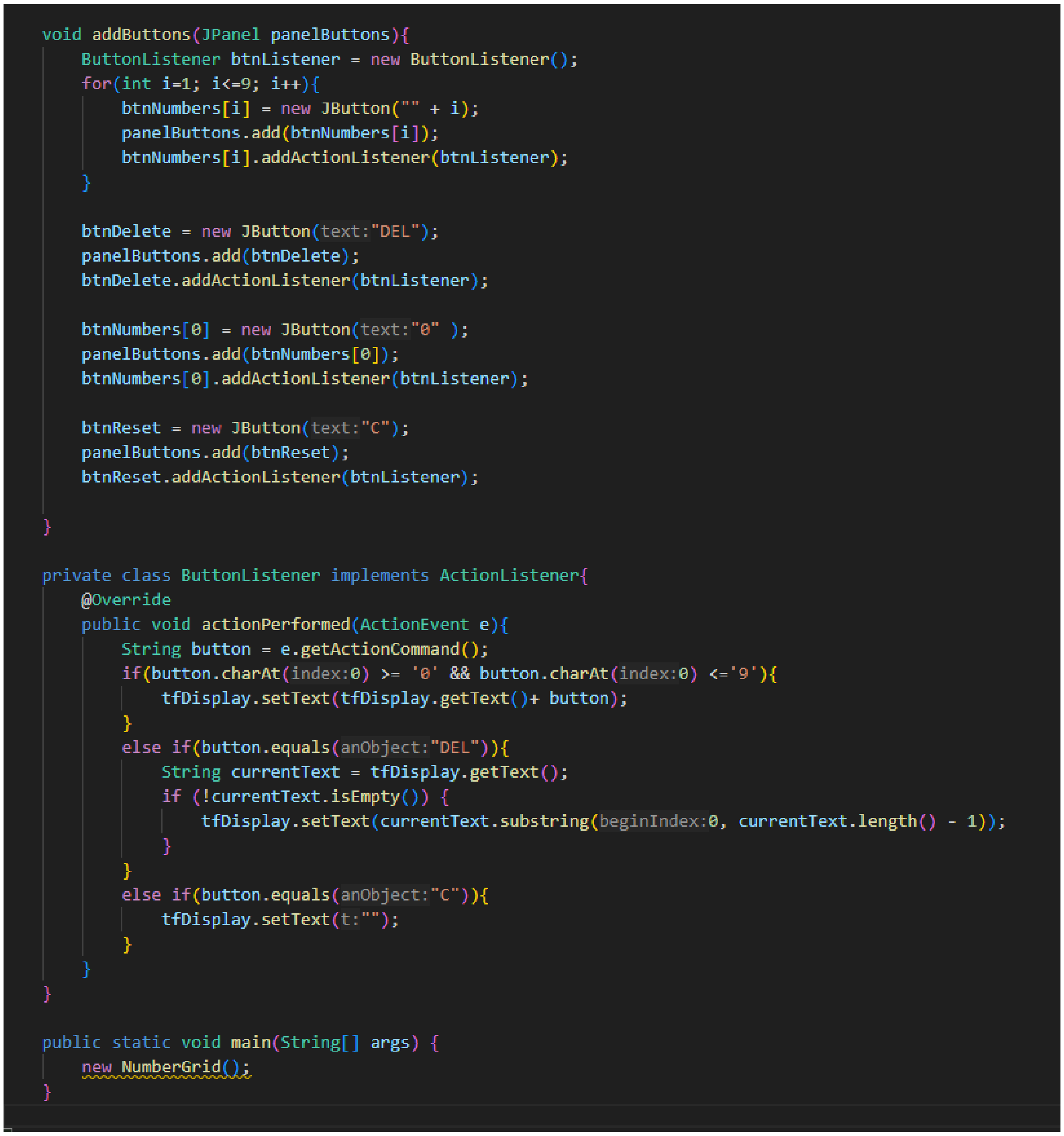
2 Organizing Swing components with Layout Managers

2.1 Code



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

import java.awt.BorderLayout;
import java.awt.ComponentOrientation;
import java.awt.Container;
import java.awt.GridLayout;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.JTextField;

public class NumberGrid extends JFrame {
    private JButton[] btnNumbers = new JButton[10];
    private JButton btnDelete, btnReset;
    private JTextField tfDisplay;

    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(title:"Number Grid");
        setSize(width:200, height:200);
        setVisible(b:true);

    }
```

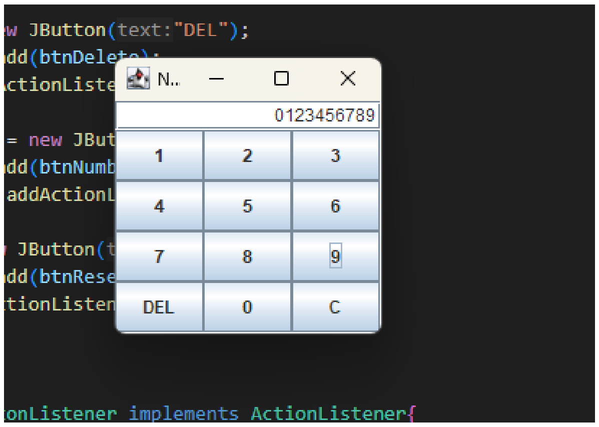
Figure 2.1: Source code of NumberGrid 1



```
void addButtons(JPanel panelButtons){
    ButtonListener btnListener = new ButtonListener();
    for(int i=1; i<=9; i++){
        btnNumbers[i] = new JButton("" + 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);

}

private class ButtonListener implements ActionListener{
    @Override
    public void actionPerformed(ActionEvent e){
        String button = e.getActionCommand();
        if(button.charAt(index:0) >= '0' && button.charAt(index:0) <='9'){
            tfDisplay.setText(tfDisplay.getText()+ button);
        }
        else if(button.equals(anObject:"DEL")){
            String currentText = tfDisplay.getText();
            if (!currentText.isEmpty()) {
                tfDisplay.setText(currentText.substring(beginIndex:0, currentText.length() - 1));
            }
        }
        else if(button.equals(anObject:"C")){
            tfDisplay.setText(t:"");
        }
    }
}

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

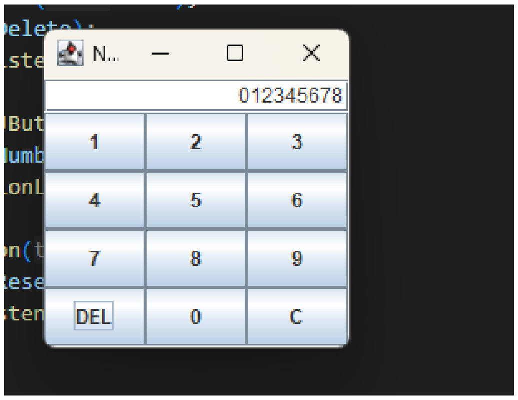
Figure 2.2: Source code of NumberGrid 2

2.2 Demo



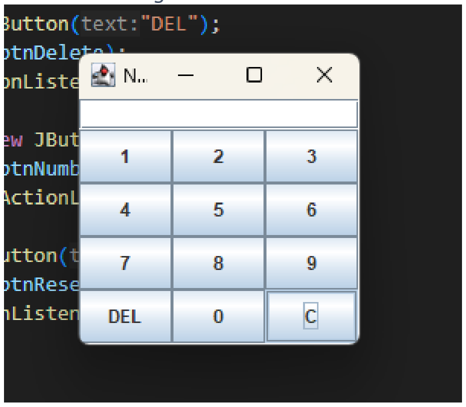
```
w JButton(text:"DEL");
dd(btnDelete);
ctionListe

= new JBut
dd(btnNumb
addActionL

 JButton(t
dd(btnRese
tionListen


onListener implements ActionListener{
```

Figure 2.3: Demo buttons 0-9



```
elete);
ste

But
umb
onL

n(t
ese
ten
```

Figure 2.4: Demo DEL button

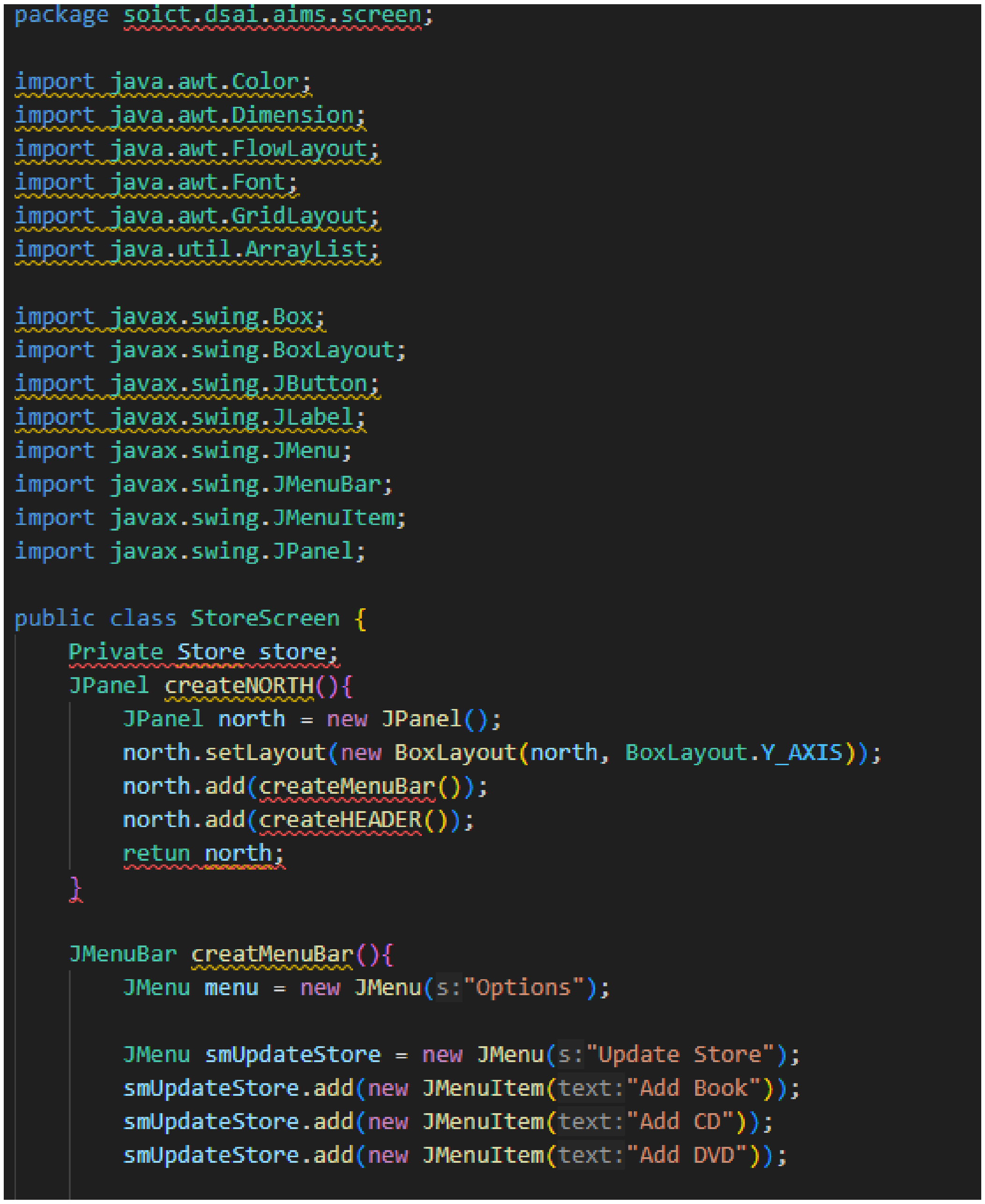


```
Button(text:"DEL");
btnDelete);
onListe

w JBut
btnNumb
ActionL

utton(t
btnRese
Listen
```

Figure 2.5: Demo C button

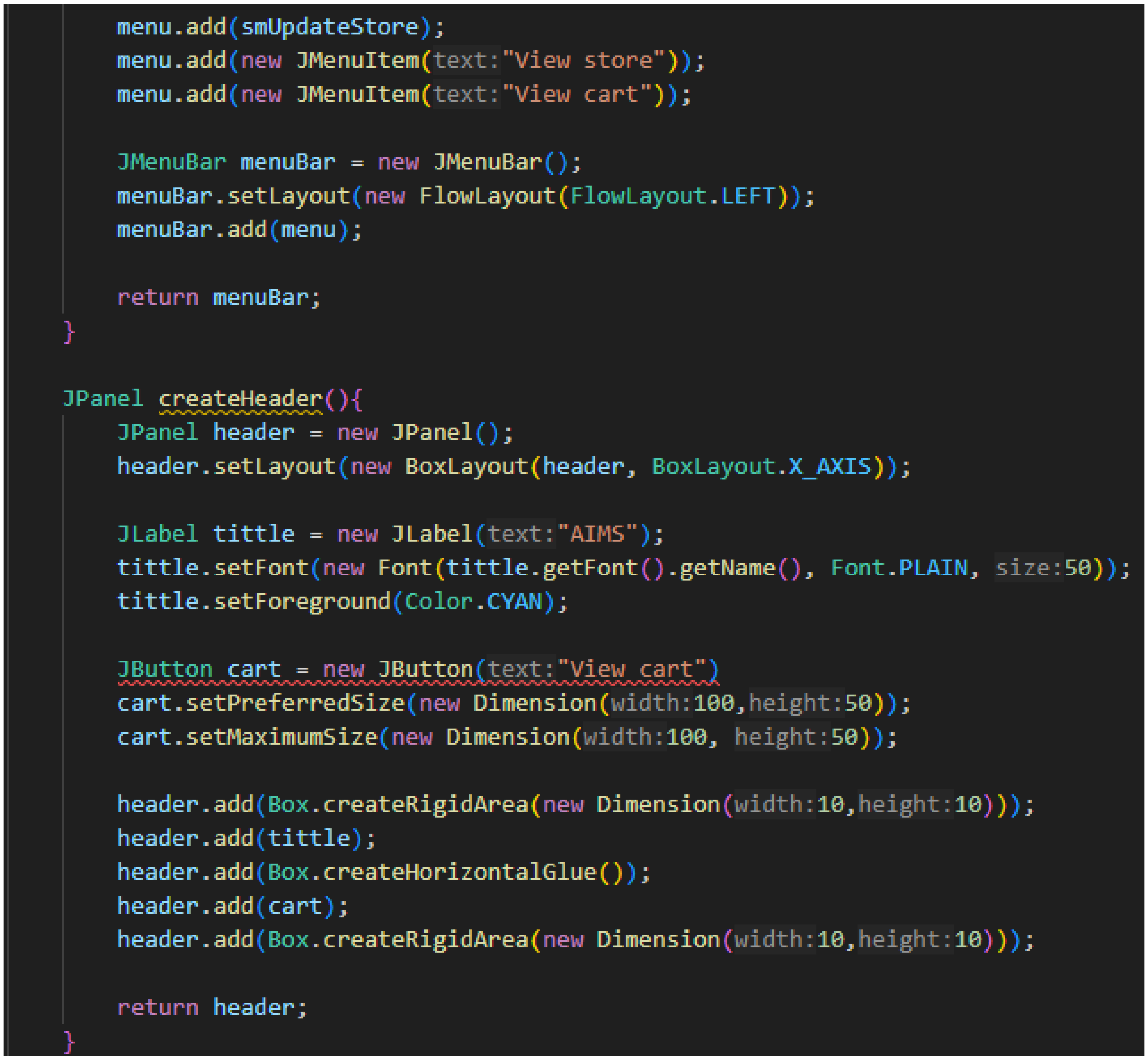
3 Create a graphical user interface for AIMS with Swing

3.1 Create class StoreScreen



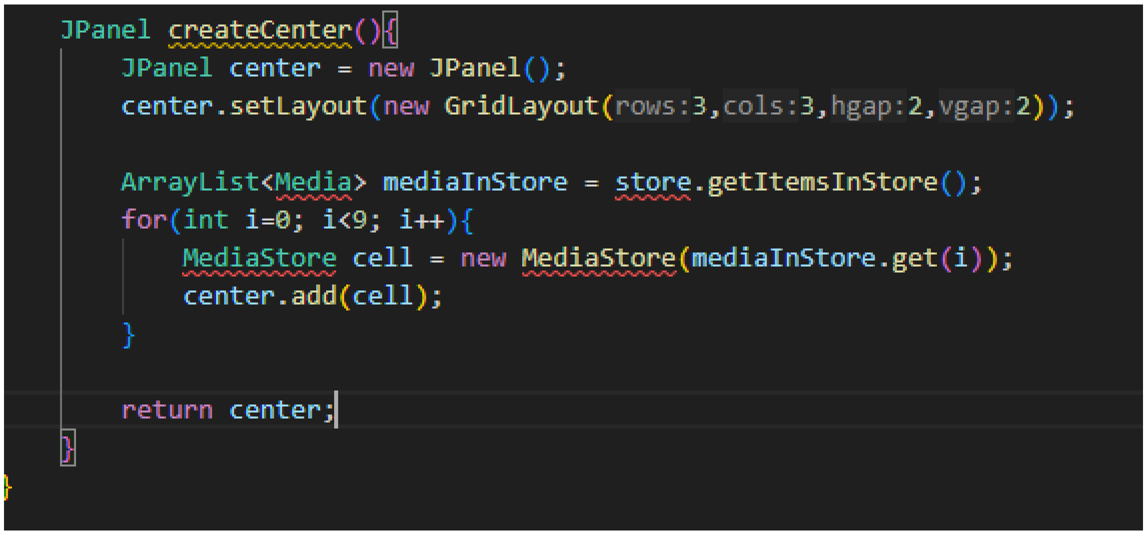
```
package soict.dsai.aims.screen;

import java.awt.Color;
import java.awt.Dimension;
import java.awt.FlowLayout;
import java.awt.Font;
import java.awt.GridLayout;
import java.util.ArrayList;

import javax.swing.Box;
import javax.swing.BoxLayout;
import javax.swing.JButton;
import javax.swing.JLabel;
import javax.swing.JMenu;
import javax.swing.JMenuBar;
import javax.swing.JMenuItem;
import javax.swing.JPanel;

public class StoreScreen {
    Private Store store;
    JPanel createNORTH(){
        JPanel north = new JPanel();
        north.setLayout(new BoxLayout(north, BoxLayout.Y_AXIS));
        north.add(createMenuBar());
        north.add(createHEADER());
        retun north;
    }

    JMenuBar creatMenuBar(){
        JMenu menu = new JMenu(s:"Options");

        JMenu smUpdateStore = new JMenu(s:"Update Store");
        smUpdateStore.add(new JMenuItem(text:"Add Book"));
        smUpdateStore.add(new JMenuItem(text:"Add CD"));
        smUpdateStore.add(new JMenuItem(text:"Add DVD"));
```

Figure 3.1: Class StoreScreen 1



```
        menu.add(smUpdateStore);
        menu.add(new JMenuItem(text:"View store"));
        menu.add(new JMenuItem(text:"View cart"));

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

        return menuBar;
    }

    JPanel createHeader(){
        JPanel header = new JPanel();
        header.setLayout(new BoxLayout(header, BoxLayout.X_AXIS));

        JLabel tittle = new JLabel(text:"AIMS");
        tittle.setFont(new Font(tittle.getFont().getName(), Font.PLAIN, size:50));
        tittle.setForeground(Color.CYAN);

        JButton cart = new JButton(text:"View cart")
        cart.setPreferredSize(new Dimension(width:100,height:50));
        cart.setMaximumSize(new Dimension(width:100, height:50));

        header.add(Box.createRigidArea(new Dimension(width:10,height:10)));
        header.add(tittle);
        header.add(Box.createHorizontalGlue());
        header.add(cart);
        header.add(Box.createRigidArea(new Dimension(width:10,height:10)));

        return header;
    }
```

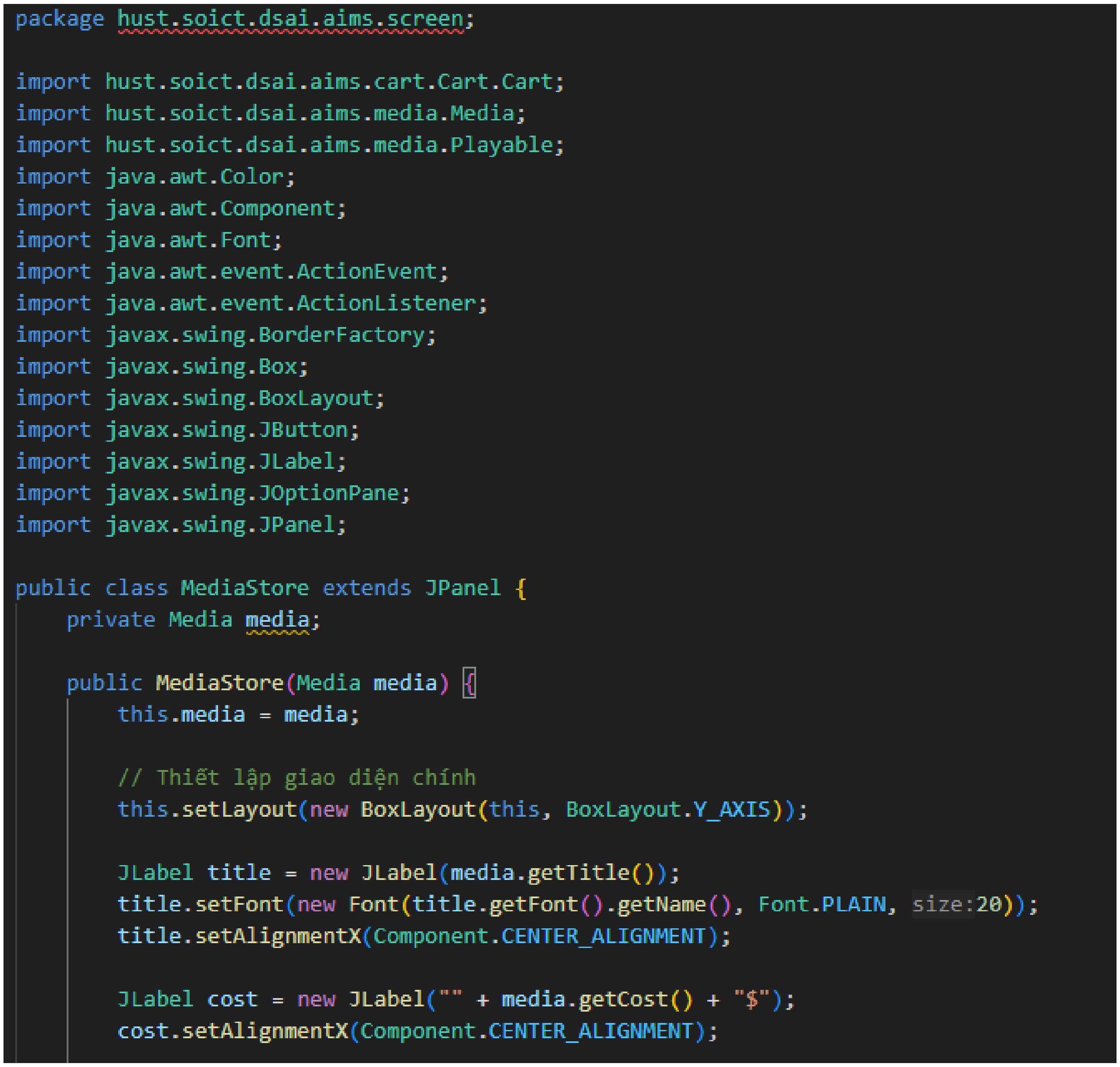
Figure 3.2: Class StoreScreen 2



```
    JPanel createCenter(){
        JPanel center = new JPanel();
        center.setLayout(new GridLayout(rows:3,cols:3,hgap:2,vgap:2));

        ArrayList<Media> mediaInStore = store.getItemsInStore();
        for(int i=0; i<9; i++){
            MediaStore cell = new MediaStore(mediaInStore.get(i));
            center.add(cell);
        }

        return center;
    }
}
```

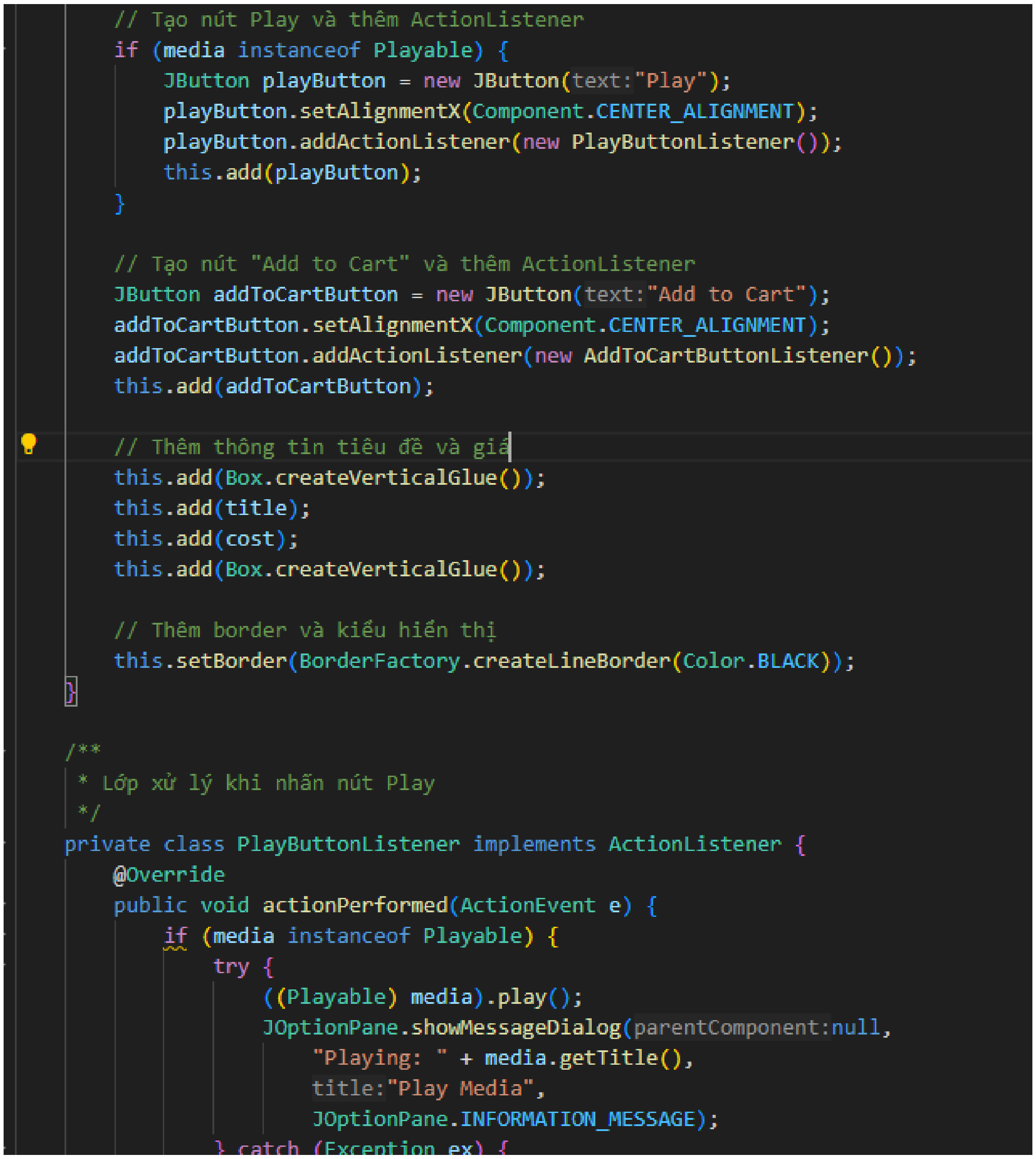
Figure 3.3: Class StoreScreen 3

3.2 Create class MediaStore



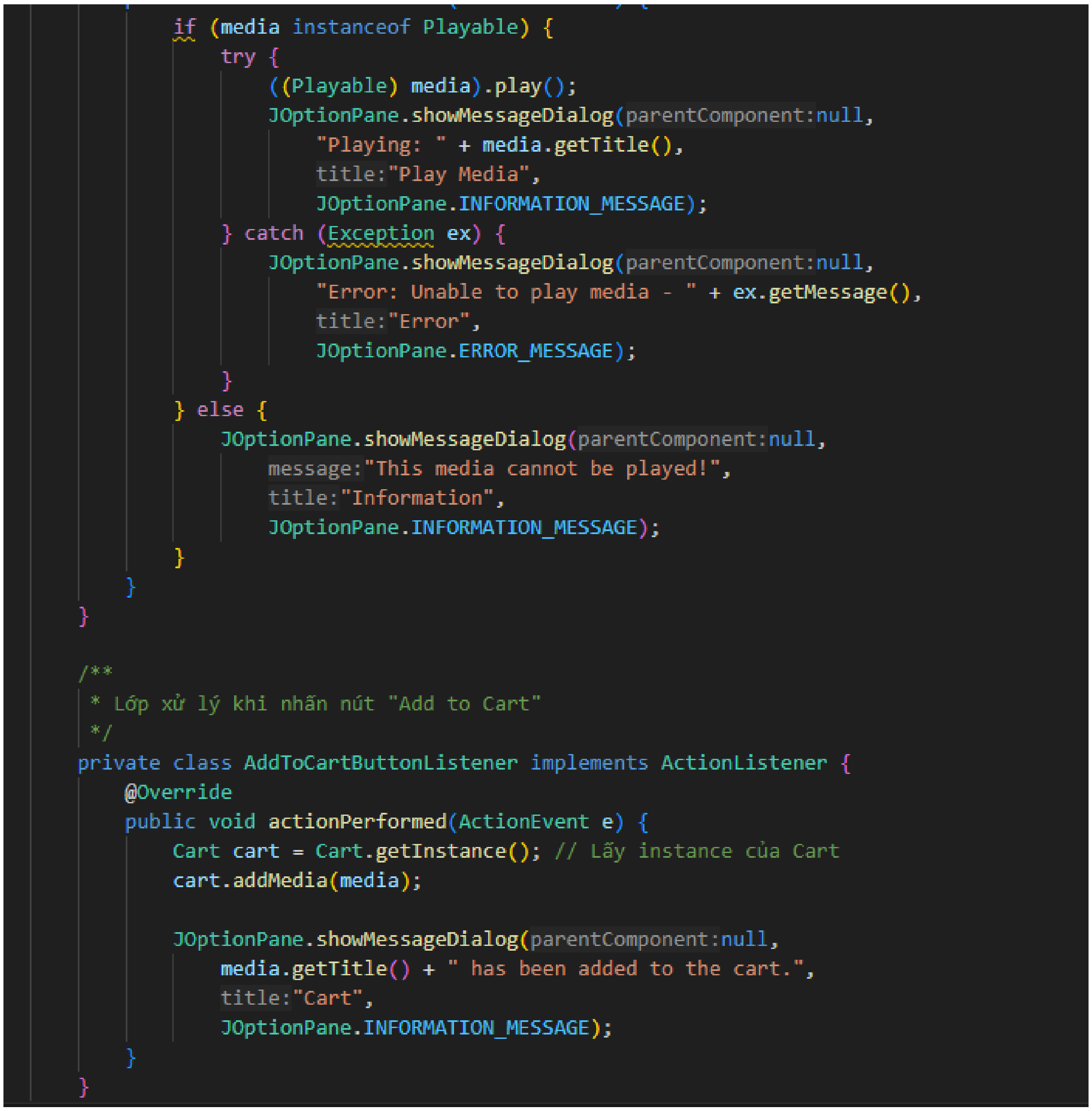
```java
package hust.soict.dsai.aims.screen;

import hust.soict.dsai.aims.cart.Cart.Cart;
import hust.soict.dsai.aims.media.Media;
import hust.soict.dsai.aims.media.Playable;
import java.awt.Color;
import java.awt.Component;
import java.awt.Font;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import javax.swing.BorderFactory;
import javax.swing.Box;
import javax.swing.BoxLayout;
import javax.swing.JButton;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.JPanel;

public class MediaStore extends JPanel {
    private Media media;

    public MediaStore(Media media) {
        this.media = media;

        // Thiết lập giao diện chính
        this.setLayout(new BoxLayout(this, BoxLayout.Y_AXIS));

        JLabel title = new JLabel(media.getTitle());
        title.setFont(new Font(title.getFont().getName(), Font.PLAIN, size:20));
        title.setAlignmentX(Component.CENTER_ALIGNMENT);

        JLabel cost = new JLabel("" + media.getCost() + "$");
        cost.setAlignmentX(Component.CENTER_ALIGNMENT);
```

Figure 3.7: Class MediaStore 1



```
        // Tạo nút Play và thêm ActionListener
        if (media instanceof Playable) {
            JButton playButton = new JButton(text:"Play");
            playButton.setAlignmentX(Component.CENTER_ALIGNMENT);
            playButton.addActionListener(new PlayButtonListener());
            this.add(playButton);
        }

        // Tạo nút "Add to Cart" và thêm ActionListener
        JButton addToCartButton = new JButton(text:"Add to Cart");
        addToCartButton.setAlignmentX(Component.CENTER_ALIGNMENT);
        addToCartButton.addActionListener(new AddToCartButtonListener());
        this.add(addToCartButton);

        // Thêm thông tin tiêu đề và giá
        this.add(Box.createVerticalGlue());
        this.add(title);
        this.add(cost);
        this.add(Box.createVerticalGlue());

        // Thêm border và kiểu hiển thị
        this.setBorder(BorderFactory.createLineBorder(Color.BLACK));
    }

    /**
     * Lớp xử lý khi nhấn nút Play
     */
    private class PlayButtonListener implements ActionListener {
        @Override
        public void actionPerformed(ActionEvent e) {
            if (media instanceof Playable) {
                try {
                    ((Playable) media).play();
                    JOptionPane.showMessageDialog(parentComponent:null,
                        "Playing: " + media.getTitle(),
                        title:"Play Media",
                        JOptionPane.INFORMATION_MESSAGE);
                } catch (Exception ex) {
```

Figure 3.8: Class MediaStore 2



```
            if (media instanceof Playable) {
                try {
                    ((Playable) media).play();
                    JOptionPane.showMessageDialog(parentComponent:null,
                        "Playing: " + media.getTitle(),
                        title:"Play Media",
                        JOptionPane.INFORMATION_MESSAGE);
                } catch (Exception ex) {
                    JOptionPane.showMessageDialog(parentComponent:null,
                        "Error: Unable to play media - " + ex.getMessage(),
                        title:"Error",
                        JOptionPane.ERROR_MESSAGE);
                }
            } else {
                JOptionPane.showMessageDialog(parentComponent:null,
                    message:"This media cannot be played!",
                    title:"Information",
                    JOptionPane.INFORMATION_MESSAGE);
            }
        }
    }

    /**
     * Lớp xử lý khi nhấn nút "Add to Cart"
     */
    private class AddToCartButtonListener implements ActionListener {
        @Override
        public void actionPerformed(ActionEvent e) {
            Cart cart = Cart.getInstance(); // Lấy instance của Cart
            cart.addMedia(media);

            JOptionPane.showMessageDialog(parentComponent:null,
                media.getTitle() + " has been added to the cart.",
                title:"Cart",
                JOptionPane.INFORMATION_MESSAGE);
        }
    }
```

Figure 3.9: Class MediaStore 3

3.3 Demo

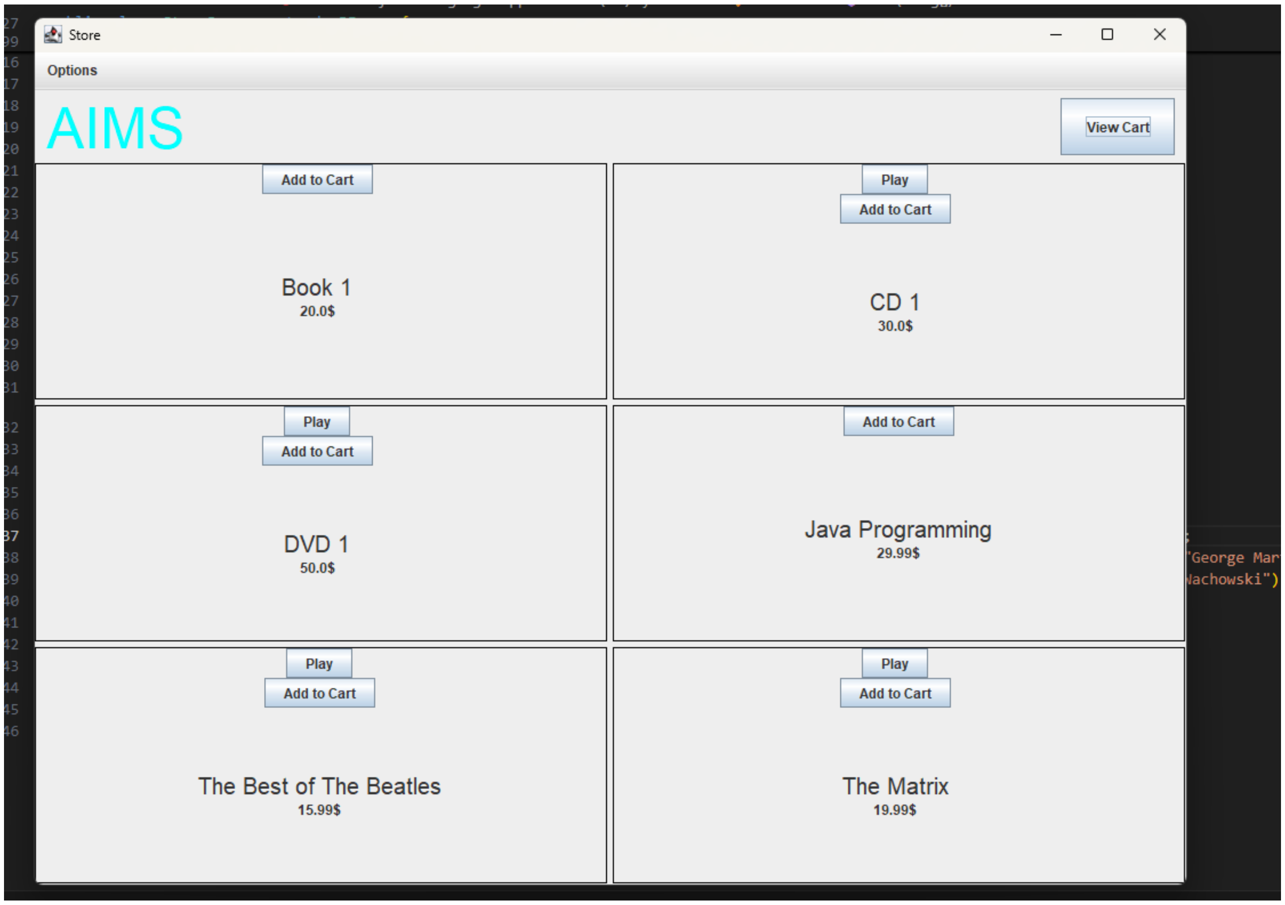
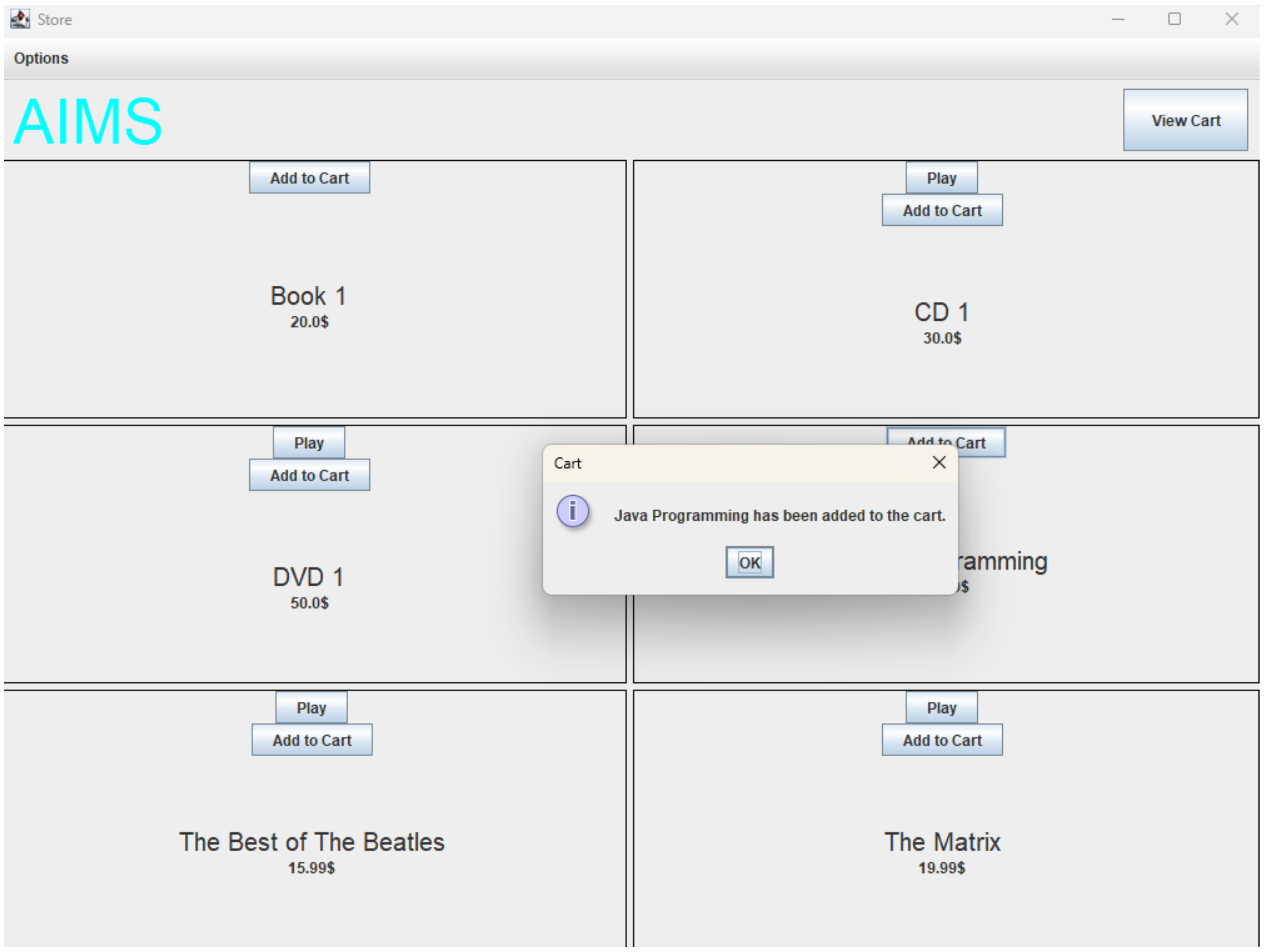
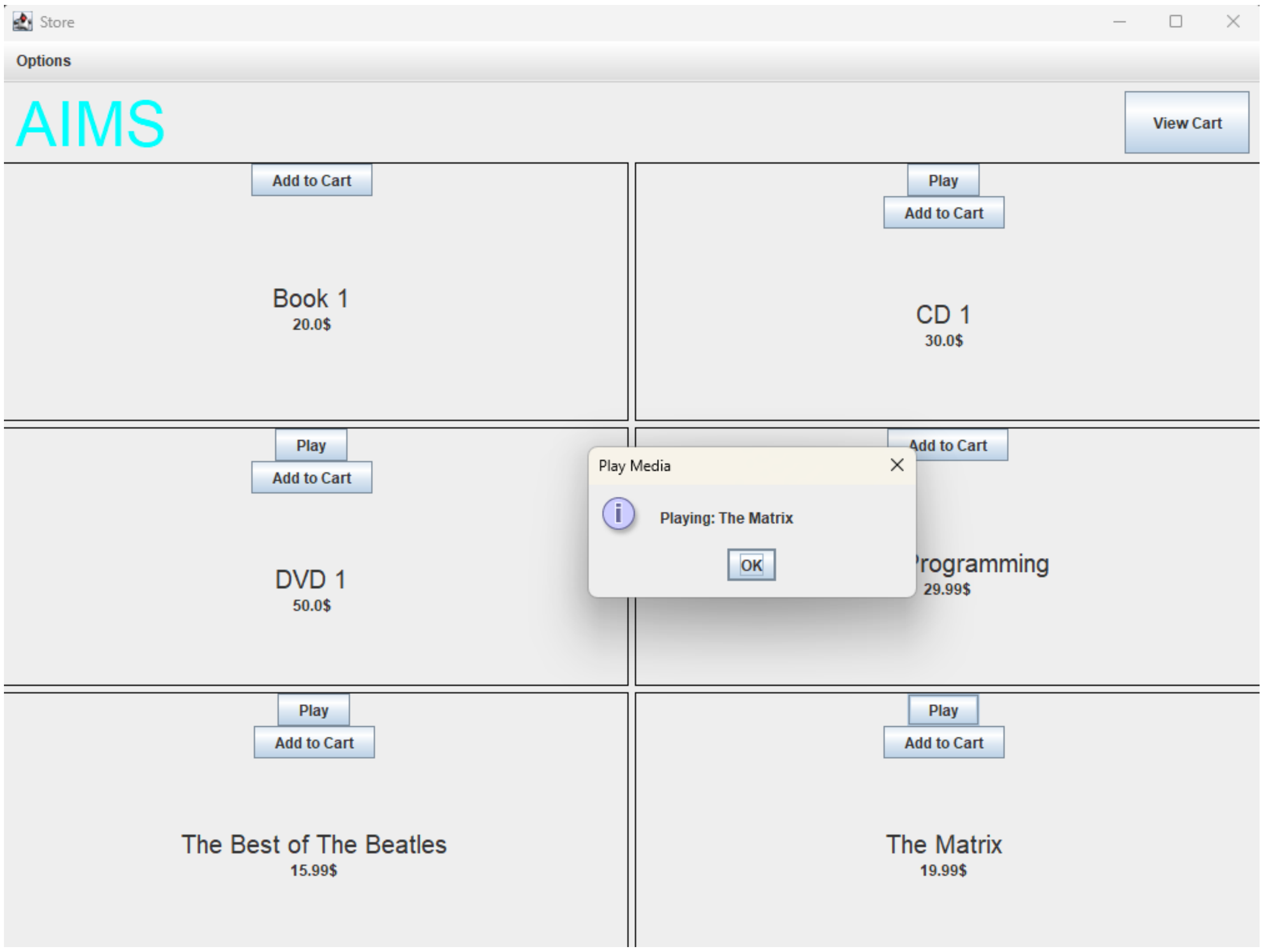


Figure 3.10: StoreScreen



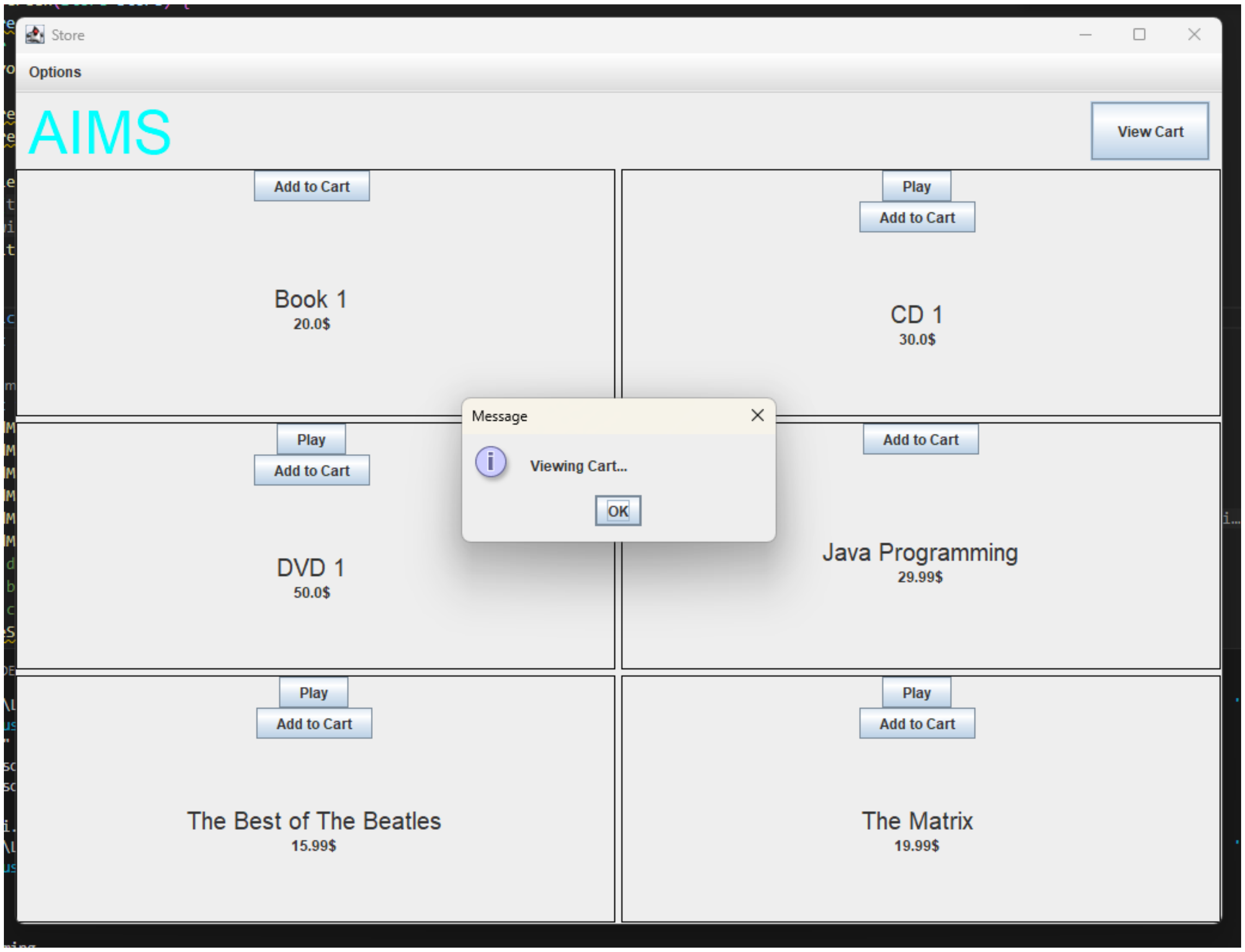
Java Programming has been added to the cart.

Figure 3.11 Demo Add to cart button



Playing: The Matrix

Figure 3.12 Demo Play button



Viewing Cart...

Figure 3.13 Demo View cart button

4 JavaFX API

4.1 Create class Painter

Figure 4.1: Class Painter

4.2 Create Painter.fxml

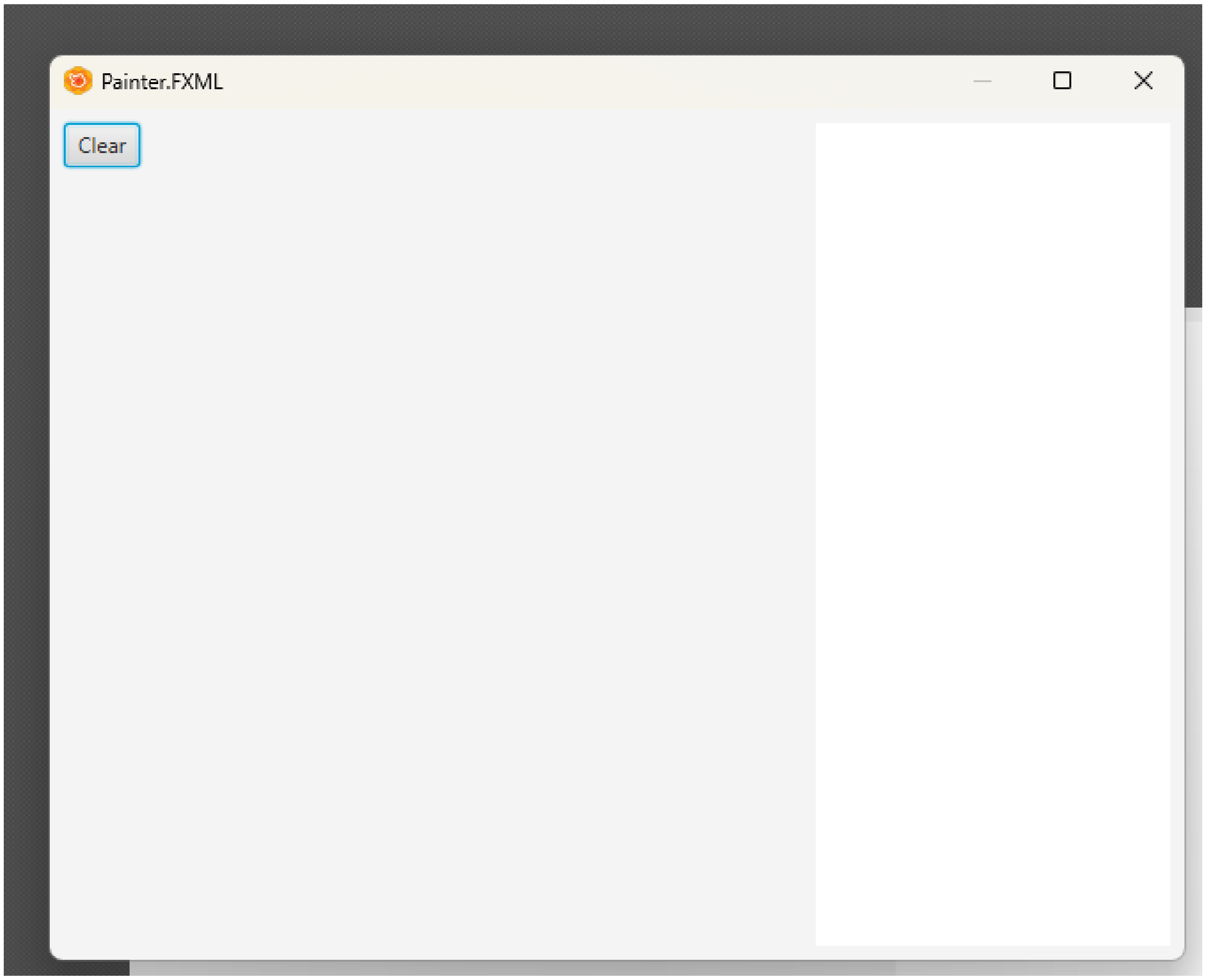
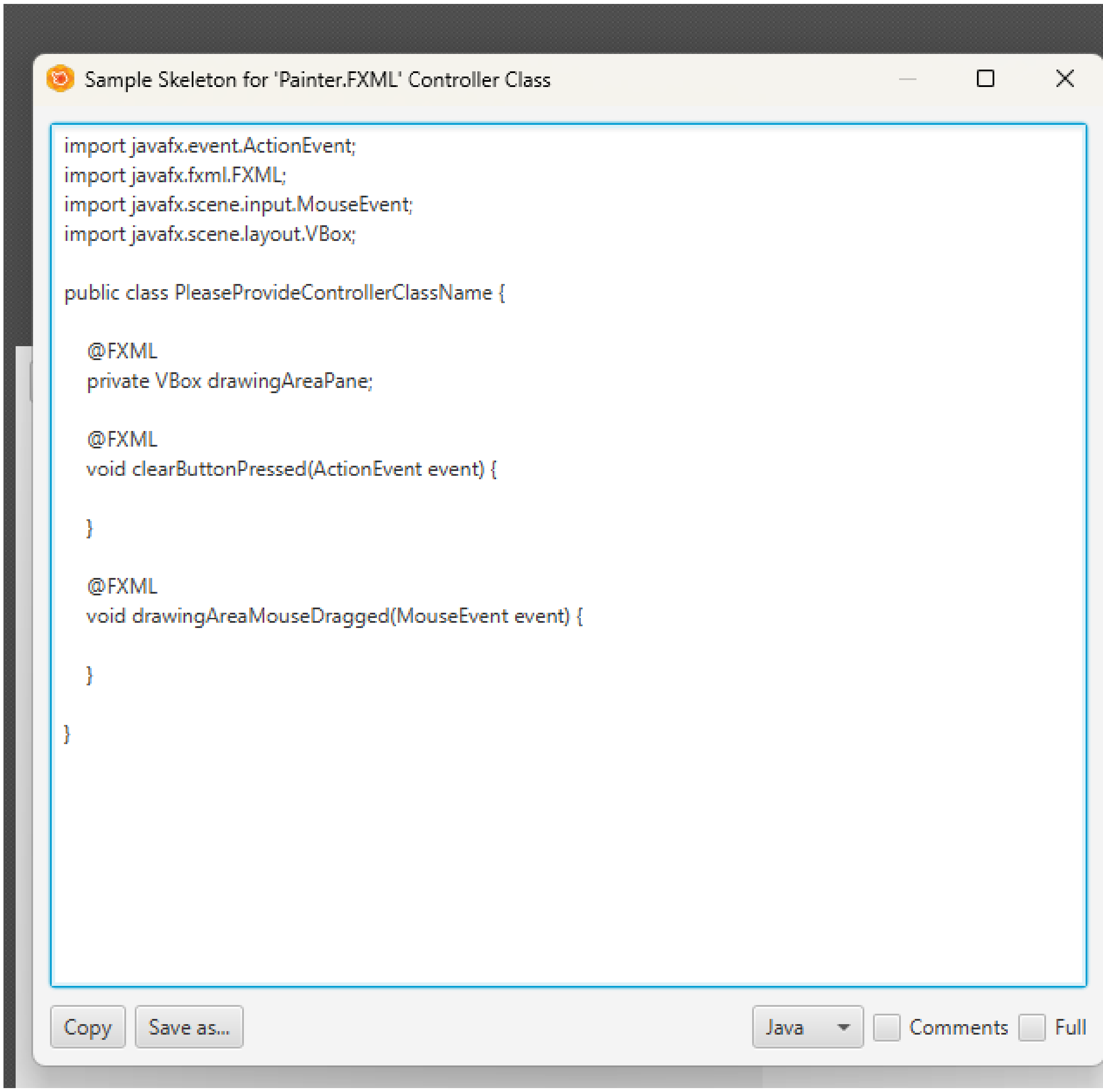


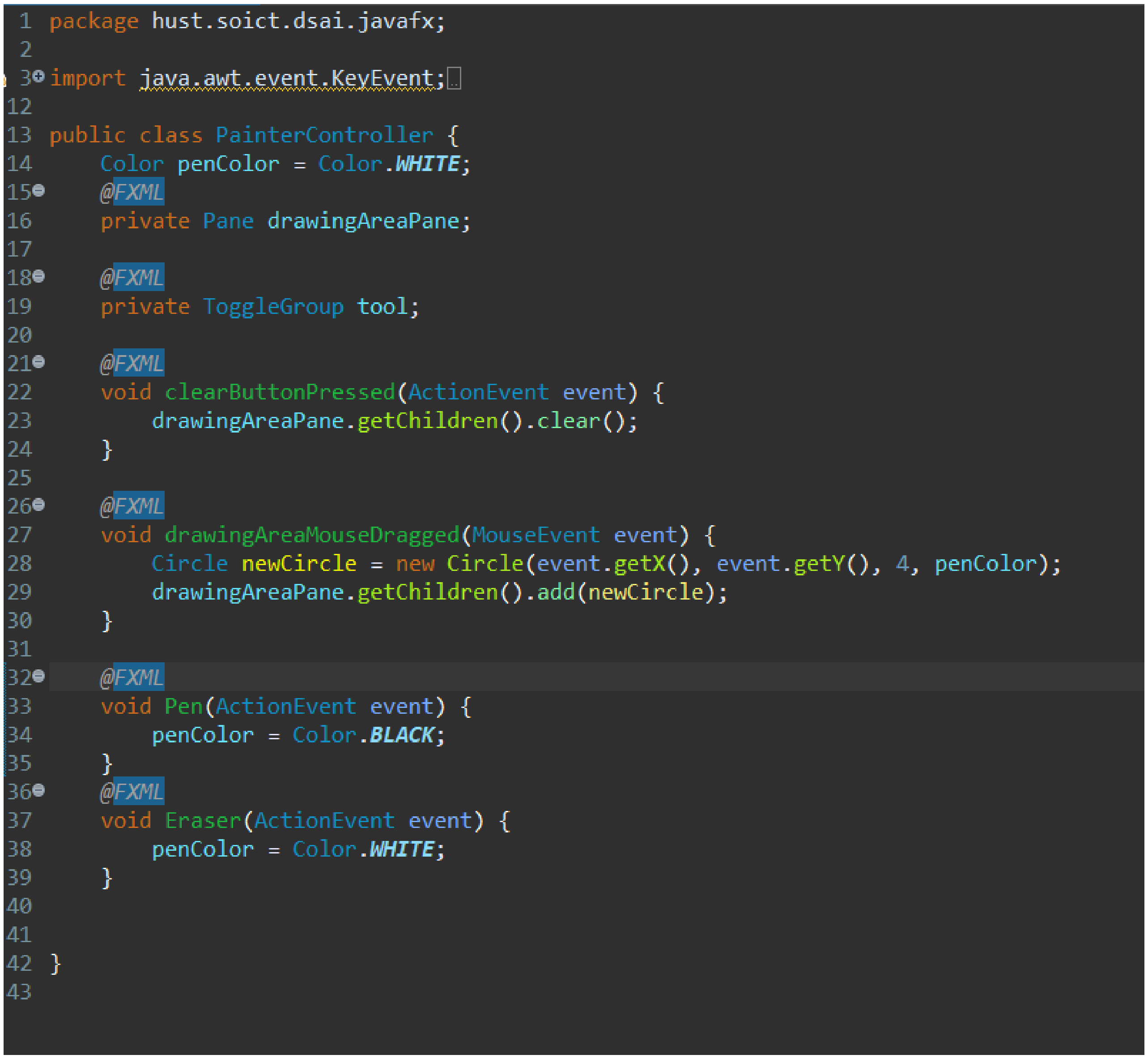
Figure 4.2: Painter.fxml 1



```
import javafx.event.ActionEvent;
import javafx.fxml.FXML;
import javafx.scene.input.MouseEvent;
import javafx.scene.layout.VBox;

public class PleaseProvideControllerClassName {

    @FXML
    private VBox drawingAreaPane;

    @FXML
    void clearButtonPressed(ActionEvent event) {

    }

    @FXML
    void drawingAreaMouseDragged(MouseEvent event) {

    }

}
```

Figure 4.3: Painter.fxml 2

4.3 Create class PainterController



```
package hust.soict.dsai.javafx;

import java.awt.event.KeyEvent;

public class PainterController {
    Color penColor = Color.WHITE;
    @FXML
    private Pane drawingAreaPane;

    @FXML
    private ToggleGroup tool;

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

    @FXML
    void drawingAreaMouseDragged(MouseEvent event) {
        Circle newCircle = new Circle(event.getX(), event.getY(), 4, penColor);
        drawingAreaPane.getChildren().add(newCircle);
    }

    @FXML
    void Pen(ActionEvent event) {
        penColor = Color.BLACK;
    }
    @FXML
    void Eraser(ActionEvent event) {
        penColor = Color.WHITE;
    }


}

```

Figure 4.4: PainterController

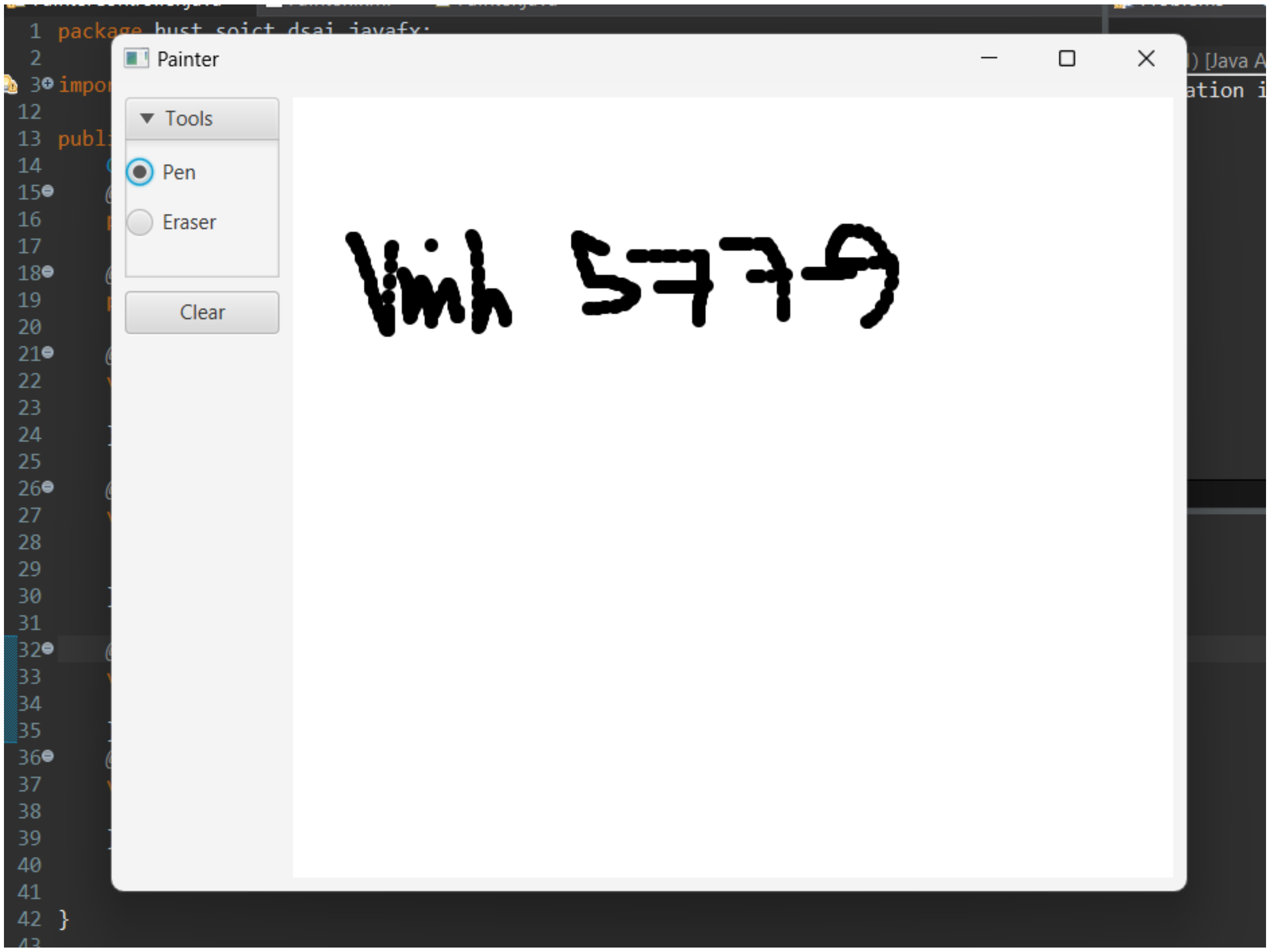


Figure 4.5: Use Pen

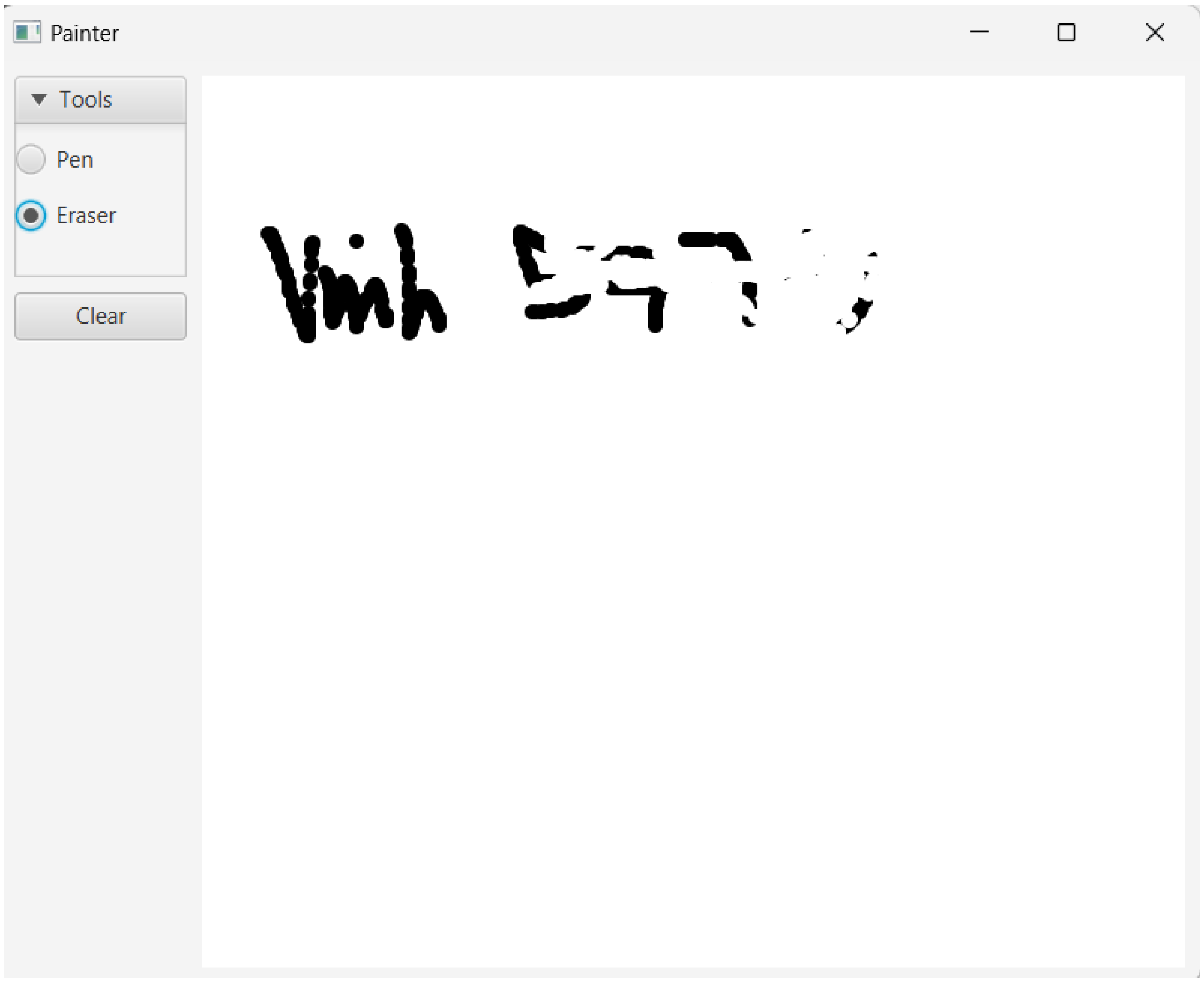


Figure 4.6: Use Eraser

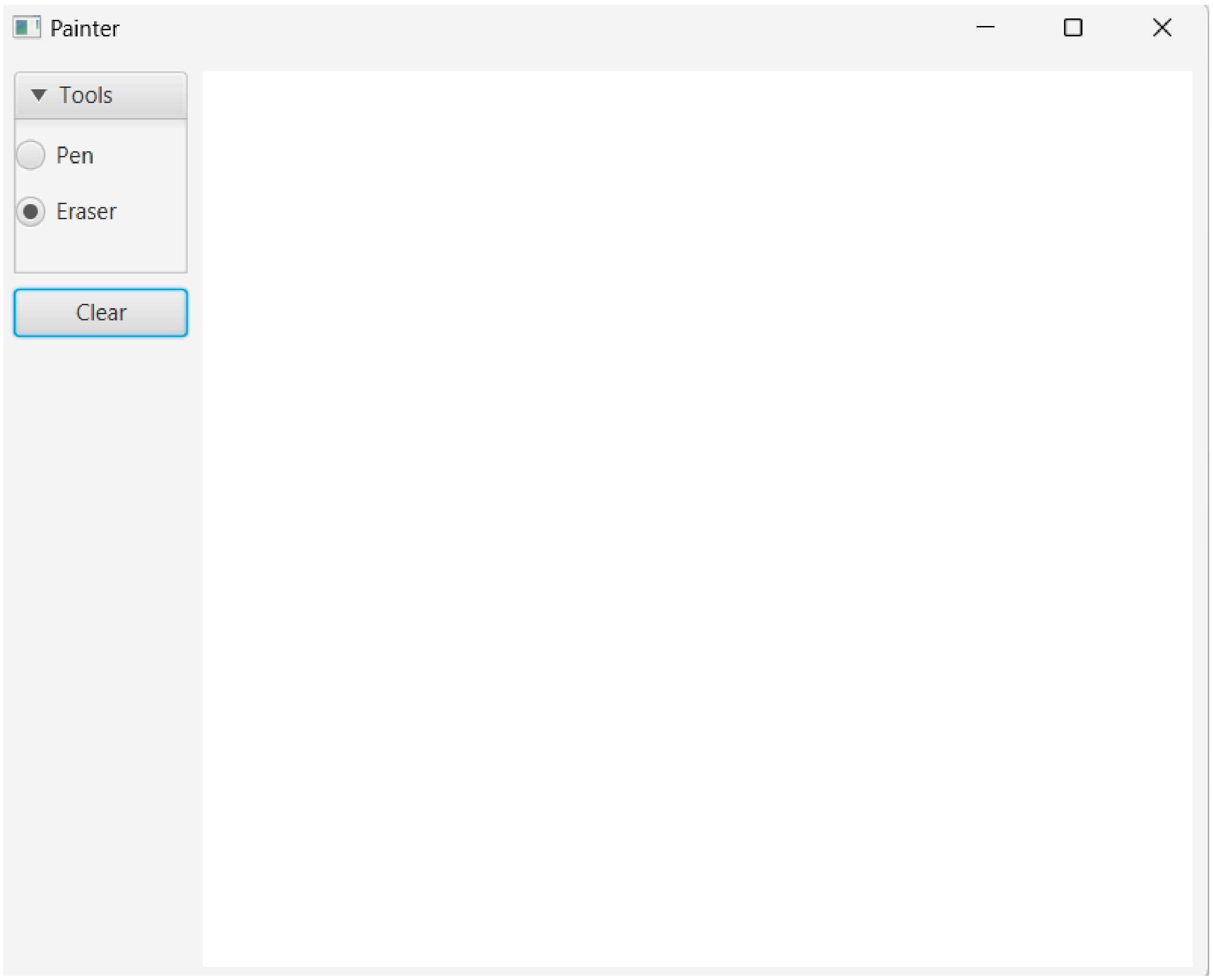
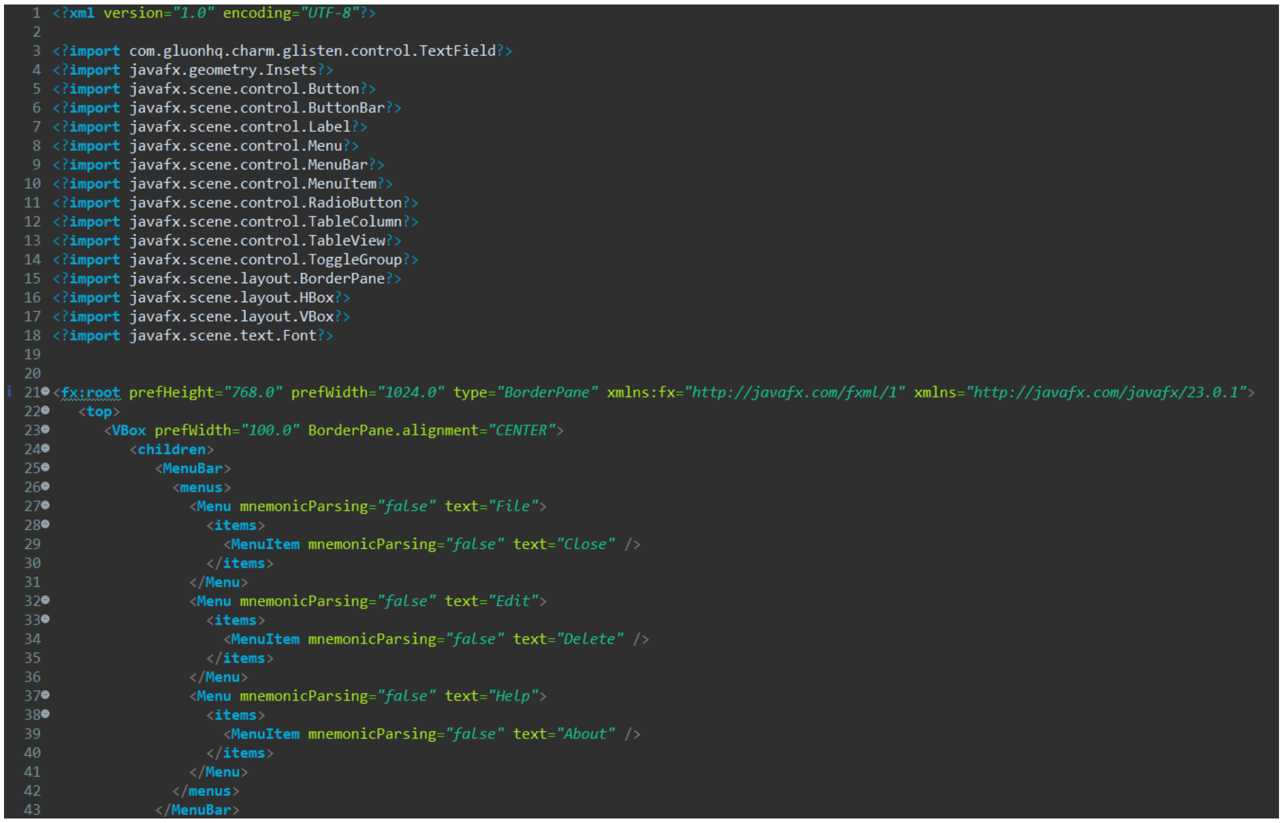


Figure 4.7: Clear button

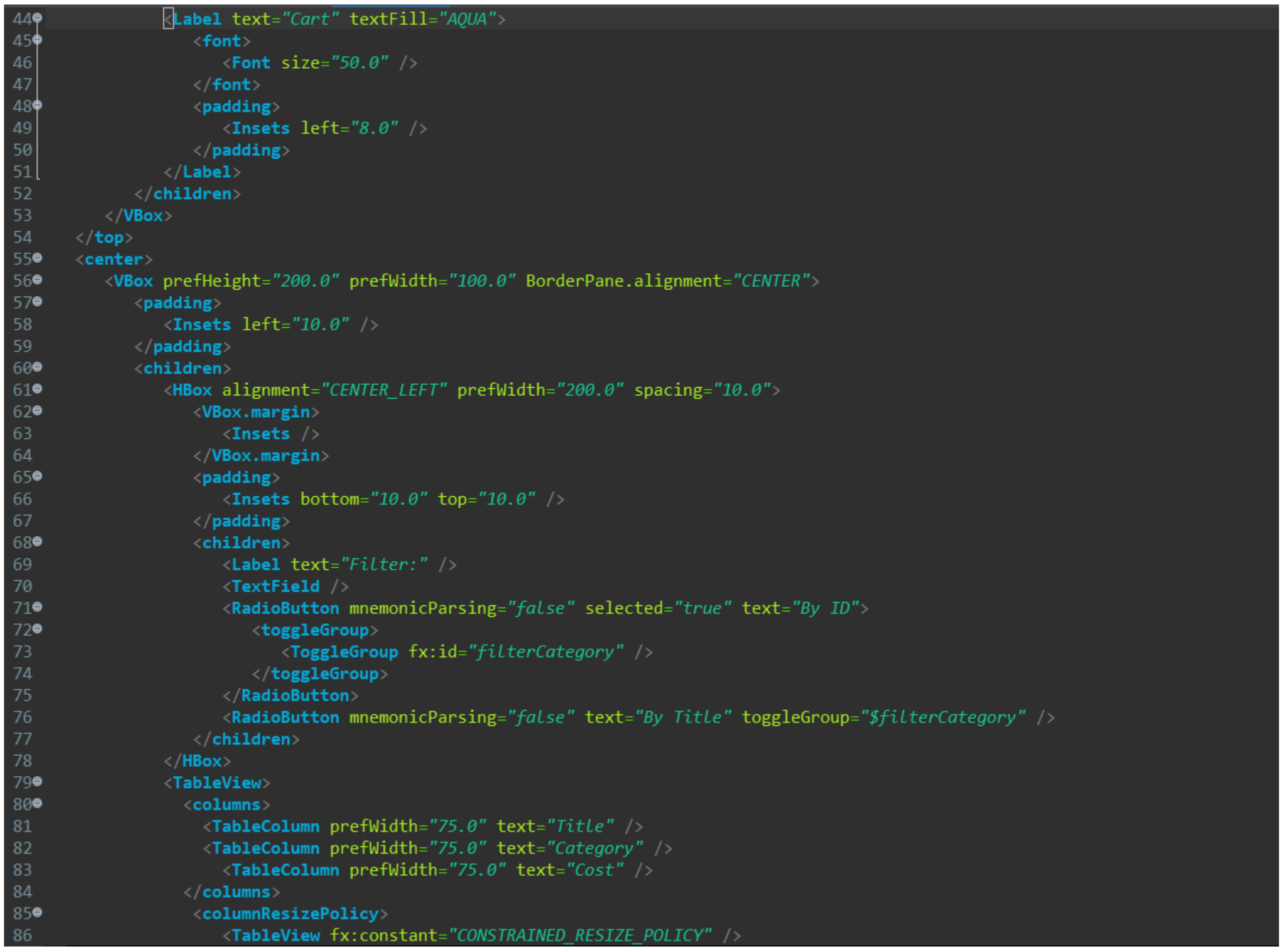
5 View Cart Screen

5.1 Create cart.fxml



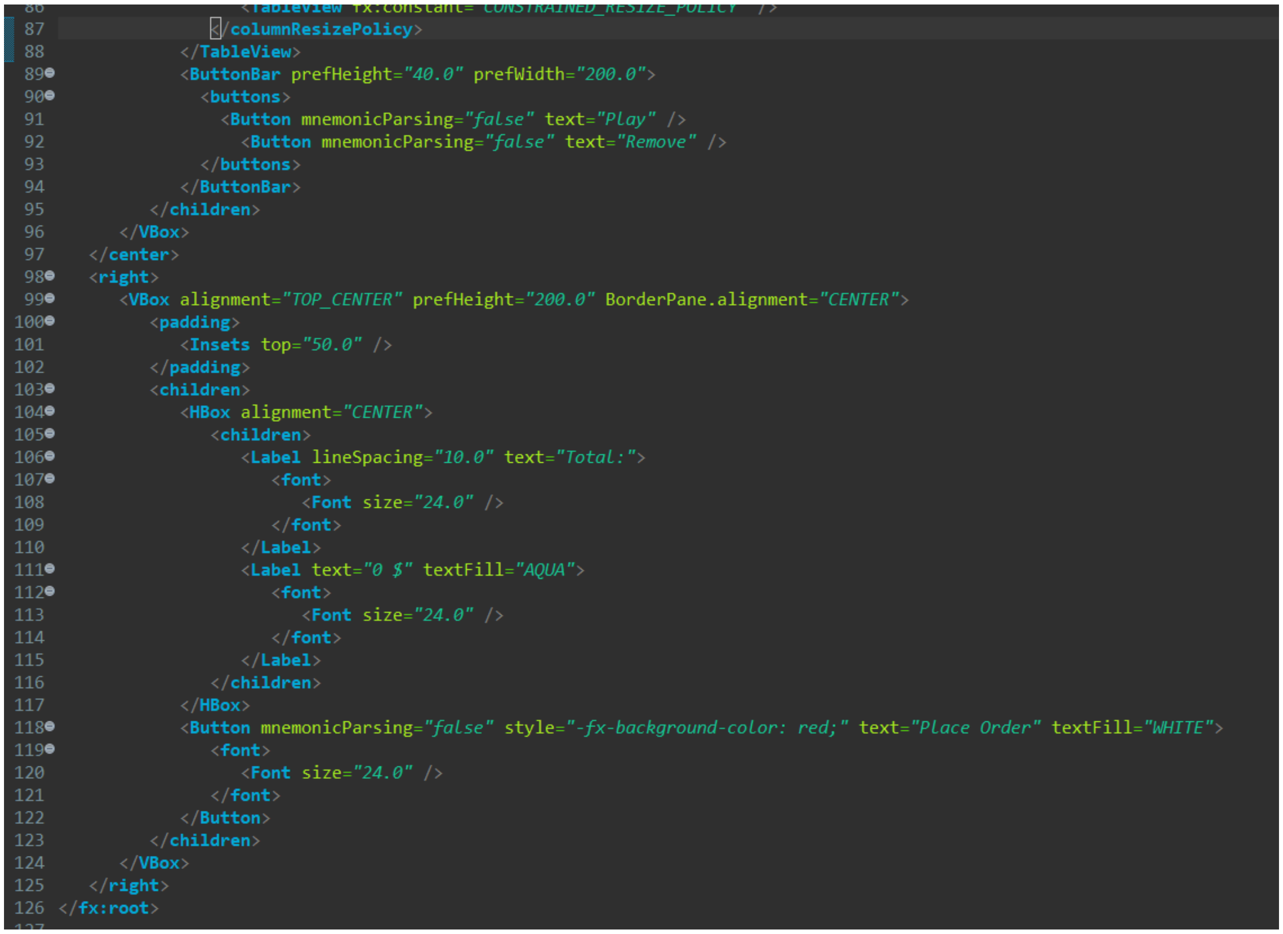
```
<?xml version="1.0" encoding="UTF-8"?>

<?import com.gluonhq.charm.glisten.control.TextField?>
<?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.ToggleGroup?>
<?import javafx.scene.layout.BorderPane?>
<?import javafx.scene.layout.HBox?>
<?import javafx.scene.layout.VBox?>
<?import javafx.scene.text.Font?>


<fx:root prefHeight="768.0" prefWidth="1024.0" type="BorderPane" xmlns:fx="http://javafx.com/fxml/1" xmlns="http://javafx.com/javafx/23.0.1">
   <top>
      <VBox prefWidth="100.0" BorderPane.alignment="CENTER">
         <children>
            <MenuBar>
              <menus>
                <Menu mnemonicParsing="false" text="File">
                  <items>
                    <MenuItem mnemonicParsing="false" text="Close" />
                  </items>
                </Menu>
                <Menu mnemonicParsing="false" text="Edit">
                  <items>
                    <MenuItem mnemonicParsing="false" text="Delete" />
                  </items>
                </Menu>
                <Menu mnemonicParsing="false" text="Help">
                  <items>
                    <MenuItem mnemonicParsing="false" text="About" />
                  </items>
                </Menu>
              </menus>
            </MenuBar>
```

Figure 5.1: Cart.fxml 1



```
                <Label text="Cart" textFill="AQUA">
                    <font>
                        <Font size="50.0" />
                    </font>
                    <padding>
                        <Insets left="8.0" />
                    </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">
                    <VBox.margin>
                        <Insets />
                    </VBox.margin>
                    <padding>
                        <Insets bottom="10.0" top="10.0" />
                    </padding>
                    <children>
                        <Label text="Filter:" />
                        <TextField />
                        <RadioButton mnemonicParsing="false" selected="true" text="By ID">
                            <toggleGroup>
                                <ToggleGroup fx:id="filterCategory" />
                            </toggleGroup>
                        </RadioButton>
                        <RadioButton mnemonicParsing="false" text="By Title" toggleGroup="$filterCategory" />
                    </children>
                </HBox>
                <TableView>
                  <columns>
                    <TableColumn prefWidth="75.0" text="Title" />
                    <TableColumn prefWidth="75.0" text="Category" />
                      <TableColumn prefWidth="75.0" text="Cost" />
                  </columns>
                   <columnResizePolicy>
                      <TableView fx:constant="CONSTRAINED_RESIZE_POLICY" />
```

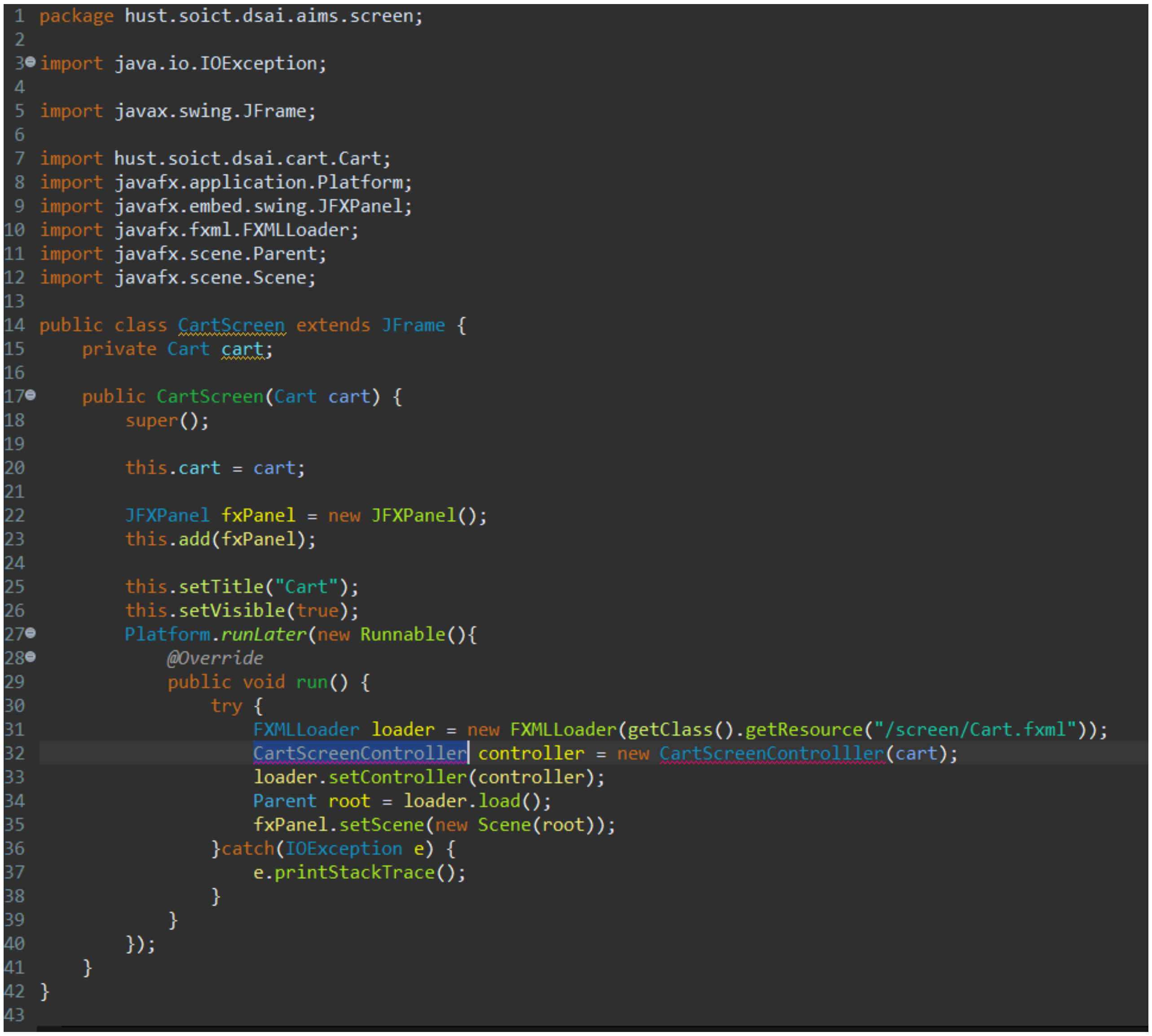
Figure 5.2: Cart.fxml 2



```
                  <TableView fx:constant="CONSTRAINED_RESIZE_POLICY" />
               </columnResizePolicy>
            </TableView>
            <ButtonBar prefHeight="40.0" prefWidth="200.0">
              <buttons>
                <Button mnemonicParsing="false" text="Play" />
                  <Button mnemonicParsing="false" 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 text="0 $" textFill="AQUA">
                     <font>
                        <Font size="24.0" />
                     </font>
                  </Label>
               </children>
            </HBox>
            <Button mnemonicParsing="false" style="-fx-background-color: red;" text="Place Order" textFill="WHITE">
               <font>
                  <Font size="24.0" />
               </font>
            </Button>
         </children>
      </VBox>
   </right>
</fx:root>
```

Figure 5.3: Cart.fxml 3

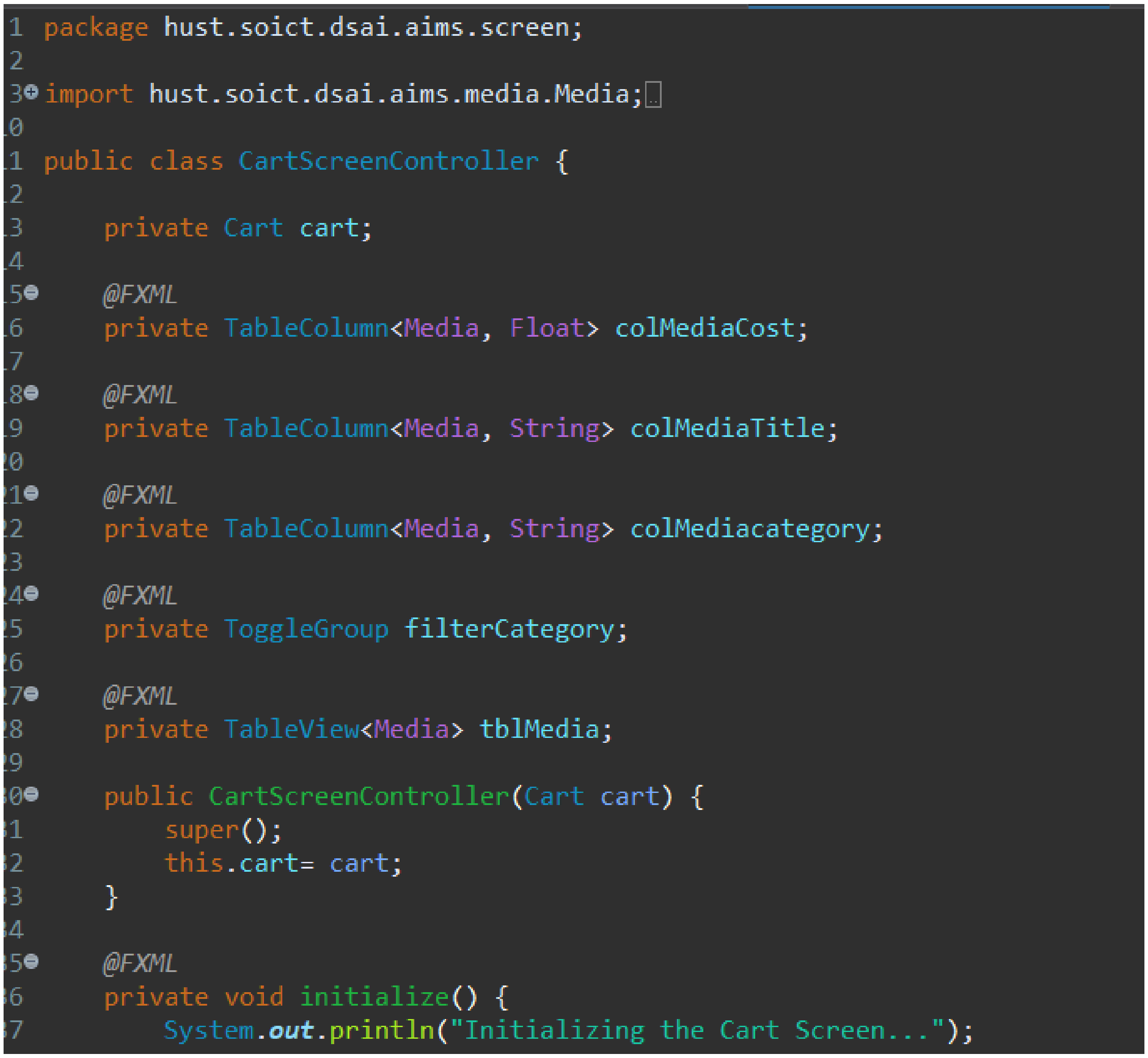
5.2 Create class CartScreen



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

import java.io.IOException;

import javax.swing.JFrame;

import hust.soict.dsai.cart.Cart;
import javafx.application.Platform;
import javafx.embed.swing.JFXPanel;
import javafx.fxml.FXMLLoader;
import javafx.scene.Parent;
import javafx.scene.Scene;

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("/screen/Cart.fxml"));
                    CartScreenController controller = new CartScreenControlller(cart);
                    loader.setController(controller);
                    Parent root = loader.load();
                    fxPanel.setScene(new Scene(root));
                }catch(IOException e) {
                    e.printStackTrace();
                }
            }
        });
    }
}
```

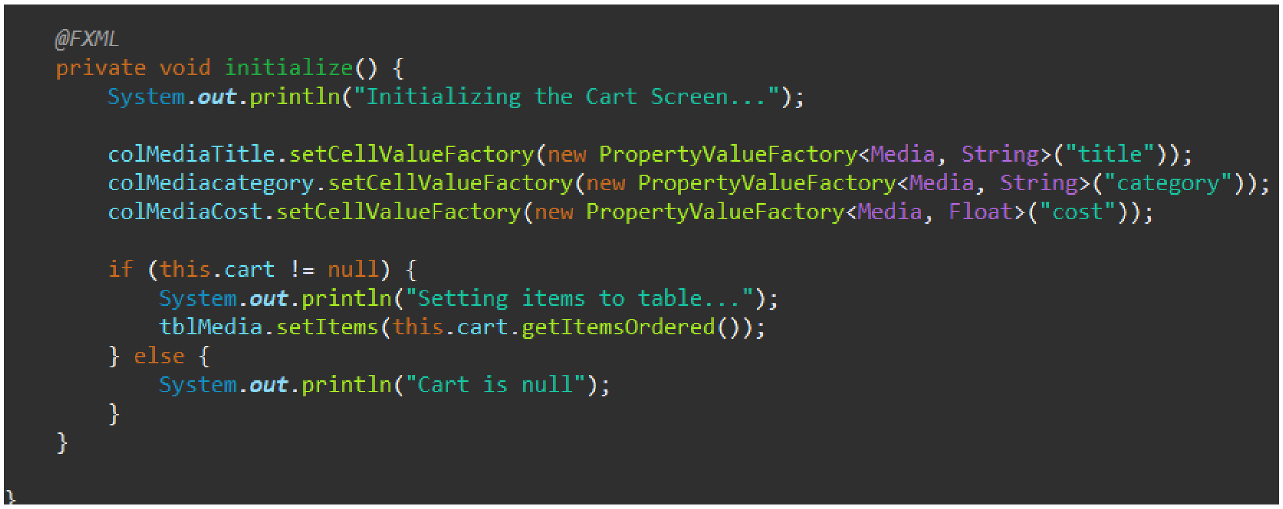
Figure 5.4: CartScreen class

5.3 Create class CartScreenController



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

import hust.soict.dsai.aims.media.Media;

public class CartScreenController {

    private Cart cart;

    @FXML
    private TableColumn<Media, Float> colMediaCost;

    @FXML
    private TableColumn<Media, String> colMediaTitle;

    @FXML
    private TableColumn<Media, String> colMediacategory;

    @FXML
    private ToggleGroup filterCategory;

    @FXML
    private TableView<Media> tblMedia;

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

    @FXML
    private void initialize() {
        System.out.println("Initializing the Cart Screen...");
```

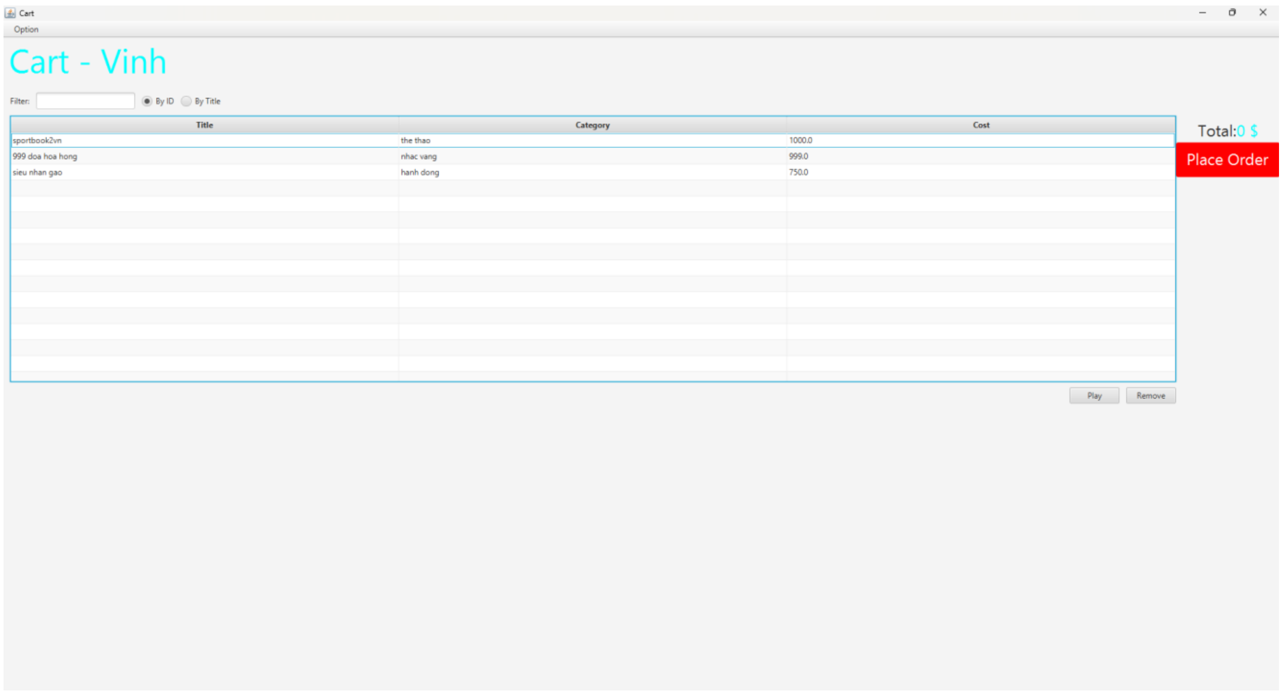
Figure 5.5: CartScreenController 1



```
    @FXML
    private void initialize() {
        System.out.println("Initializing the Cart Screen...");

        colMediaTitle.setCellValueFactory(new PropertyValueFactory<Media, String>("title"));
        colMediacategory.setCellValueFactory(new PropertyValueFactory<Media, String>("category"));
        colMediaCost.setCellValueFactory(new PropertyValueFactory<Media, Float>("cost"));

        if (this.cart != null) {
            System.out.println("Setting items to table...");
            tblMedia.setItems(this.cart.getItemsOrdered());
        } else {
            System.out.println("Cart is null");
        }
    }

}
```

Figure 5.6: CartScreenController 2

5.4 Demo

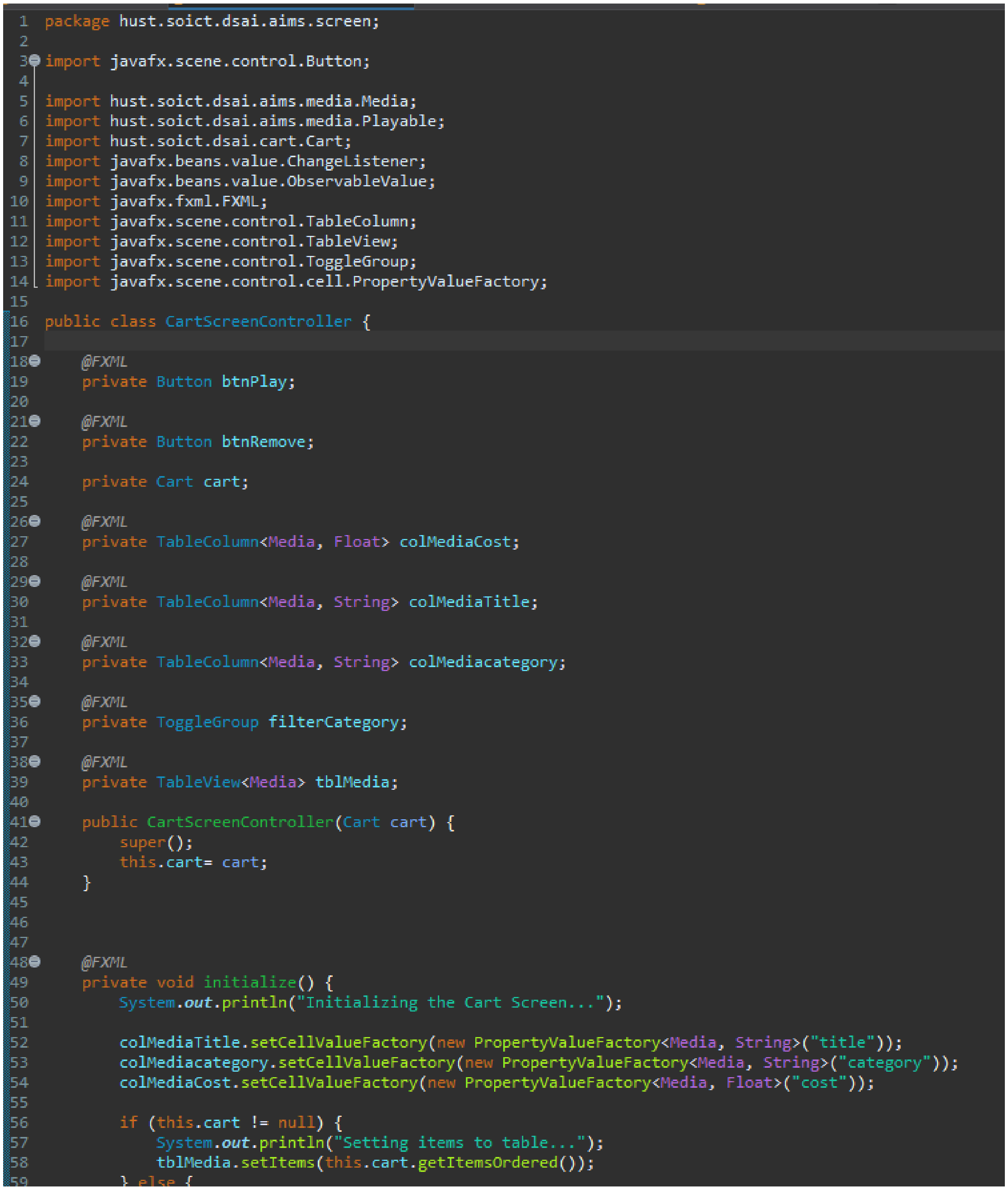


Title	Category	Cost
sportbook2vn	the thao	1000.0
999 doa hoa hong	nhac vang	999.0
sieu nhan gao	hanh dong	750.0

Figure 5.7: Demo CartScreen

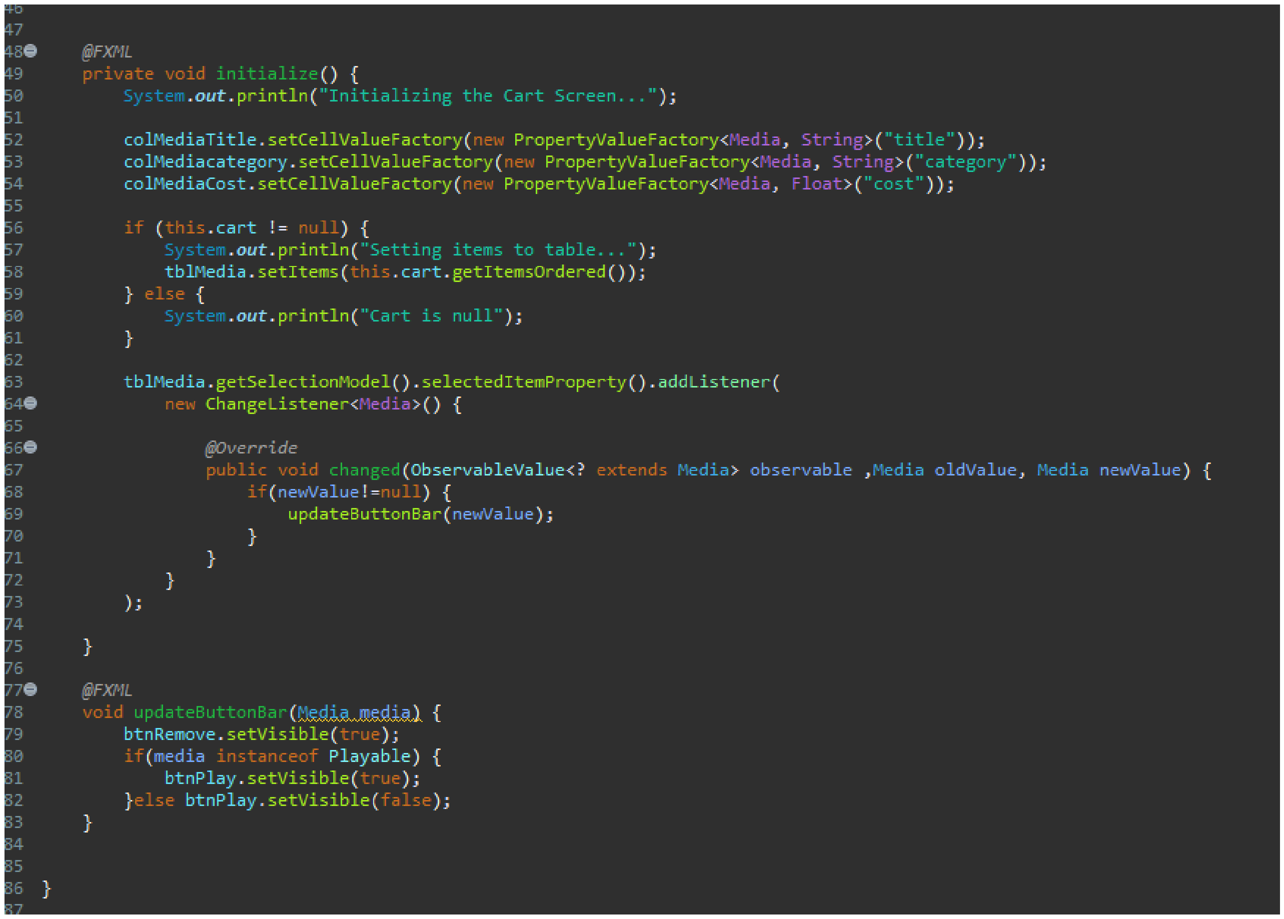
6 Updating buttons based on selected item in TableView – ChangeListener

6.1 Edit class CartScreenController



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

import javafx.scene.control.Button;

import hust.soict.dsai.aims.media.Media;
import hust.soict.dsai.aims.media.Playable;
import hust.soict.dsai.cart.Cart;
import javafx.beans.value.ChangeListener;
import javafx.beans.value.ObservableValue;
import javafx.fxml.FXML;
import javafx.scene.control.TableColumn;
import javafx.scene.control.TableView;
import javafx.scene.control.ToggleGroup;
import javafx.scene.control.cell.PropertyValueFactory;

public class CartScreenController {

    @FXML
    private Button btnPlay;

    @FXML
    private Button btnRemove;

    private Cart cart;

    @FXML
    private TableColumn<Media, Float> colMediaCost;

    @FXML
    private TableColumn<Media, String> colMediaTitle;

    @FXML
    private TableColumn<Media, String> colMediacategory;

    @FXML
    private ToggleGroup filterCategory;

    @FXML
    private TableView<Media> tblMedia;

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



    @FXML
    private void initialize() {
        System.out.println("Initializing the Cart Screen...");

        colMediaTitle.setCellValueFactory(new PropertyValueFactory<Media, String>("title"));
        colMediacategory.setCellValueFactory(new PropertyValueFactory<Media, String>("category"));
        colMediaCost.setCellValueFactory(new PropertyValueFactory<Media, Float>("cost"));

        if (this.cart != null) {
            System.out.println("Setting items to table...");
            tblMedia.setItems(this.cart.getItemsOrdered());
        } else {
```

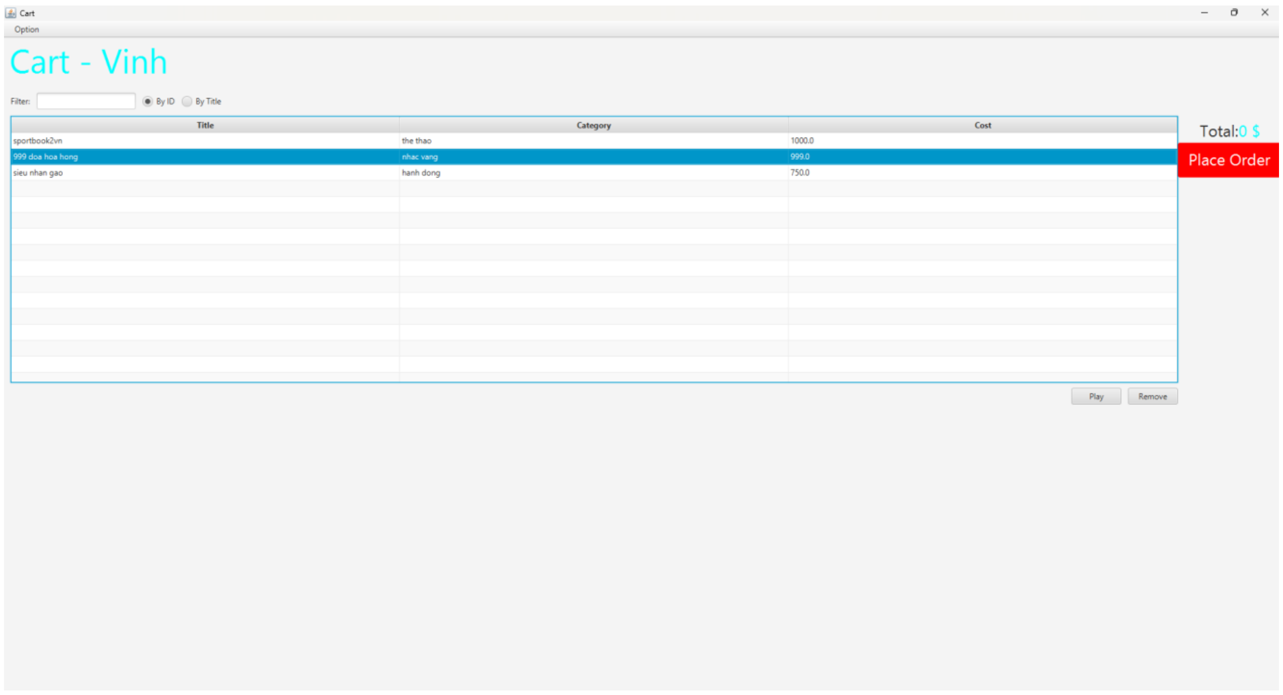
Figure 6.1: CartScreenController 1



```java

    @FXML
    private void initialize() {
        System.out.println("Initializing the Cart Screen...");

        colMediaTitle.setCellValueFactory(new PropertyValueFactory<Media, String>("title"));
        colMediacategory.setCellValueFactory(new PropertyValueFactory<Media, String>("category"));
        colMediaCost.setCellValueFactory(new PropertyValueFactory<Media, Float>("cost"));

        if (this.cart != null) {
            System.out.println("Setting items to table...");
            tblMedia.setItems(this.cart.getItemsOrdered());
        } else {
            System.out.println("Cart is null");
        }

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

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

    }

    @FXML
    void updateButtonBar(Media media) {
        btnRemove.setVisible(true);
        if(media instanceof Playable) {
            btnPlay.setVisible(true);
        }else btnPlay.setVisible(false);
    }


}
```

Figure 6.2: CartScreenController 2

6.2 Demo



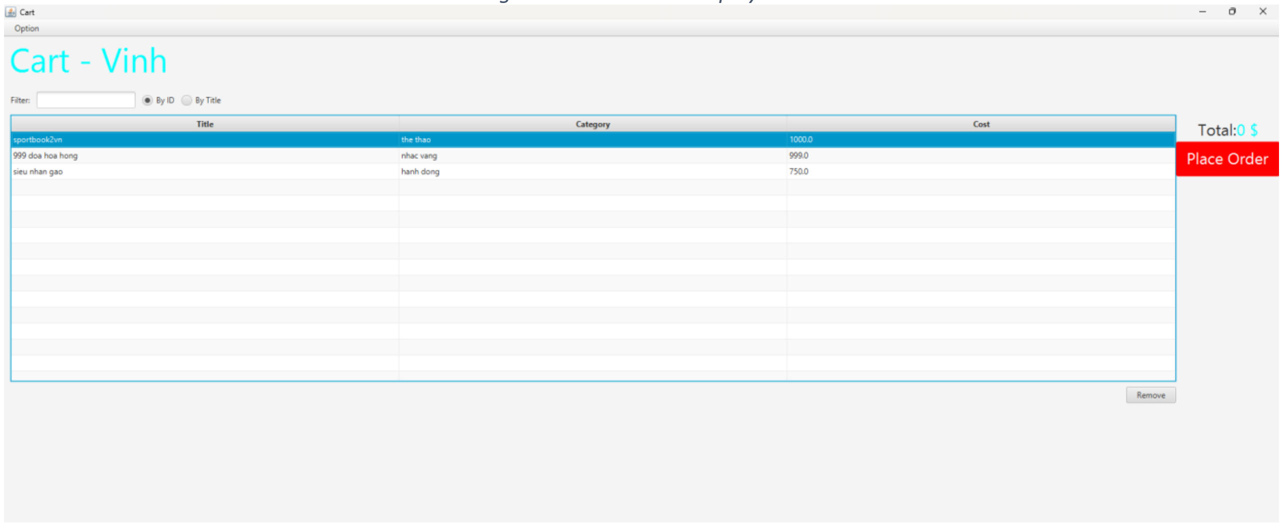
Cart - Vinh

Filter: [● By ID] ○ By Title

Title	Category	Cost
sportbook2vn	the thao	1000.0
999 doa hoa hong	nhac vang	999.0
sieu nhan gao	hanh dong	750.0

Total:0 $

Figure 6.3: Demo media playable



Cart - Vinh

Filter: [✓ By ID] By Title

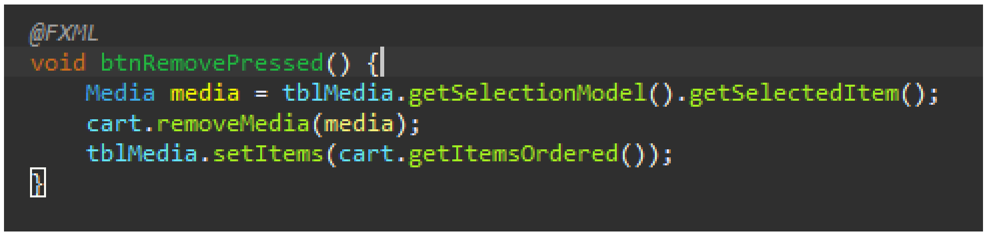
Title	Category	Cost
sportbook2vn	the thao	1000.0
999 doa hoa hong	nhac vang	999.0
sieu nhan gao	hanh dong	750.0

Total:0 $

Figure 6.4: Demo media unplayable

7 Deleting a media

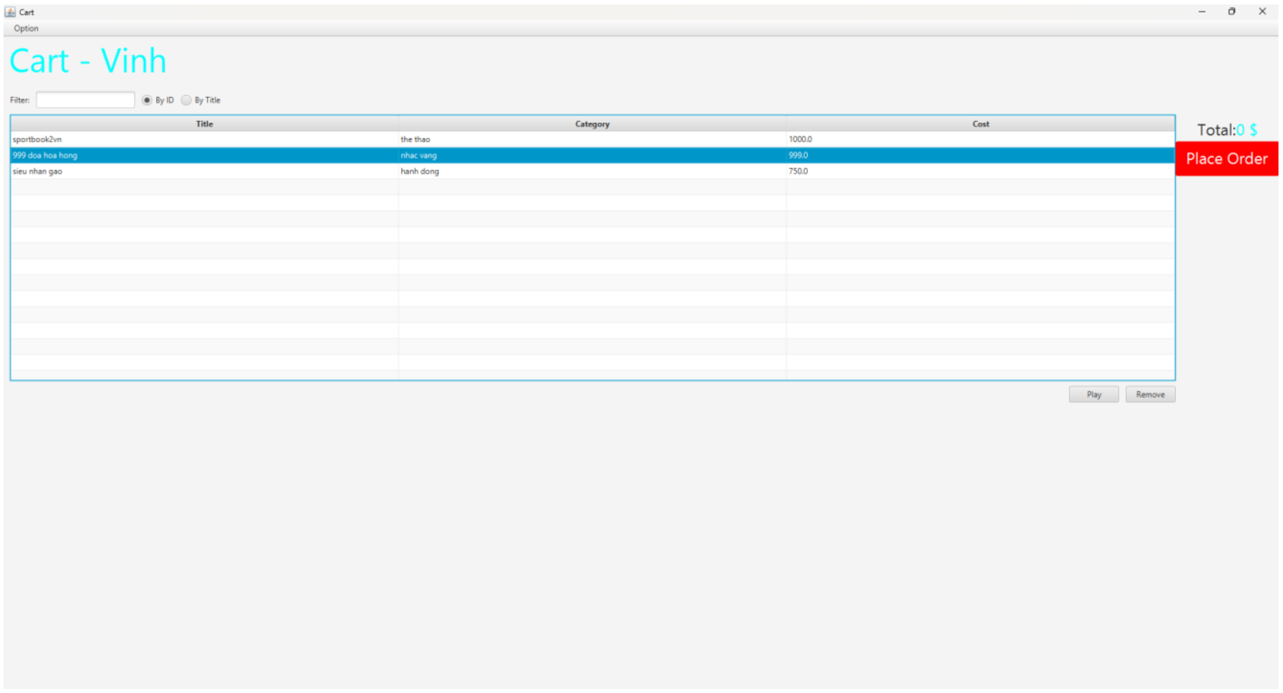
7.1 Code



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

Figure 7.1: btnRemovePressed Method

7.2 Demo



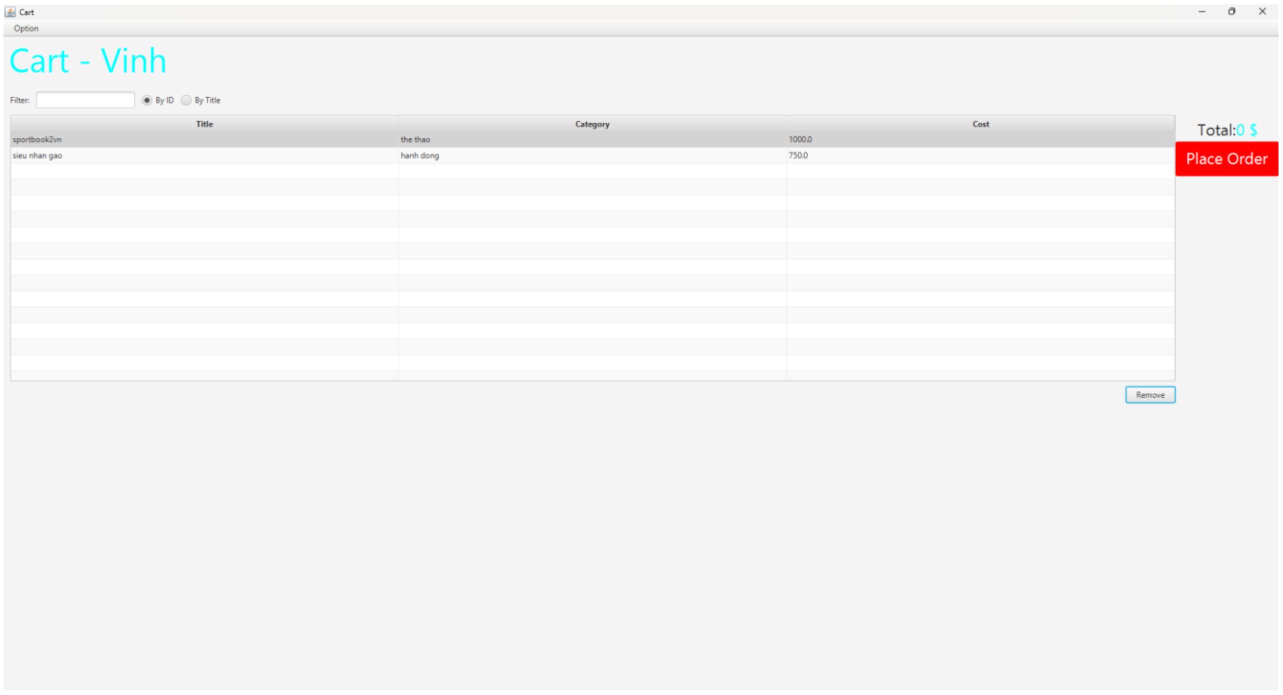
Cart - Vinh

Filter: [✓ By ID] By Title

Title	Category	Cost
sportbook2vn	the thao	1000.0
999 doa hoa hong	nhac vang	999.0
sieu nhan gao	hanh dong	750.0

Total:0 $

Figure 7.2: button Remove



Cart - Vinh

Filter: [✓ By ID] By Title

Title	Category	Cost
sportbook2vn	the thao	1000.0
sieu nhan gao	hanh dong	750.0

Total:0 $

Figure 7.3: button Remove

8 Complete the Aims GUI application

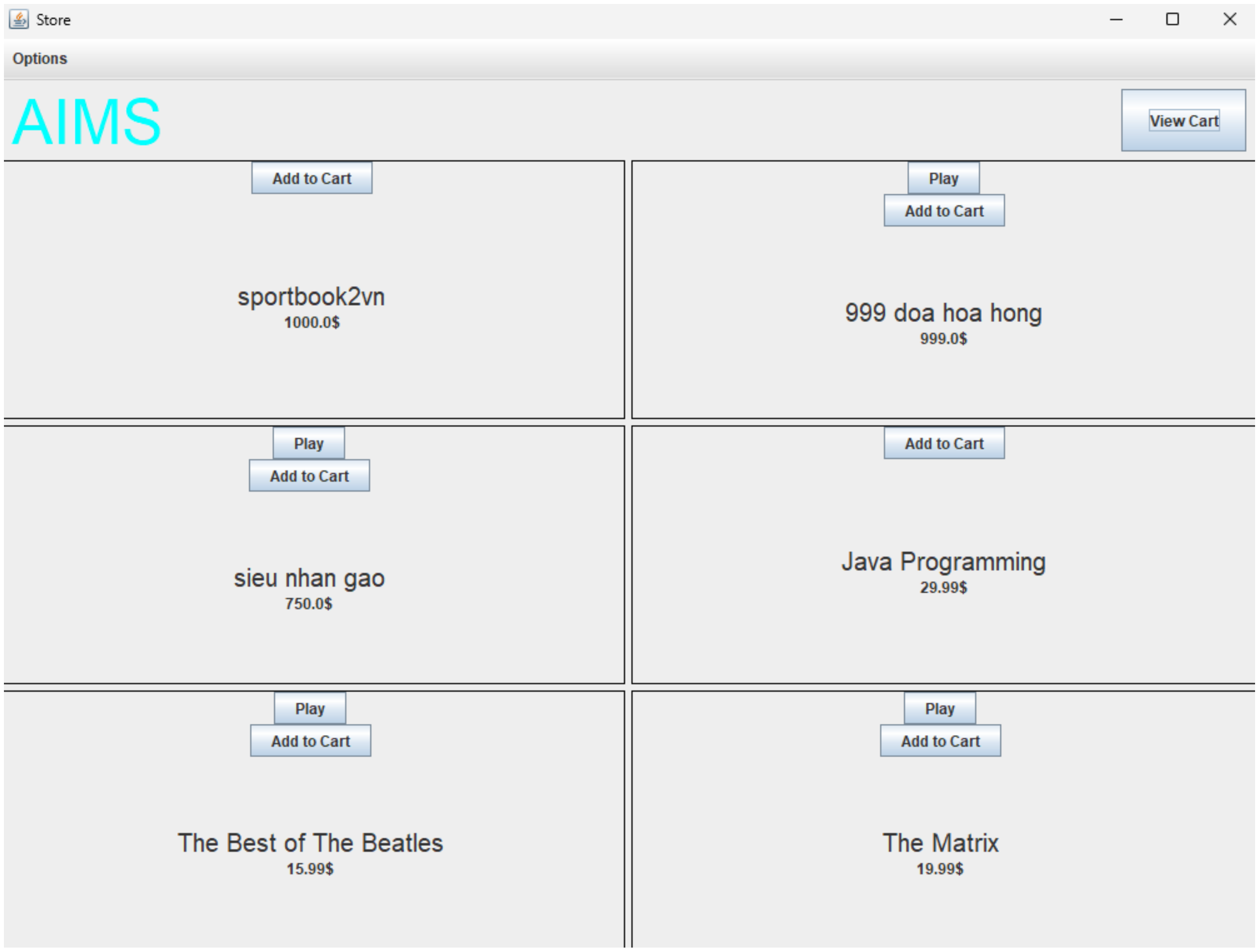
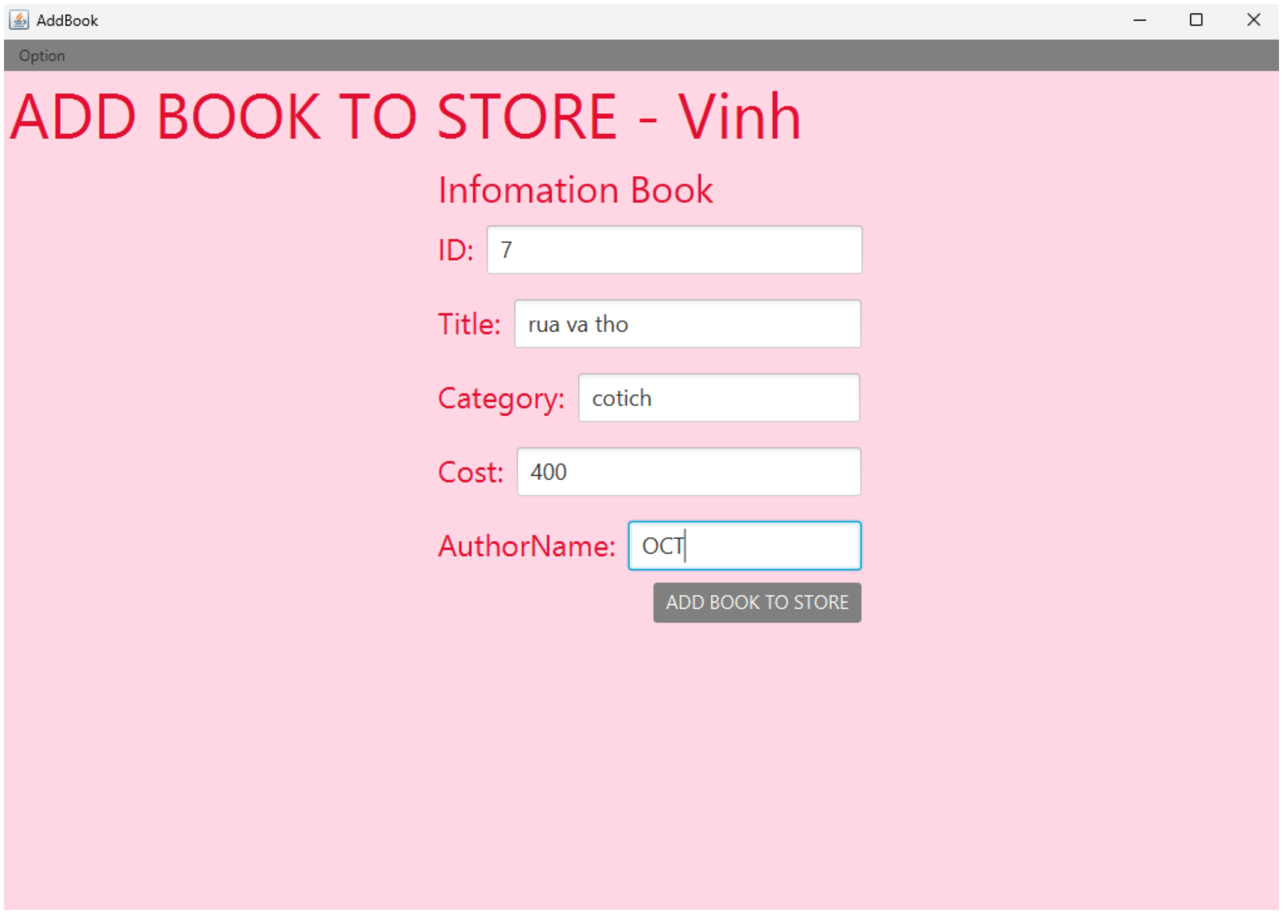


Figure 8.1: Store before add book



ID: 7

Title: rua va tho

Category: cotich

Cost: 400

AuthorName: OCT

Figure 8.2: Add book

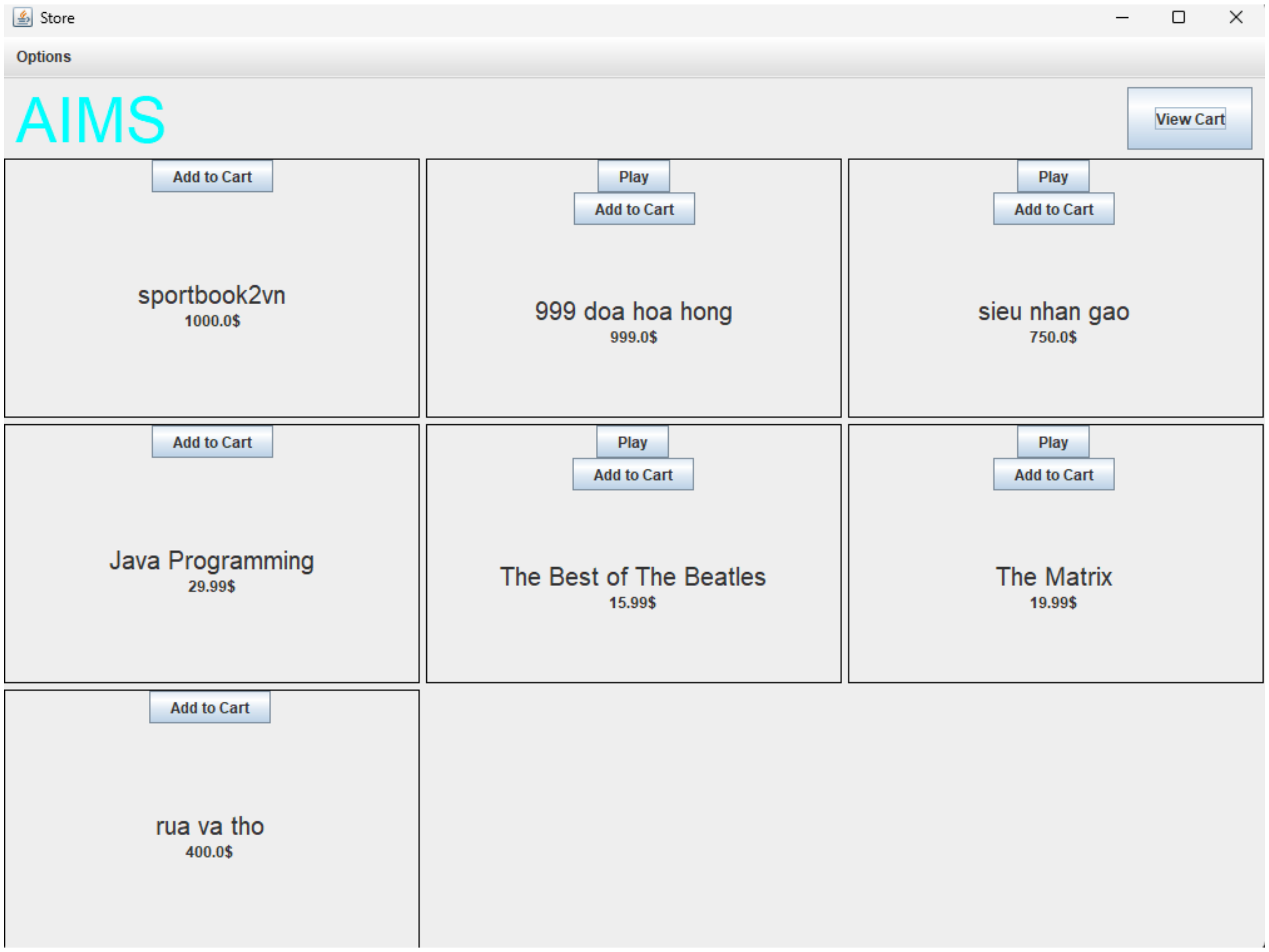
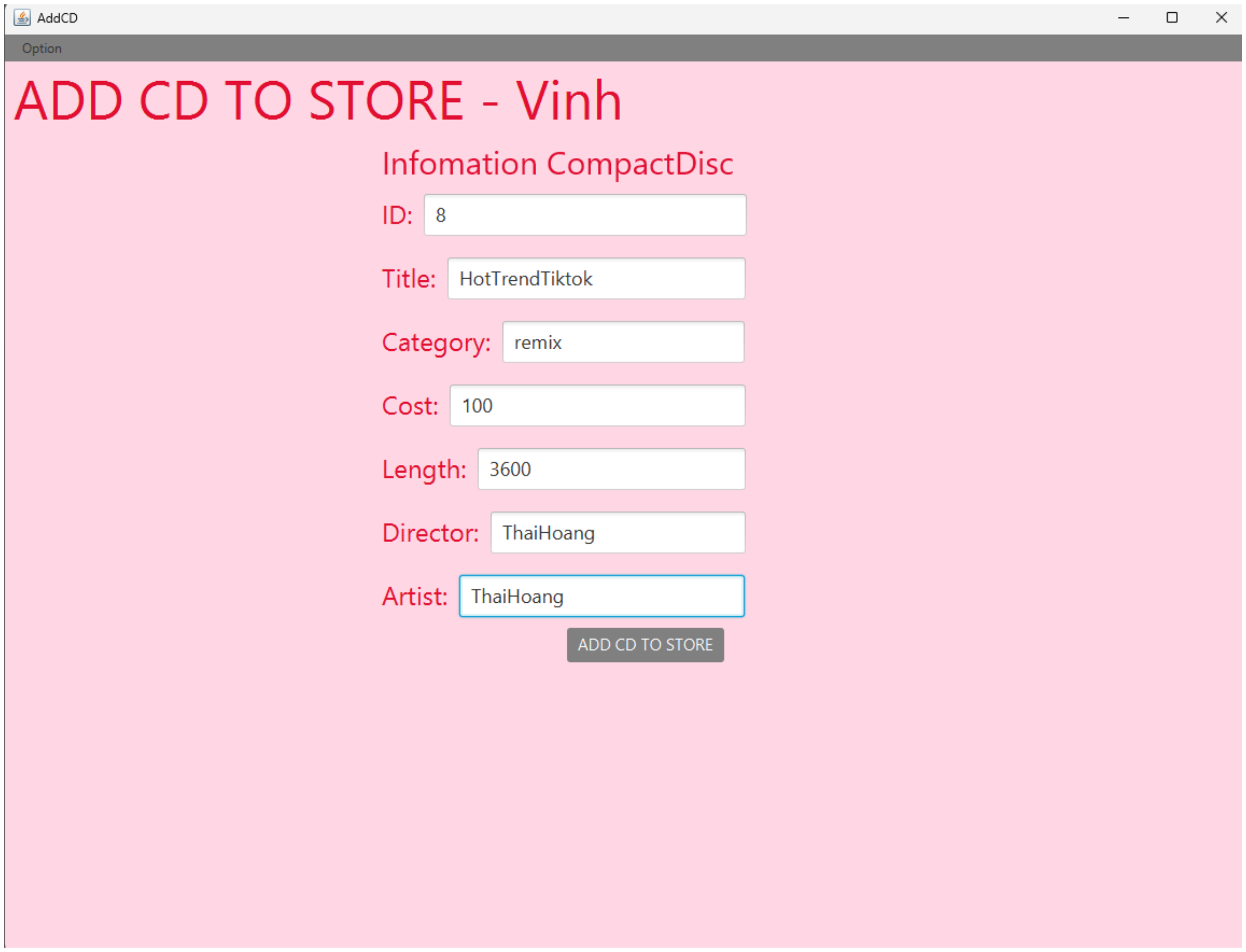


Figure 8.3: Store after add book



ADD CD TO STORE - Vinh

Infomation CompactDisc

ID: 8

Title: HotTrendTiktok

Category: remix

Cost: 100

Length: 3600

Director: ThaiHoang

Artist: ThaiHoang

Figure 8.4: Add CD

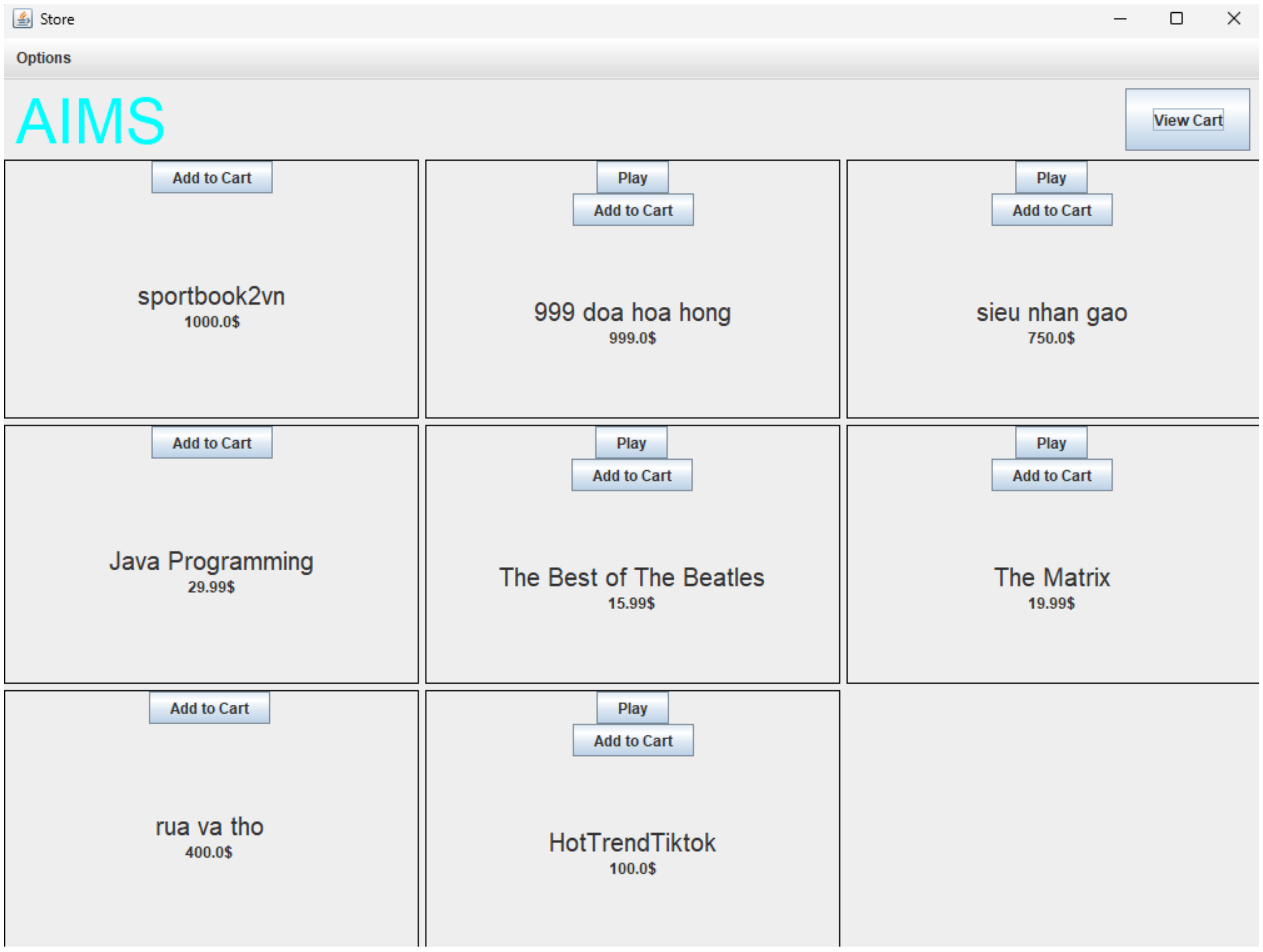
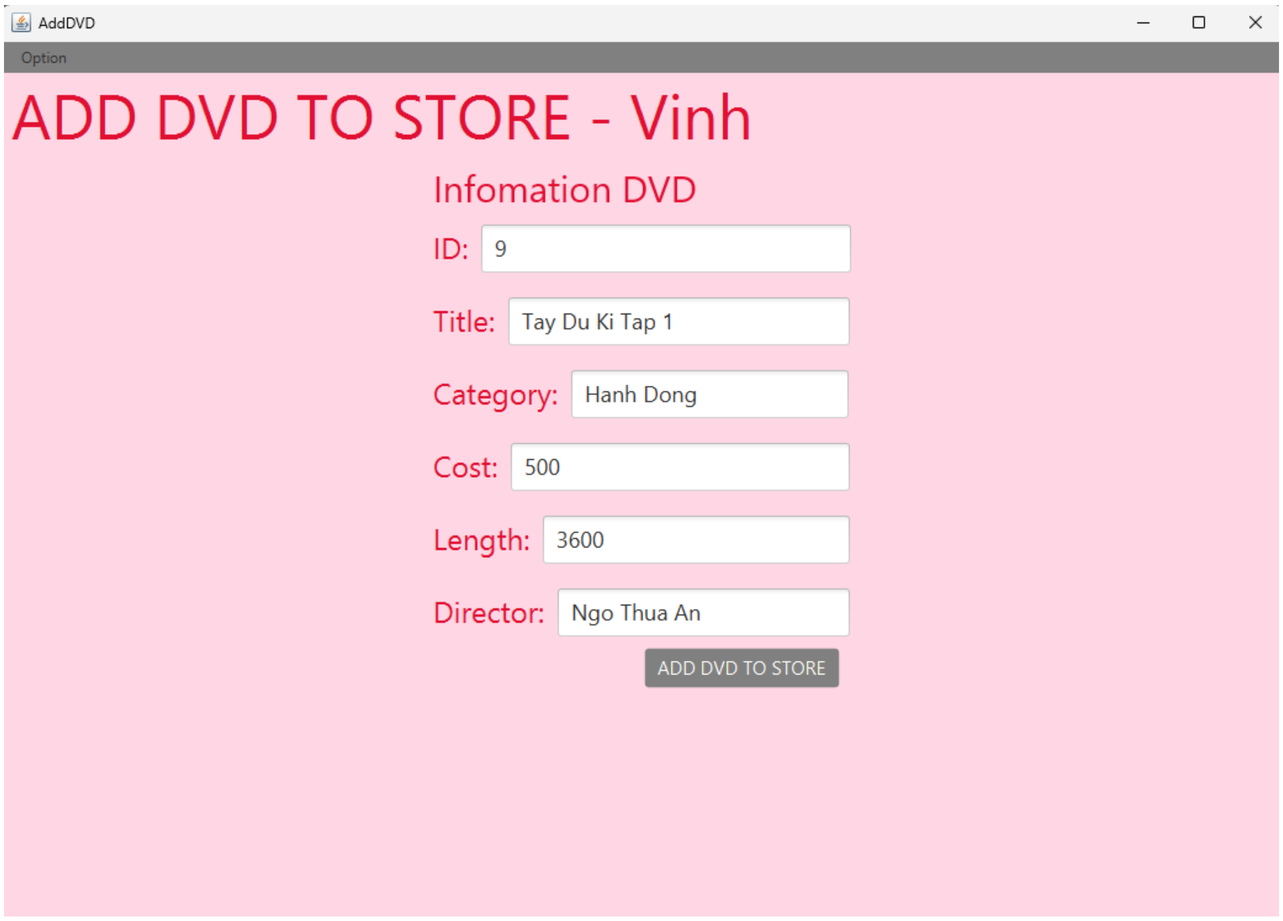


Figure 8.5: Store after add CD



ID: 9

Title: Tay Du Ki Tap 1

Category: Hanh Dong

Cost: 500

Length: 3600

Director: Ngo Thua An

Figure 8.6 Add DVD

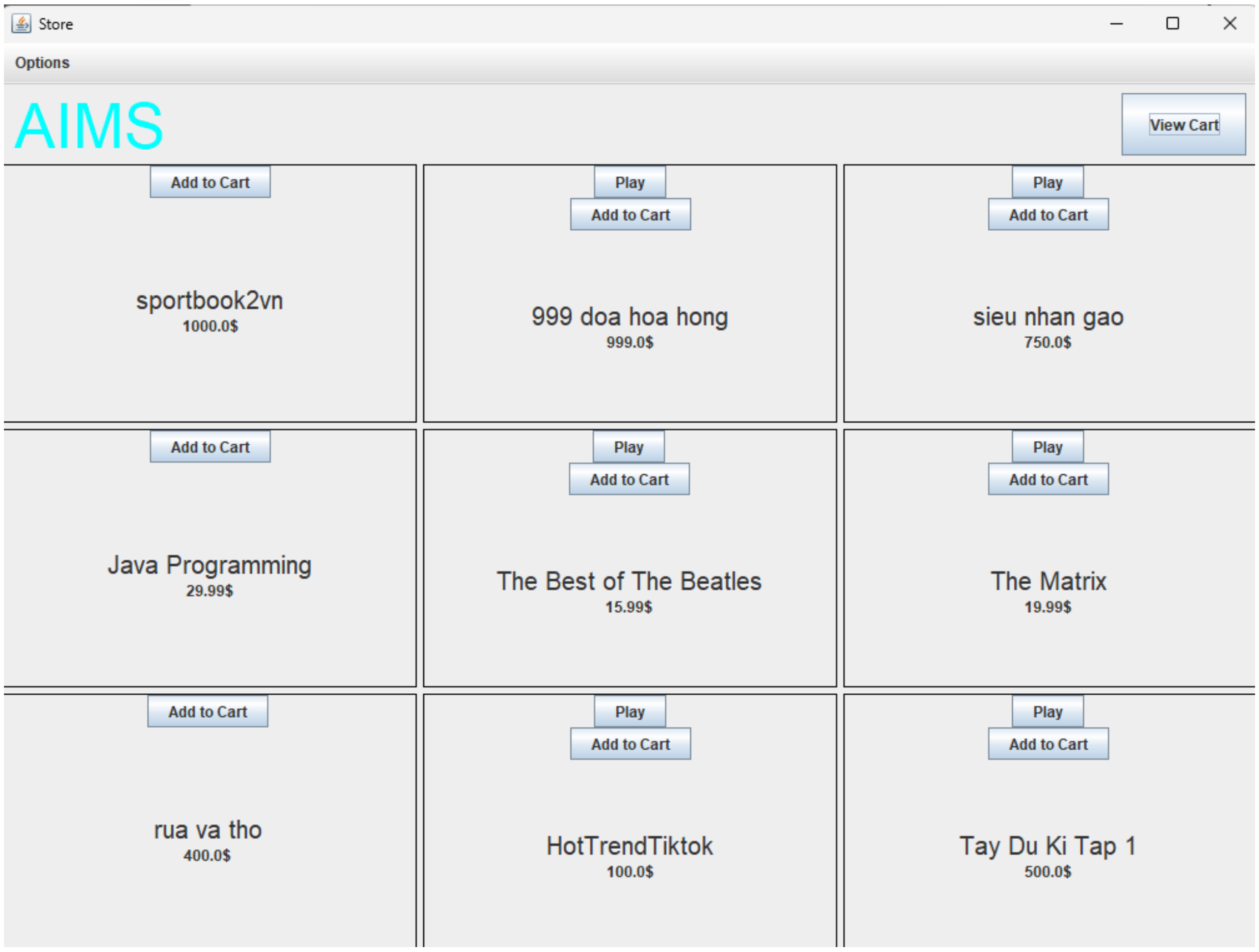
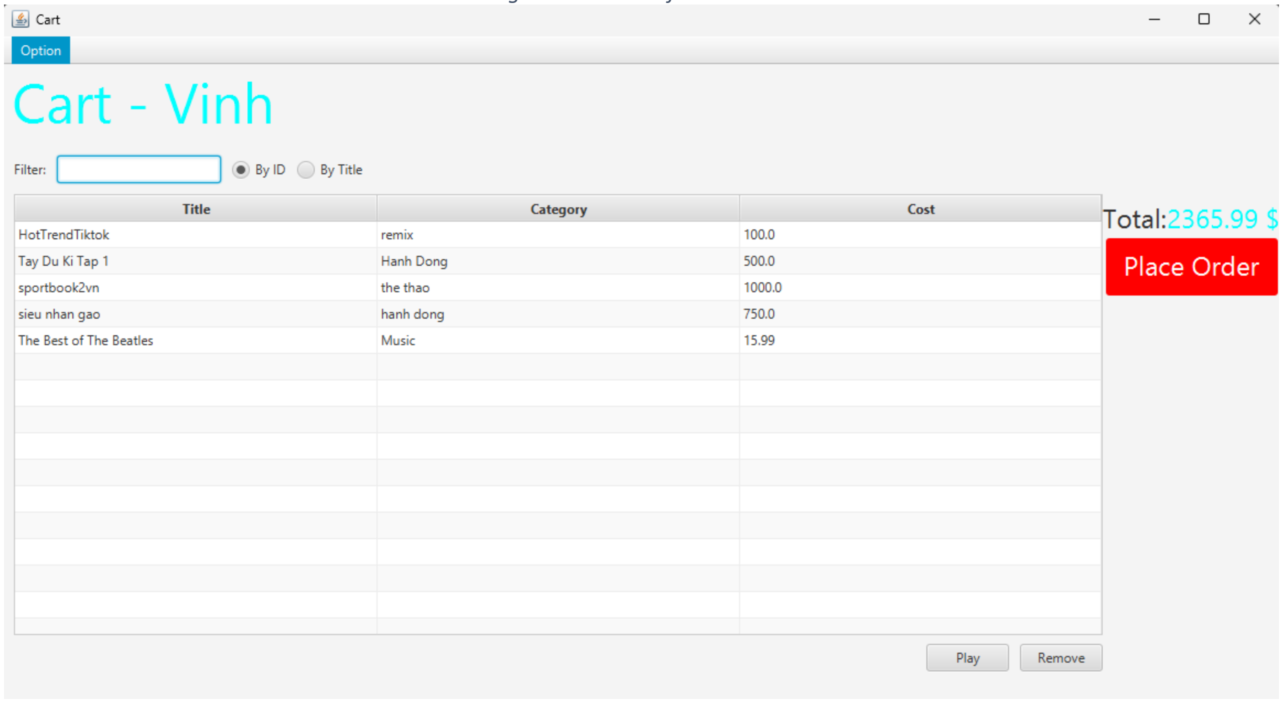
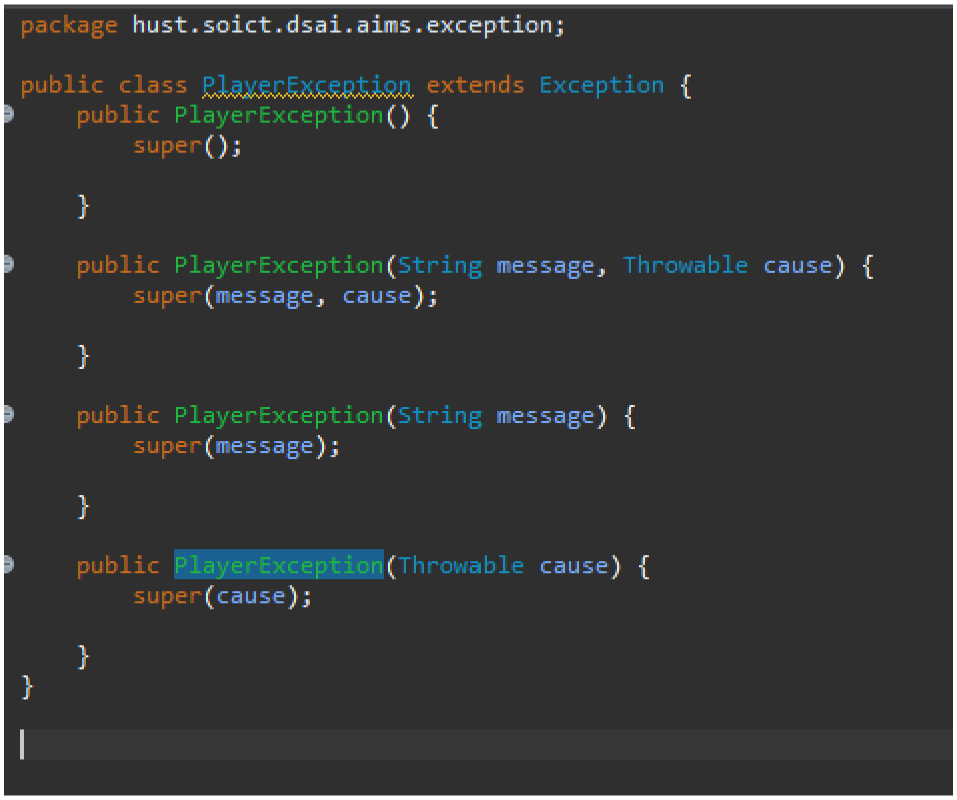


Figure 8.7: Store after add DVD



Title	Category	Cost
HotTrendTiktok	remix	100.0
Tay Du Ki Tap 1	Hanh Dong	500.0
sportbook2vn	the thao	1000.0
sieu nhan gao	hanh dong	750.0
The Best of The Beatles	Music	15.99

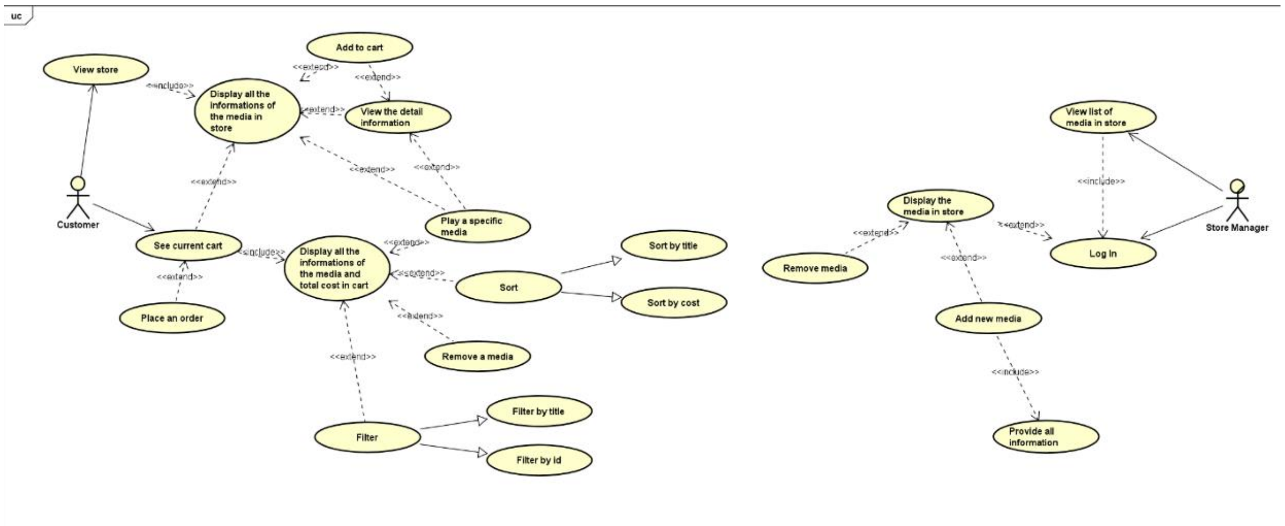
Figure 8.8: Cart



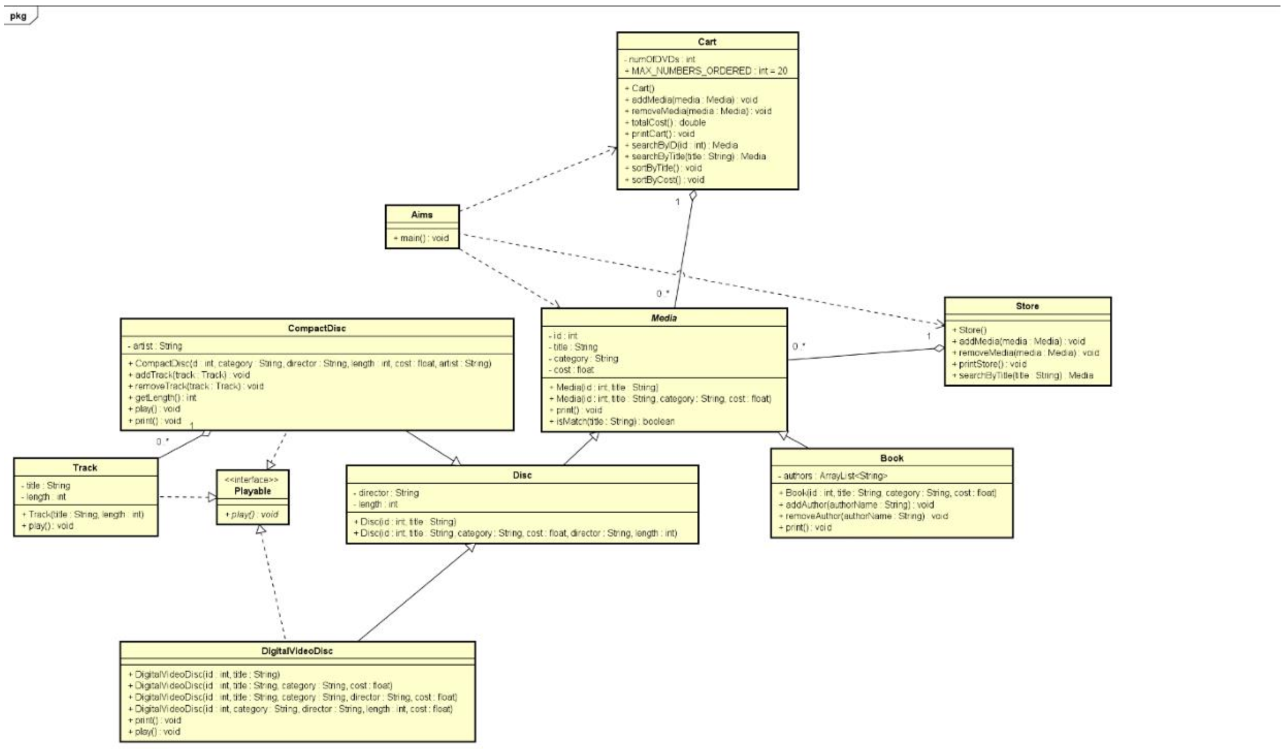
```java
package hust.soict.dsai.aims.exception;

public class PlayerException extends Exception {
	public PlayerException() {
		super();

	}

	public PlayerException(String message, Throwable cause) {
		super(message, cause);

	}

	public PlayerException(String message) {
		super(message);

	}

	public PlayerException(Throwable cause) {
		super(cause);

	}
}
```

Figure 8.9: Exception

9 Use case Diagram



10 Class Diagram


pkg
Cart
- numOfDVDs : int
+ MAX_NUMBERS_ORDERED : int = 20
+ Cart()
+ addMedia(media : Media) : void
+ removeMedia(media : Media) : void
+ totalCost() : double
+ printCart() : void
+ searchByID(id : int) : Media
+ searchByTitle(title : String) : Media
+ sortByTitle() : void
+ sortByCost() : void
Aims
+ main() : void
1
0..*
Media
- id : int
- title : String
- category : String
- cost : float
+ Media(id : int, title : String)
+ Media(id : int, title : String, category : String, cost : float)
+ print() : void
+ isMatch(title : String) : boolean
0..*
1
Store
+ Store()
+ addMedia(media : Media) : void
+ removeMedia(media : Media) : void
+ printStore() : void
+ searchByTitle(title : String) : Media
CompactDisc
- artist : String
+ CompactDisc(id : int, category : String, director : String, length : int, cost : float, artist : String)
+ addTrack(track : Track) : void
+ removeTrack(track : Track) : void
+ getLength() : int
+ play() : void
+ print() : void
1
0..*
Track
- title : String
- length : int
+ Track(title : String, length : int)
+ play() : void
<<interface>>
Playable
+ play() : void
Disc
- director : String
- length : int
+ Disc(id : int, title : String)
+ Disc(id : int, title : String, category : String, cost : float, director : String, length : int)
Book
- authors : ArrayList<String>
+ Book(id : int, title : String, category : String, cost : float)
+ addAuthor(authorName : String) : void
+ removeAuthor(authorName : String) : void
+ print() : void
DigitalVideoDisc
+ DigitalVideoDisc(id : int, title : String)
+ DigitalVideoDisc(id : int, title : String, category : String, cost : float)
+ DigitalVideoDisc(id : int, title : String, category : String, director : String, cost : float)
+ DigitalVideoDisc(id : int, category : String, director : String, length : int, cost : float)
+ print() : void
+ play() : void