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

BÁO CÁO THỰC HÀNH

IT1130-744528-2024.1

BÀI THỰC HÀNH 05

Họ và tên sv: Nguyễn Trọng Hinh

Lớp: **K67-ITE6**

GVHD: Lê Thị Hoa

TA: **Đặng Mạnh Cường**

Hà Nội 12/2024

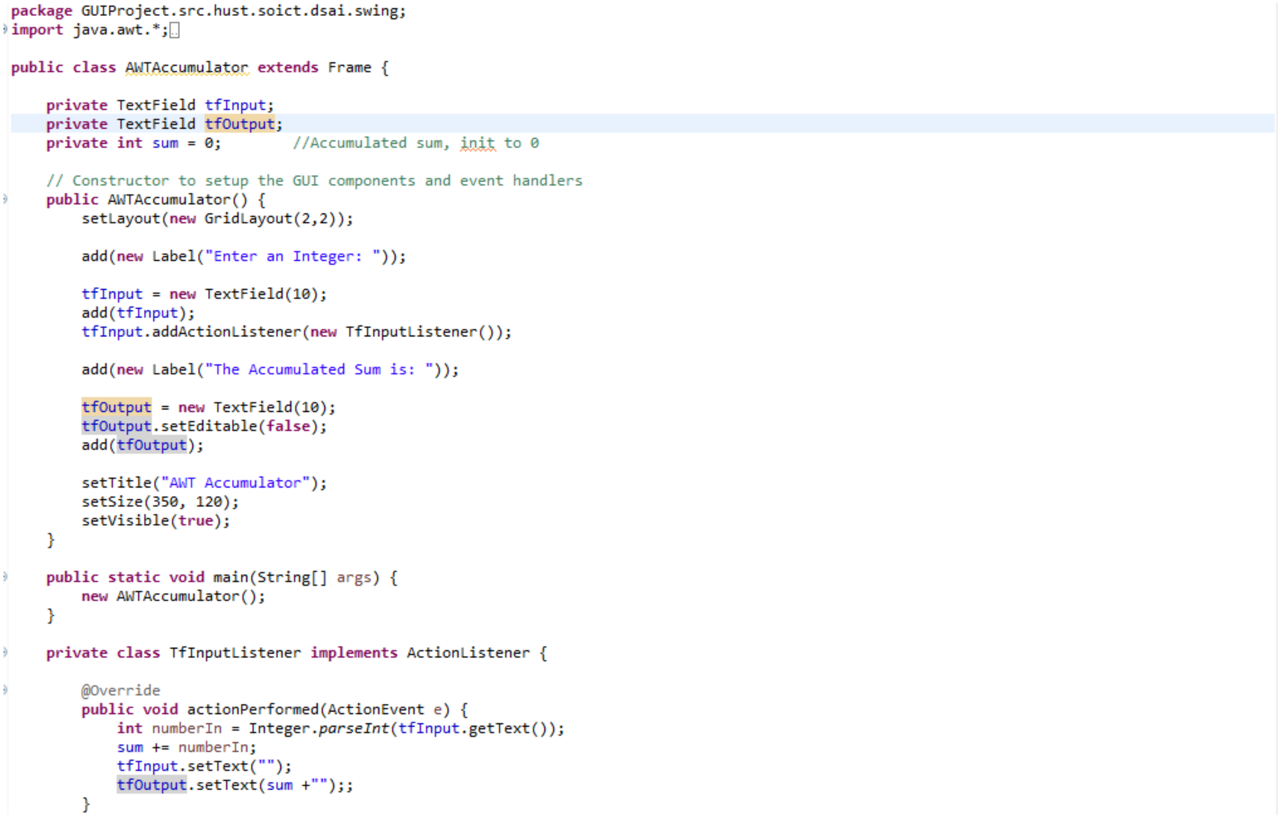
BÁO CÁO THỰC HÀNH LAB 5
LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG

Nội dung

Danh mục hình ảnh :

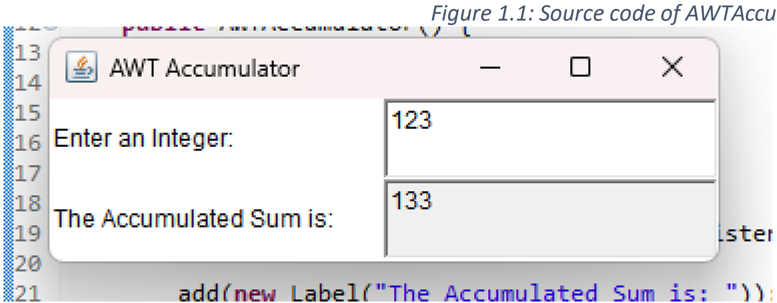
1. Swing components

1.1 AWTAccumulator



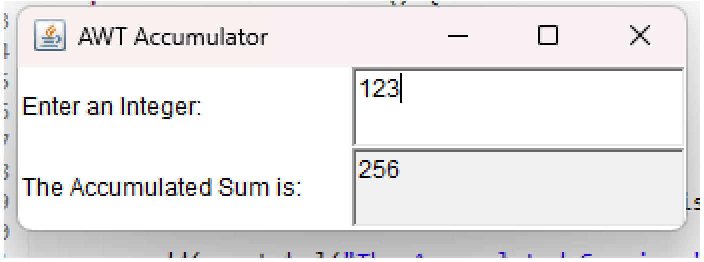
```
package GUIProject.src.hust.soict.dsai.swing;
import java.awt.*;

public class AWTAccumulator extends Frame {

    private TextField tfInput;
    private TextField tfOutput;
    private int sum = 0;        //Accumulated sum, init to 0

    // Constructor to setup the GUI components and event handlers
    public AWTAccumulator() {
        setLayout(new GridLayout(2,2));

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

        tfInput = new TextField(10);
        add(tfInput);
        tfInput.addActionListener(new TfInputListener());

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

        tfOutput = new TextField(10);
        tfOutput.setEditable(false);
        add(tfOutput);

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

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

    private class TfInputListener implements ActionListener {

        @Override
        public void actionPerformed(ActionEvent e) {
            int numberIn = Integer.parseInt(tfInput.getText());
            sum += numberIn;
            tfInput.setText("");
            tfOutput.setText(sum +"");;
        }
```

Figure 1.1: Source code of AWTAccumulator



Enter an Integer: 123

The Accumulated Sum is: 133

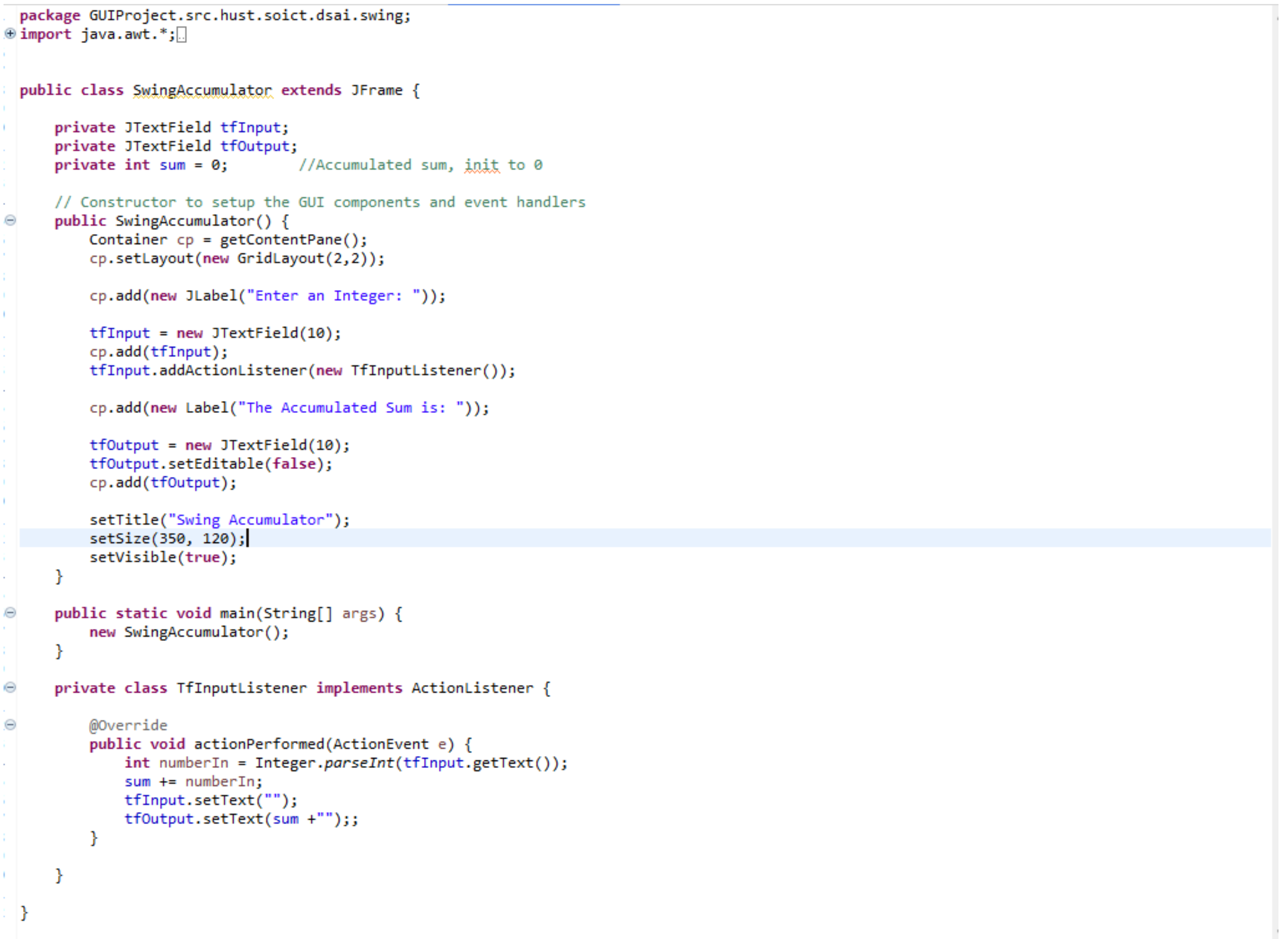


Enter an Integer: 123

The Accumulated Sum is: 256

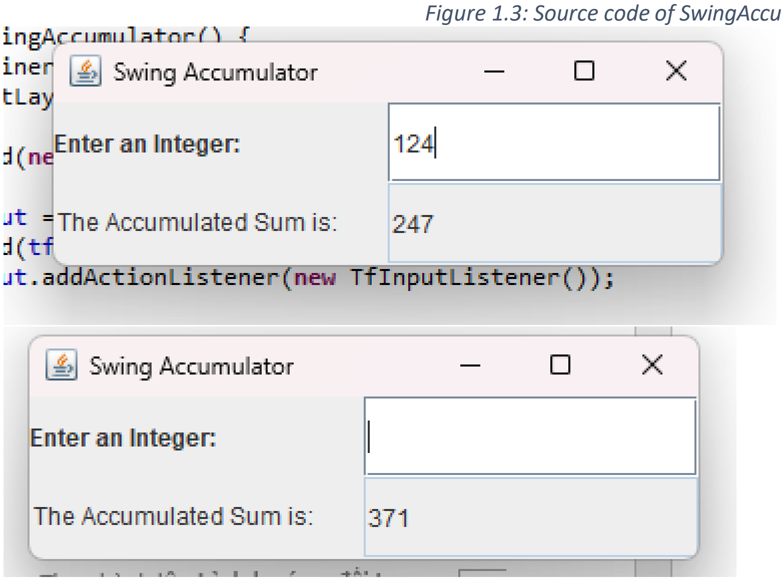
Figure 1.2: Demo of AWTAccumulator

1.2 SwingAccummulator



```java
package GUIProject.src.hust.soict.dsai.swing;
import java.awt.*;


public class SwingAccumulator extends JFrame {

    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();
        cp.setLayout(new GridLayout(2,2));

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

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

        cp.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 e) {
            int numberIn = Integer.parseInt(tfInput.getText());
            sum += numberIn;
            tfInput.setText("");
            tfOutput.setText(sum +"");;
        }

    }

}
```

Figure 1.3: Source code of SwingAccumulator



Enter an Integer: 124

The Accumulated Sum is: 247

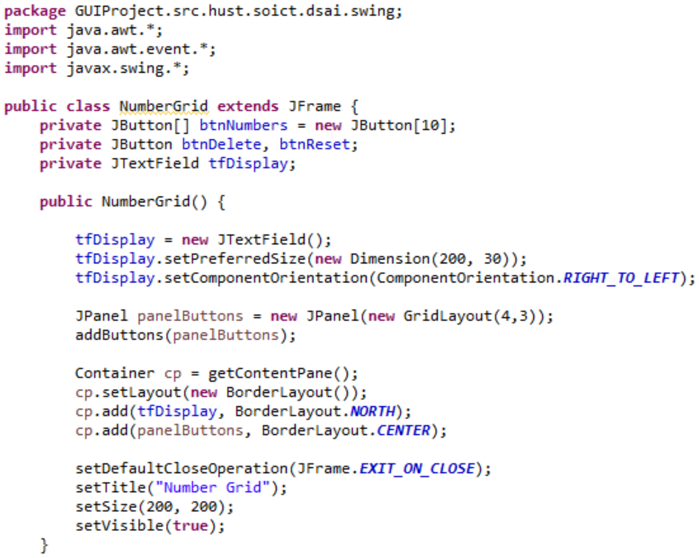
Enter an Integer:

The Accumulated Sum is: 371

Figure 1.4: Demo of SwingAccumulator

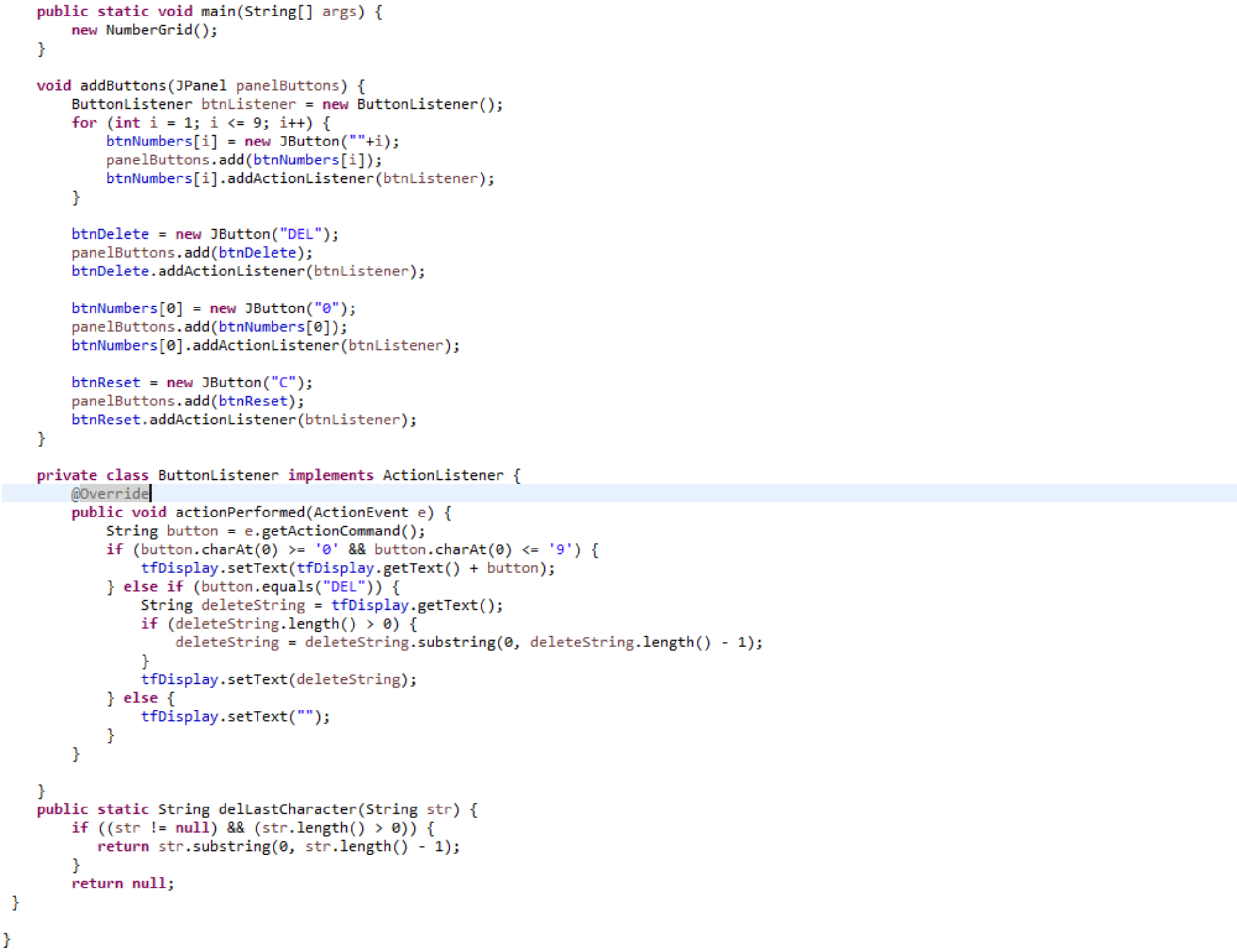
2.Organnizing Swing components with Layout Managers

2.1 Code



```
package GUIProject.src.hust.soict.dsai.swing;
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

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

    public NumberGrid() {

        tfDisplay = new JTextField();
        tfDisplay.setPreferredSize(new Dimension(200, 30));
        tfDisplay.setComponentOrientation(ComponentOrientation.RIGHT_TO_LEFT);

        JPanel panelButtons = new JPanel(new GridLayout(4,3));
        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(200, 200);
        setVisible(true);
    }
```

Figure 2.1: Source code of NumberGrid 1



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

    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("DEL");
        panelButtons.add(btnDelete);
        btnDelete.addActionListener(btnListener);

        btnNumbers[0] = new JButton("0");
        panelButtons.add(btnNumbers[0]);
        btnNumbers[0].addActionListener(btnListener);

        btnReset = new JButton("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(0) >= '0' && button.charAt(0) <= '9') {
                tfDisplay.setText(tfDisplay.getText() + button);
            } else if (button.equals("DEL")) {
                String deleteString = tfDisplay.getText();
                if (deleteString.length() > 0) {
                    deleteString = deleteString.substring(0, deleteString.length() - 1);
                }
                tfDisplay.setText(deleteString);
            } else {
                tfDisplay.setText("");
            }
        }

    }
    public static String delLastCharacter(String str) {
        if ((str != null) && (str.length() > 0)) {
           return str.substring(0, str.length() - 1);
        }
        return null;
 }

}
```

Figure 2.2: Source code of NumberGrid 2

2.2 Demo

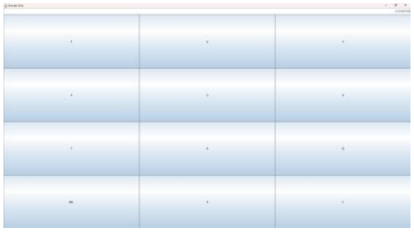


Figure 2.3: Demo buttons 0-9

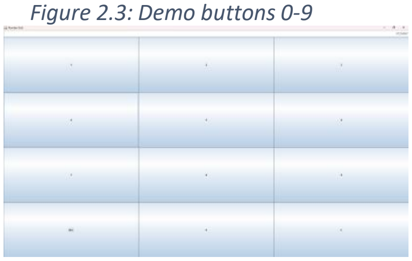


Figure 2.4: Demo DEL button

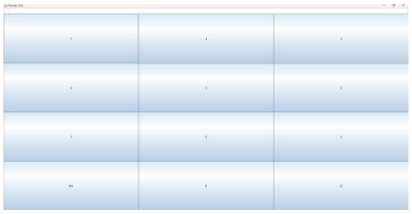
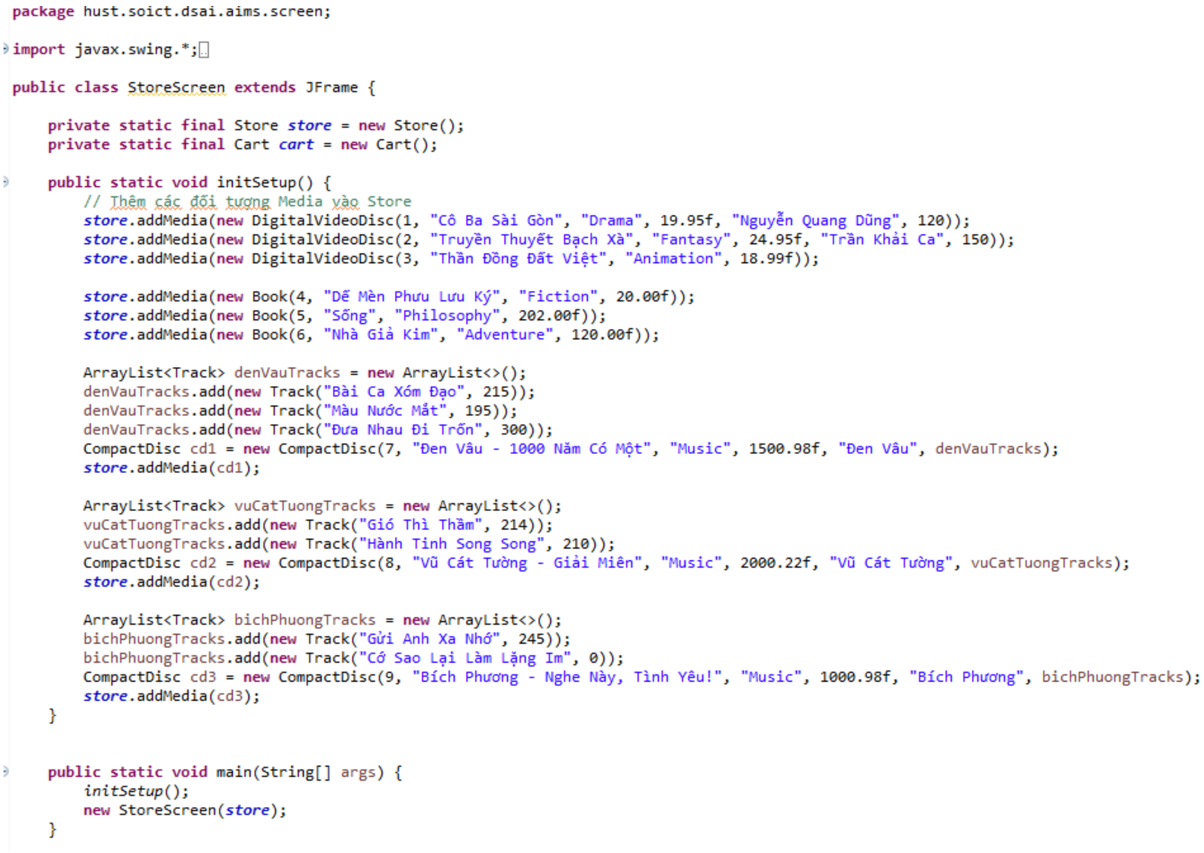


Figure 2.5: Demo C button

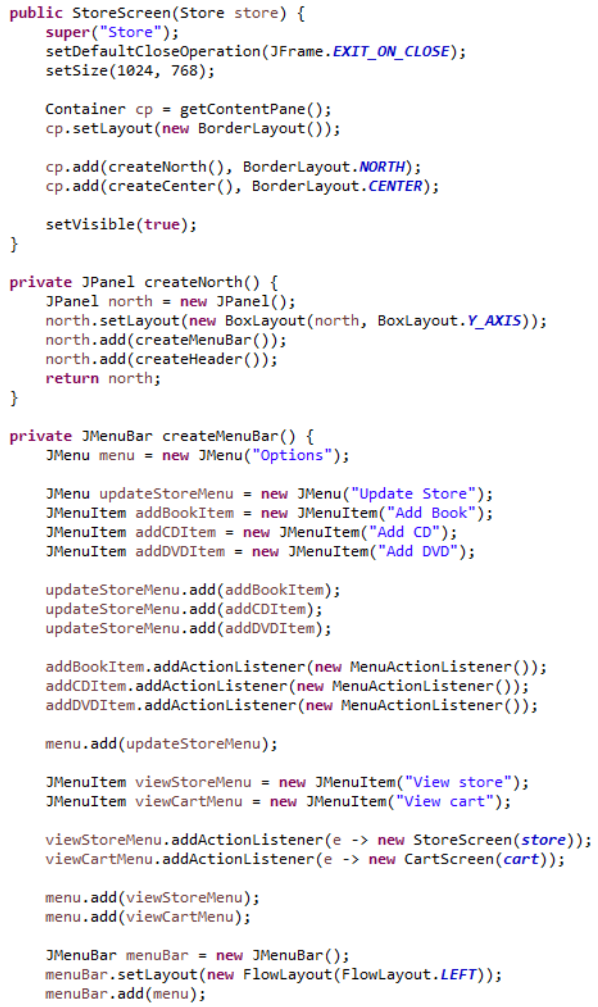
3. Create a graphical user interface for AIMS with Swing

3.1 Create class StoreScreen



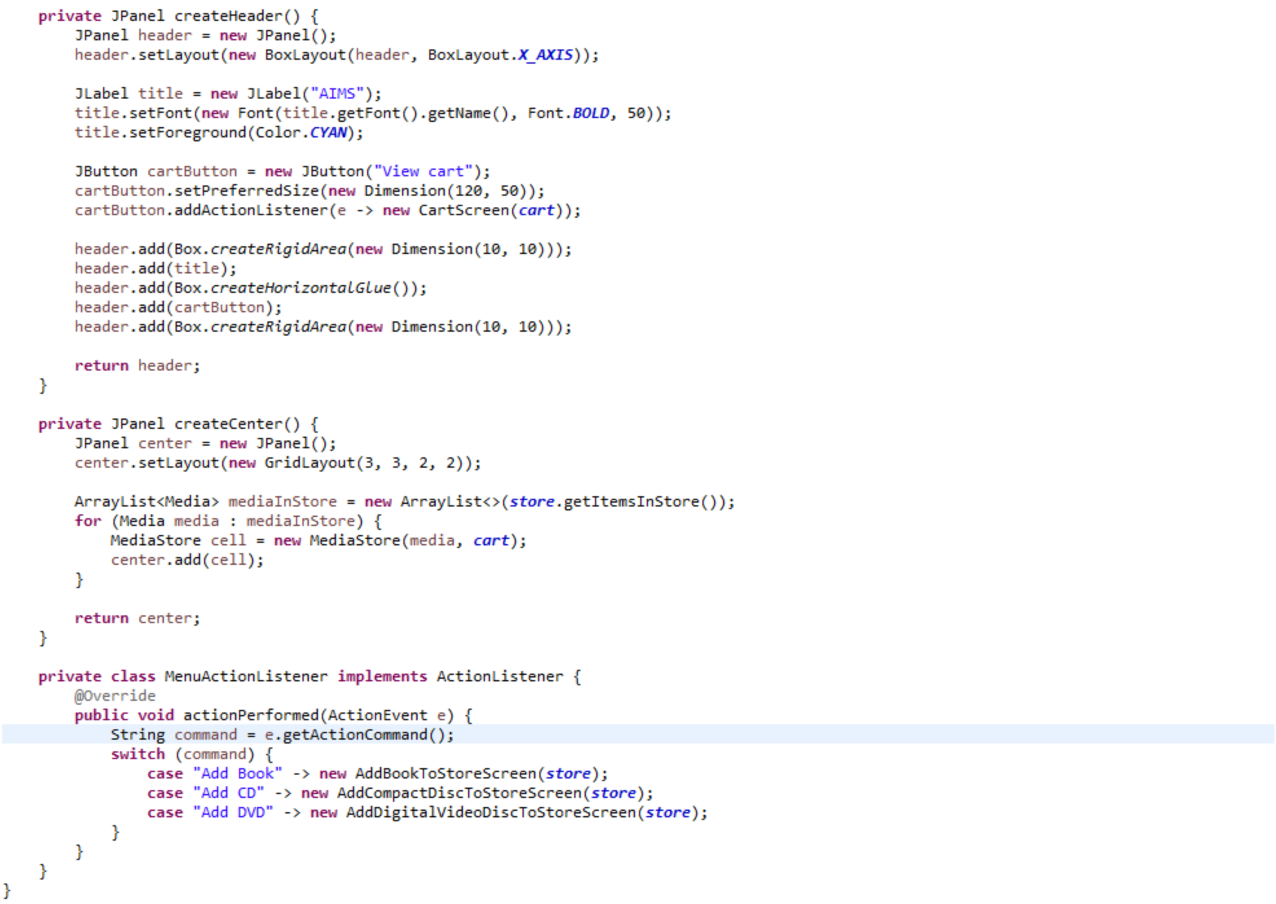
```
package hust.soict.dsai.aims.screen;

import javax.swing.*;

public class StoreScreen extends JFrame {

    private static final Store store = new Store();
    private static final Cart cart = new Cart();

    public static void initSetup() {
        // Thêm các đối tượng Media vào Store
        store.addMedia(new DigitalVideoDisc(1, "Cô Ba Sài Gòn", "Drama", 19.95f, "Nguyễn Quang Dũng", 120));
        store.addMedia(new DigitalVideoDisc(2, "Truyền Thuyết Bạch Xà", "Fantasy", 24.95f, "Trần Khải Ca", 150));
        store.addMedia(new DigitalVideoDisc(3, "Thần Đồng Đất Việt", "Animation", 18.99f));

        store.addMedia(new Book(4, "Dế Mèn Phưu Lưu Ký", "Fiction", 20.00f));
        store.addMedia(new Book(5, "Sống", "Philosophy", 202.00f));
        store.addMedia(new Book(6, "Nhà Giả Kim", "Adventure", 120.00f));

        ArrayList<Track> denVauTracks = new ArrayList<>();
        denVauTracks.add(new Track("Bài Ca Xóm Đạo", 215));
        denVauTracks.add(new Track("Màu Nước Mắt", 195));
        denVauTracks.add(new Track("Đưa Nhau Đi Trốn", 300));
        CompactDisc cd1 = new CompactDisc(7, "Đen Vâu - 1000 Năm Có Một", "Music", 1500.98f, "Đen Vâu", denVauTracks);
        store.addMedia(cd1);

        ArrayList<Track> vuCatTuongTracks = new ArrayList<>();
        vuCatTuongTracks.add(new Track("Gió Thì Thầm", 214));
        vuCatTuongTracks.add(new Track("Hành Tinh Song Song", 210));
        CompactDisc cd2 = new CompactDisc(8, "Vũ Cát Tường - Giải Miên", "Music", 2000.22f, "Vũ Cát Tường", vuCatTuongTracks);
        store.addMedia(cd2);

        ArrayList<Track> bichPhuongTracks = new ArrayList<>();
        bichPhuongTracks.add(new Track("Gửi Anh Xa Nhớ", 245));
        bichPhuongTracks.add(new Track("Cớ Sao Lại Làm Lặng Im", 0));
        CompactDisc cd3 = new CompactDisc(9, "Bích Phương - Nghe Này, Tình Yêu!", "Music", 1000.98f, "Bích Phương", bichPhuongTracks);
        store.addMedia(cd3);
    }


    public static void main(String[] args) {
        initSetup();
        new StoreScreen(store);
    }
```

Figure 3.1: Class StoreScreen 1



```
public StoreScreen(Store store) {
    super("Store");
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    setSize(1024, 768);

    Container cp = getContentPane();
    cp.setLayout(new BorderLayout());

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

    setVisible(true);
}

private JPanel createNorth() {
    JPanel north = new JPanel();
    north.setLayout(new BoxLayout(north, BoxLayout.Y_AXIS));
    north.add(createMenuBar());
    north.add(createHeader());
    return north;
}

private JMenuBar createMenuBar() {
    JMenu menu = new JMenu("Options");

    JMenu updateStoreMenu = new JMenu("Update Store");
    JMenuItem addBookItem = new JMenuItem("Add Book");
    JMenuItem addCDItem = new JMenuItem("Add CD");
    JMenuItem addDVDItem = new JMenuItem("Add DVD");

    updateStoreMenu.add(addBookItem);
    updateStoreMenu.add(addCDItem);
    updateStoreMenu.add(addDVDItem);

    addBookItem.addActionListener(new MenuActionListener());
    addCDItem.addActionListener(new MenuActionListener());
    addDVDItem.addActionListener(new MenuActionListener());

    menu.add(updateStoreMenu);

    JMenuItem viewStoreMenu = new JMenuItem("View store");
    JMenuItem viewCartMenu = new JMenuItem("View cart");

    viewStoreMenu.addActionListener(e -> new StoreScreen(store));
    viewCartMenu.addActionListener(e -> new CartScreen(cart));

    menu.add(viewStoreMenu);
    menu.add(viewCartMenu);

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

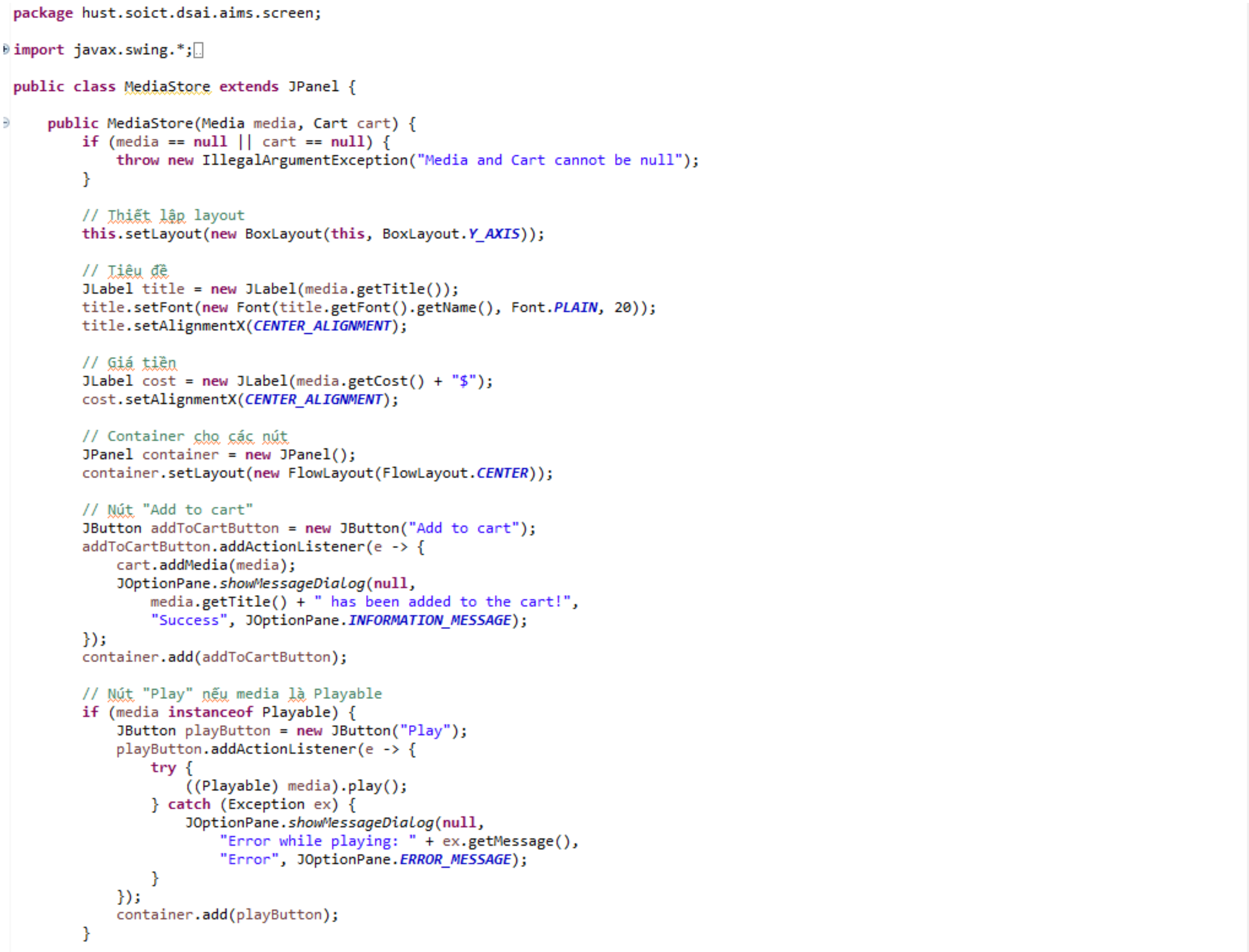
Figure 3.2: Class StoreScreen 2



```
    private 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.BOLD, 50));
        title.setForeground(Color.CYAN);

        JButton cartButton = new JButton("View cart");
        cartButton.setPreferredSize(new Dimension(120, 50));
        cartButton.addActionListener(e -> new CartScreen(cart));

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

        return header;
    }

    private JPanel createCenter() {
        JPanel center = new JPanel();
        center.setLayout(new GridLayout(3, 3, 2, 2));

        ArrayList<Media> mediaInStore = new ArrayList<>(store.getItemsInStore());
        for (Media media : mediaInStore) {
            MediaStore cell = new MediaStore(media, cart);
            center.add(cell);
        }

        return center;
    }

    private class MenuActionListener implements ActionListener {
        @Override
        public void actionPerformed(ActionEvent e) {
            String command = e.getActionCommand();
            switch (command) {
                case "Add Book" -> new AddBookToStoreScreen(store);
                case "Add CD" -> new AddCompactDiscToStoreScreen(store);
                case "Add DVD" -> new AddDigitalVideoDiscToStoreScreen(store);
            }
        }
    }
}
```

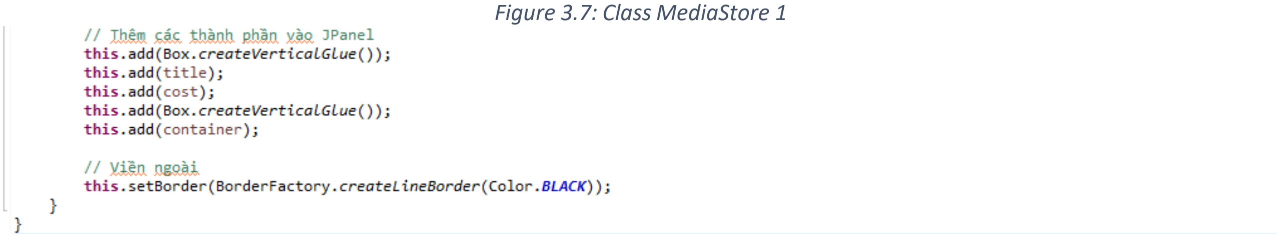
Figure 3.3: Class StoreScreen

3.2 Create class MediaStore



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

import javax.swing.*;

public class MediaStore extends JPanel {

    public MediaStore(Media media, Cart cart) {
        if (media == null || cart == null) {
            throw new IllegalArgumentException("Media and Cart cannot be null");
        }

        // Thiết lập layout
        this.setLayout(new BoxLayout(this, BoxLayout.Y_AXIS));

        // Tiêu đề
        JLabel title = new JLabel(media.getTitle());
        title.setFont(new Font(title.getFont().getName(), Font.PLAIN, 20));
        title.setAlignmentX(CENTER_ALIGNMENT);

        // Giá tiền
        JLabel cost = new JLabel(media.getCost() + "$");
        cost.setAlignmentX(CENTER_ALIGNMENT);

        // Container cho các nút
        JPanel container = new JPanel();
        container.setLayout(new FlowLayout(FlowLayout.CENTER));

        // Nút "Add to cart"
        JButton addToCartButton = new JButton("Add to cart");
        addToCartButton.addActionListener(e -> {
            cart.addMedia(media);
            JOptionPane.showMessageDialog(null,
                media.getTitle() + " has been added to the cart!",
                "Success", JOptionPane.INFORMATION_MESSAGE);
        });
        container.add(addToCartButton);

        // Nút "Play" nếu media là Playable
        if (media instanceof Playable) {
            JButton playButton = new JButton("Play");
            playButton.addActionListener(e -> {
                try {
                    ((Playable) media).play();
                } catch (Exception ex) {
                    JOptionPane.showMessageDialog(null,
                        "Error while playing: " + ex.getMessage(),
                        "Error", JOptionPane.ERROR_MESSAGE);
                }
            });
            container.add(playButton);
        }
```

Figure 3.7: Class MediaStore 1



```
        // Thêm các thành phần vào JPanel
        this.add(Box.createVerticalGlue());
        this.add(title);
        this.add(cost);
        this.add(Box.createVerticalGlue());
        this.add(container);

        // Viền ngoài
        this.setBorder(BorderFactory.createLineBorder(Color.BLACK));
    }
}
```

Figure 3.8: Class MediaStore 2

3.3 Demo

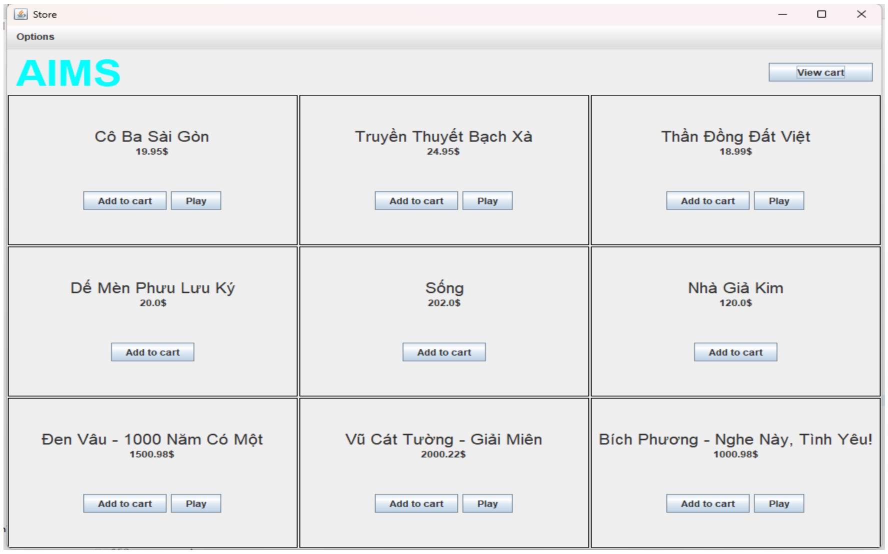
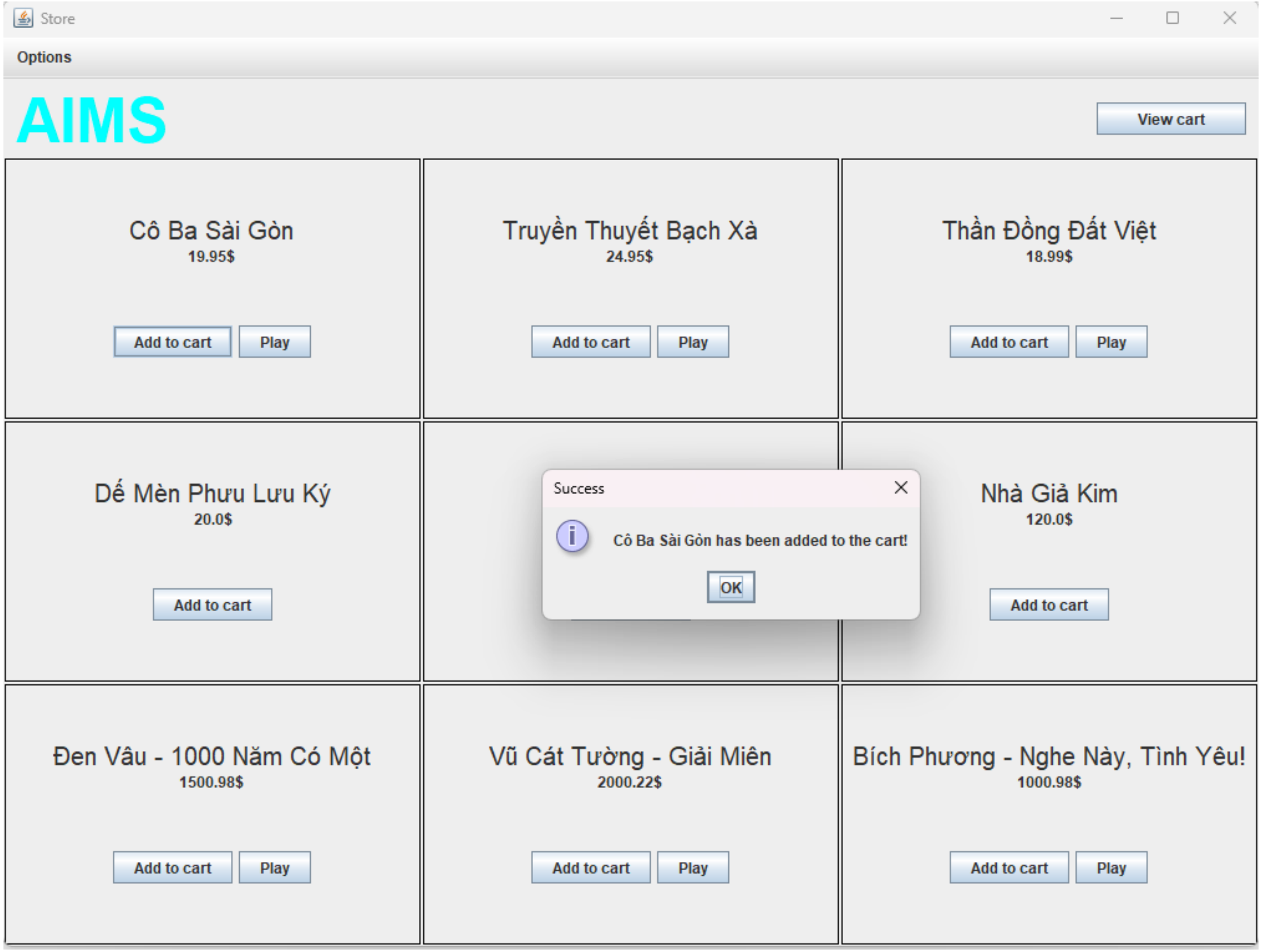
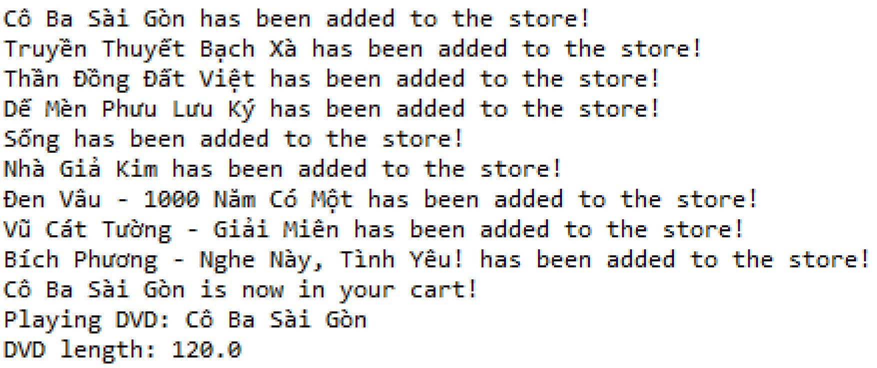


Figure 3.9: StoreScreen



Cô Ba Sài Gòn has been added to the cart!

Figure 3.10 Demo Add to cart button

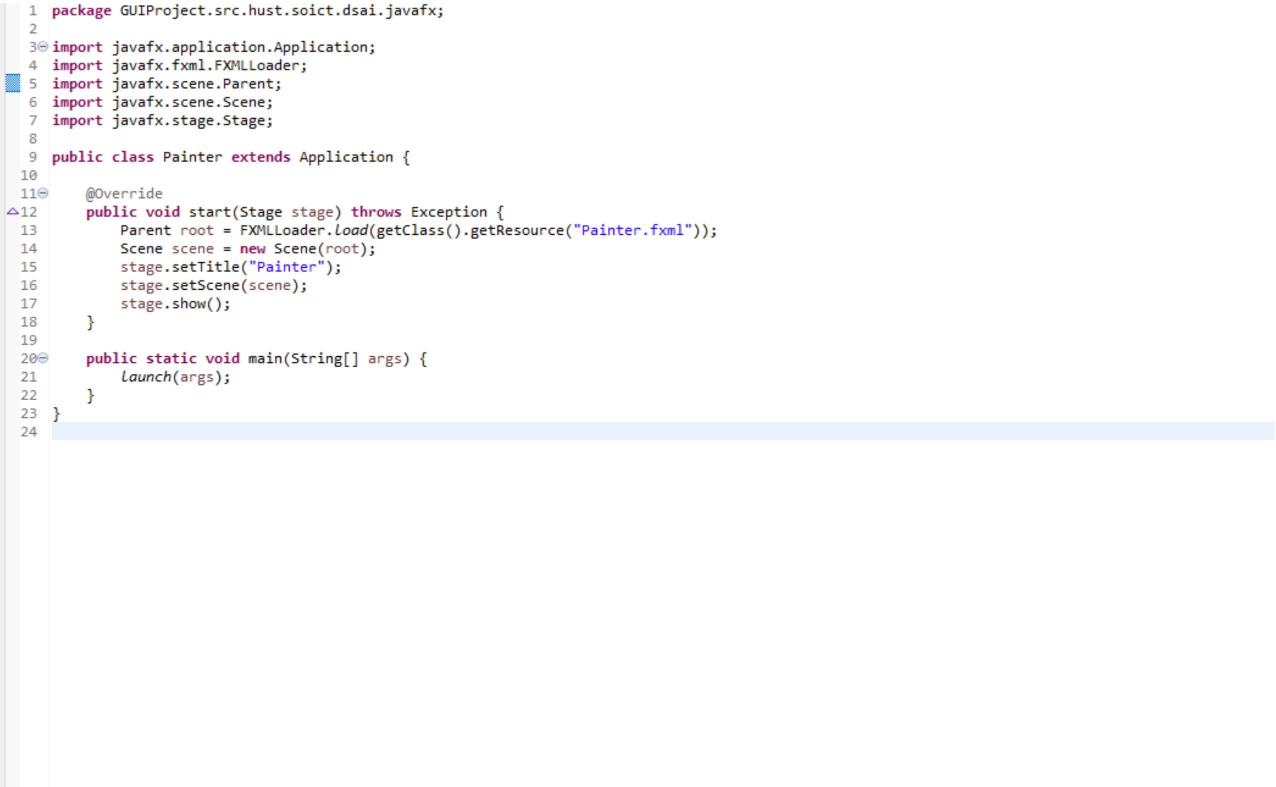


```
Cô Ba Sài Gòn has been added to the store!
Truyền Thuyết Bạch Xà has been added to the store!
Thần Đồng Đất Việt has been added to the store!
Dế Mèn Phưu Lưu Ký has been added to the store!
Sống has been added to the store!
Nhà Giả Kim has been added to the store!
Đen Vâu - 1000 Năm Có Một has been added to the store!
Vũ Cát Tường - Giải Miên has been added to the store!
Bích Phương - Nghe Này, Tình Yêu! has been added to the store!
Cô Ba Sài Gòn is now in your cart!
Playing DVD: Cô Ba Sài Gòn
DVD length: 120.0
```

Figure 3.11 Demo Play button

4. JavaFX API

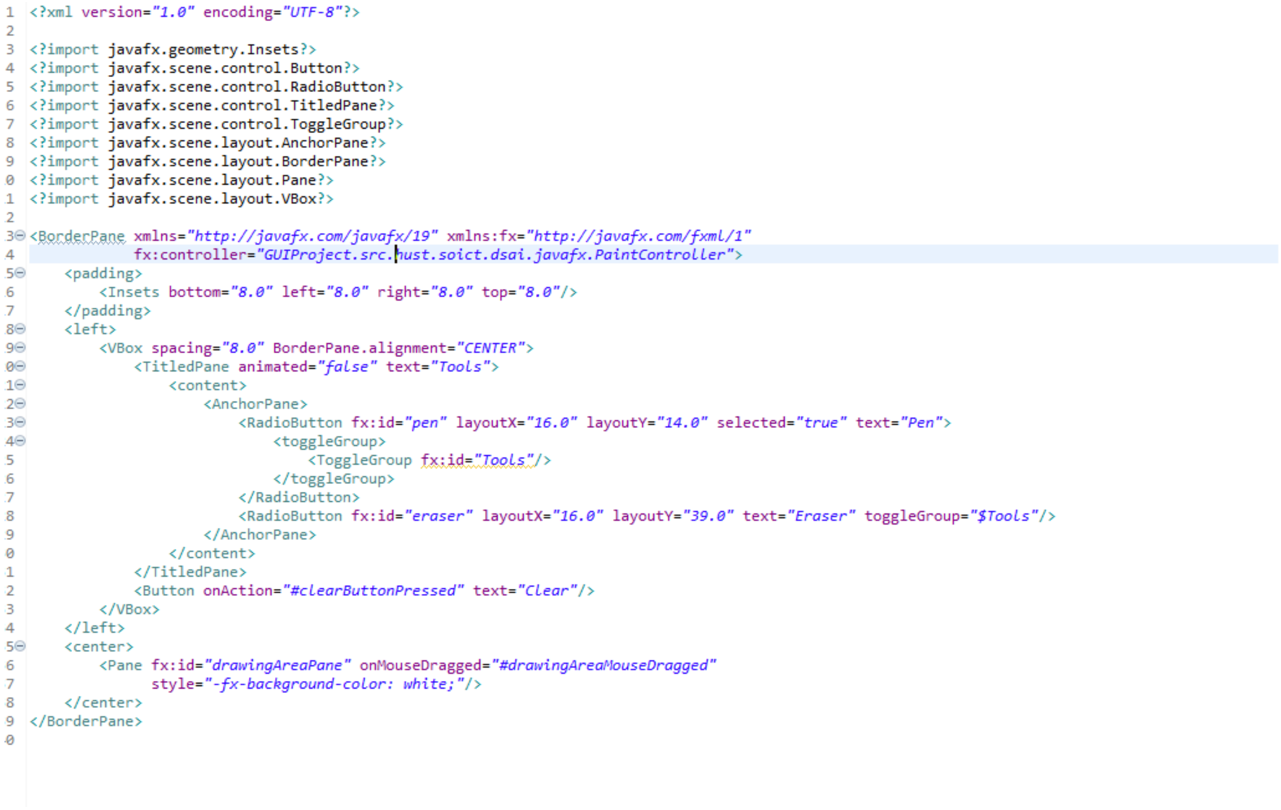
4.1 Create class Painter



```java
package GUIProject.src.hust.soict.dsai.javafx;

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

public class Painter extends Application {

    @Override
    public void start(Stage stage) throws Exception {
        Parent root = FXMLLoader.load(getClass().getResource("Painter.fxml"));
        Scene scene = new Scene(root);
        stage.setTitle("Painter");
        stage.setScene(scene);
        stage.show();
    }

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

Figure 4.1: Class Painter

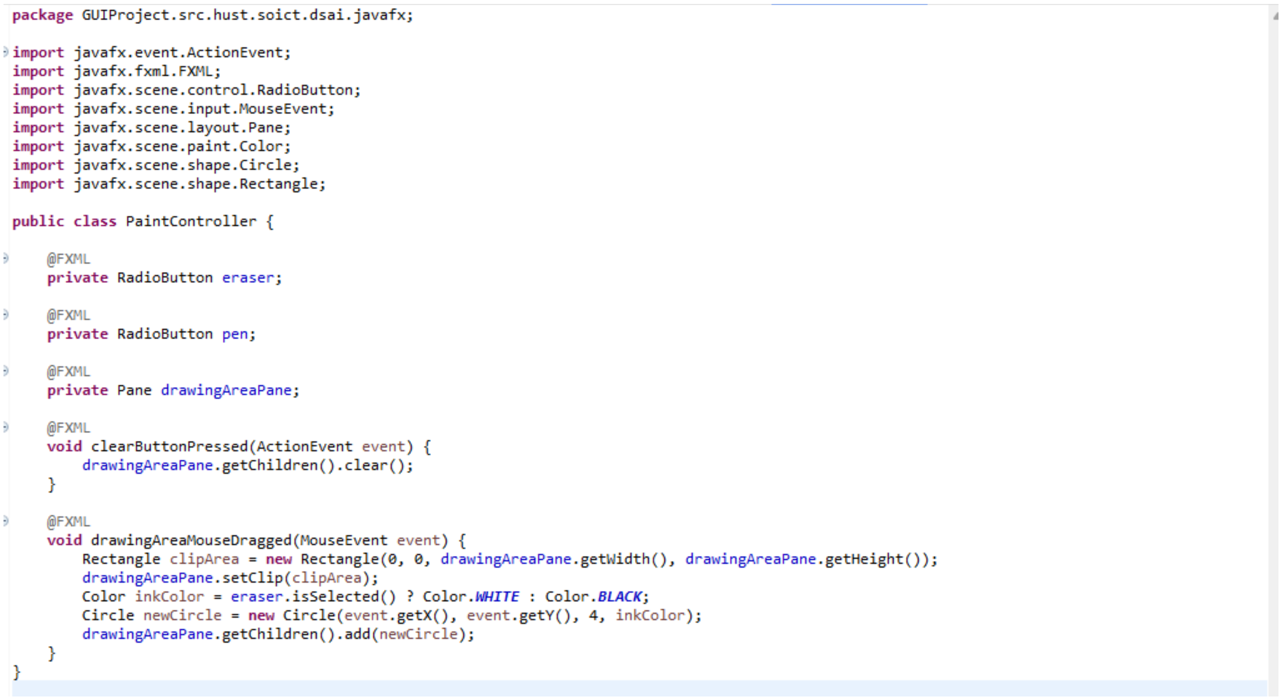
4.2 Create Painter.fxml



```
<?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 xmlns="http://javafx.com/javafx/19" xmlns:fx="http://javafx.com/fxml/1"
            fx:controller="GUIProject.src.hust.soict.dsai.javafx.PaintController">
    <padding>
        <Insets bottom="8.0" left="8.0" right="8.0" top="8.0"/>
    </padding>
    <left>
        <VBox spacing="8.0" BorderPane.alignment="CENTER">
            <TitledPane animated="false" text="Tools">
                <content>
                    <AnchorPane>
                        <RadioButton fx:id="pen" layoutX="16.0" layoutY="14.0" selected="true" text="Pen">
                            <toggleGroup>
                                <ToggleGroup fx:id="Tools"/>
                            </toggleGroup>
                        </RadioButton>
                        <RadioButton fx:id="eraser" layoutX="16.0" layoutY="39.0" text="Eraser" toggleGroup="$Tools"/>
                    </AnchorPane>
                </content>
            </TitledPane>
            <Button onAction="#clearButtonPressed" text="Clear"/>
        </VBox>
    </left>
    <center>
        <Pane fx:id="drawingAreaPane" onMouseDragged="#drawingAreaMouseDragged"
              style="-fx-background-color: white;"/>
    </center>
</BorderPane>
```

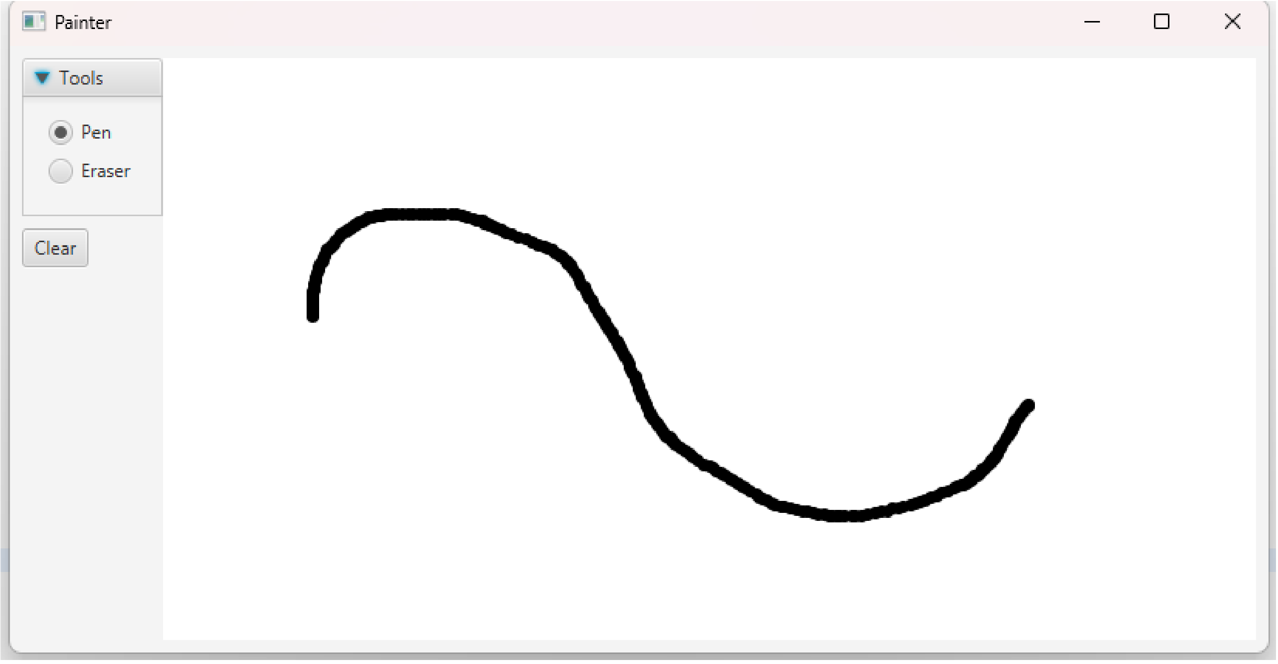
Figure 4.2: Painter.fxml 1

4.3 Create class PainterController



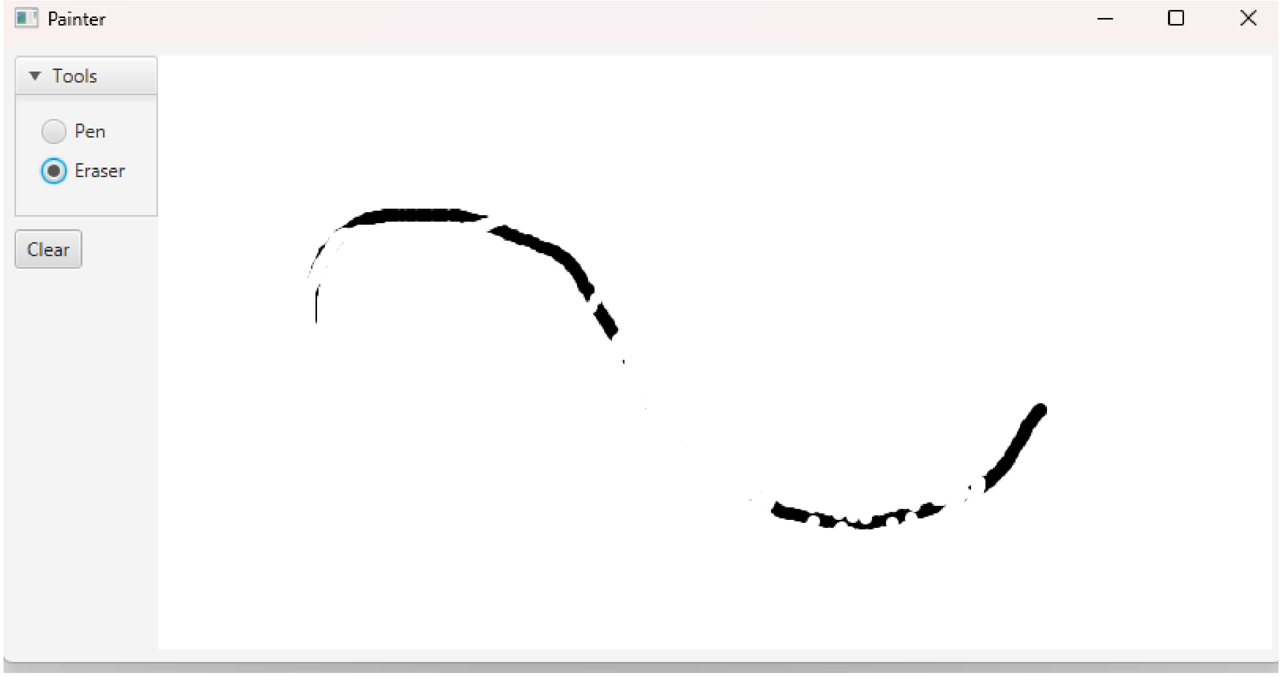
```
package GUIProject.src.hust.soict.dsai.javafx;

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

public class PaintController {

    @FXML
    private RadioButton eraser;

    @FXML
    private RadioButton pen;

    @FXML
    private Pane drawingAreaPane;

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

    @FXML
    void drawingAreaMouseDragged(MouseEvent event) {
        Rectangle clipArea = new Rectangle(0, 0, drawingAreaPane.getWidth(), drawingAreaPane.getHeight());
        drawingAreaPane.setClip(clipArea);
        Color inkColor = eraser.isSelected() ? Color.WHITE : Color.BLACK;
        Circle newCircle = new Circle(event.getX(), event.getY(), 4, inkColor);
        drawingAreaPane.getChildren().add(newCircle);
    }
}
```

Figure 4.4: PainterController



● Pen
○ Eraser

Figure 4.5: Use Pen



○ Pen
● Eraser

Figure 4.6: Use Eraser

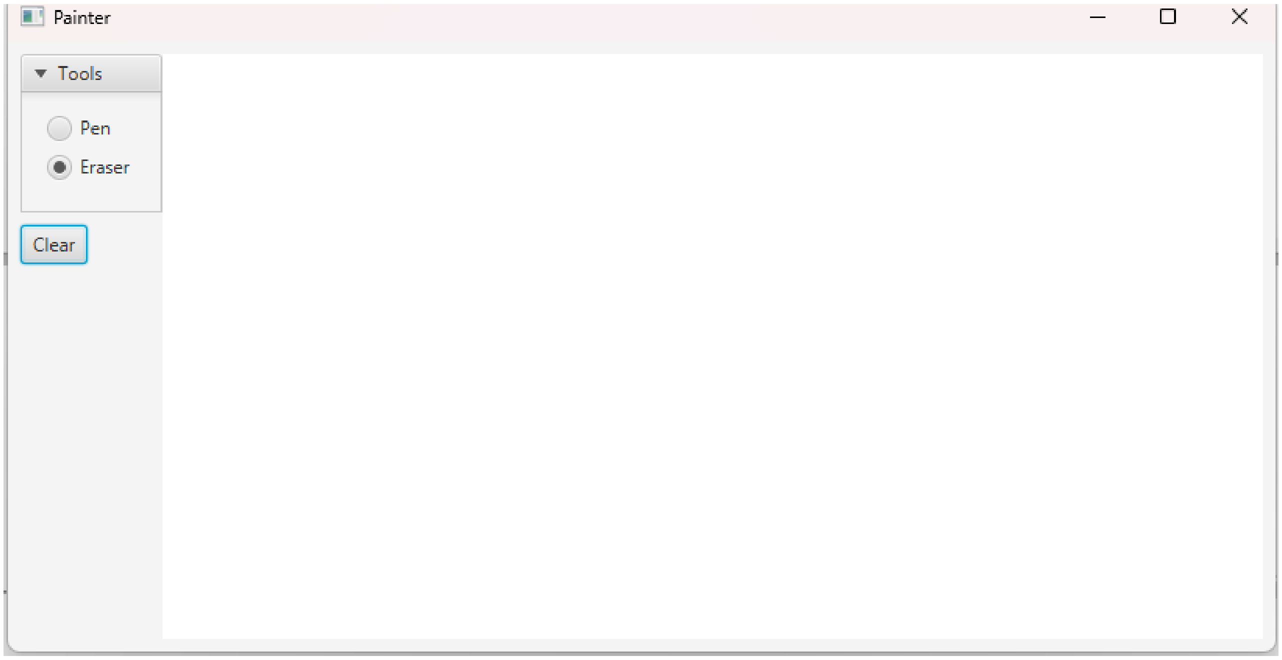
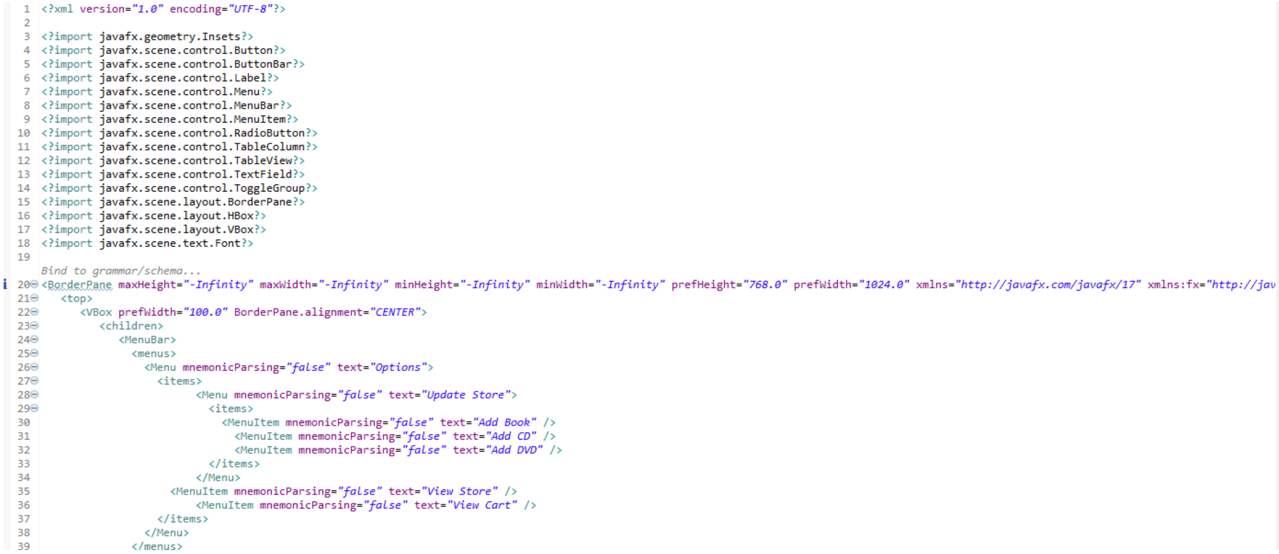


Figure 4.7: Clear button

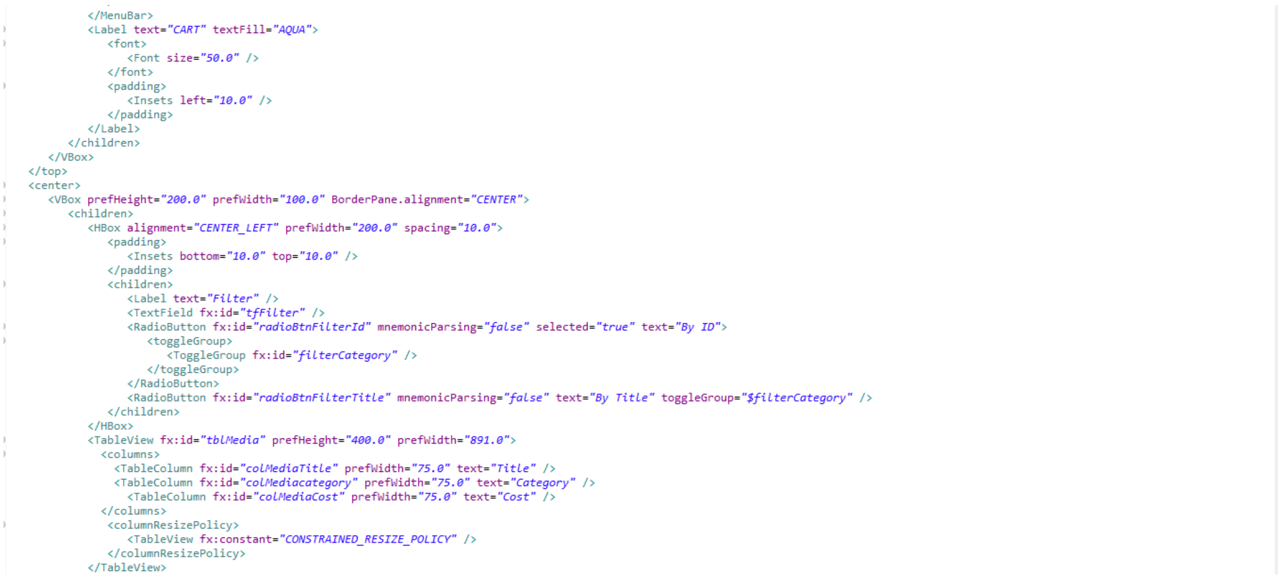
5. View Cart Screen

5.1 Create cart.fxml



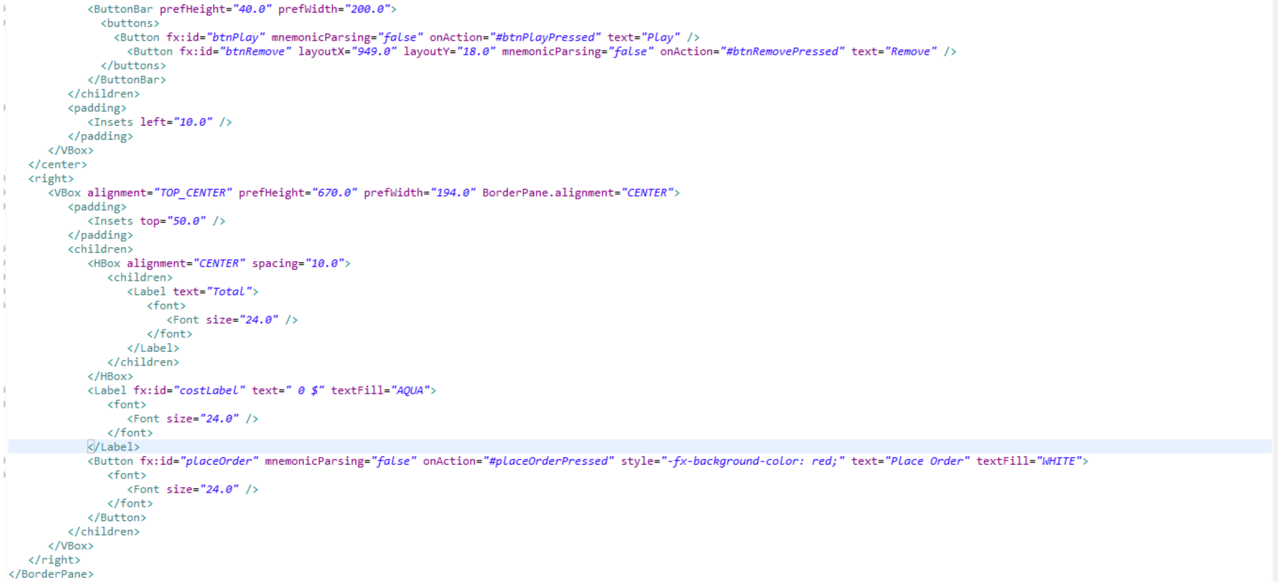
```
<?xml version="1.0" encoding="UTF-8"?>

<?import 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.BorderPane?>
<?import javafx.scene.layout.HBox?>
<?import javafx.scene.layout.VBox?>
<?import javafx.scene.text.Font?>

Bind to grammar/schema...
<BorderPane maxHeight="-Infinity" maxWidth="-Infinity" minHeight="-Infinity" minWidth="-Infinity" prefHeight="768.0" prefWidth="1024.0" xmlns="http://javafx.com/javafx/17" xmlns:fx="http://jav
   <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>
```

Figure 5.1: Cart.fxml 1



```
            </MenuBar>
            <Label text="CART" textFill="AQUA">
               <font>
                  <Font size="50.0" />
               </font>
               <padding>
                  <Insets left="10.0" />
               </padding>
            </Label>
         </children>
      </VBox>
   </top>
   <center>
      <VBox prefHeight="200.0" prefWidth="100.0" BorderPane.alignment="CENTER">
         <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" prefHeight="400.0" prefWidth="891.0">
              <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>
```

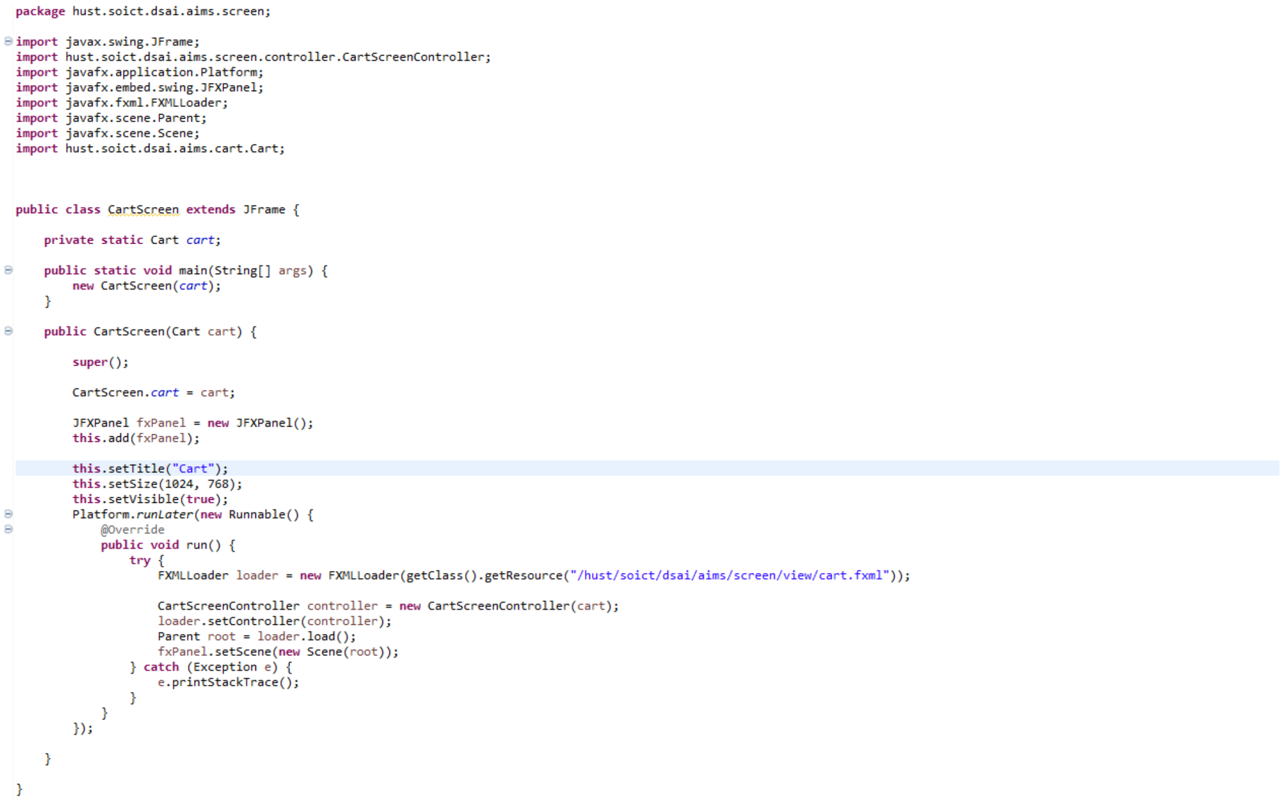
Figure 5.2: Cart.fxml 2



```
            <ButtonBar prefHeight="40.0" prefWidth="200.0">
              <buttons>
                <Button fx:id="btnPlay" mnemonicParsing="false" onAction="#btnPlayPressed" text="Play" />
                  <Button fx:id="btnRemove" layoutX="949.0" layoutY="18.0" mnemonicParsing="false" onAction="#btnRemovePressed" text="Remove" />
              </buttons>
            </ButtonBar>
         </children>
         <padding>
            <Insets left="10.0" />
         </padding>
      </VBox>
   </center>
   <right>
      <VBox alignment="TOP_CENTER" prefHeight="670.0" prefWidth="194.0" BorderPane.alignment="CENTER">
         <padding>
            <Insets top="50.0" />
         </padding>
         <children>
            <HBox alignment="CENTER" spacing="10.0">
               <children>
                  <Label text="Total">
                     <font>
                        <Font size="24.0" />
                     </font>
                  </Label>
               </children>
            </HBox>
            <Label fx:id="costLabel" text=" 0 $" textFill="AQUA">
               <font>
                  <Font size="24.0" />
               </font>
            </Label>
            <Button fx:id="placeOrder" mnemonicParsing="false" onAction="#placeOrderPressed" style="-fx-background-color: red;" text="Place Order" textFill="WHITE">
               <font>
                  <Font size="24.0" />
               </font>
            </Button>
         </children>
      </VBox>
   </right>
</BorderPane>
```

Figure 5.3: Cart.fxml 3

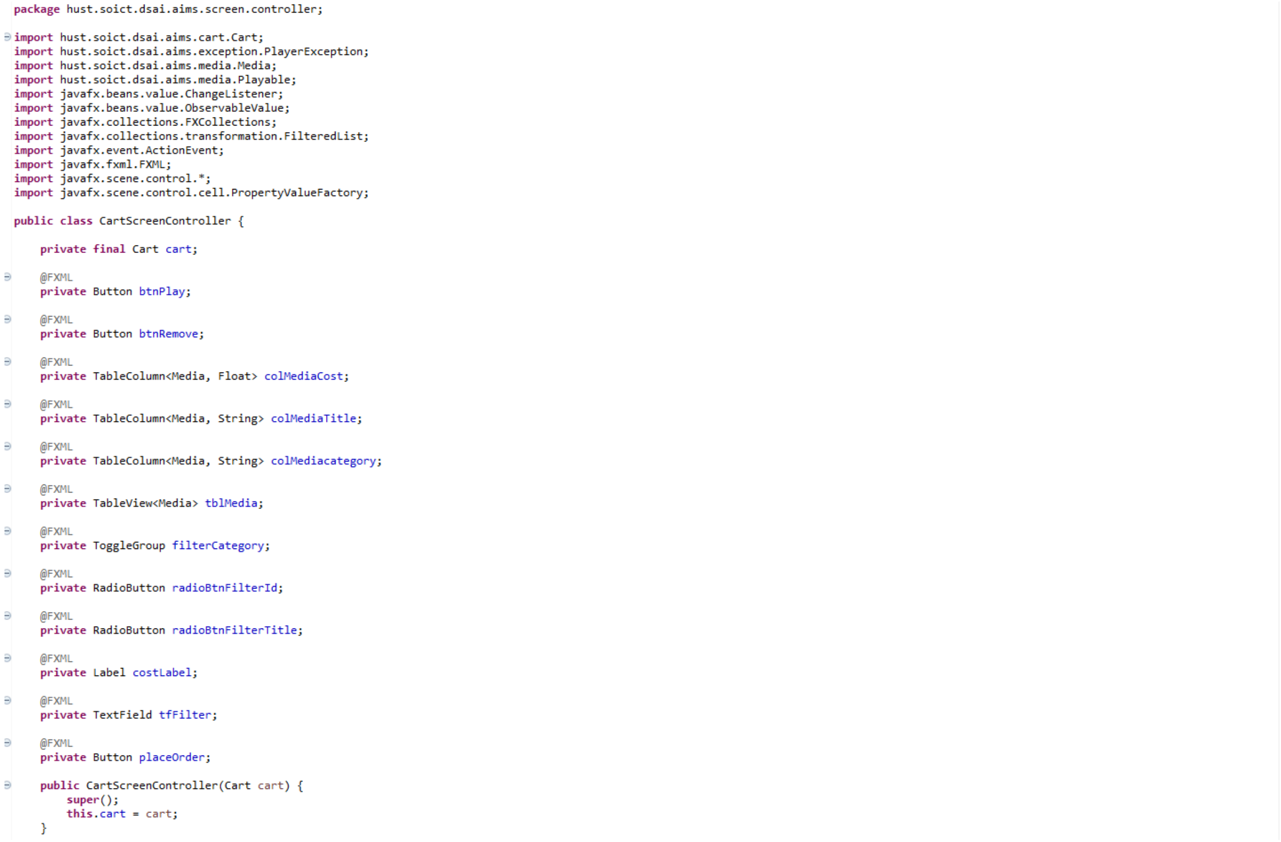
5.2 Create class CartScreen



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

import javax.swing.JFrame;
import hust.soict.dsai.aims.screen.controller.CartScreenController;
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;


public class CartScreen extends JFrame {

    private static Cart cart;

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

    public CartScreen(Cart cart) {

        super();

        CartScreen.cart = cart;

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

        this.setTitle("Cart");
        this.setSize(1024, 768);
        this.setVisible(true);
        Platform.runLater(new Runnable() {
            @Override
            public void run() {
                try {
                    FXMLLoader loader = new FXMLLoader(getClass().getResource("/hust/soict/dsai/aims/screen/view/cart.fxml"));

                    CartScreenController controller = new CartScreenController(cart);
                    loader.setController(controller);
                    Parent root = loader.load();
                    fxPanel.setScene(new Scene(root));
                } catch (Exception e) {
                    e.printStackTrace();
                }
            }
        });

    }

}
```

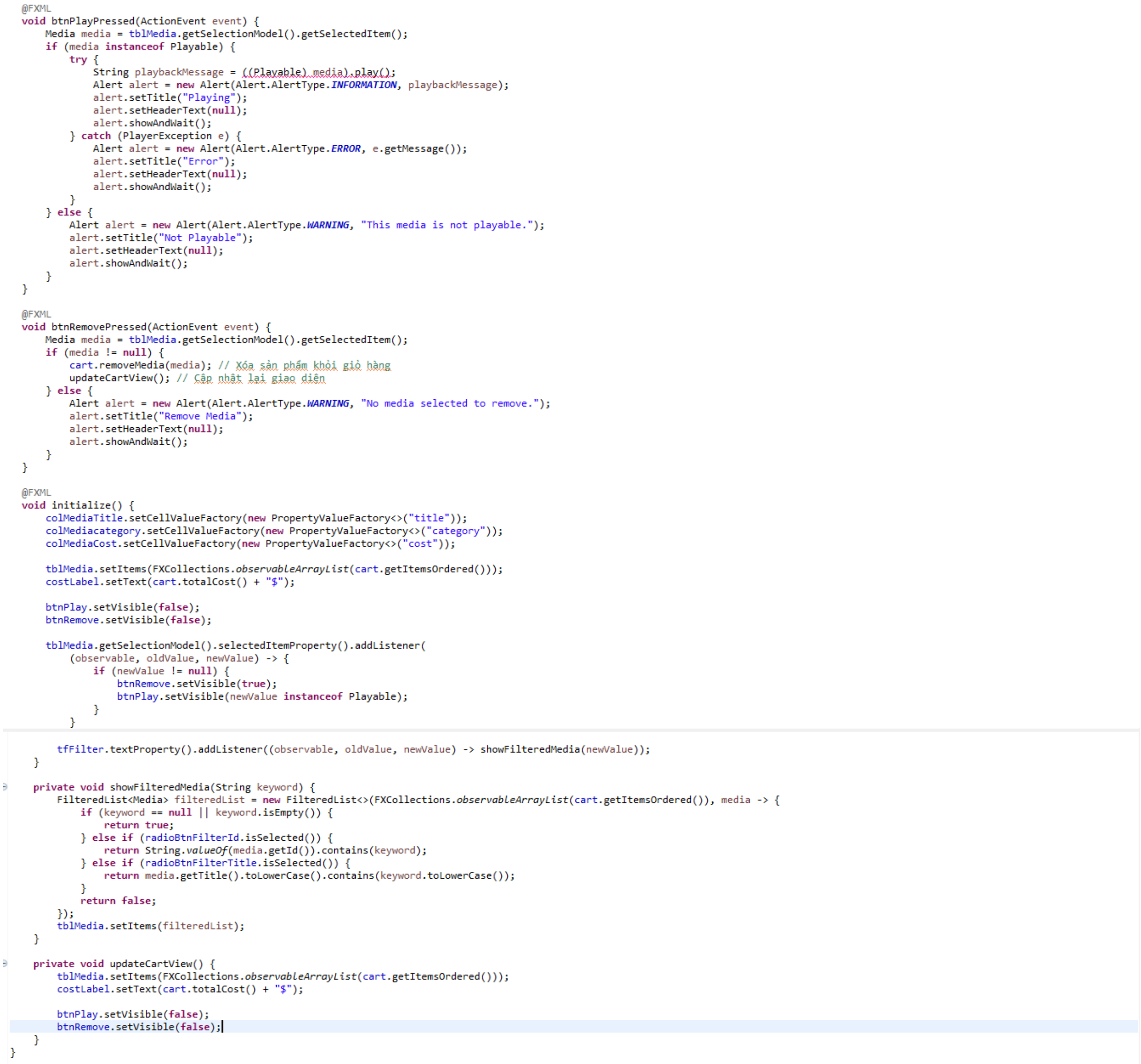
Figure 5.4: CartScreen class

5.3 Create class CartScreenController



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

import hust.soict.dsai.aims.cart.Cart;
import hust.soict.dsai.aims.exception.PlayerException;
import hust.soict.dsai.aims.media.Media;
import hust.soict.dsai.aims.media.Playable;
import javafx.beans.value.ChangeListener;
import javafx.beans.value.ObservableValue;
import javafx.collections.FXCollections;
import javafx.collections.transformation.FilteredList;
import javafx.event.ActionEvent;
import javafx.fxml.FXML;
import javafx.scene.control.*;
import javafx.scene.control.cell.PropertyValueFactory;

public class CartScreenController {

    private final Cart cart;

    @FXML
    private Button btnPlay;

    @FXML
    private Button btnRemove;

    @FXML
    private TableColumn<Media, Float> colMediaCost;

    @FXML
    private TableColumn<Media, String> colMediaTitle;

    @FXML
    private TableColumn<Media, String> colMediacategory;

    @FXML
    private TableView<Media> tblMedia;

    @FXML
    private ToggleGroup filterCategory;

    @FXML
    private RadioButton radioBtnFilterId;

    @FXML
    private RadioButton radioBtnFilterTitle;

    @FXML
    private Label costLabel;

    @FXML
    private TextField tfFilter;

    @FXML
    private Button placeOrder;

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

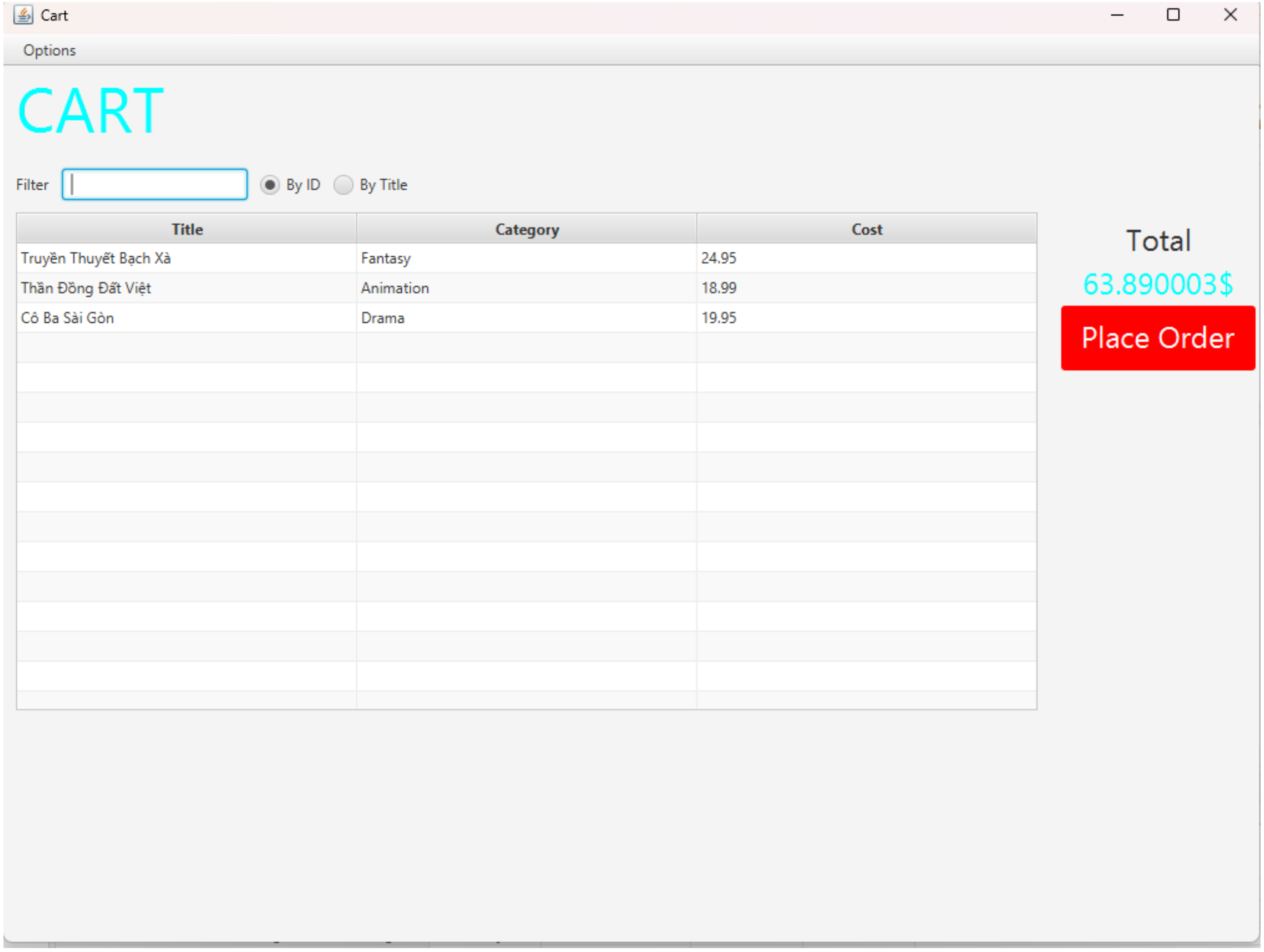
Figure 5.5: CartScreenController 1



```
    @FXML
    void btnPlayPressed(ActionEvent event) {
        Media media = tblMedia.getSelectionModel().getSelectedItem();
        if (media instanceof Playable) {
            try {
                String playbackMessage = ((Playable) media).play();
                Alert alert = new Alert(Alert.AlertType.INFORMATION, playbackMessage);
                alert.setTitle("Playing");
                alert.setHeaderText(null);
                alert.showAndWait();
            } catch (PlayerException e) {
                Alert alert = new Alert(Alert.AlertType.ERROR, e.getMessage());
                alert.setTitle("Error");
                alert.setHeaderText(null);
                alert.showAndWait();
            }
        } else {
            Alert alert = new Alert(Alert.AlertType.WARNING, "This media is not playable.");
            alert.setTitle("Not Playable");
            alert.setHeaderText(null);
            alert.showAndWait();
        }
    }

    @FXML
    void btnRemovePressed(ActionEvent event) {
        Media media = tblMedia.getSelectionModel().getSelectedItem();
        if (media != null) {
            cart.removeMedia(media); // Xóa sản phẩm khỏi giỏ hàng
            updateCartView(); // Cập nhật lại giao diện
        } else {
            Alert alert = new Alert(Alert.AlertType.WARNING, "No media selected to remove.");
            alert.setTitle("Remove Media");
            alert.setHeaderText(null);
            alert.showAndWait();
        }
    }

    @FXML
    void initialize() {
        colMediaTitle.setCellValueFactory(new PropertyValueFactory<>("title"));
        colMediacategory.setCellValueFactory(new PropertyValueFactory<>("category"));
        colMediaCost.setCellValueFactory(new PropertyValueFactory<>("cost"));

        tblMedia.setItems(FXCollections.observableArrayList(cart.getItemsOrdered()));
        costLabel.setText(cart.totalCost() + "$");

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

        tblMedia.getSelectionModel().selectedItemProperty().addListener(
            (observable, oldValue, newValue) -> {
                if (newValue != null) {
                    btnRemove.setVisible(true);
                    btnPlay.setVisible(newValue instanceof Playable);
                }
            }
        tfFilter.textProperty().addListener((observable, oldValue, newValue) -> showFilteredMedia(newValue));
    }

    private void showFilteredMedia(String keyword) {
        FilteredList<Media> filteredList = new FilteredList<>(FXCollections.observableArrayList(cart.getItemsOrdered()), media -> {
            if (keyword == null || keyword.isEmpty()) {
                return true;
            } else if (radioBtnFilterId.isSelected()) {
                return String.valueOf(media.getId()).contains(keyword);
            } else if (radioBtnFilterTitle.isSelected()) {
                return media.getTitle().toLowerCase().contains(keyword.toLowerCase());
            }
            return false;
        });
        tblMedia.setItems(filteredList);
    }

    private void updateCartView() {
        tblMedia.setItems(FXCollections.observableArrayList(cart.getItemsOrdered()));
        costLabel.setText(cart.totalCost() + "$");

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

Figure 5.6: CartScreenController 2

5.4 Demo



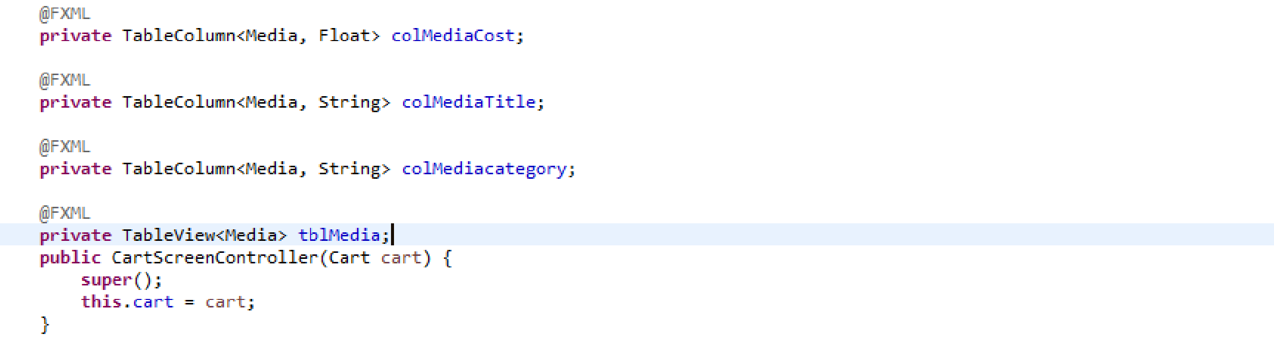
Title	Category	Cost
Truyền Thuyết Bạch Xà	Fantasy	24.95
Thần Đồng Đất Việt	Animation	18.99
Cô Ba Sài Gòn	Drama	19.95

63.890003$

Figure 5.9: Demo CartScreen

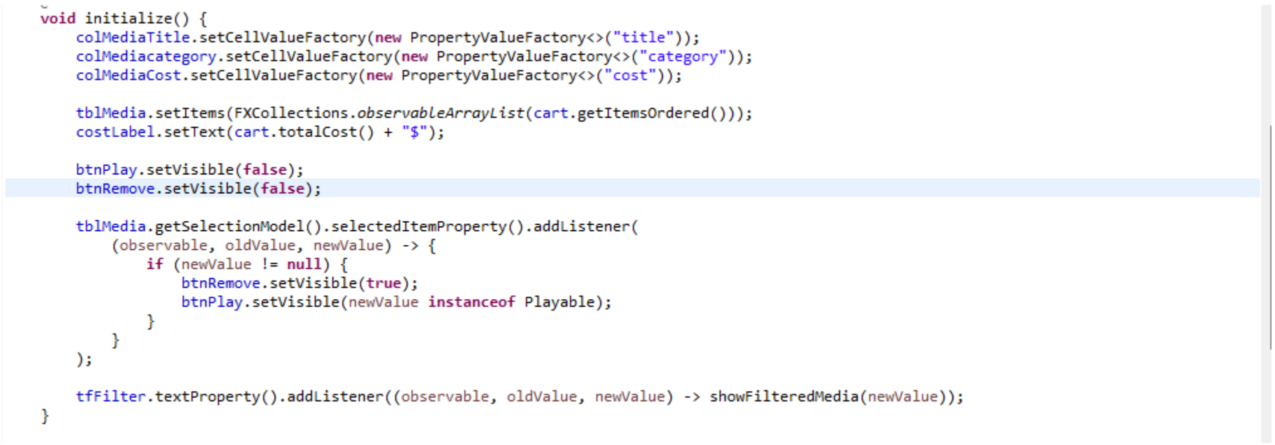
6 Updating buttons based on selected item in TableView – ChangeListener

6.1 Edit class CartScreenController



```
@FXML
private TableColumn<Media, Float> colMediaCost;

@FXML
private TableColumn<Media, String> colMediaTitle;

@FXML
private TableColumn<Media, String> colMediacategory;

@FXML
private TableView<Media> tblMedia;
public CartScreenController(Cart cart) {
    super();
    this.cart = cart;
}
```

Figure 6.1: CartScreenController 1

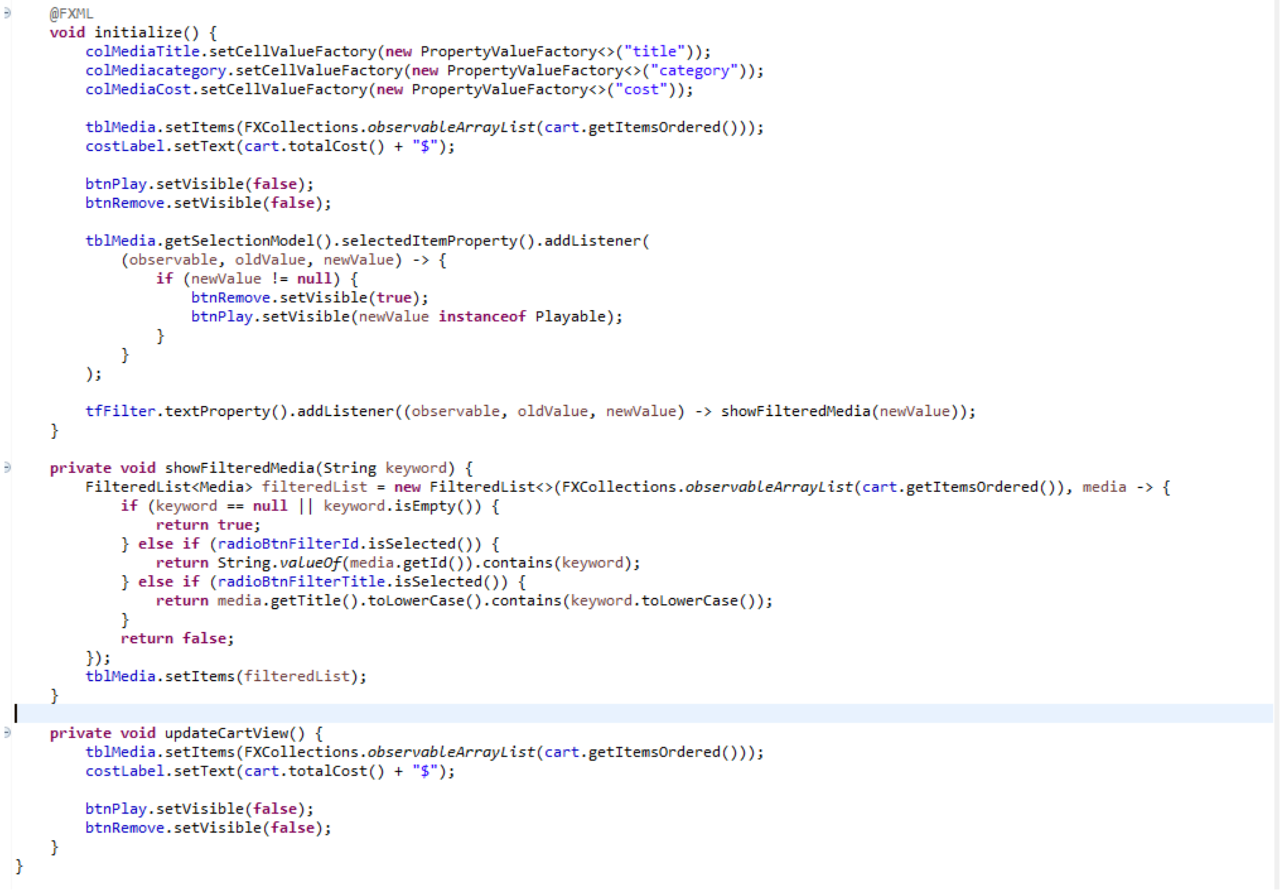


```java
void initialize() {
    colMediaTitle.setCellValueFactory(new PropertyValueFactory<>("title"));
    colMediacategory.setCellValueFactory(new PropertyValueFactory<>("category"));
    colMediaCost.setCellValueFactory(new PropertyValueFactory<>("cost"));

    tblMedia.setItems(FXCollections.observableArrayList(cart.getItemsOrdered()));
    costLabel.setText(cart.totalCost() + "$");

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

    tblMedia.getSelectionModel().selectedItemProperty().addListener(
        (observable, oldValue, newValue) -> {
            if (newValue != null) {
                btnRemove.setVisible(true);
                btnPlay.setVisible(newValue instanceof Playable);
            }
        }
    );

    tfFilter.textProperty().addListener((observable, oldValue, newValue) -> showFilteredMedia(newValue));
}
```

Figure 6.1: CartScreenController 2

7. Deleting a media

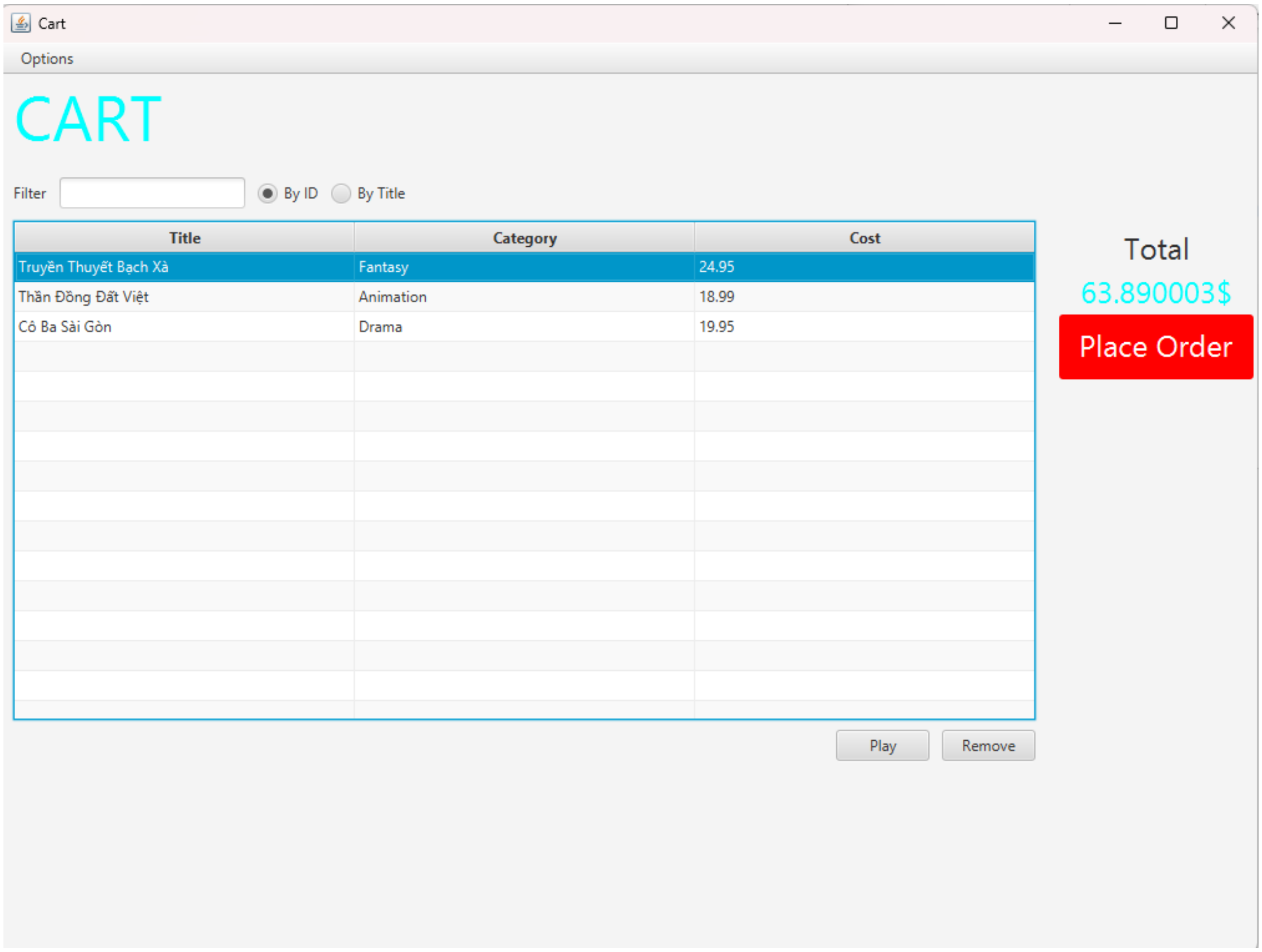
7.1 Code



```java
@FXML
void initialize() {
    colMediaTitle.setCellValueFactory(new PropertyValueFactory<>("title"));
    colMediacategory.setCellValueFactory(new PropertyValueFactory<>("category"));
    colMediaCost.setCellValueFactory(new PropertyValueFactory<>("cost"));

    tblMedia.setItems(FXCollections.observableArrayList(cart.getItemsOrdered()));
    costLabel.setText(cart.totalCost() + "$");

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

    tblMedia.getSelectionModel().selectedItemProperty().addListener(
        (observable, oldValue, newValue) -> {
            if (newValue != null) {
                btnRemove.setVisible(true);
                btnPlay.setVisible(newValue instanceof Playable);
            }
        }
    );

    tfFilter.textProperty().addListener((observable, oldValue, newValue) -> showFilteredMedia(newValue));
}

private void showFilteredMedia(String keyword) {
    FilteredList<Media> filteredList = new FilteredList<>(FXCollections.observableArrayList(cart.getItemsOrdered()), media -> {
        if (keyword == null || keyword.isEmpty()) {
            return true;
        } else if (radioBtnFilterId.isSelected()) {
            return String.valueOf(media.getId()).contains(keyword);
        } else if (radioBtnFilterTitle.isSelected()) {
            return media.getTitle().toLowerCase().contains(keyword.toLowerCase());
        }
        return false;
    });
    tblMedia.setItems(filteredList);
}

private void updateCartView() {
    tblMedia.setItems(FXCollections.observableArrayList(cart.getItemsOrdered()));
    costLabel.setText(cart.totalCost() + "$");

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

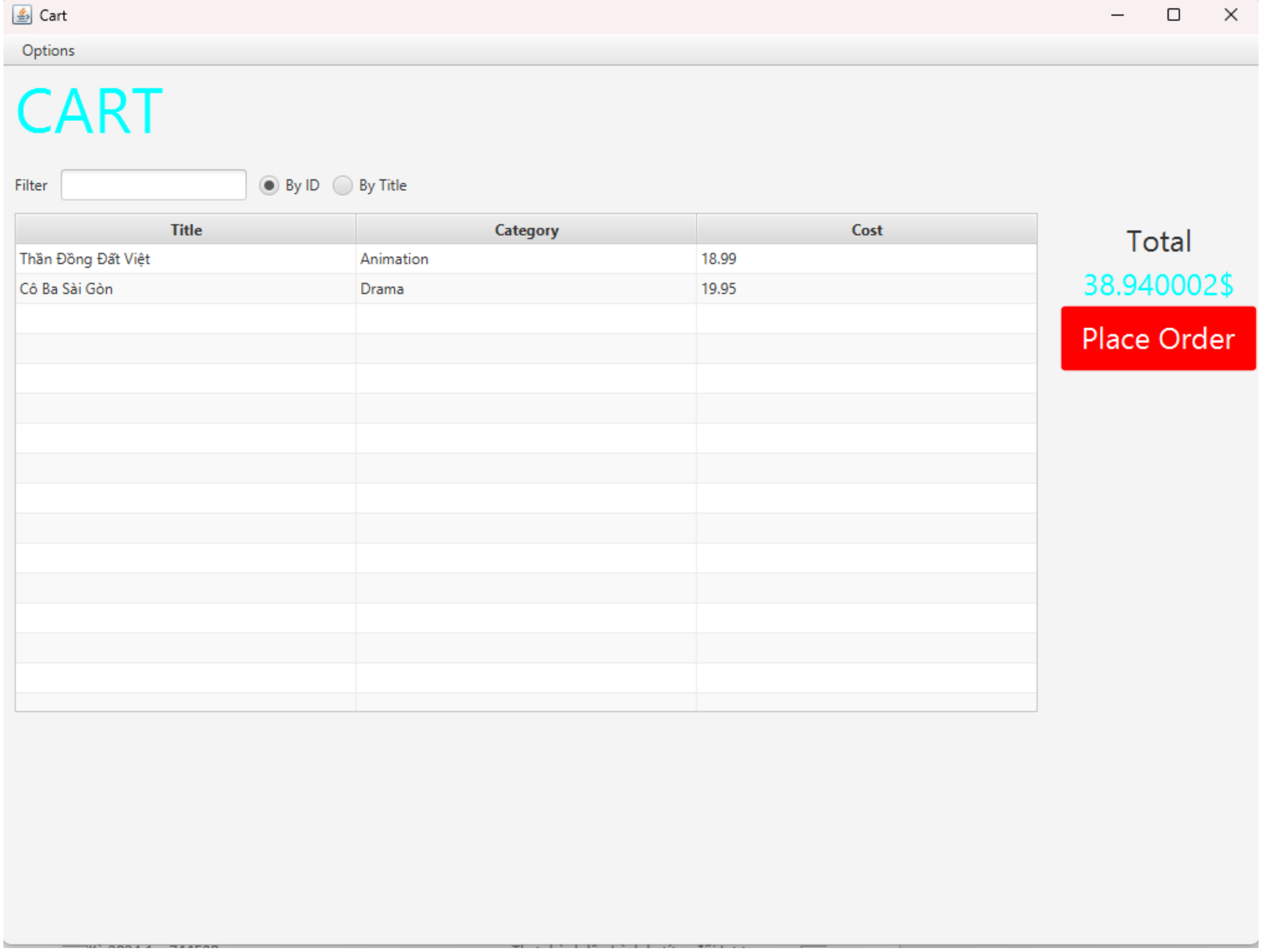
Figure 7.1: btnRemovePressed Method

7.2 Demo



Title	Category	Cost
Truyền Thuyết Bạch Xà	Fantasy	24.95
Thần Đồng Đất Việt	Animation	18.99
Cô Ba Sài Gòn	Drama	19.95

Figure 7.2: button Remove



CART

Filter ● By ID ○ By Title

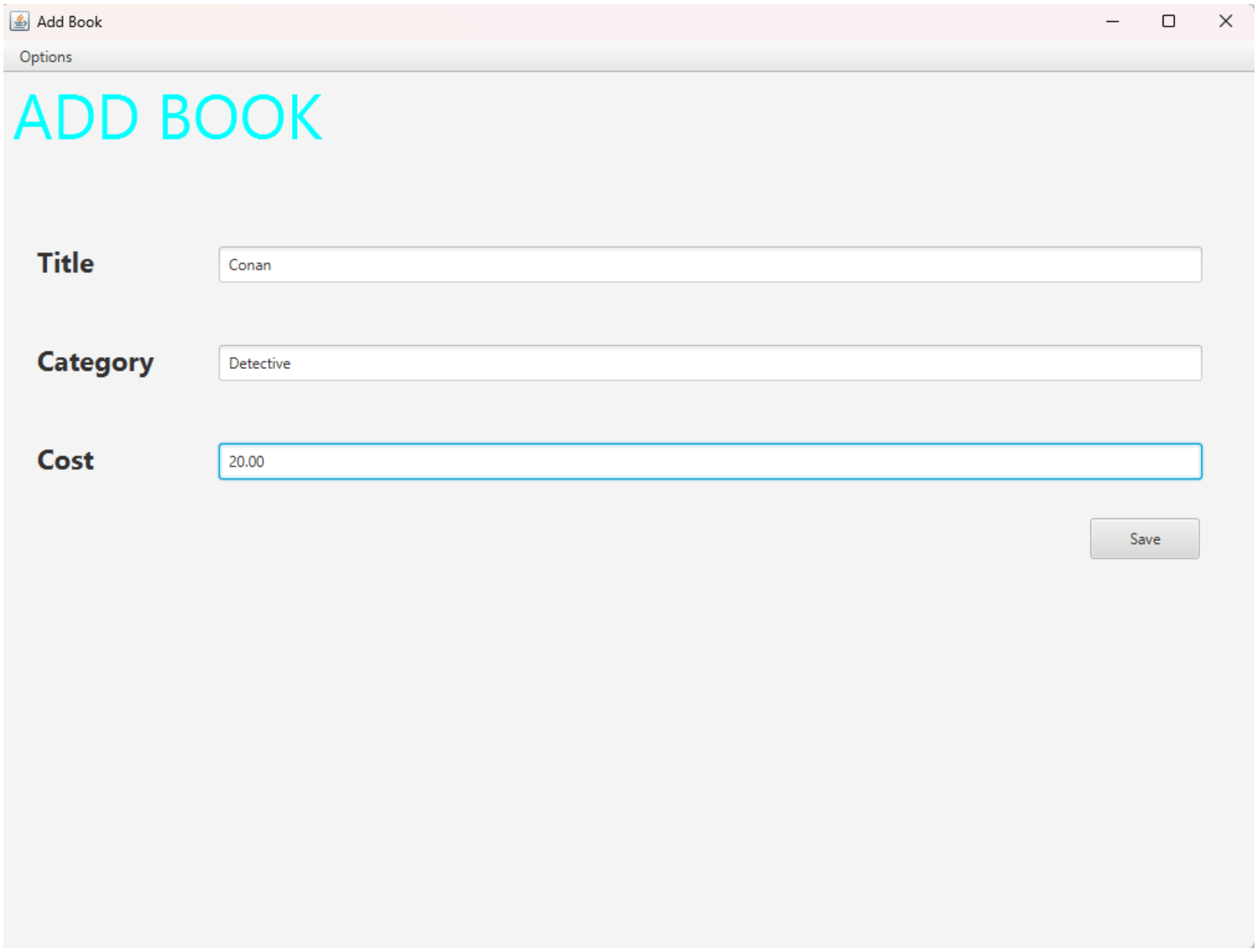
Title	Category	Cost
Thần Đồng Đất Việt	Animation	18.99
Cô Ba Sài Gòn	Drama	19.95

Total
38.940002$

Figure 7.3: button Remove

8. Complete the Aims GUI application

8.1



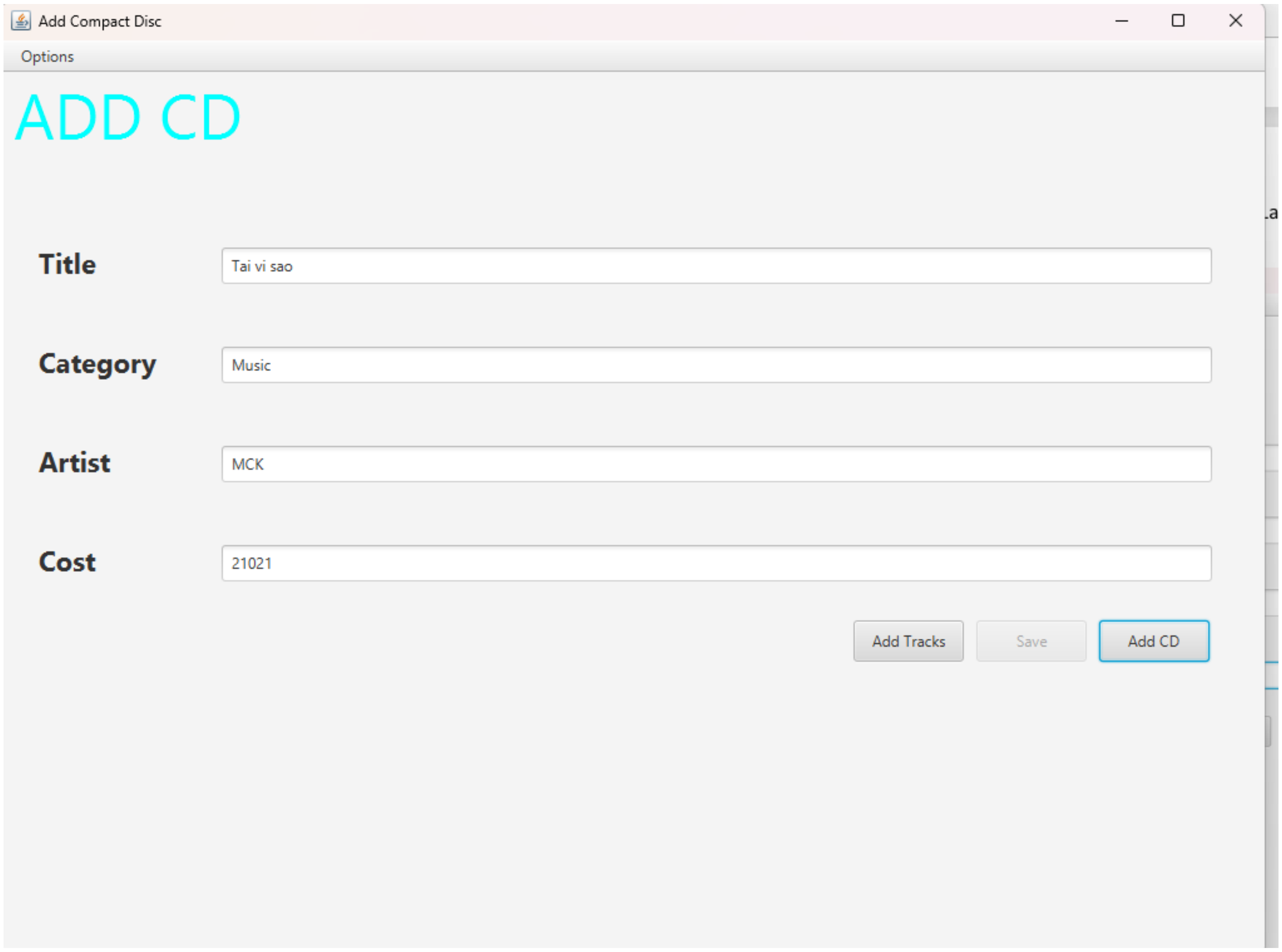
ADD BOOK

Title: Conan

Category: Detective

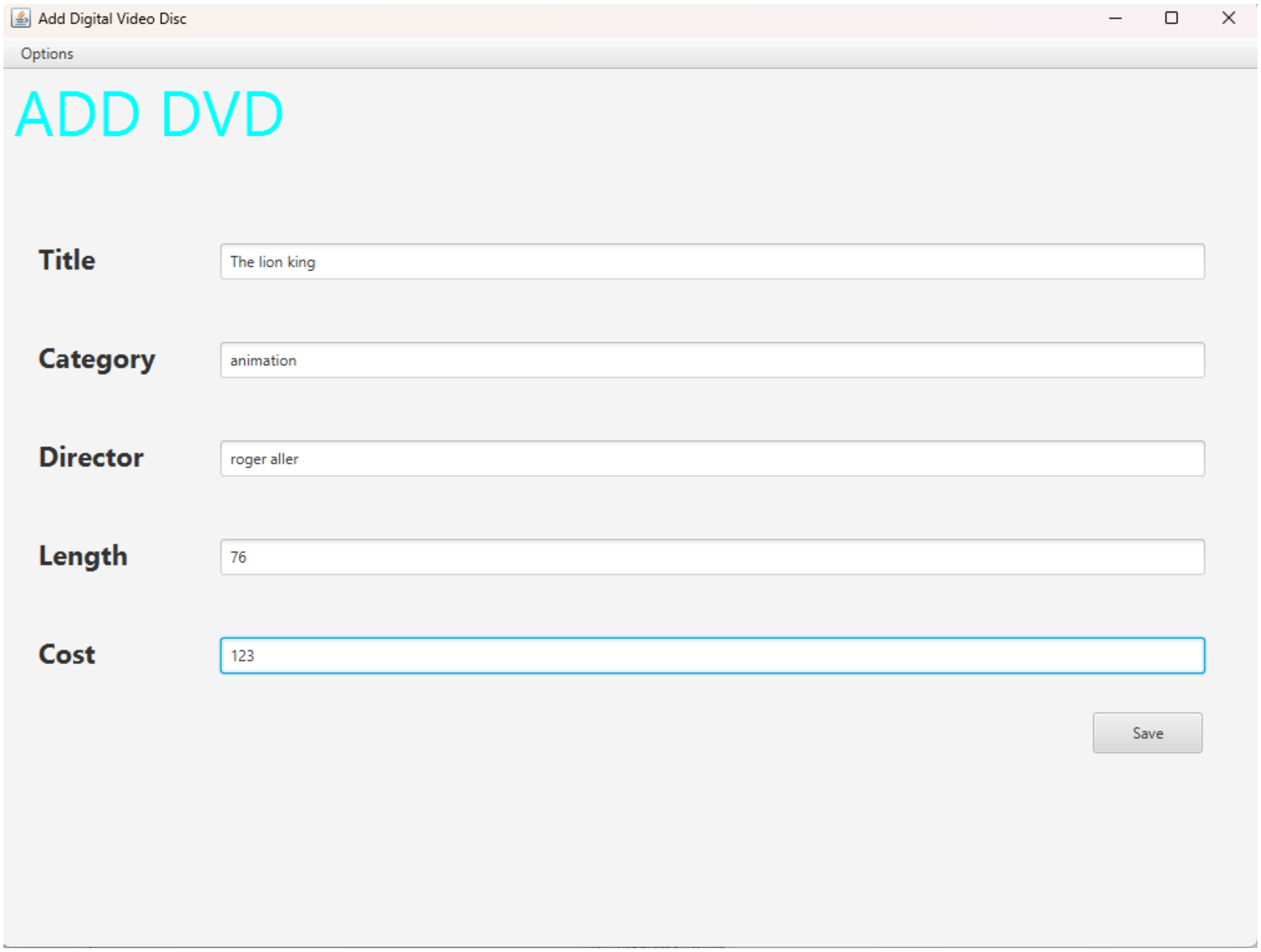
Cost: 20.00

Figure 8.1: Add Book



Title	Tai vi sao
Category	Music
Artist	MCK
Cost	21021

Figure 8.2: Add CD



ADD DVD

Title: The lion king

Category: animation

Director: roger aller

Length: 76

Cost: 123

Figure 8.3: Add DVD