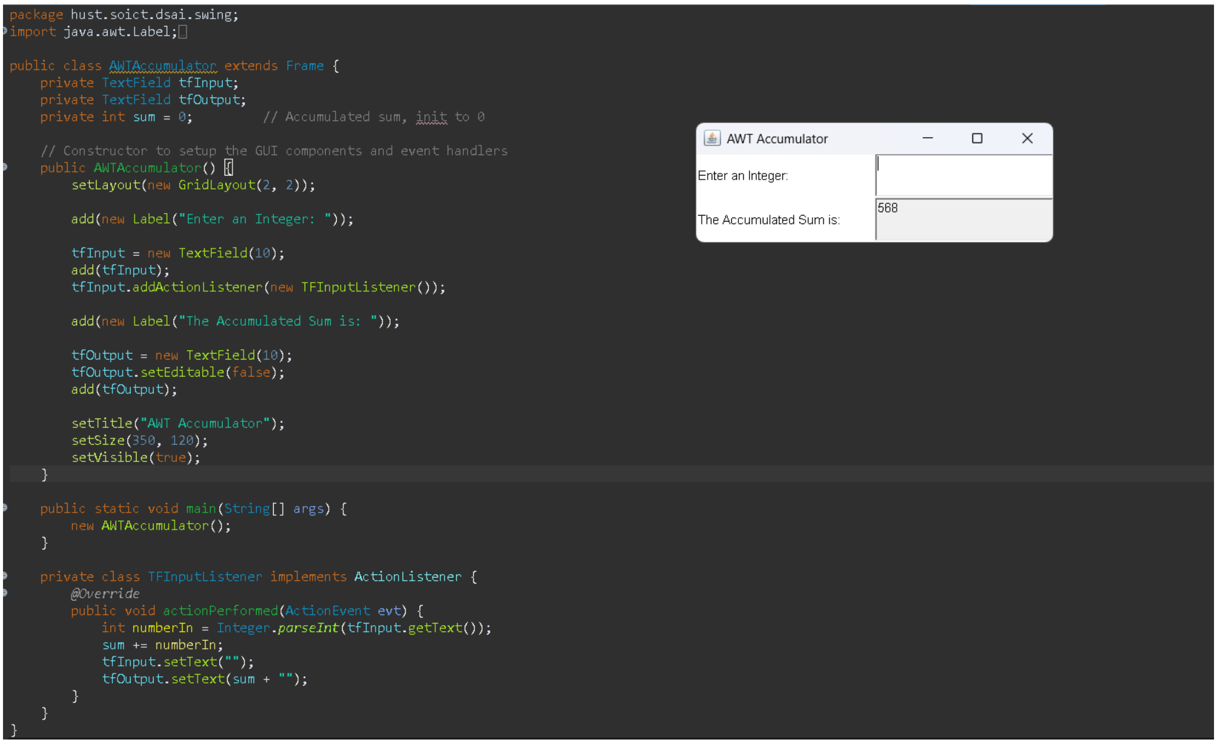
Báo cáo Object Oriented Programming – lab05

1, Swing Component

- AWTAccumulator Source code and result:



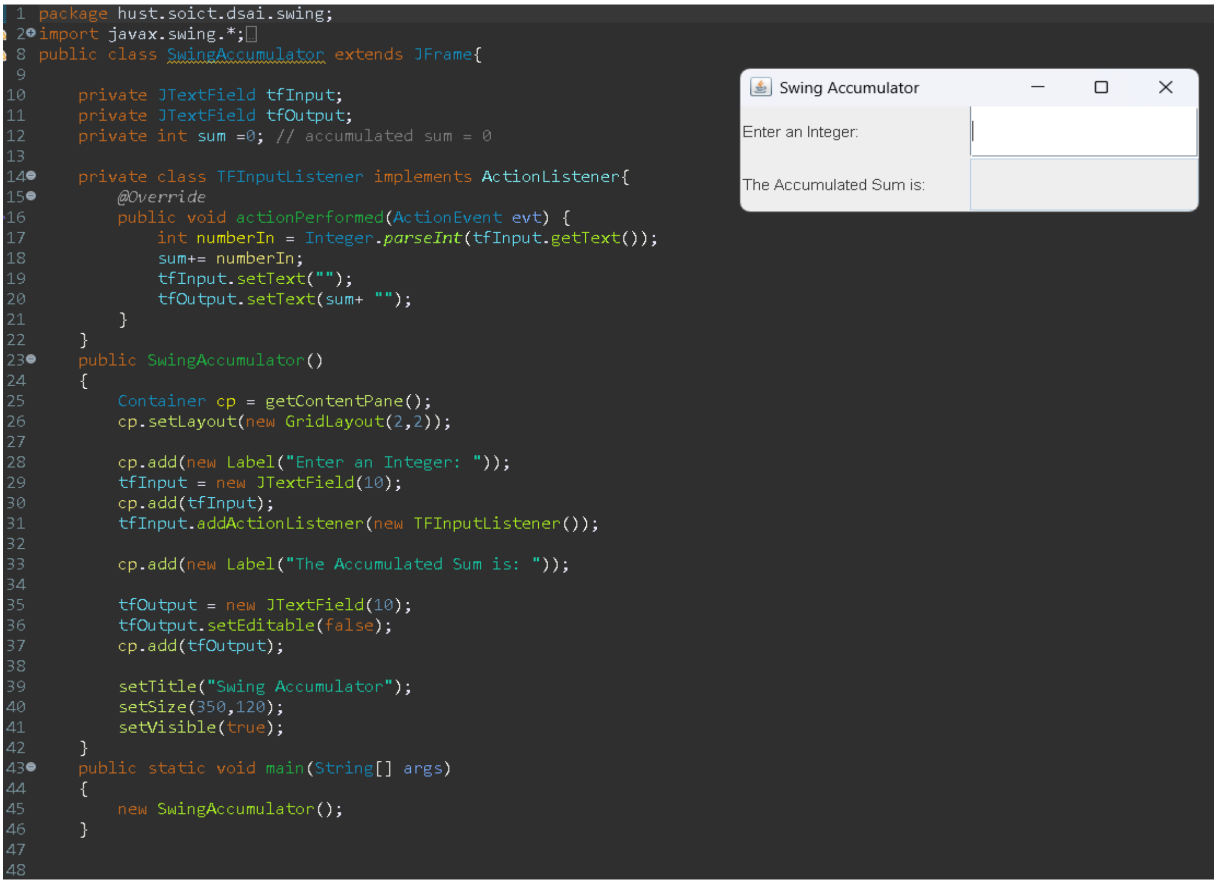
```java
package hust.soict.dsai.swing;
import java.awt.Label;

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

Enter an Integer:

The Accumulated Sum is: 568

2, Create a graphical user interface for AIMS with Swing

2.1 Swing Accumulator soruce code and result

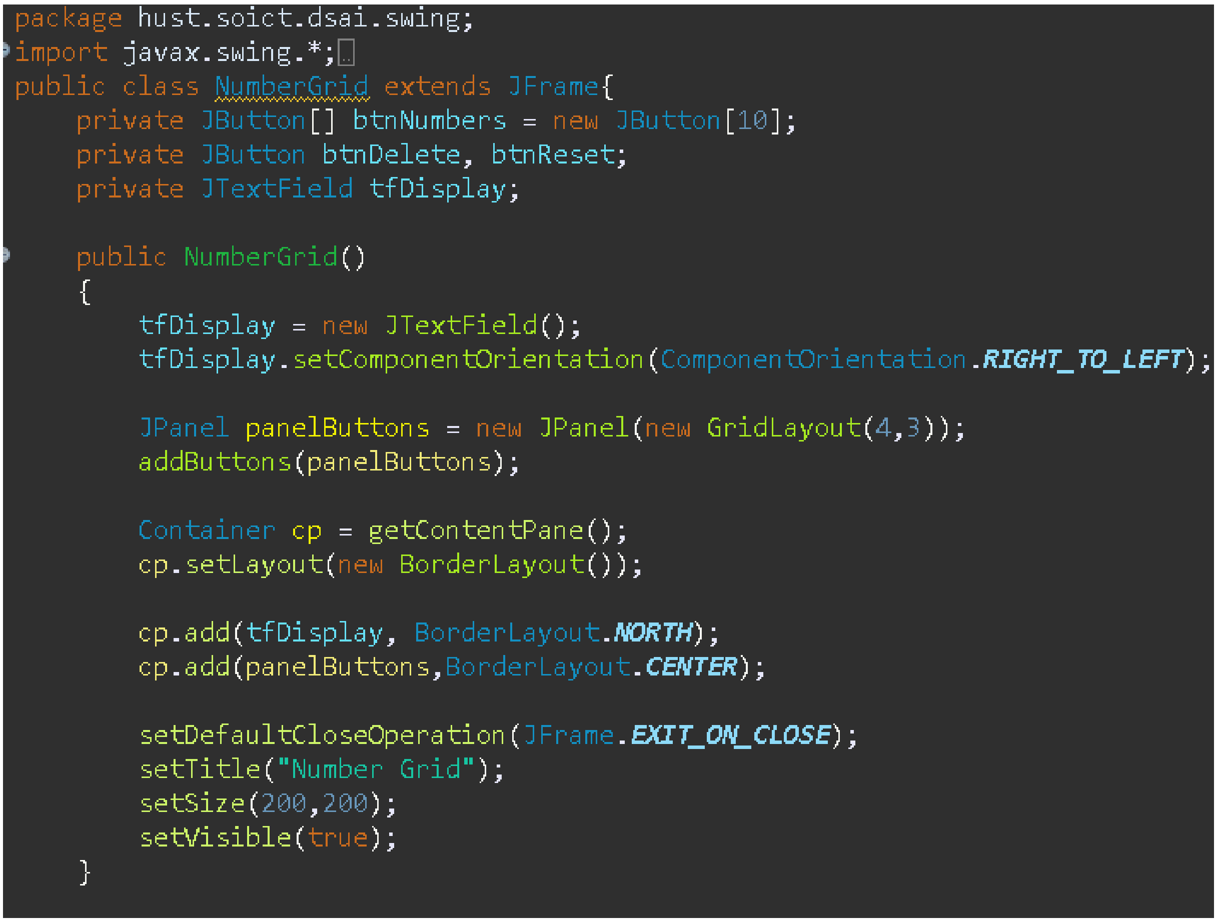


```java
package hust.soict.dsai.swing;
import javax.swing.*;
public class SwingAccumulator extends JFrame{

    private JTextField tfInput;
    private JTextField tfOutput;
    private int sum =0; // accumulated sum = 0

    private class TFInputListener implements ActionListener{
        @Override
        public void actionPerformed(ActionEvent evt) {
            int numberIn = Integer.parseInt(tfInput.getText());
            sum+= numberIn;
            tfInput.setText("");
            tfOutput.setText(sum+ "");
        }
    }
    public SwingAccumulator()
    {
        Container cp = getContentPane();
        cp.setLayout(new GridLayout(2,2));

        cp.add(new Label("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();
    }

```

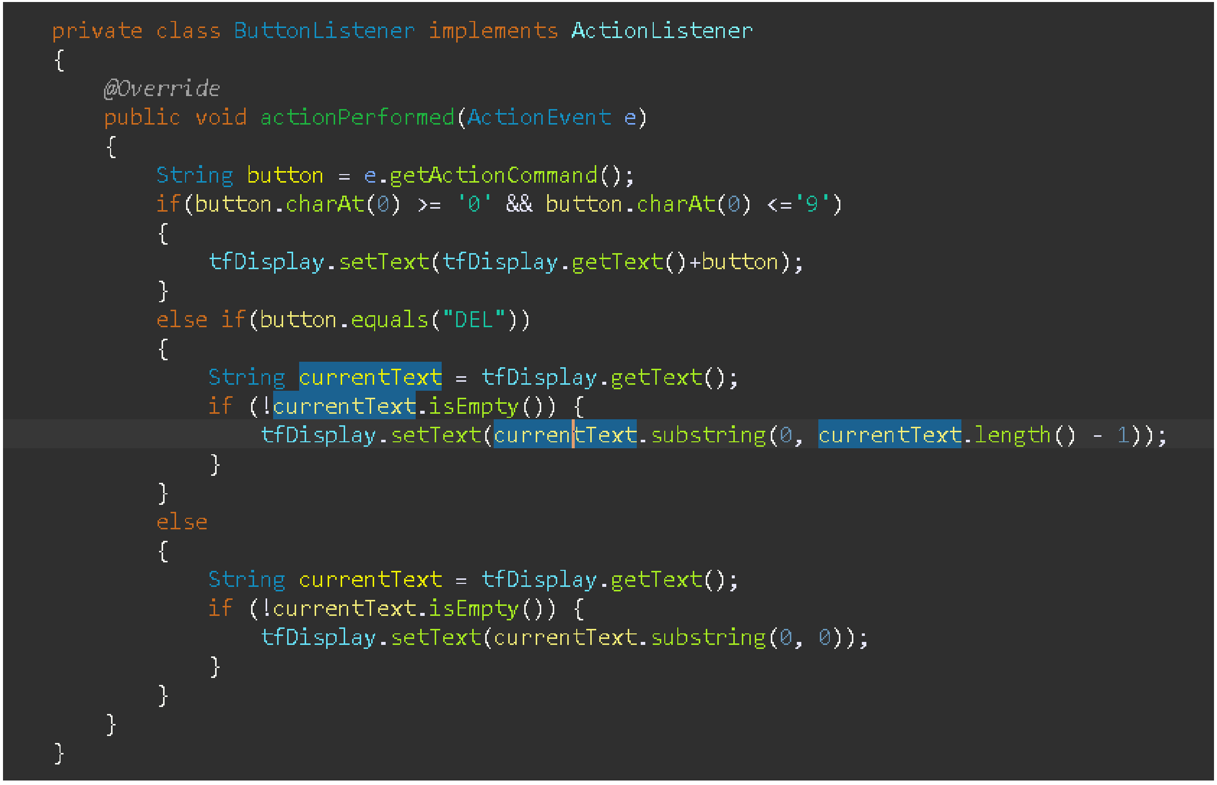
Enter an Integer:

The Accumulated Sum is:

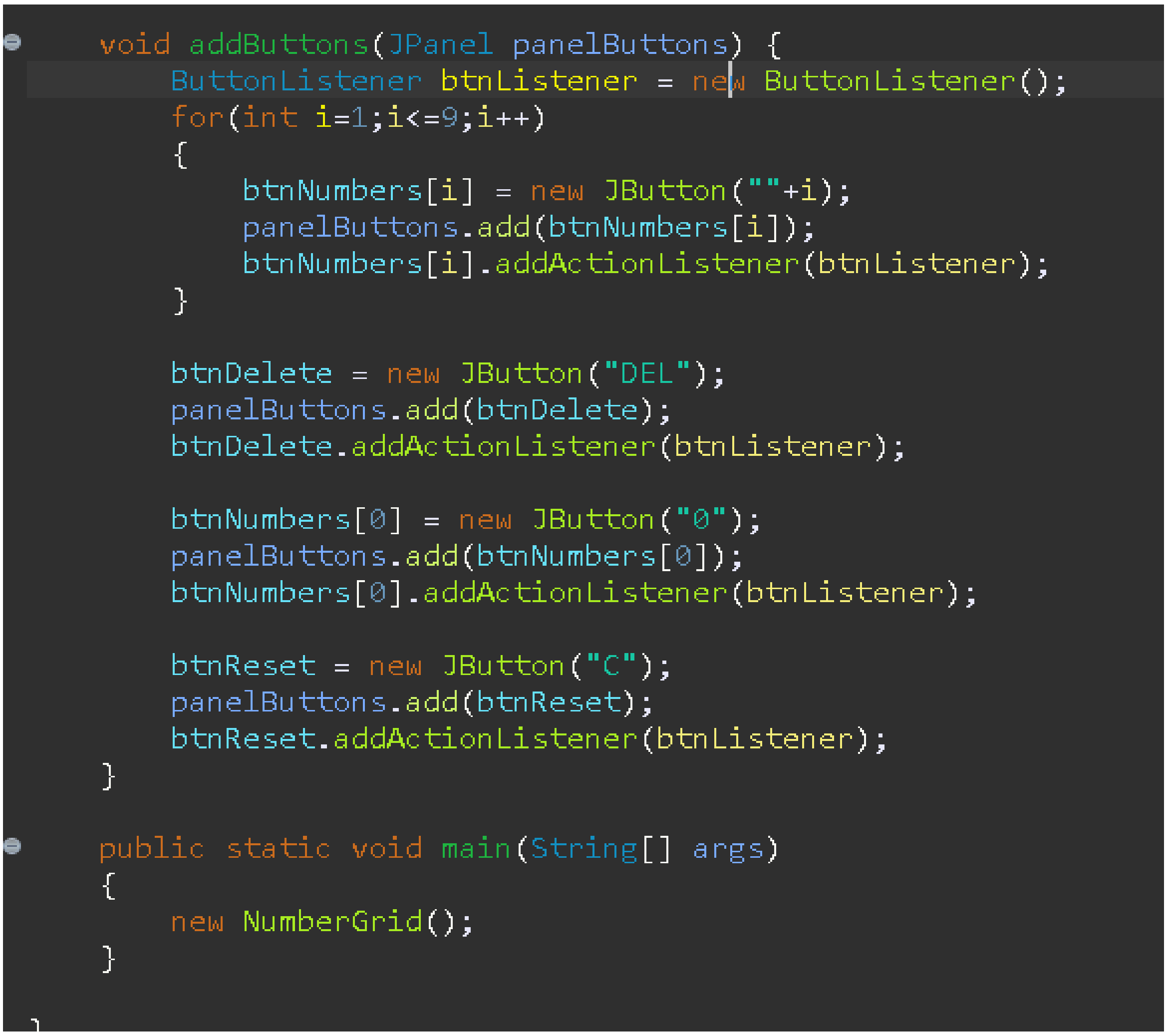
2.2, NumberGrid source code and result



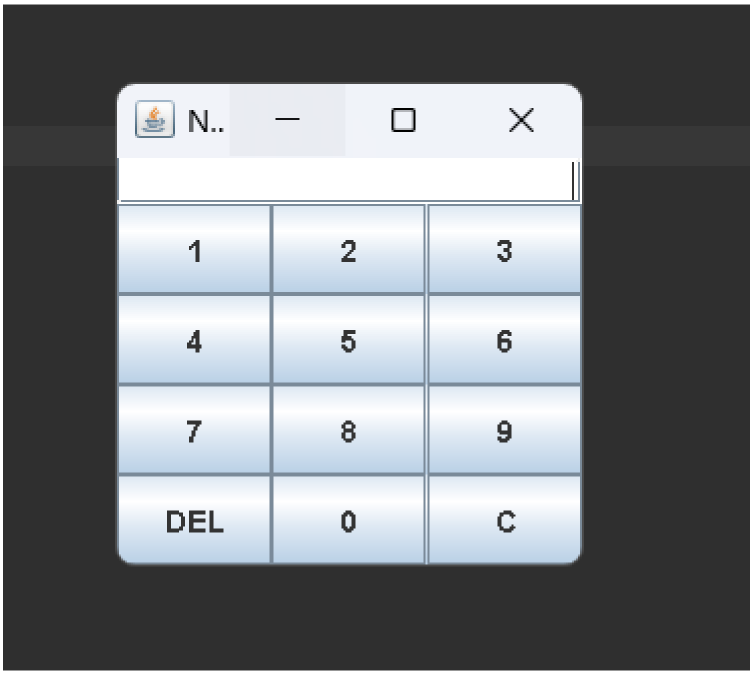
```
package hust.soict.dsai.swing;
import javax.swing.*;
public class NumberGrid extends JFrame{
    private JButton[] btnNumbers = new JButton[10];
    private JButton btnDelete, btnReset;
    private JTextField tfDisplay;

    public NumberGrid()
    {
        tfDisplay = new JTextField();
        tfDisplay.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);
    }
```



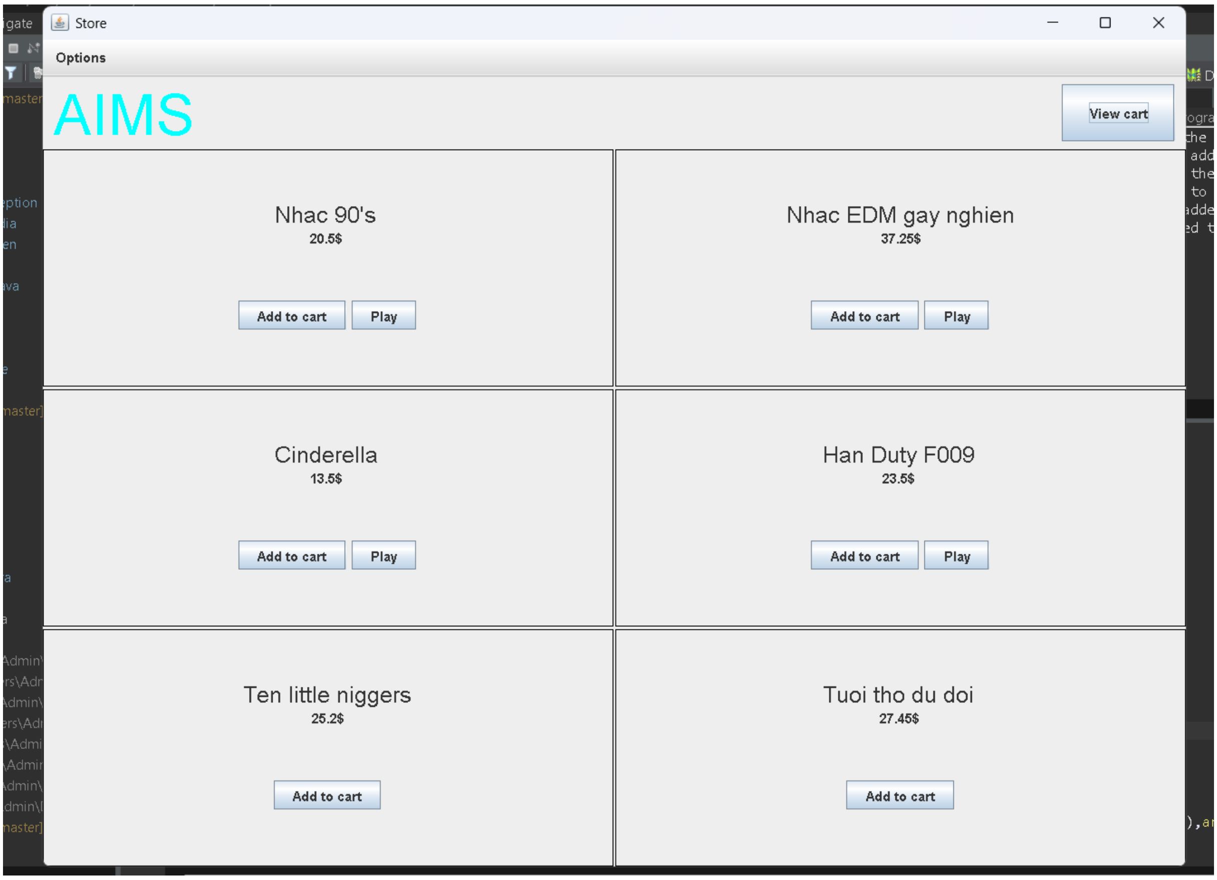
```
    private class ButtonListener implements ActionListener
    {
        @Override
        public void actionPerformed(ActionEvent e)
        {
            String button = e.getActionCommand();
            if(button.charAt(0) >= '0' && button.charAt(0) <='9')
            {
                tfDisplay.setText(tfDisplay.getText()+button);
            }
            else if(button.equals("DEL"))
            {
                String currentText = tfDisplay.getText();
                if (!currentText.isEmpty()) {
                    tfDisplay.setText(currentText.substring(0, currentText.length() - 1));
                }
            }
            else
            {
                String currentText = tfDisplay.getText();
                if (!currentText.isEmpty()) {
                    tfDisplay.setText(currentText.substring(0, 0));
                }
            }
        }
    }
```



```
    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);
    }

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

}
```



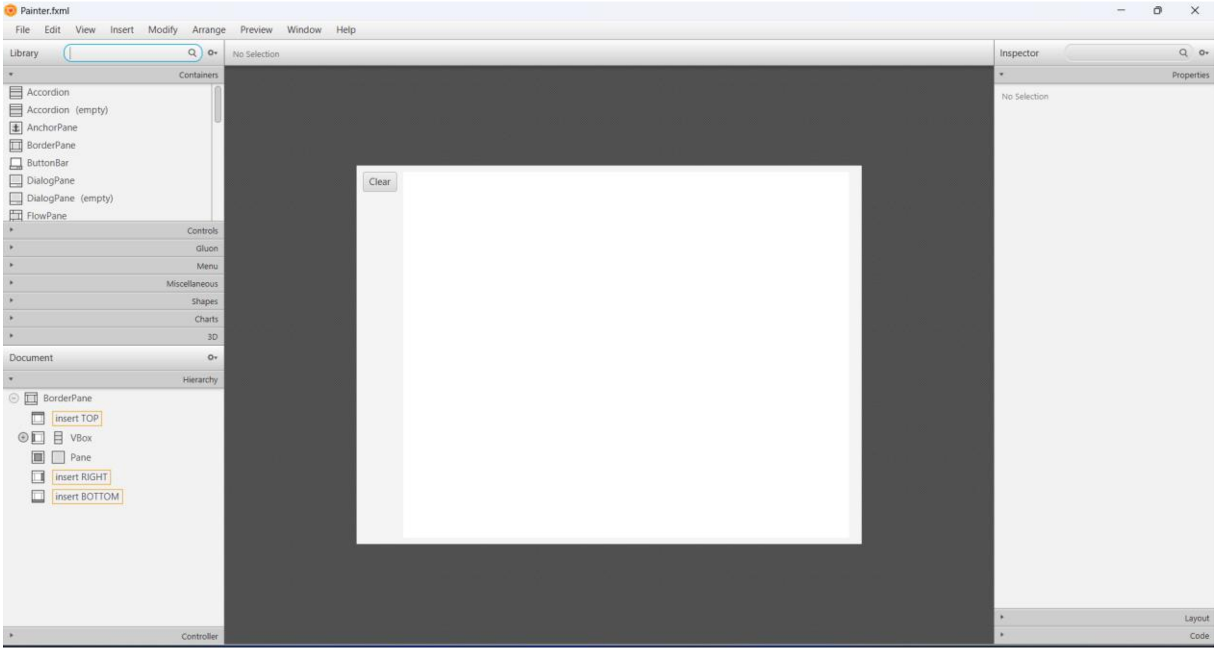
2.3 View store Screen



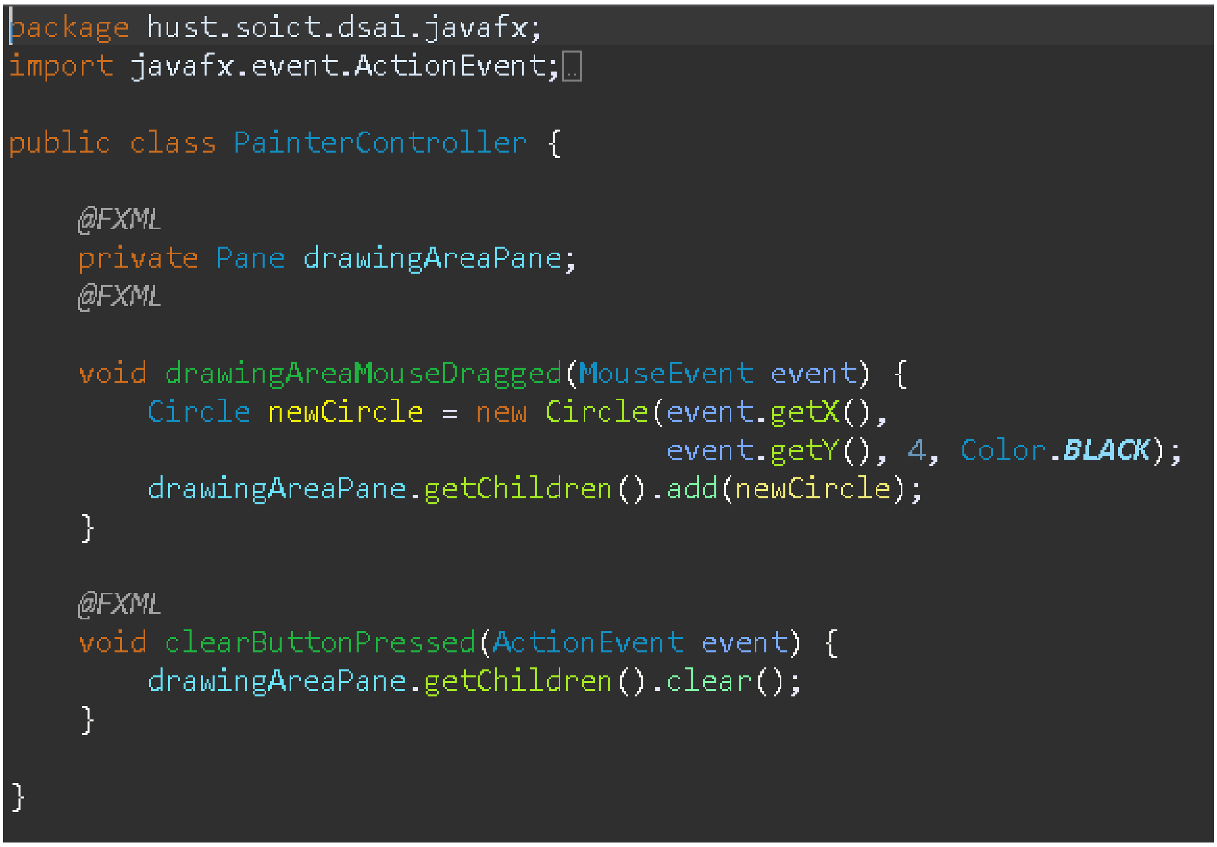
3. JavaFX API

3.1, Painter Application

- Painter.fxml opened in Scene Buider

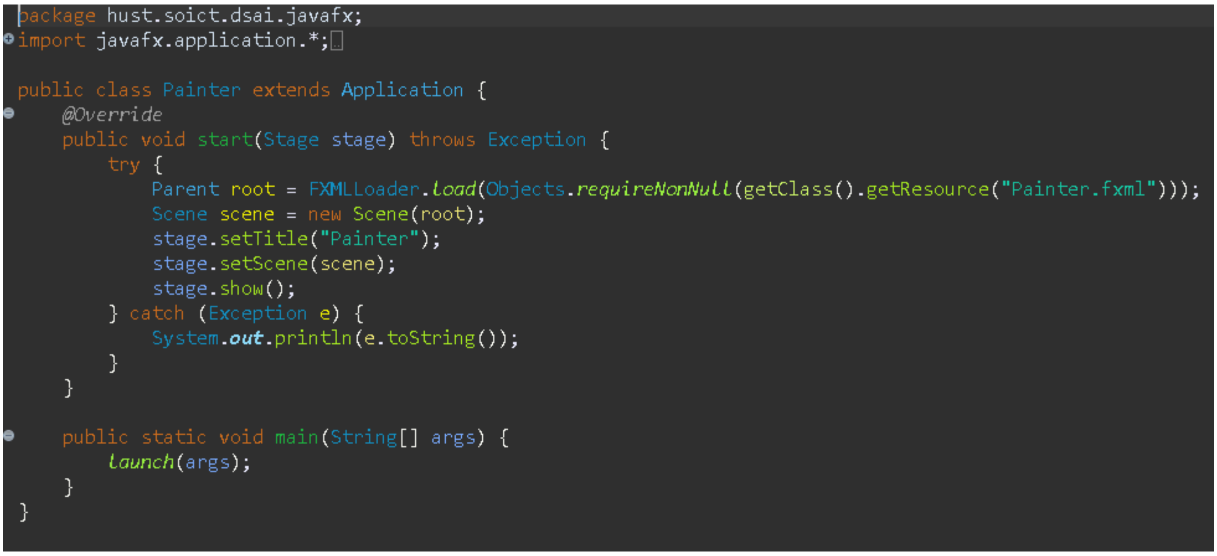


- Painter Controller Code:



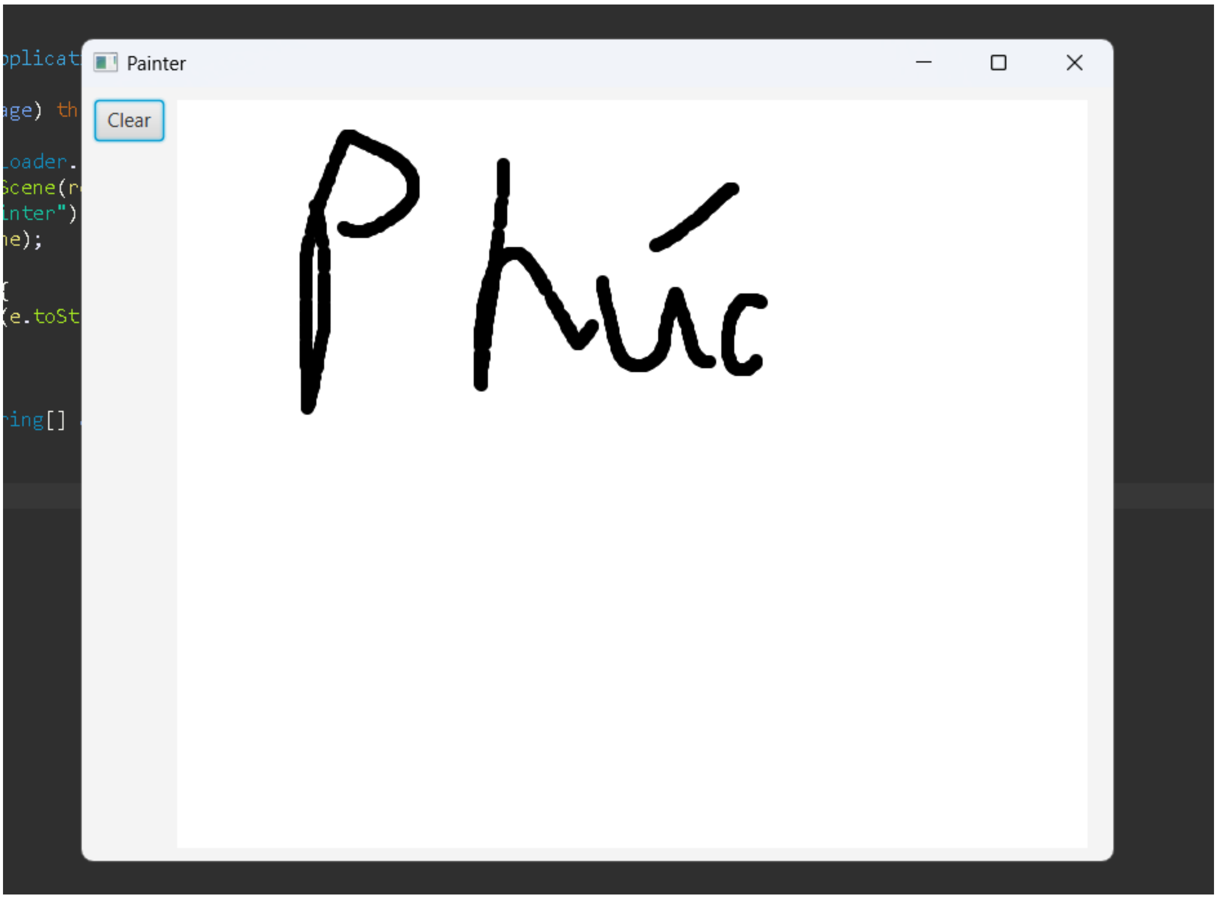
```
package hust.soict.dsai.javafx;
import javafx.event.ActionEvent;

public class PainterController {

    @FXML
    private Pane drawingAreaPane;
    @FXML

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

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

}
```

- Painter Application



```
package hust.soict.dsai.javafx;
import javafx.application.*;

public class Painter extends Application {
    @Override
    public void start(Stage stage) throws Exception {
        try {
            Parent root = FXMLLoader.load(Objects.requireNonNull(getClass().getResource("Painter.fxml")));
            Scene scene = new Scene(root);
            stage.setTitle("Painter");
            stage.setScene(scene);
            stage.show();
        } catch (Exception e) {
            System.out.println(e.toString());
        }
    }

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

- Result:

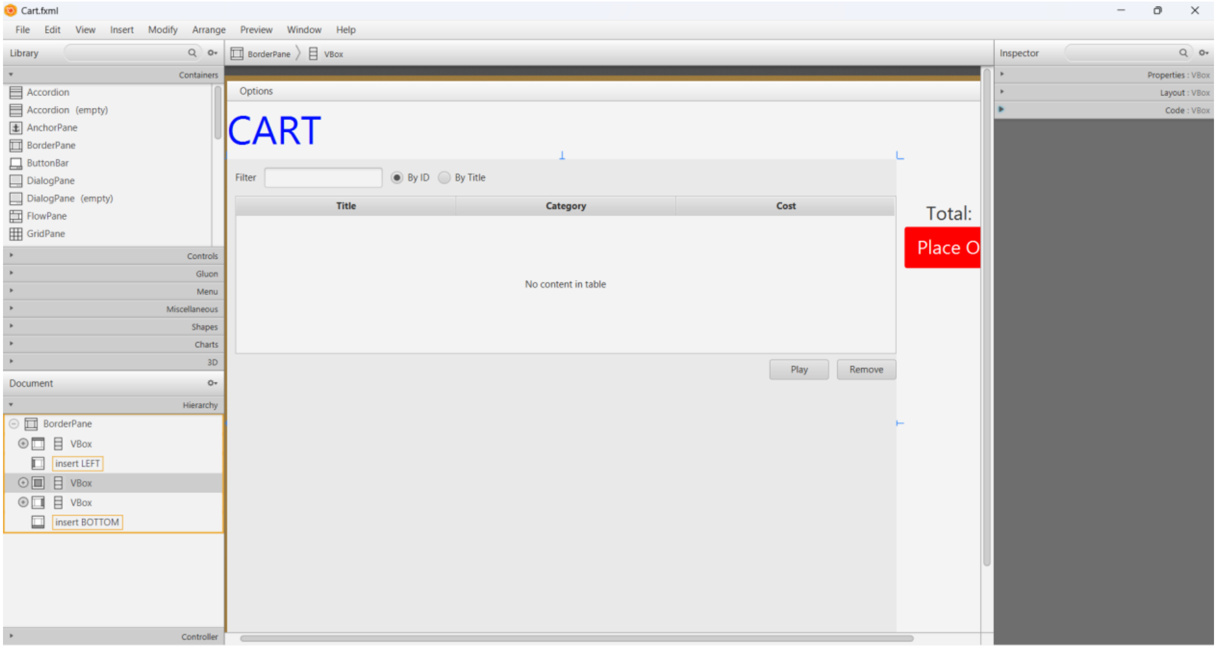


```
pplicat
age) th
oader.
cene(r
inter")
e);
{
(e.toSt

ing[]
```

3.2, Setting up the View Cart Screen with ScreenBuilder

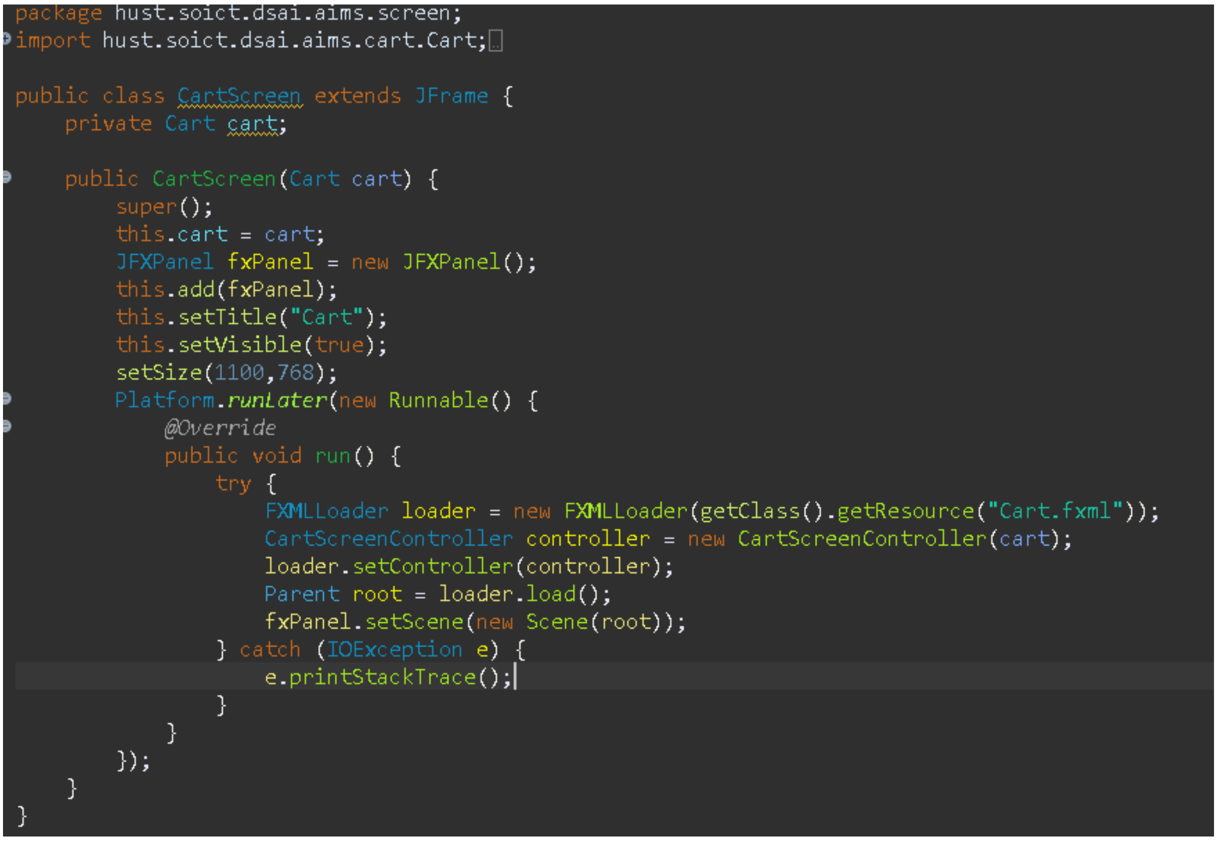
- Setting up the BorderPane&Top Area

Total Result:



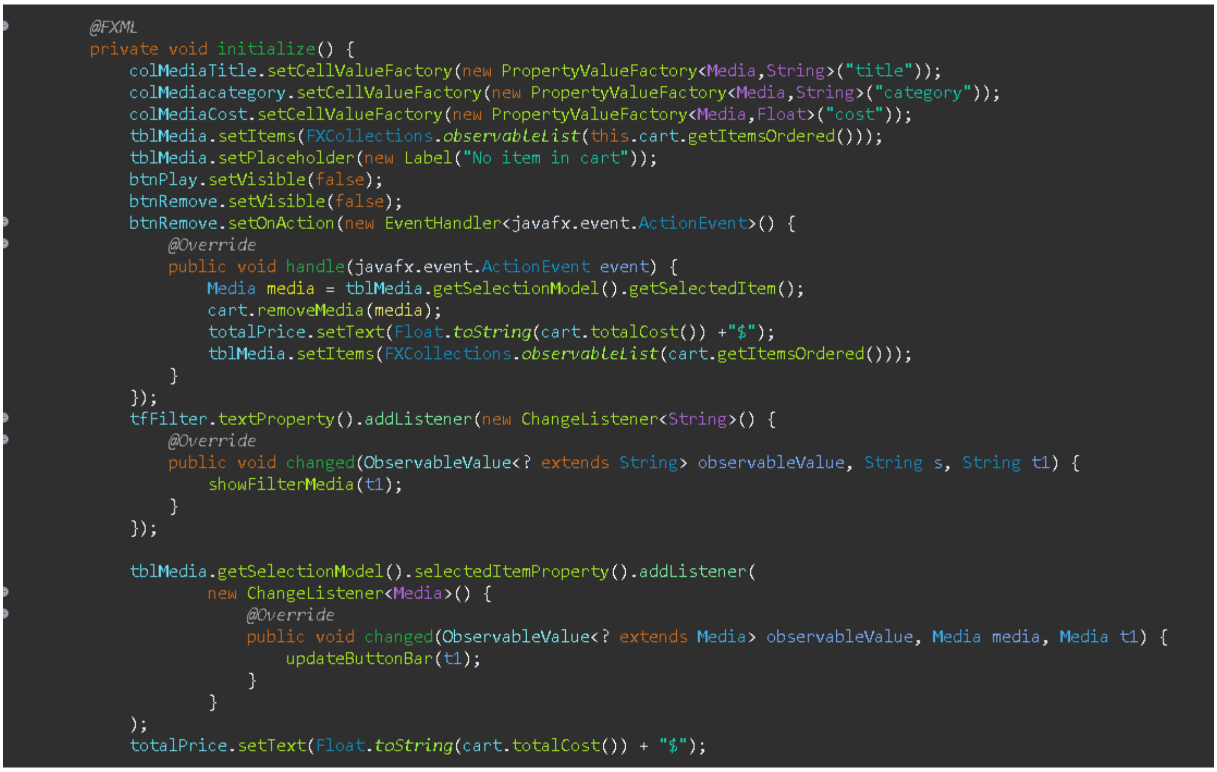
4, Integrating JavaFX into Swing Application

4.1 Source Code for CartScreen

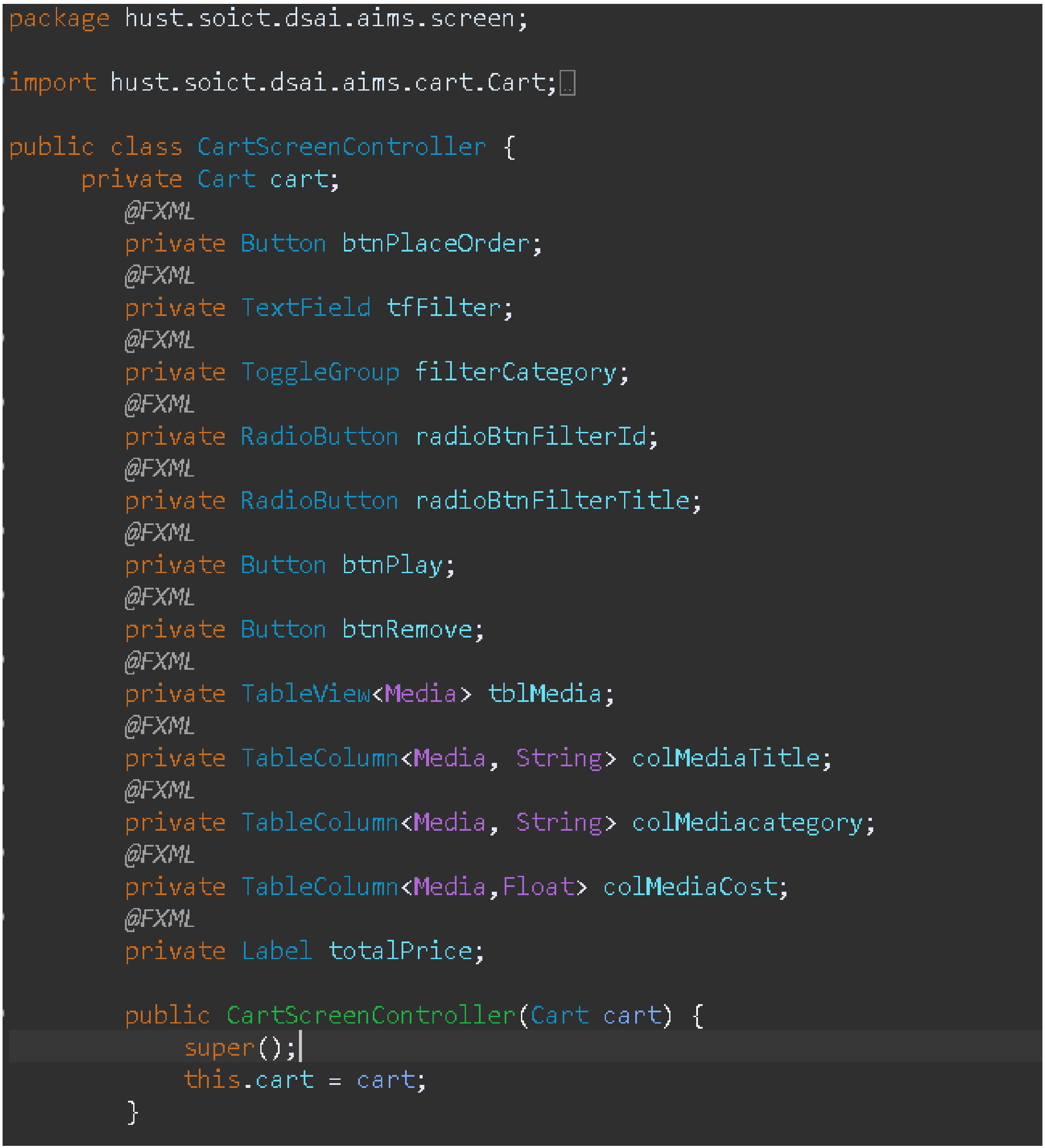


```
package hust.soict.dsai.aims.screen;
import hust.soict.dsai.aims.cart.Cart;

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);
        setSize(1100,768);
        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();
                }
            }
        });
    }
}
```

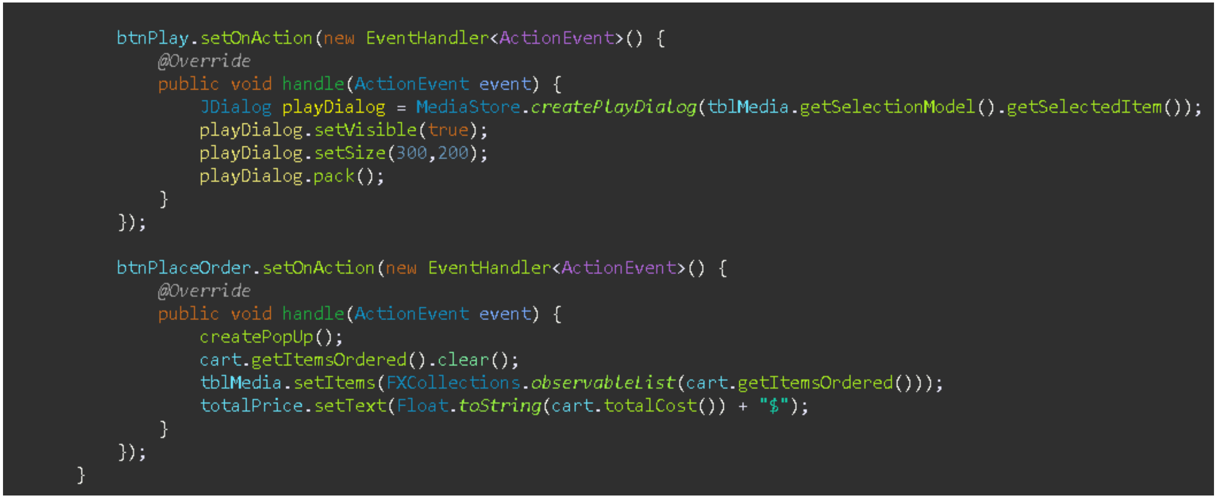
4.2 Source code for CartScreenController



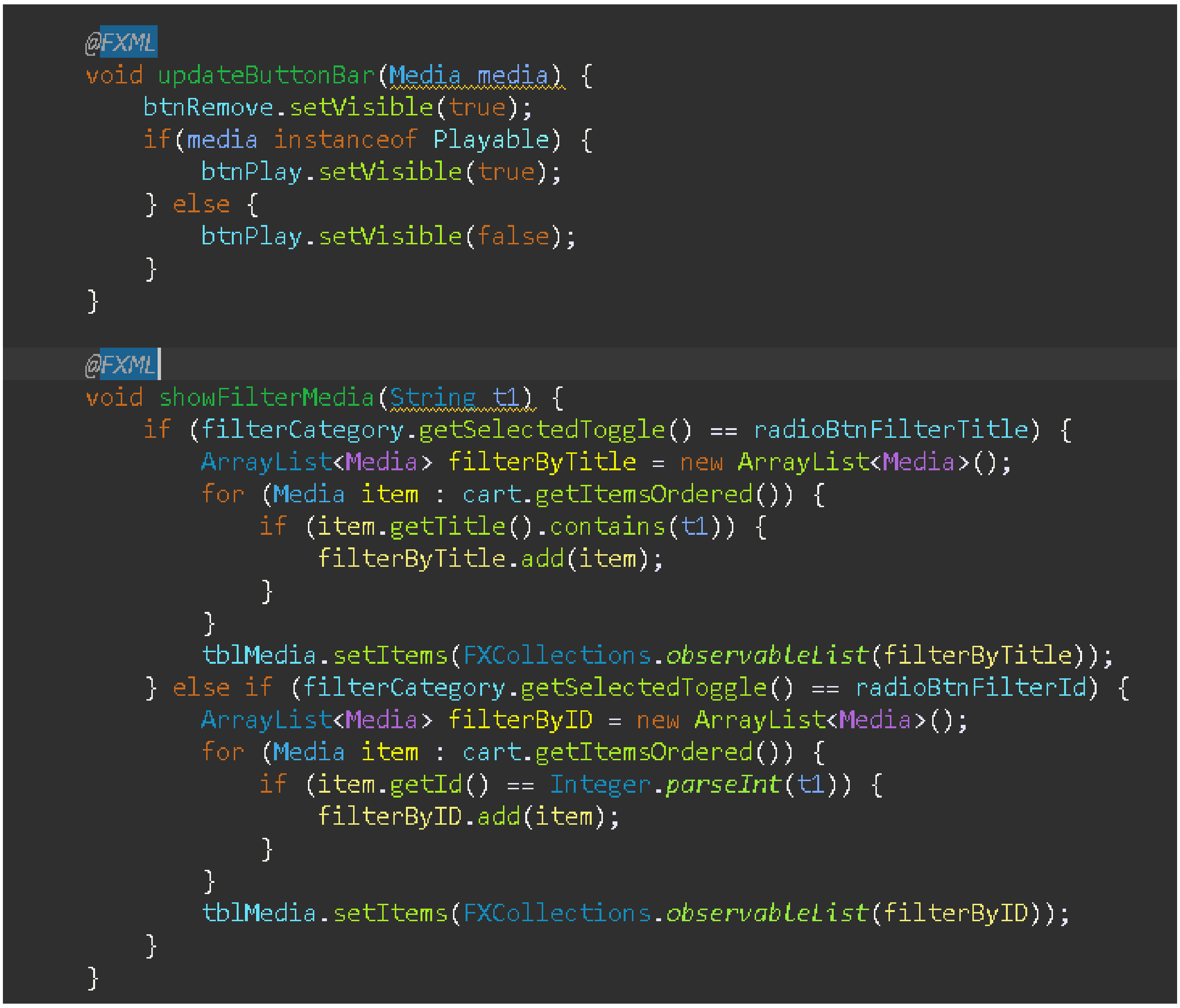
```
    @FXML
    private void initialize() {
        colMediaTitle.setCellValueFactory(new PropertyValueFactory<Media,String>("title"));
        colMediacategory.setCellValueFactory(new PropertyValueFactory<Media,String>("category"));
        colMediaCost.setCellValueFactory(new PropertyValueFactory<Media,Float>("cost"));
        tblMedia.setItems(FXCollections.observableList(this.cart.getItemsOrdered()));
        tblMedia.setPlaceholder(new Label("No item in cart"));
        btnPlay.setVisible(false);
        btnRemove.setVisible(false);
        btnRemove.setOnAction(new EventHandler<javafx.event.ActionEvent>() {
            @Override
            public void handle(javafx.event.ActionEvent event) {
                Media media = tblMedia.getSelectionModel().getSelectedItem();
                cart.removeMedia(media);
                totalPrice.setText(Float.toString(cart.totalCost()) +"$");
                tblMedia.setItems(FXCollections.observableList(cart.getItemsOrdered()));
            }
        });
        tfFilter.textProperty().addListener(new ChangeListener<String>() {
            @Override
            public void changed(ObservableValue<? extends String> observableValue, String s, String t1) {
                showFilterMedia(t1);
            }
        });

        tblMedia.getSelectionModel().selectedItemProperty().addListener(
                new ChangeListener<Media>() {
                    @Override
                    public void changed(ObservableValue<? extends Media> observableValue, Media media, Media t1) {
                        updateButtonBar(t1);
                    }
                }
        );
        totalPrice.setText(Float.toString(cart.totalCost()) + "$");
```



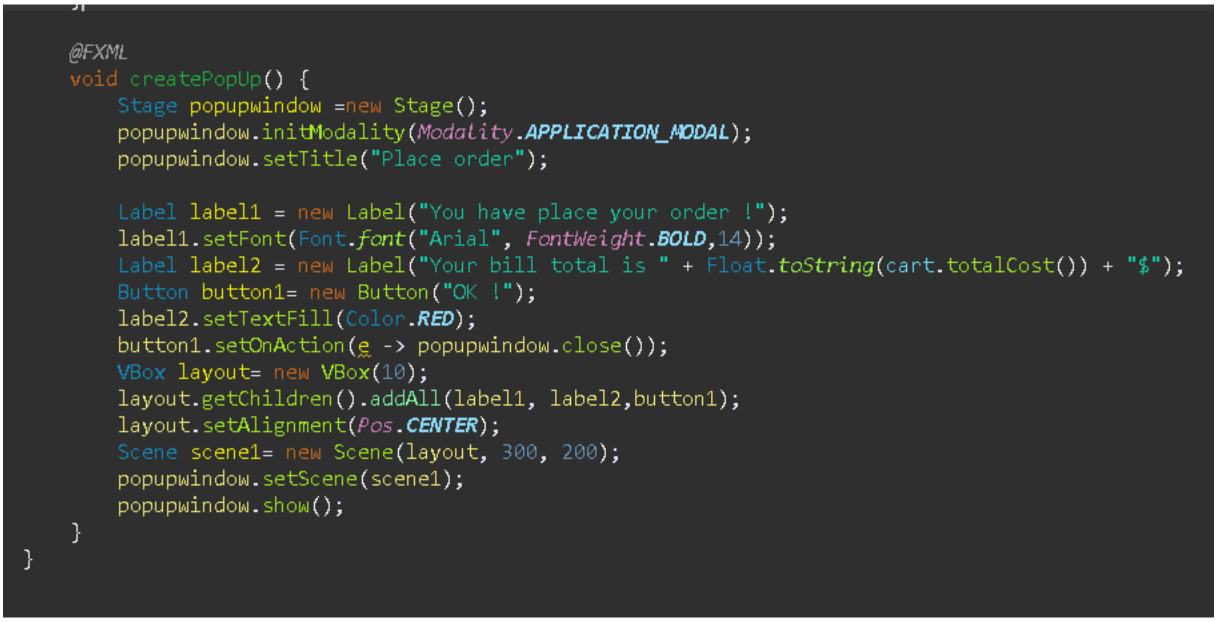
```
package hust.soict.dsai.aims.screen;

import hust.soict.dsai.aims.cart.Cart;

public class CartScreenController {
	private Cart cart;
		@FXML
		private Button btnPlaceOrder;
		@FXML
		private TextField tfFilter;
		@FXML
		private ToggleGroup filterCategory;
		@FXML
		private RadioButton radioBtnFilterId;
		@FXML
		private RadioButton radioBtnFilterTitle;
		@FXML
		private Button btnPlay;
		@FXML
		private Button btnRemove;
		@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 Label totalPrice;

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



```
        btnPlay.setOnAction(new EventHandler<ActionEvent>() {
            @Override
            public void handle(ActionEvent event) {
                JDialog playDialog = MediaStore.createPlayDialog(tblMedia.getSelectionModel().getSelectedItem());
                playDialog.setVisible(true);
                playDialog.setSize(300,200);
                playDialog.pack();
            }
        });

        btnPlaceOrder.setOnAction(new EventHandler<ActionEvent>() {
            @Override
            public void handle(ActionEvent event) {
                createPopUp();
                cart.getItemsOrdered().clear();
                tblMedia.setItems(FXCollections.observableList(cart.getItemsOrdered()));
                totalPrice.setText(Float.toString(cart.totalCost()) + "$");
            }
        });
    }
```



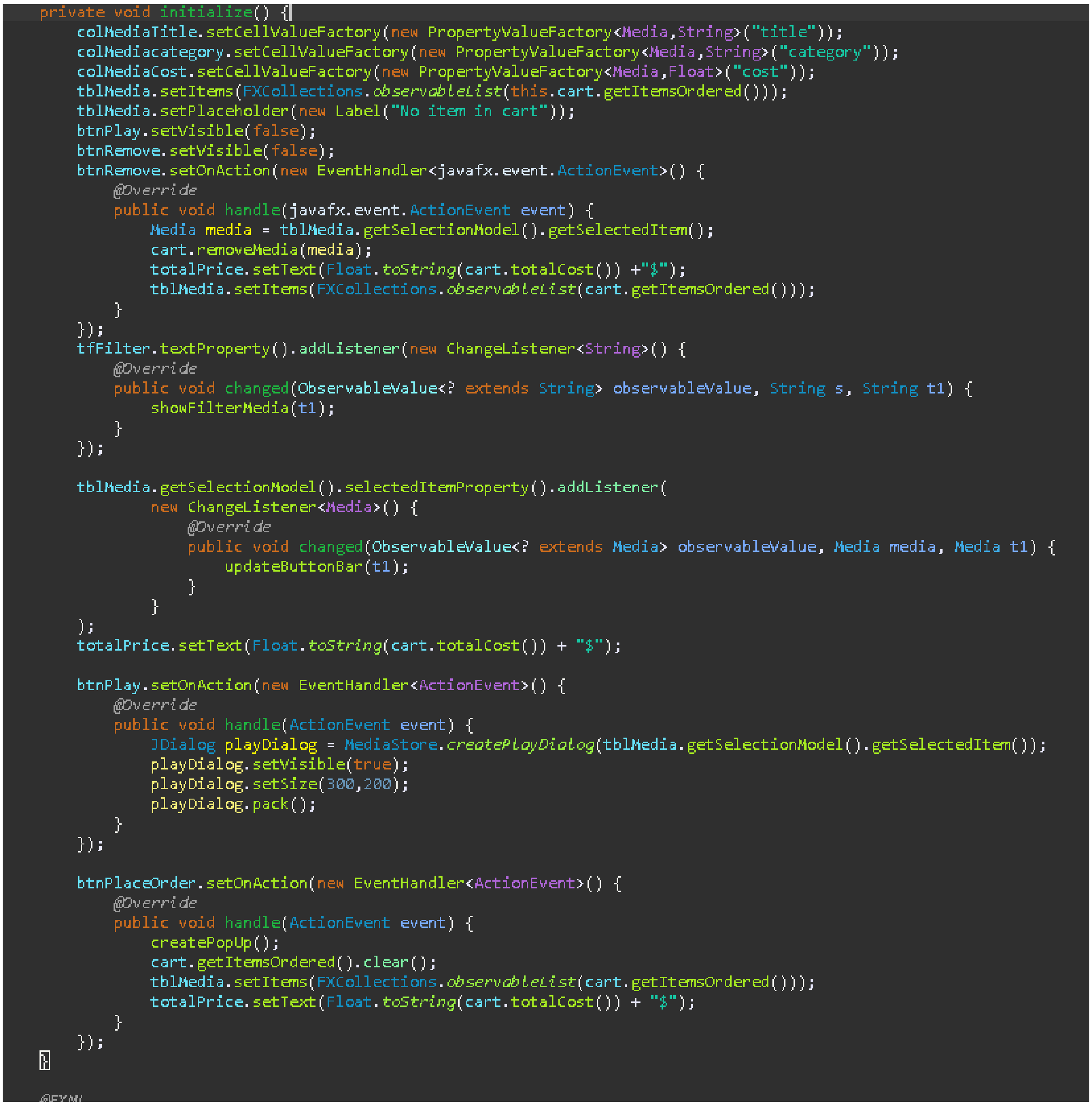
```
    @FXML
    void updateButtonBar(Media media) {
        btnRemove.setVisible(true);
        if(media instanceof Playable) {
            btnPlay.setVisible(true);
        } else {
            btnPlay.setVisible(false);
        }
    }

    @FXML
    void showFilterMedia(String t1) {
        if (filterCategory.getSelectedToggle() == radioBtnFilterTitle) {
            ArrayList<Media> filterByTitle = new ArrayList<Media>();
            for (Media item : cart.getItemsOrdered()) {
                if (item.getTitle().contains(t1)) {
                    filterByTitle.add(item);
                }
            }
            tblMedia.setItems(FXCollections.observableList(filterByTitle));
        } else if (filterCategory.getSelectedToggle() == radioBtnFilterId) {
            ArrayList<Media> filterByID = new ArrayList<Media>();
            for (Media item : cart.getItemsOrdered()) {
                if (item.getId() == Integer.parseInt(t1)) {
                    filterByID.add(item);
                }
            }
            tblMedia.setItems(FXCollections.observableList(filterByID));
        }
    }
```



```
    @FXML
    void createPopUp() {
        Stage popupwindow =new Stage();
        popupwindow.initModality(Modality.APPLICATION_MODAL);
        popupwindow.setTitle("Place order");

        Label label1 = new Label("You have place your order !");
        label1.setFont(Font.font("Arial", FontWeight.BOLD,14));
        Label label2 = new Label("Your bill total is " + Float.toString(cart.totalCost()) + "$");
        Button button1= new Button("OK !");
        label2.setTextFill(Color.RED);
        button1.setOnAction(e -> popupwindow.close());
        VBox layout= new VBox(10);
        layout.getChildren().addAll(label1, label2,button1);
        layout.setAlignment(Pos.CENTER);
        Scene scene1= new Scene(layout, 300, 200);
        popupwindow.setScene(scene1);
        popupwindow.show();
    }
}
```

5. **Updating buttons based on selected item in TableView – ChangeListener**

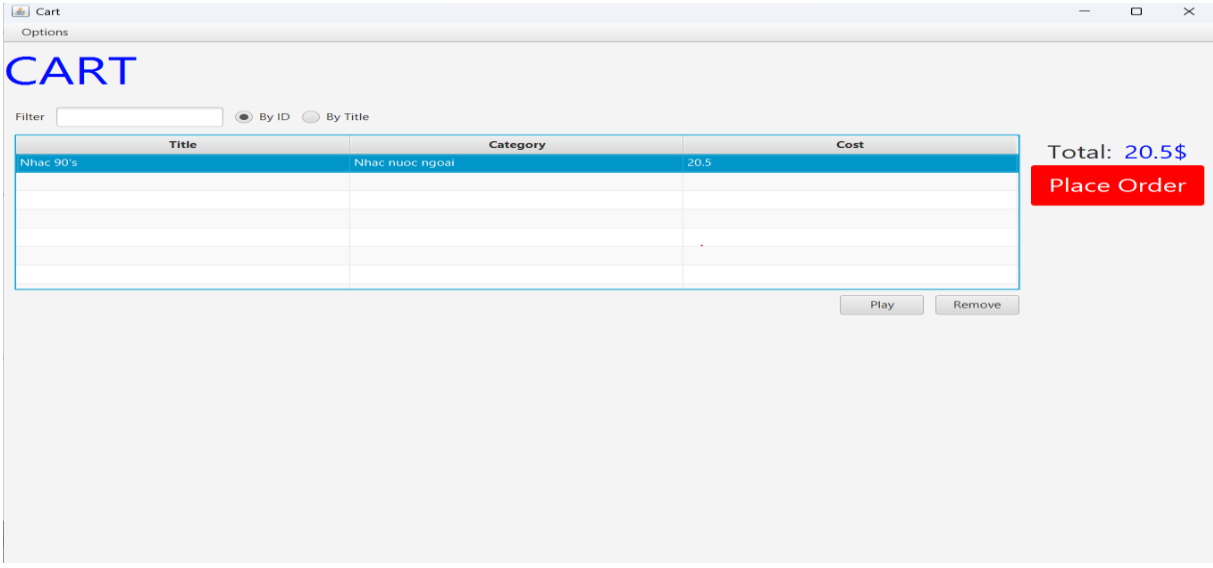
5.1,initialize() function



```
    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(FXCollections.observableList(this.cart.getItemsOrdered()));
        tblMedia.setPlaceholder(new Label("No item in cart"));
        btnPlay.setVisible(false);
        btnRemove.setVisible(false);
        btnRemove.setOnAction(new EventHandler<javafx.event.ActionEvent>() {
            @Override
            public void handle(javafx.event.ActionEvent event) {
                Media media = tblMedia.getSelectionModel().getSelectedItem();
                cart.removeMedia(media);
                totalPrice.setText(Float.toString(cart.totalCost()) +"$");
                tblMedia.setItems(FXCollections.observableList(cart.getItemsOrdered()));
            }
        });
        tfFilter.textProperty().addListener(new ChangeListener<String>() {
            @Override
            public void changed(ObservableValue<? extends String> observableValue, String s, String t1) {
                showFilterMedia(t1);
            }
        });

        tblMedia.getSelectionModel().selectedItemProperty().addListener(
                new ChangeListener<Media>() {
                    @Override
                    public void changed(ObservableValue<? extends Media> observableValue, Media media, Media t1) {
                        updateButtonBar(t1);
                    }
                }
        );
        totalPrice.setText(Float.toString(cart.totalCost()) + "$");

        btnPlay.setOnAction(new EventHandler<ActionEvent>() {
            @Override
            public void handle(ActionEvent event) {
                JDialog playDialog = MediaStore.createPlayDialog(tblMedia.getSelectionModel().getSelectedItem());
                playDialog.setVisible(true);
                playDialog.setSize(300,200);
                playDialog.pack();
            }
        });

        btnPlaceOrder.setOnAction(new EventHandler<ActionEvent>() {
            @Override
            public void handle(ActionEvent event) {
                createPopUp();
                cart.getItemsOrdered().clear();
                tblMedia.setItems(FXCollections.observableList(cart.getItemsOrdered()));
                totalPrice.setText(Float.toString(cart.totalCost()) + "$");
            }
        });
    }

    @FXML
```

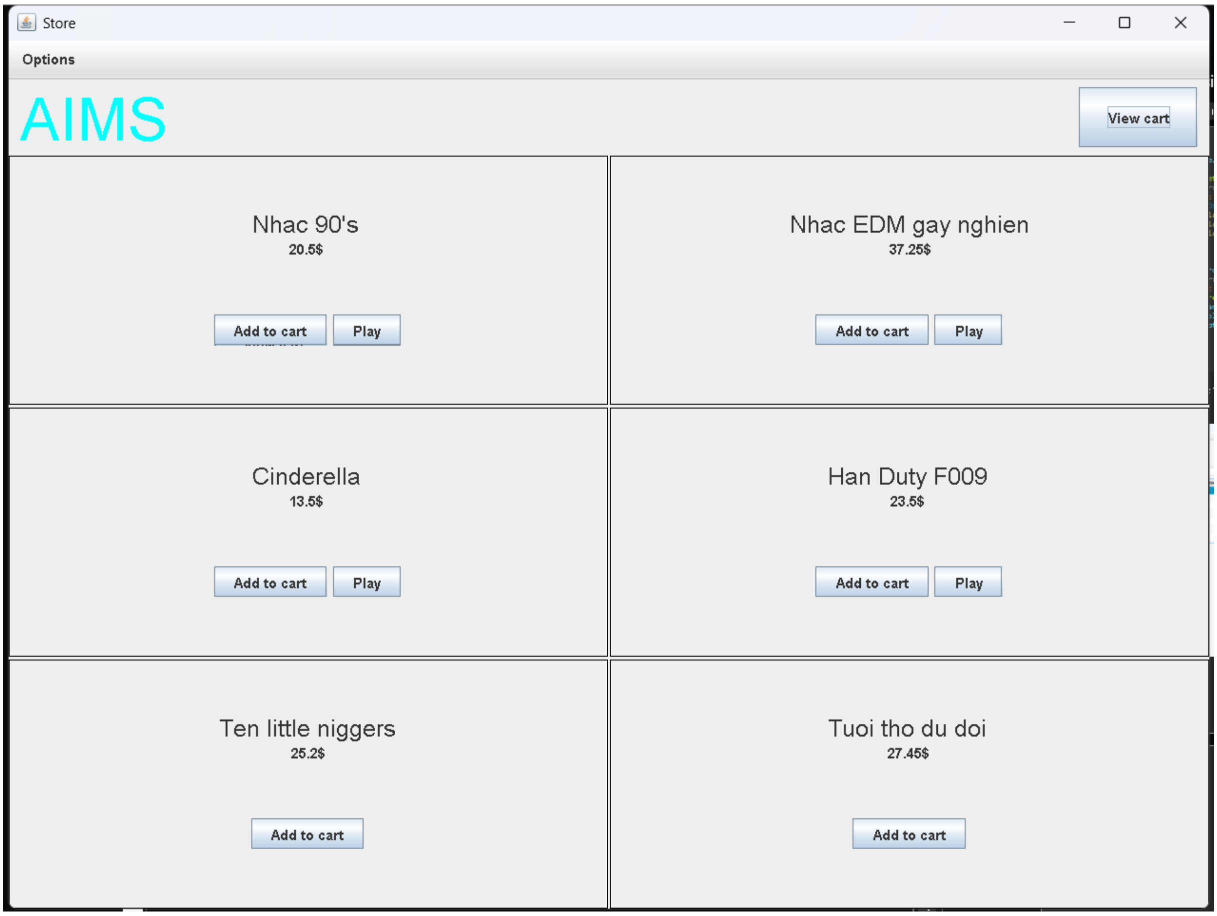
5.2, Complete the AIM GUI application

- Cart Screen:



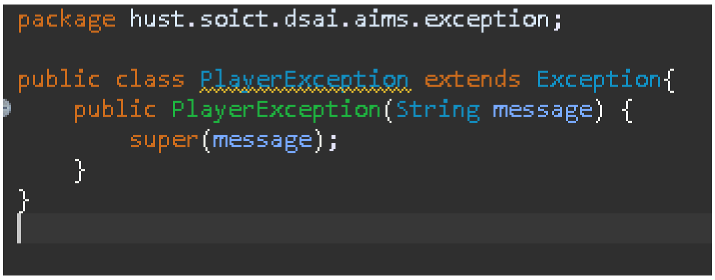
Title	Category	Cost
Nhac 90's	Nhac nuoc ngoai	20.5

- Store Screen:



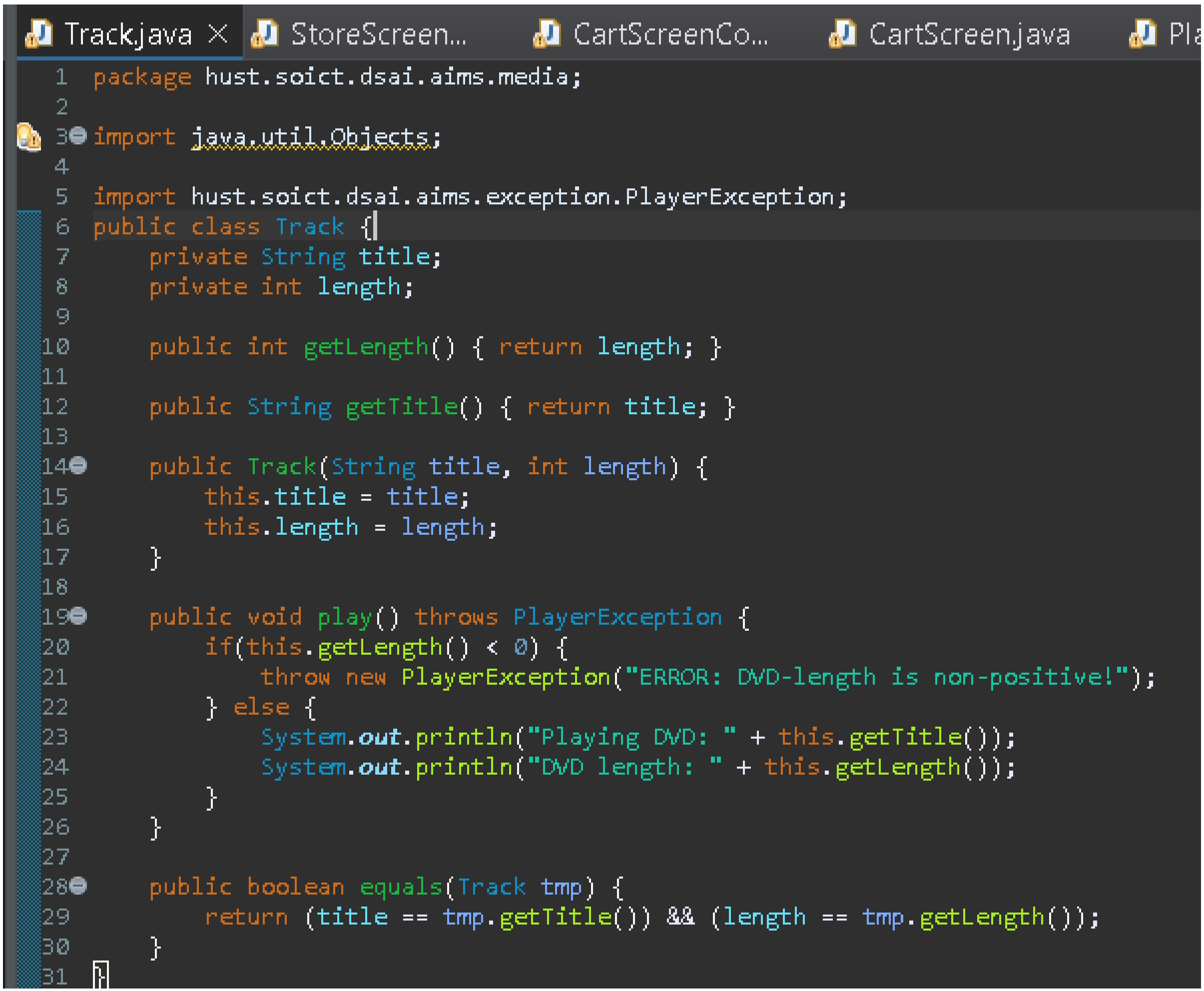
6, Create a class which inherits from **Exception**

6.1 PlayerException Source code

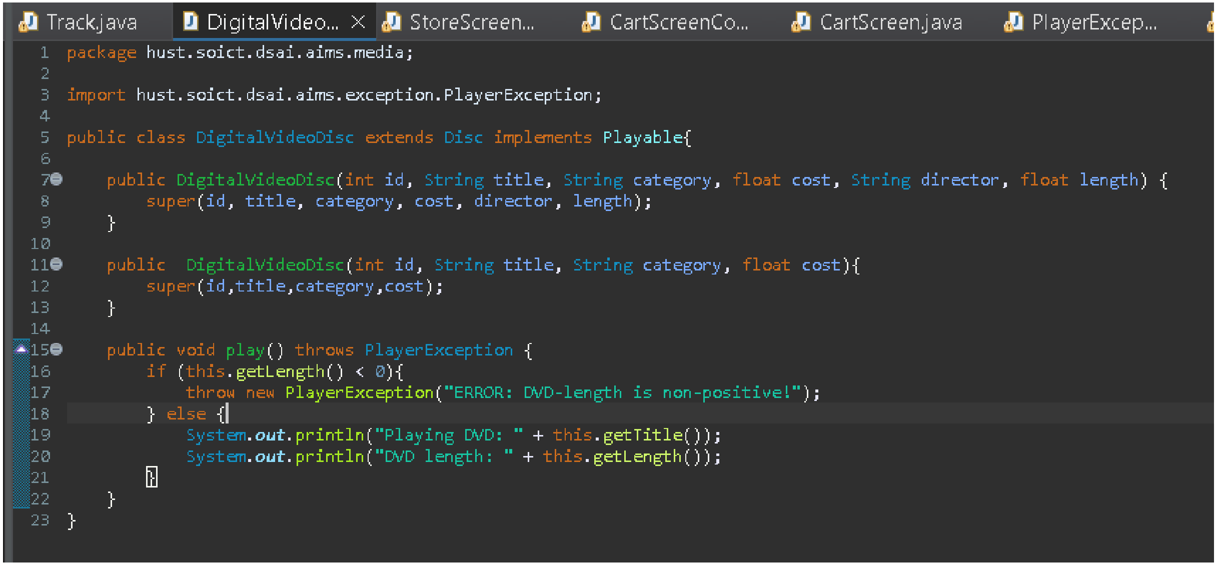


```
package hust.soict.dsai.aims.exception;

public class PlayerException extends Exception{
    public PlayerException(String message) {
        super(message);
    }
}
```

6.2 play() update in DigitaVideoDisc and Track

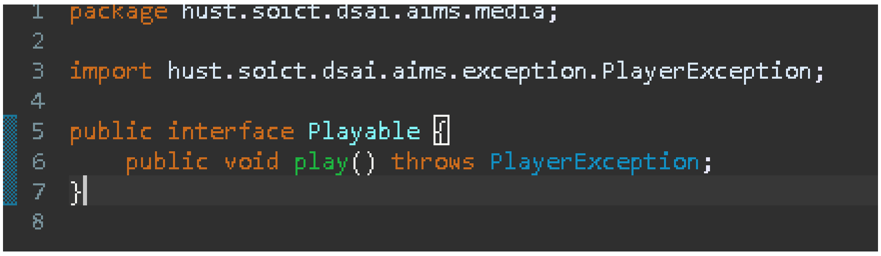


```java
package hust.soict.dsai.aims.media;

import java.util.Objects;

import hust.soict.dsai.aims.exception.PlayerException;
public class Track {
    private String title;
    private int length;

    public int getLength() { return length; }

    public String getTitle() { return title; }

    public Track(String title, int length) {
        this.title = title;
        this.length = length;
    }

    public void play() throws PlayerException {
        if(this.getLength() < 0) {
            throw new PlayerException("ERROR: DVD-length is non-positive!");
        } else {
            System.out.println("Playing DVD: " + this.getTitle());
            System.out.println("DVD length: " + this.getLength());
        }
    }

    public boolean equals(Track tmp) {
        return (title == tmp.getTitle()) && (length == tmp.getLength());
    }
}
```

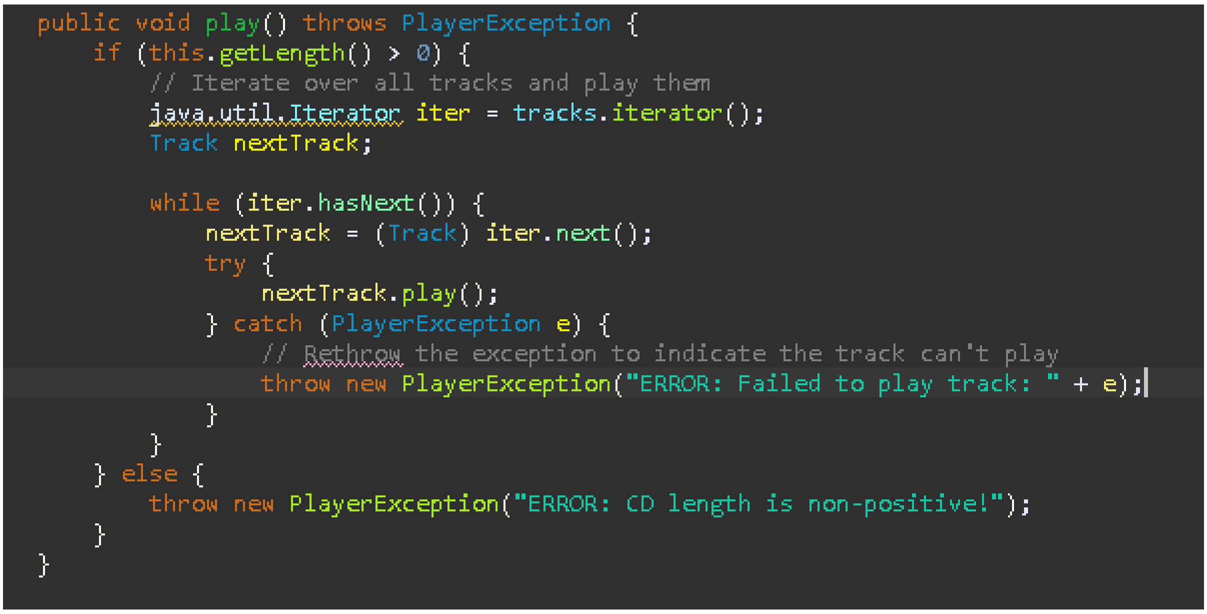


```java
package hust.soict.dsai.aims.media;

import hust.soict.dsai.aims.exception.PlayerException;

public class DigitalVideoDisc extends Disc implements Playable{

    public DigitalVideoDisc(int id, String title, String category, float cost, String director, float length) {
        super(id, title, category, cost, director, length);
    }

    public  DigitalVideoDisc(int id, String title, String category, float cost){
        super(id,title,category,cost);
    }

    public void play() throws PlayerException {
        if (this.getLength() < 0){
            throw new PlayerException("ERROR: DVD-length is non-positive!");
        } else {
            System.out.println("Playing DVD: " + this.getTitle());
            System.out.println("DVD length: " + this.getLength());
        }
    }
}
```

6.3, Update play() in the Playable Interface and in CompactDisc

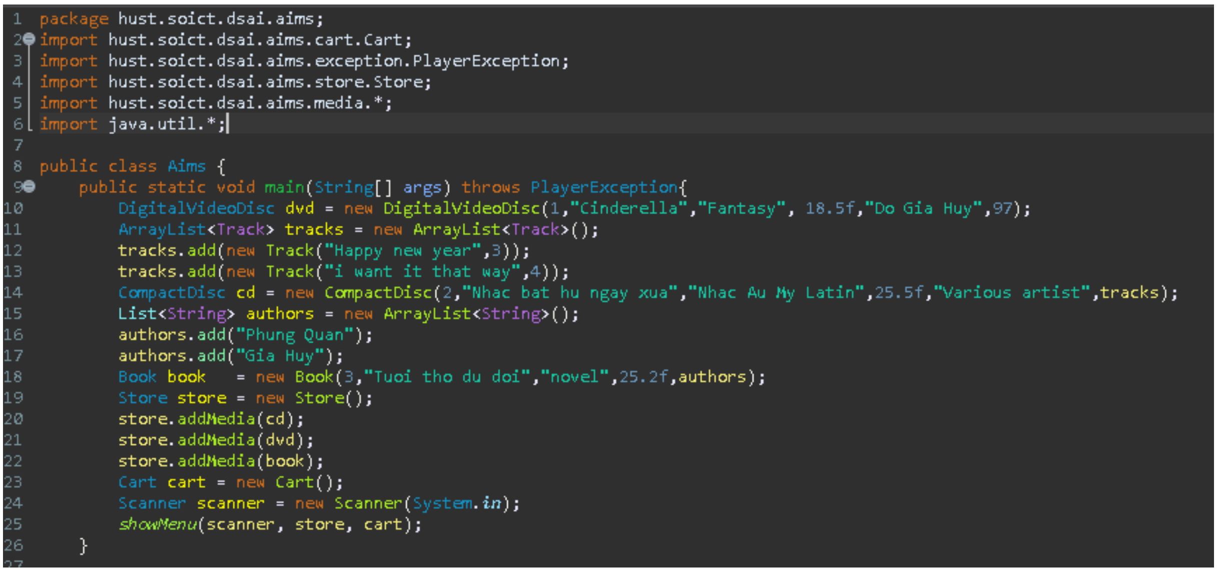


```java
package hust.soict.dsai.aims.media;

import hust.soict.dsai.aims.exception.PlayerException;

public interface Playable {
    public void play() throws PlayerException;
}

```

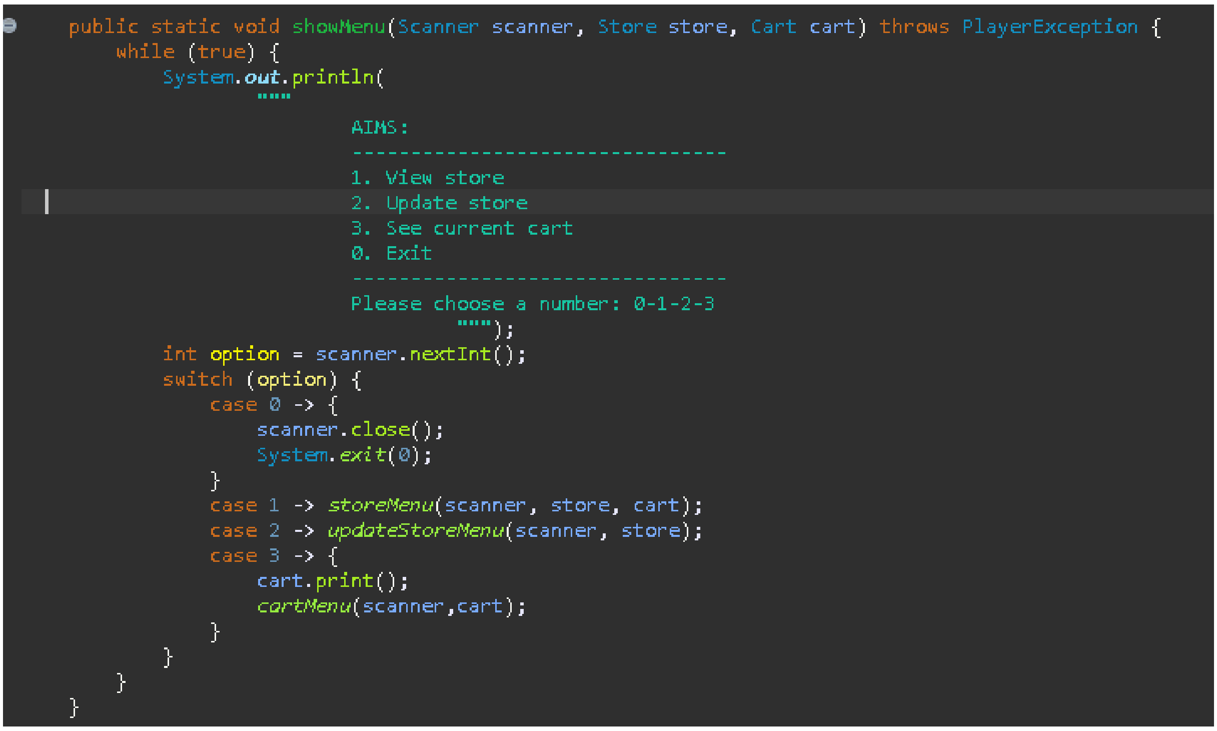


```
public void play() throws PlayerException {
    if (this.getLength() > 0) {
        // Iterate over all tracks and play them
        java.util.Iterator iter = tracks.iterator();
        Track nextTrack;

        while (iter.hasNext()) {
            nextTrack = (Track) iter.next();
            try {
                nextTrack.play();
            } catch (PlayerException e) {
                // Rethrow the exception to indicate the track can't play
                throw new PlayerException("ERROR: Failed to play track: " + e);
            }
        }
    } else {
        throw new PlayerException("ERROR: CD length is non-positive!");
    }
}
```

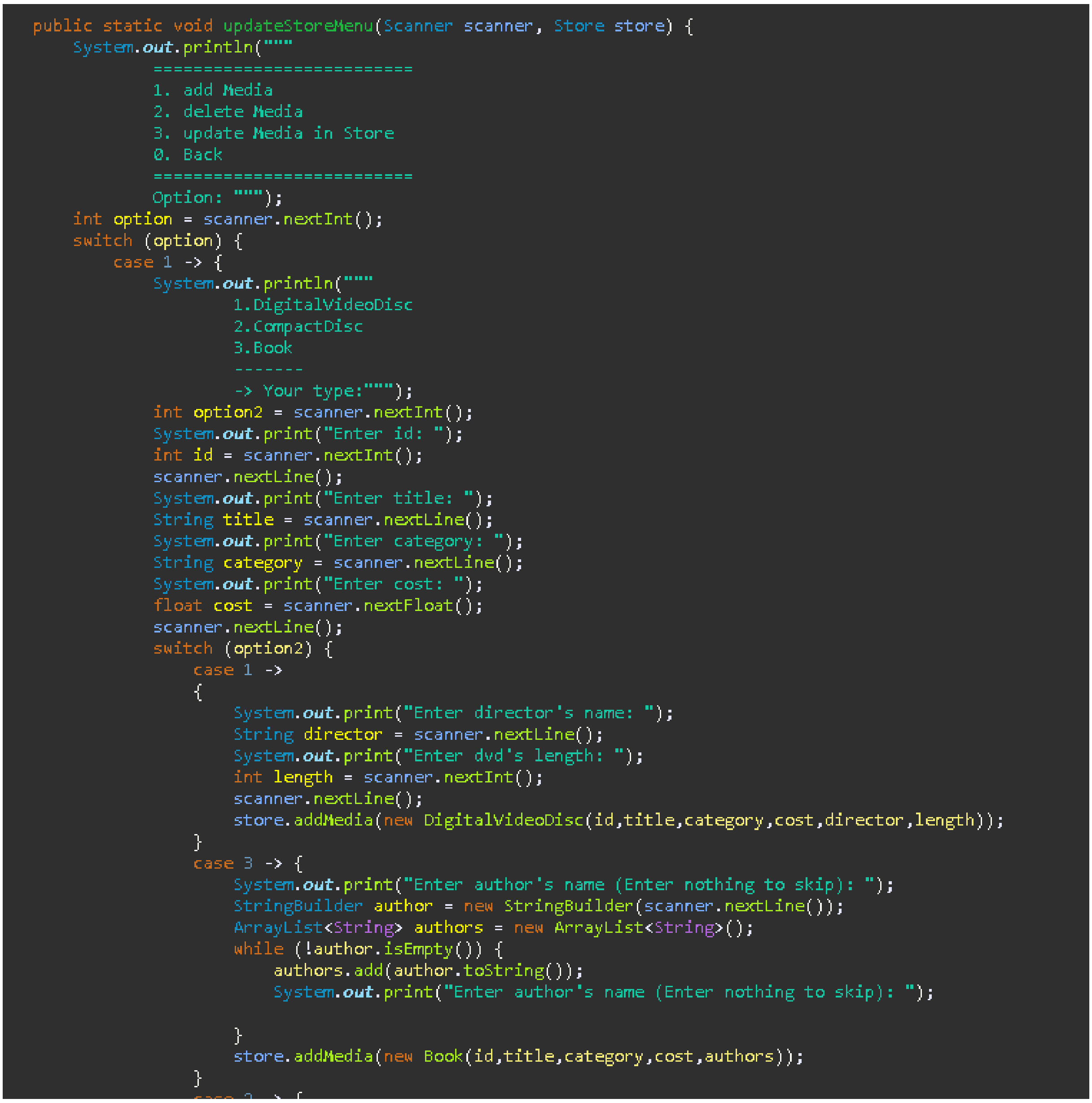
7. Update the AIM class



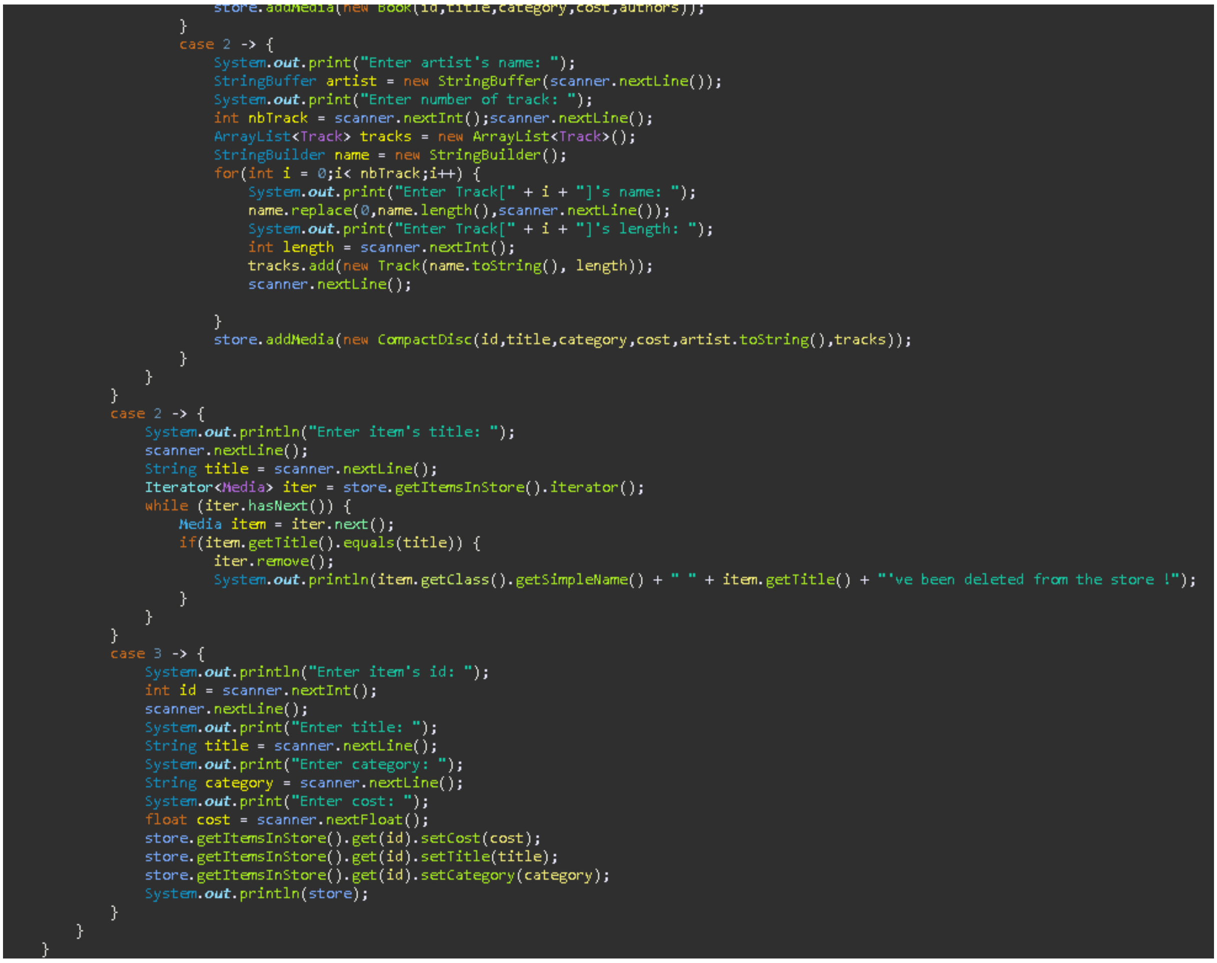
```java
package hust.soict.dsai.aims;
import hust.soict.dsai.aims.cart.Cart;
import hust.soict.dsai.aims.exception.PlayerException;
import hust.soict.dsai.aims.store.Store;
import hust.soict.dsai.aims.media.*;
import java.util.*;

public class Aims {
    public static void main(String[] args) throws PlayerException{
        DigitalVideoDisc dvd = new DigitalVideoDisc(1,"Cinderella","Fantasy", 18.5f,"Do Gia Huy",97);
        ArrayList<Track> tracks = new ArrayList<Track>();
        tracks.add(new Track("Happy new year",3));
        tracks.add(new Track("i want it that way",4));
        CompactDisc cd = new CompactDisc(2,"Nhac bat hu ngay xua","Nhac Au My Latin",25.5f,"Various artist",tracks);
        List<String> authors = new ArrayList<String>();
        authors.add("Phung Quan");
        authors.add("Gia Huy");
        Book book   = new Book(3,"Tuoi tho du doi","novel",25.2f,authors);
        Store store = new Store();
        store.addMedia(cd);
        store.addMedia(dvd);
        store.addMedia(book);
        Cart cart = new Cart();
        Scanner scanner = new Scanner(System.in);
        showMenu(scanner, store, cart);
    }
```



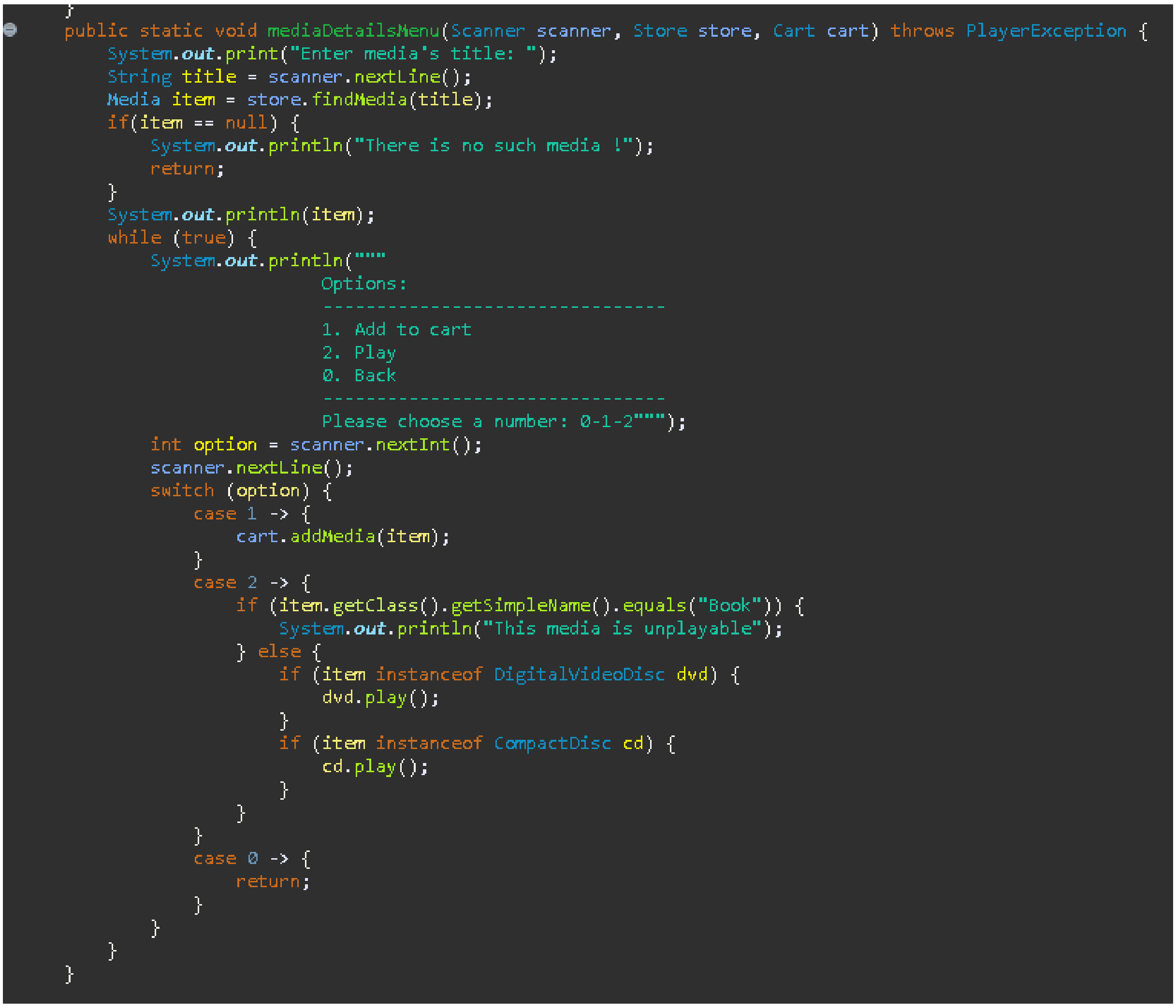
```java
    public static void showMenu(Scanner scanner, Store store, Cart cart) throws PlayerException {
        while (true) {
            System.out.println(
                    """
                            AIMS:
                            --------------------------------
                            1. View store
                            2. Update store
                            3. See current cart
                            0. Exit
                            --------------------------------
                            Please choose a number: 0-1-2-3
                                     """);
            int option = scanner.nextInt();
            switch (option) {
                case 0 -> {
                    scanner.close();
                    System.exit(0);
                }
                case 1 -> storeMenu(scanner, store, cart);
                case 2 -> updateStoreMenu(scanner, store);
                case 3 -> {
                    cart.print();
                    cartMenu(scanner,cart);
                }
            }
        }
    }
```



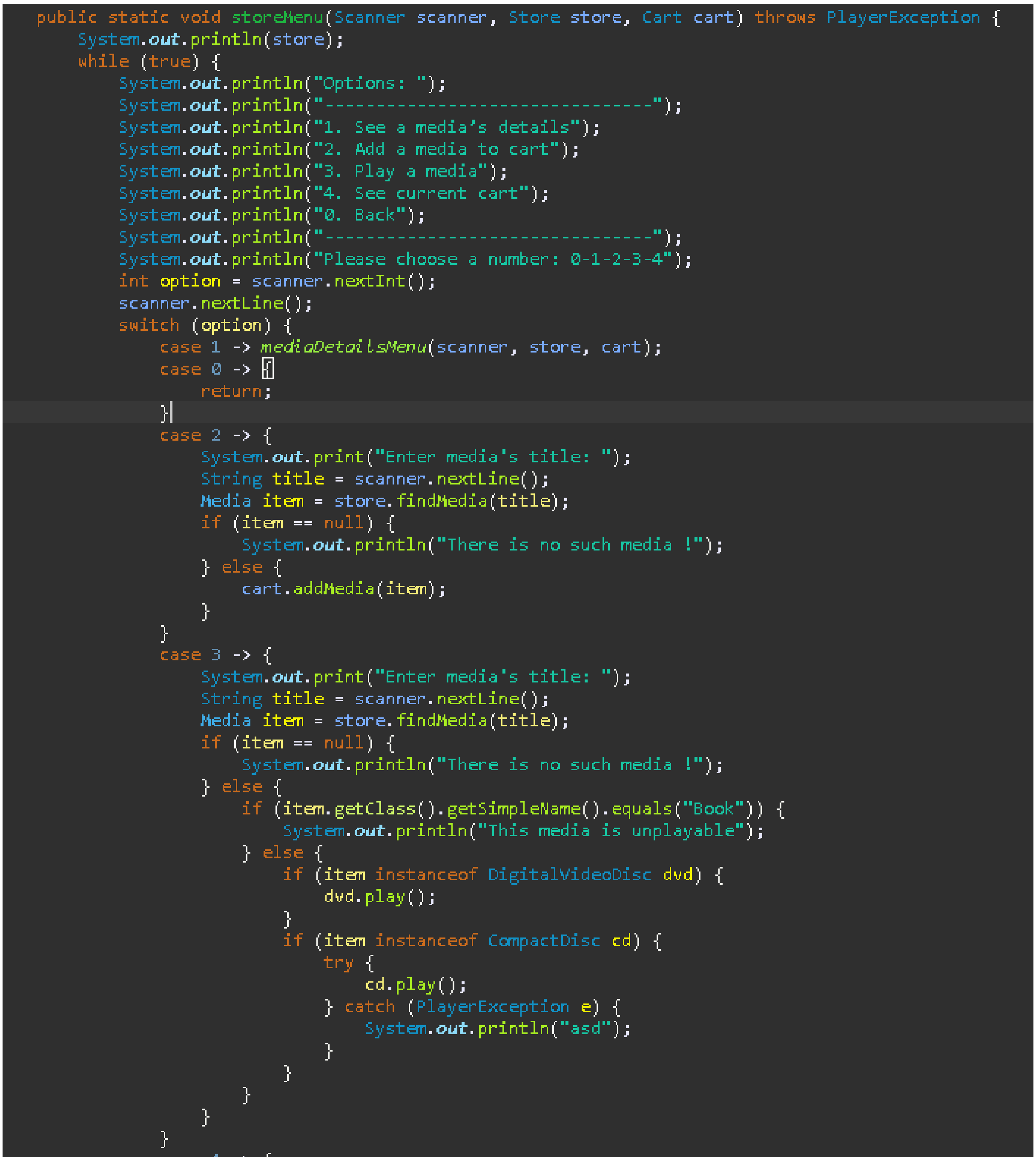
```
public static void updateStoreMenu(Scanner scanner, Store store) {
    System.out.println("""
            ==========================
            1. add Media
            2. delete Media
            3. update Media in Store
            0. Back
            ==========================
            Option: """);
    int option = scanner.nextInt();
    switch (option) {
        case 1 -> {
            System.out.println("""
                    1.DigitalVideoDisc
                    2.CompactDisc
                    3.Book
                    -------
                    -> Your type:""");
            int option2 = scanner.nextInt();
            System.out.print("Enter id: ");
            int id = scanner.nextInt();
            scanner.nextLine();
            System.out.print("Enter title: ");
            String title = scanner.nextLine();
            System.out.print("Enter category: ");
            String category = scanner.nextLine();
            System.out.print("Enter cost: ");
            float cost = scanner.nextFloat();
            scanner.nextLine();
            switch (option2) {
                case 1 ->
                {
                    System.out.print("Enter director's name: ");
                    String director = scanner.nextLine();
                    System.out.print("Enter dvd's length: ");
                    int length = scanner.nextInt();
                    scanner.nextLine();
                    store.addMedia(new DigitalVideoDisc(id,title,category,cost,director,length));
                }
                case 3 -> {
                    System.out.print("Enter author's name (Enter nothing to skip): ");
                    StringBuilder author = new StringBuilder(scanner.nextLine());
                    ArrayList<String> authors = new ArrayList<String>();
                    while (!author.isEmpty()) {
                        authors.add(author.toString());
                        System.out.print("Enter author's name (Enter nothing to skip): ");

                    }
                    store.addMedia(new Book(id,title,category,cost,authors));
                }
                case 2 -> {
```



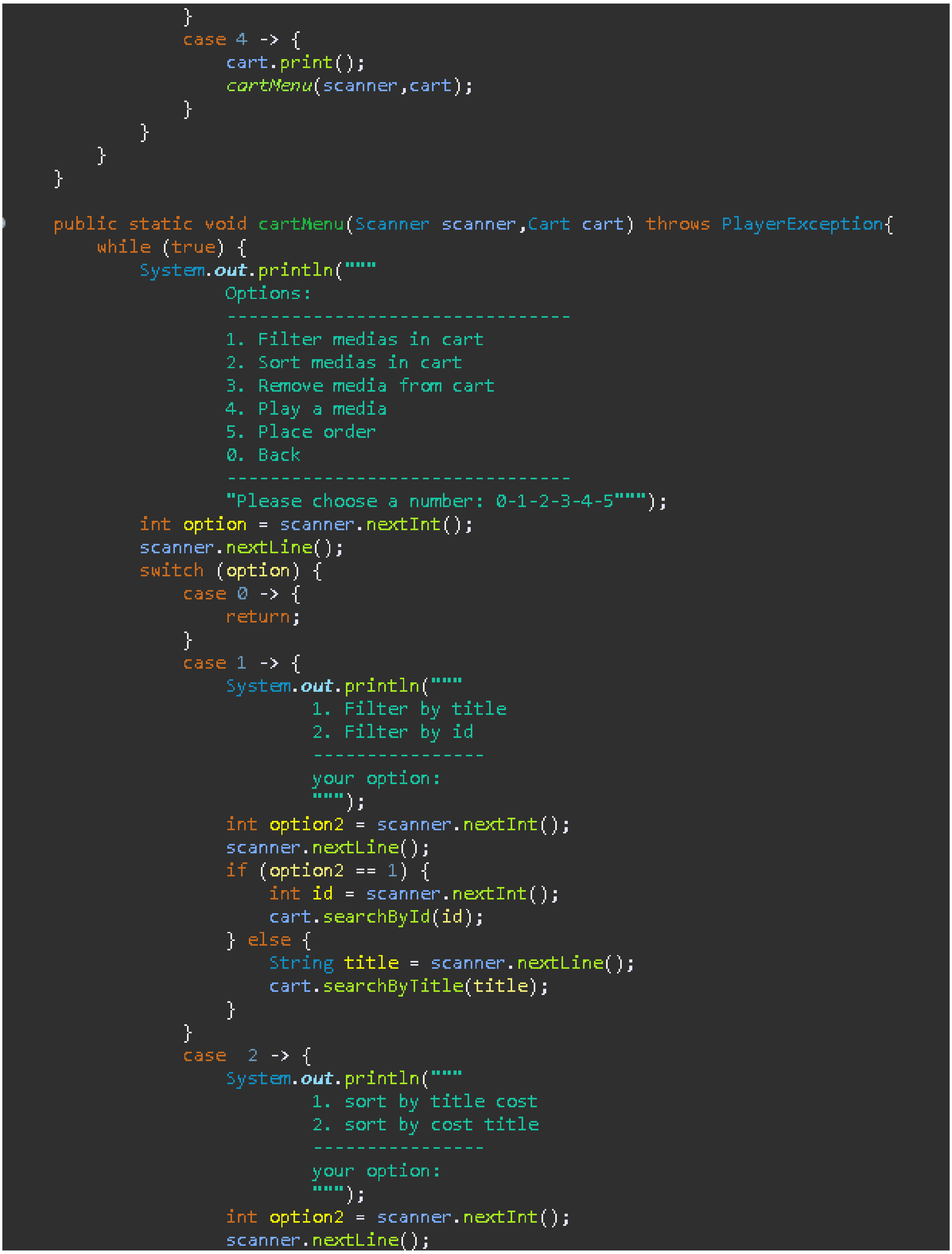
```
                    store.addMedia(new Book(id,title,category,cost,authors));
                }
                case 2 -> {
                    System.out.print("Enter artist's name: ");
                    StringBuffer artist = new StringBuffer(scanner.nextLine());
                    System.out.print("Enter number of track: ");
                    int nbTrack = scanner.nextInt();scanner.nextLine();
                    ArrayList<Track> tracks = new ArrayList<Track>();
                    StringBuilder name = new StringBuilder();
                    for(int i = 0;i< nbTrack;i++) {
                        System.out.print("Enter Track[" + i + "]'s name: ");
                        name.replace(0,name.length(),scanner.nextLine());
                        System.out.print("Enter Track[" + i + "]'s length: ");
                        int length = scanner.nextInt();
                        tracks.add(new Track(name.toString(), length));
                        scanner.nextLine();

                    }
                    store.addMedia(new CompactDisc(id,title,category,cost,artist.toString(),tracks));
                }
            }
        }
        case 2 -> {
            System.out.println("Enter item's title: ");
            scanner.nextLine();
            String title = scanner.nextLine();
            Iterator<Media> iter = store.getItemsInStore().iterator();
            while (iter.hasNext()) {
                Media item = iter.next();
                if(item.getTitle().equals(title)) {
                    iter.remove();
                    System.out.println(item.getClass().getSimpleName() + " " + item.getTitle() + "'ve been deleted from the store !");
                }
            }
        }
        case 3 -> {
            System.out.println("Enter item's id: ");
            int id = scanner.nextInt();
            scanner.nextLine();
            System.out.print("Enter title: ");
            String title = scanner.nextLine();
            System.out.print("Enter category: ");
            String category = scanner.nextLine();
            System.out.print("Enter cost: ");
            float cost = scanner.nextFloat();
            store.getItemsInStore().get(id).setCost(cost);
            store.getItemsInStore().get(id).setTitle(title);
            store.getItemsInStore().get(id).setCategory(category);
            System.out.println(store);
        }
    }
}
```



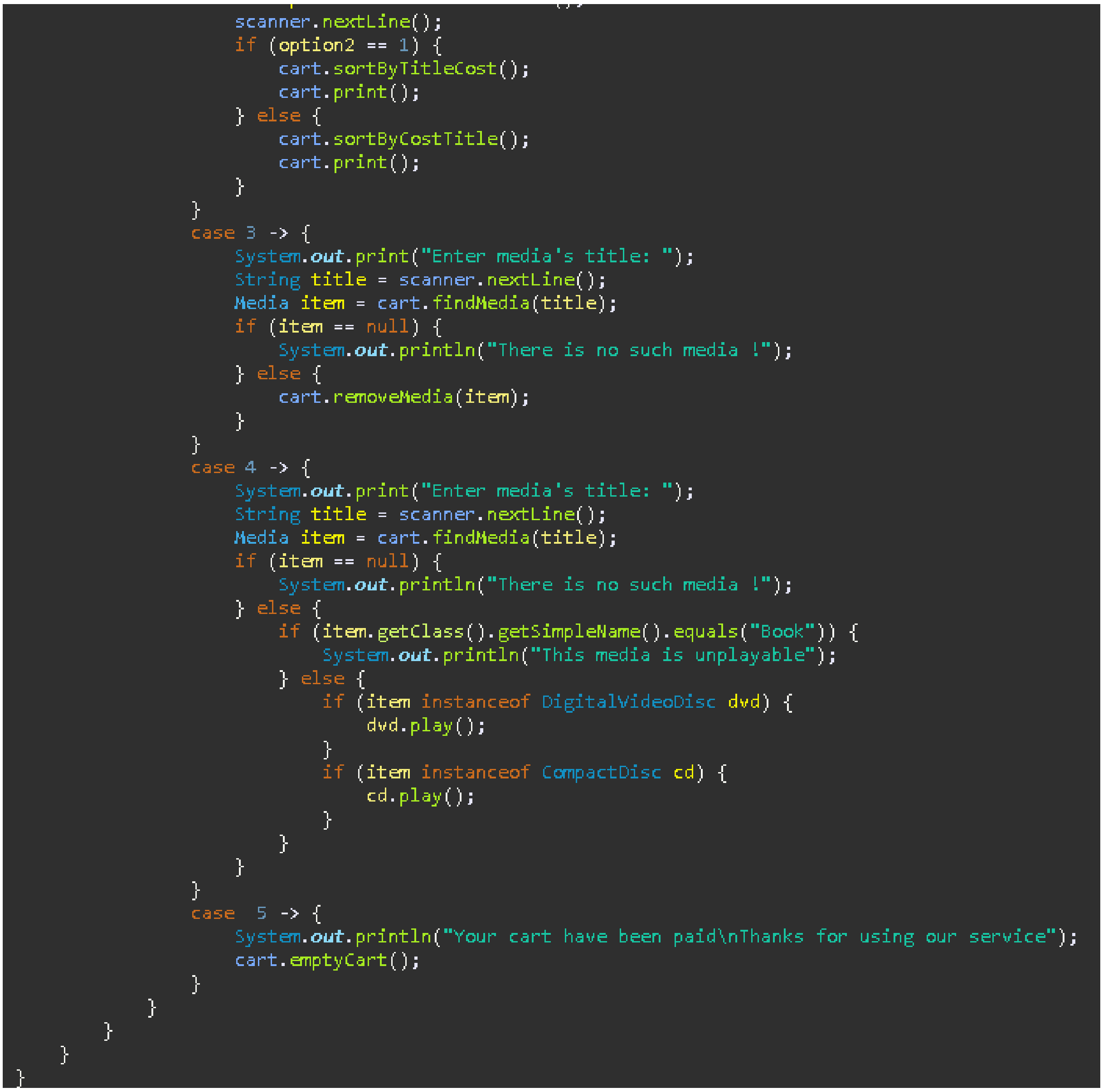
```
    }
    public static void mediaDetailsMenu(Scanner scanner, Store store, Cart cart) throws PlayerException {
        System.out.print("Enter media's title: ");
        String title = scanner.nextLine();
        Media item = store.findMedia(title);
        if(item == null) {
            System.out.println("There is no such media !");
            return;
        }
        System.out.println(item);
        while (true) {
            System.out.println("""
                            Options:
                            --------------------------------
                            1. Add to cart
                            2. Play
                            0. Back
                            --------------------------------
                            Please choose a number: 0-1-2""");
            int option = scanner.nextInt();
            scanner.nextLine();
            switch (option) {
                case 1 -> {
                    cart.addMedia(item);
                }
                case 2 -> {
                    if (item.getClass().getSimpleName().equals("Book")) {
                        System.out.println("This media is unplayable");
                    } else {
                        if (item instanceof DigitalVideoDisc dvd) {
                            dvd.play();
                        }
                        if (item instanceof CompactDisc cd) {
                            cd.play();
                        }
                    }
                }
                case 0 -> {
                    return;
                }
            }
        }
    }
```



```
public static void storeMenu(Scanner scanner, Store store, Cart cart) throws PlayerException {
    System.out.println(store);
    while (true) {
        System.out.println("Options: ");
        System.out.println("--------------------------------");
        System.out.println("1. See a media’s details");
        System.out.println("2. Add a media to cart");
        System.out.println("3. Play a media");
        System.out.println("4. See current cart");
        System.out.println("0. Back");
        System.out.println("--------------------------------");
        System.out.println("Please choose a number: 0-1-2-3-4");
        int option = scanner.nextInt();
        scanner.nextLine();
        switch (option) {
            case 1 -> mediaDetailsMenu(scanner, store, cart);
            case 0 -> {
                return;
            }
            case 2 -> {
                System.out.print("Enter media's title: ");
                String title = scanner.nextLine();
                Media item = store.findMedia(title);
                if (item == null) {
                    System.out.println("There is no such media !");
                } else {
                    cart.addMedia(item);
                }
            }
            case 3 -> {
                System.out.print("Enter media's title: ");
                String title = scanner.nextLine();
                Media item = store.findMedia(title);
                if (item == null) {
                    System.out.println("There is no such media !");
                } else {
                    if (item.getClass().getSimpleName().equals("Book")) {
                        System.out.println("This media is unplayable");
                    } else {
                        if (item instanceof DigitalVideoDisc dvd) {
                            dvd.play();
                        }
                        if (item instanceof CompactDisc cd) {
                            try {
                                cd.play();
                            } catch (PlayerException e) {
                                System.out.println("asd");
                            }
                        }
                    }
                }
            }
```



```java
                }
                case 4 -> {
                    cart.print();
                    cartMenu(scanner,cart);
                }
            }
        }
    }

    public static void cartMenu(Scanner scanner,Cart cart) throws PlayerException{
        while (true) {
            System.out.println("""
                    Options:
                    --------------------------------
                    1. Filter medias in cart
                    2. Sort medias in cart
                    3. Remove media from cart
                    4. Play a media
                    5. Place order
                    0. Back
                    --------------------------------
                    "Please choose a number: 0-1-2-3-4-5""");
            int option = scanner.nextInt();
            scanner.nextLine();
            switch (option) {
                case 0 -> {
                    return;
                }
                case 1 -> {
                    System.out.println("""
                            1. Filter by title
                            2. Filter by id
                            ----------------
                            your option:
                            """);
                    int option2 = scanner.nextInt();
                    scanner.nextLine();
                    if (option2 == 1) {
                        int id = scanner.nextInt();
                        cart.searchById(id);
                    } else {
                        String title = scanner.nextLine();
                        cart.searchByTitle(title);
                    }
                }
                case  2 -> {
                    System.out.println("""
                            1. sort by title cost
                            2. sort by cost title
                            ----------------
                            your option:
                            """);
                    int option2 = scanner.nextInt();
                    scanner.nextLine();
```



```
                    scanner.nextLine();
                    if (option2 == 1) {
                        cart.sortByTitleCost();
                        cart.print();
                    } else {
                        cart.sortByCostTitle();
                        cart.print();
                    }
                }
                case 3 -> {
                    System.out.print("Enter media's title: ");
                    String title = scanner.nextLine();
                    Media item = cart.findMedia(title);
                    if (item == null) {
                        System.out.println("There is no such media !");
                    } else {
                        cart.removeMedia(item);
                    }
                }
                case 4 -> {
                    System.out.print("Enter media's title: ");
                    String title = scanner.nextLine();
                    Media item = cart.findMedia(title);
                    if (item == null) {
                        System.out.println("There is no such media !");
                    } else {
                        if (item.getClass().getSimpleName().equals("Book")) {
                            System.out.println("This media is unplayable");
                        } else {
                            if (item instanceof DigitalVideoDisc dvd) {
                                dvd.play();
                            }
                            if (item instanceof CompactDisc cd) {
                                cd.play();
                            }
                        }
                    }
                }
                case  5 -> {
                    System.out.println("Your cart have been paid\nThanks for using our service");
                    cart.emptyCart();
                }
            }
        }
    }
}
```