Student name: Lê Ngọc Anh Vũ

Student ID: 20236014

REPORT LAB 05

3 JavaFX API

Note: For the exercises in this lab (excluding the AIMS exercises), you will continue to use the `GUIProject`, and put all your source code in a package called "**`hust.soict.dsai.javafx`**" for DS & AI. You might need to add the JavaFX library to this project if you are using the JDK version after 1.8.
Note: From this section onwards, it is assumed that you are a DS-AI student, so your folder structure will contain the "**`dsai`**" package. If you are an HEDSPI or ICT student, you should replace the "**`dsai`**" string with "**`hedspi`**" or "**`globalict`**".

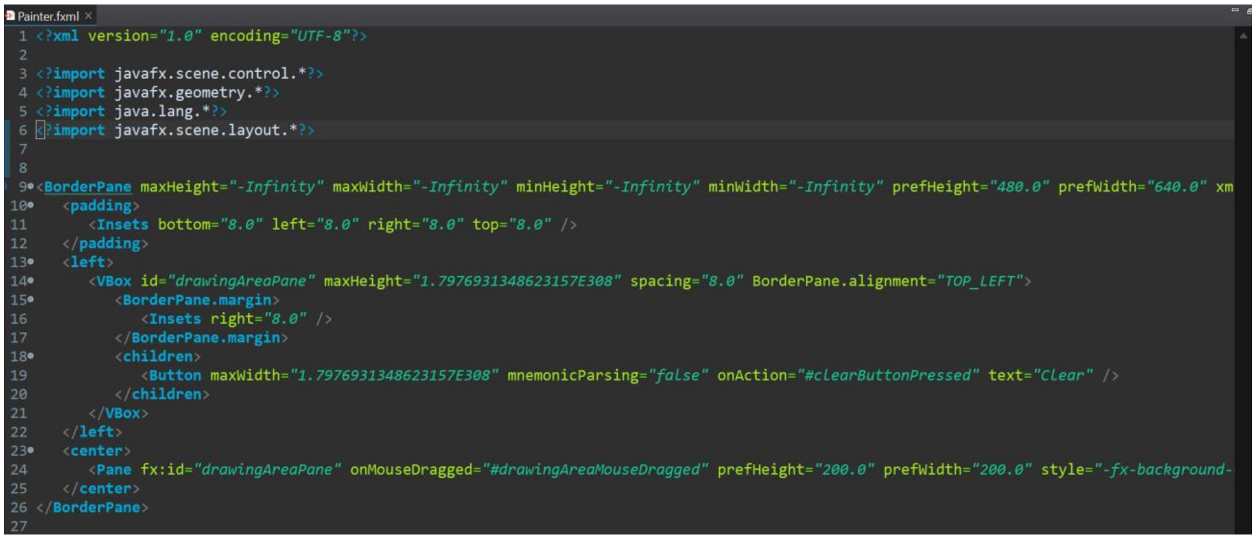
In this exercise, we revisit the components of a JavaFX application by implementing a simple `Painter` app that allows the user to draw on a white canvas with their mouse.

3.1 Create the FXML file

3.1.1 Create and open the FXML file in Scene Builder from Eclipse:

3.1.2 Building the GUI:

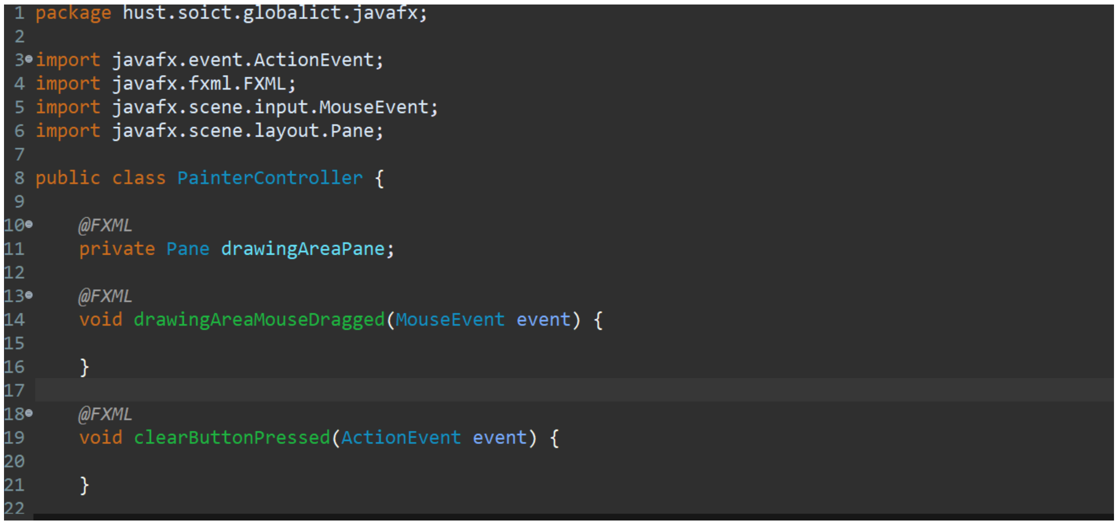
After using Scene Builder:



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

<?import javafx.scene.control.*?>
<?import javafx.geometry.*?>
<?import java.lang.*?>
<?import javafx.scene.layout.*?>


<BorderPane maxHeight="-Infinity" maxWidth="-Infinity" minHeight="-Infinity" minWidth="-Infinity" prefHeight="480.0" prefWidth="640.0" xm
   <padding>
      <Insets bottom="8.0" left="8.0" right="8.0" top="8.0" />
   </padding>
   <left>
      <VBox id="drawingAreaPane" maxHeight="1.7976931348623157E308" spacing="8.0" BorderPane.alignment="TOP_LEFT">
         <BorderPane.margin>
            <Insets right="8.0" />
         </BorderPane.margin>
         <children>
            <Button maxWidth="1.7976931348623157E308" mnemonicParsing="false" onAction="#clearButtonPressed" text="Clear" />
         </children>
      </VBox>
   </left>
   <center>
      <Pane fx:id="drawingAreaPane" onMouseDragged="#drawingAreaMouseDragged" prefHeight="200.0" prefWidth="200.0" style="-fx-background-
   </center>
</BorderPane>

```

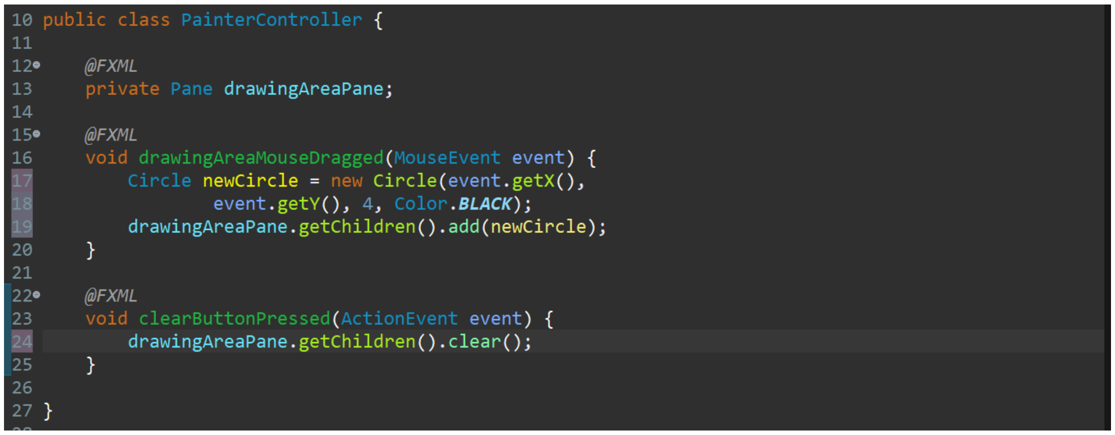
3.2 Create the controller class

In the same package as the FXML, create a Java class called `PainterController`.



```java
package hust.soict.globalict.javafx;

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

public class PainterController {

    @FXML
    private Pane drawingAreaPane;

    @FXML
    void drawingAreaMouseDragged(MouseEvent event) {

    }

    @FXML
    void clearButtonPressed(ActionEvent event) {

    }

```

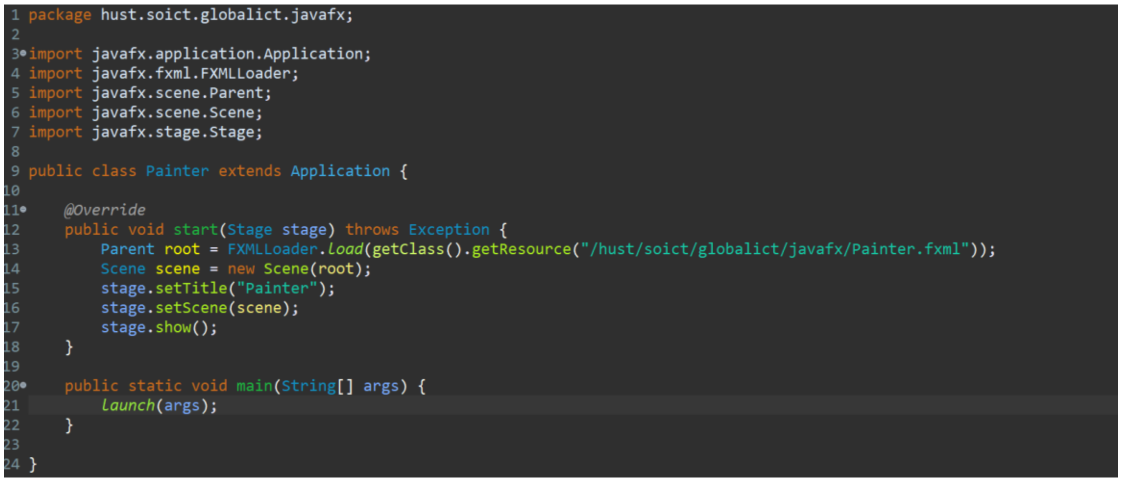
Next, we will implement the event-handling functions.



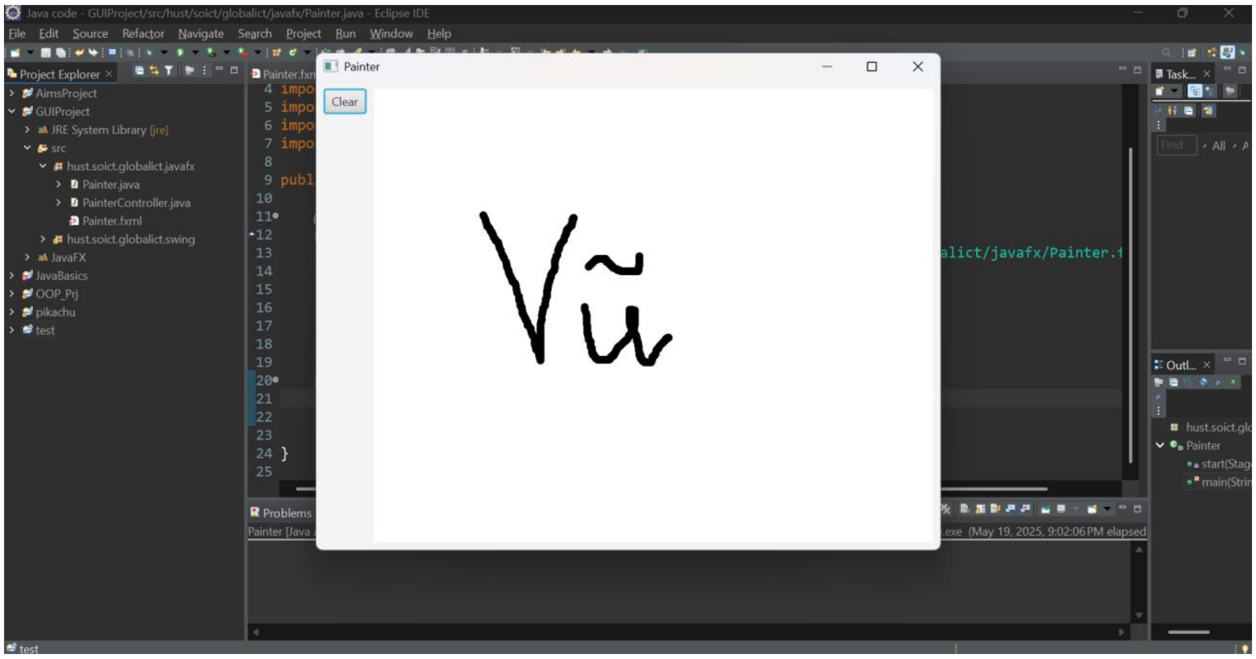
```java
public class PainterController {

    @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();
    }

}
```

3.3 Create the application

Create a class named `Painter` in the same package as the FXML and the controller class. The source code is provided below:



```java
package hust.soict.globalict.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("/hust/soict/globalict/javafx/Painter.fxml"));
        Scene scene = new Scene(root);
        stage.setTitle("Painter");
        stage.setScene(scene);
        stage.show();
    }

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

}
```



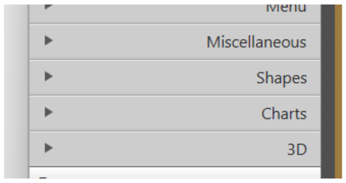
3.4 *Practice exercise:*

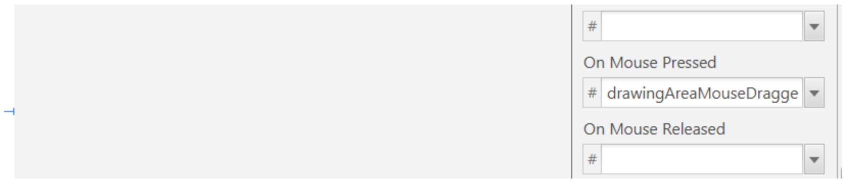
3.4.1 Draw when mouse down

In the current version of the `Painter` app, if the user just presses down on the mouse without dragging it, nothing will appear on the canvas (because we only add the handling method for the "On Mouse Dragged" entry for the Pane). The expected output should be a dot appearing at the position of the mouse.

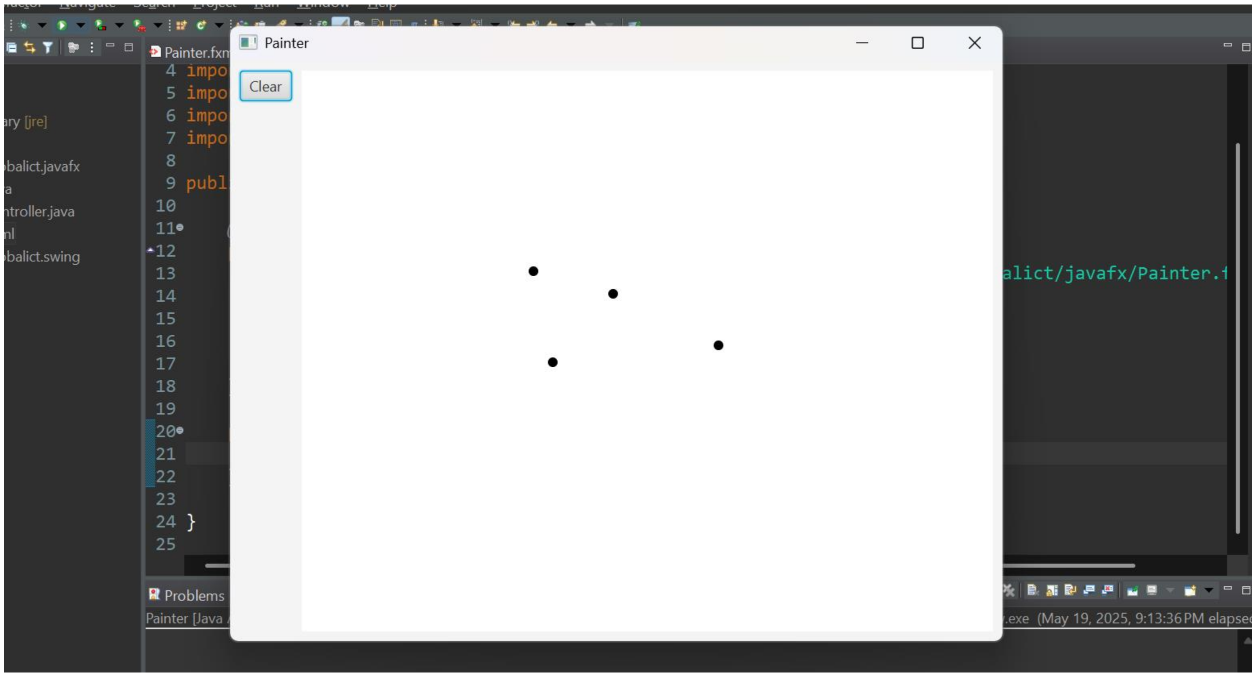
Your task is to improve the Painter application so the output will be as the one expected. You shouldn't have to change your source code for this, rather just the FXML file through the Scene Builder GUI. Since the event-handling for mouse pressed and mouse dragged are the same, they can share the same event-handler method (drawingAreaMouseDragged). You just need to pick the appropriate entry among the ones below to set the event-handler method name to. (On Mouse Dragged and another entry will have the same method name).

<u>Note</u>: After modifying the FXML file, you might need to refresh your Java project to make sure the latest changes are updated. You can do so by right-clicking on your project in Project Explorer > Refresh.

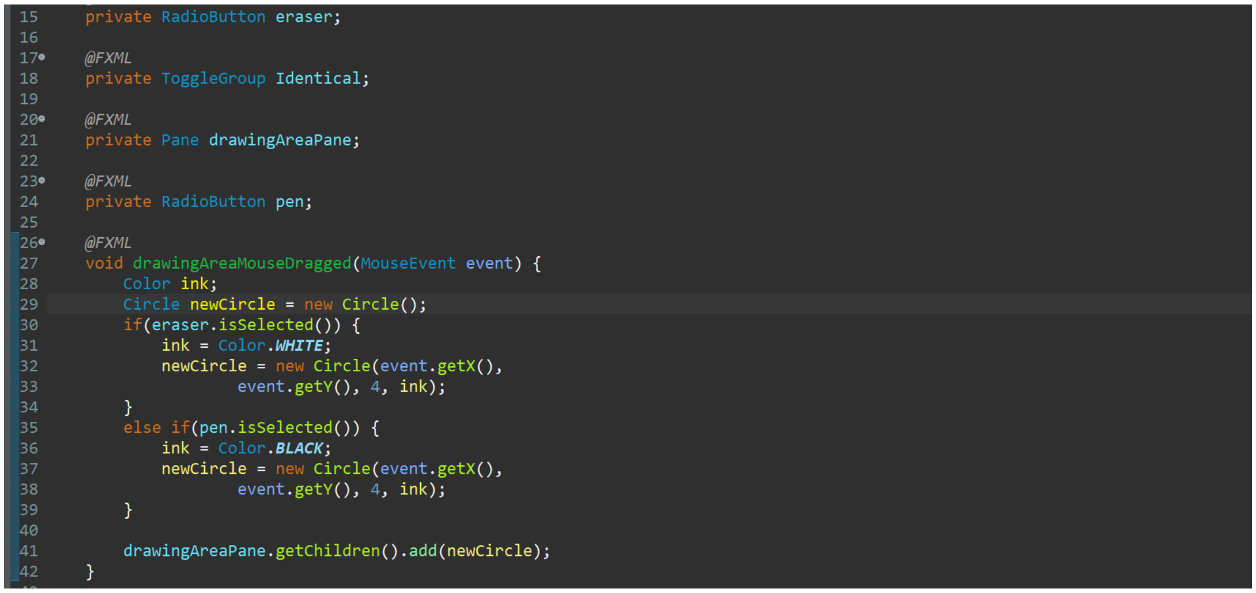




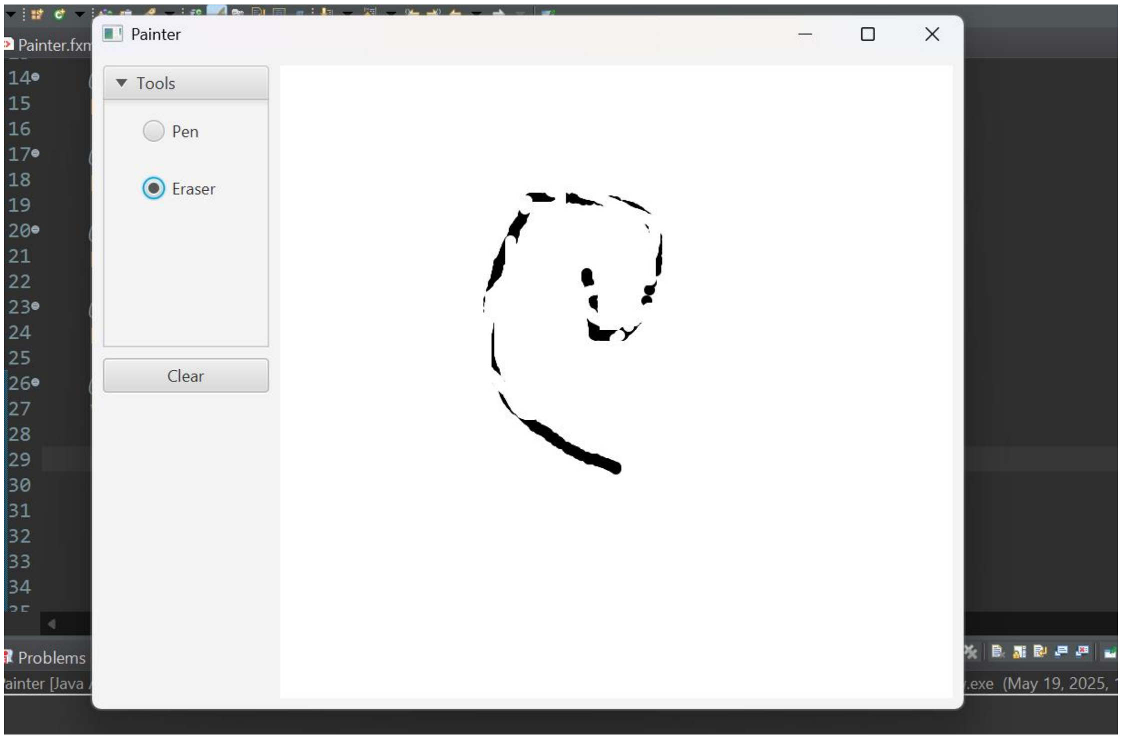
On Mouse Pressed
# drawingAreaMouseDragge
On Mouse Released
#



3.4.2 Add the Eraser functionality

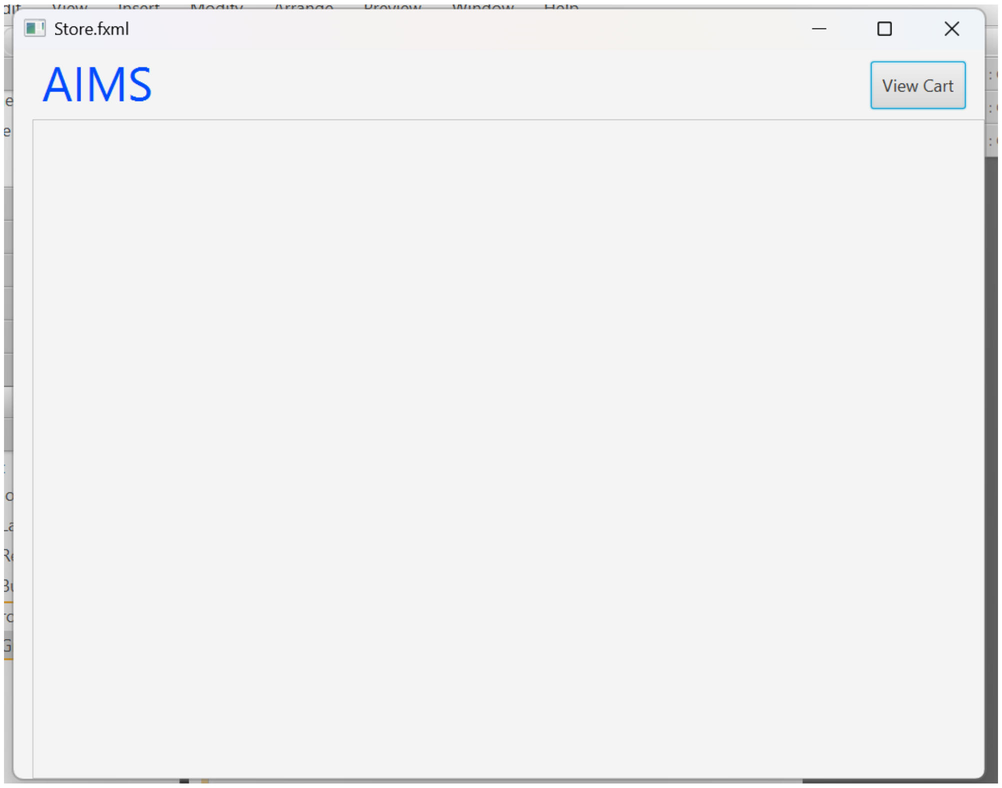


```java
    private RadioButton eraser;

    @FXML
    private ToggleGroup Identical;

    @FXML
    private Pane drawingAreaPane;

    @FXML
    private RadioButton pen;

    @FXML
    void drawingAreaMouseDragged(MouseEvent event) {
        Color ink;
        Circle newCircle = new Circle();
        if(eraser.isSelected()) {
            ink = Color.WHITE;
            newCircle = new Circle(event.getX(),
                    event.getY(), 4, ink);
        }
        else if(pen.isSelected()) {
            ink = Color.BLACK;
            newCircle = new Circle(event.getX(),
                    event.getY(), 4, ink);
        }

        drawingAreaPane.getChildren().add(newCircle);
    }
```



5 Build View Store Screen for AIMS Customer Application with JavaFX

5.1 Set up View Store Screen with Scene Builder

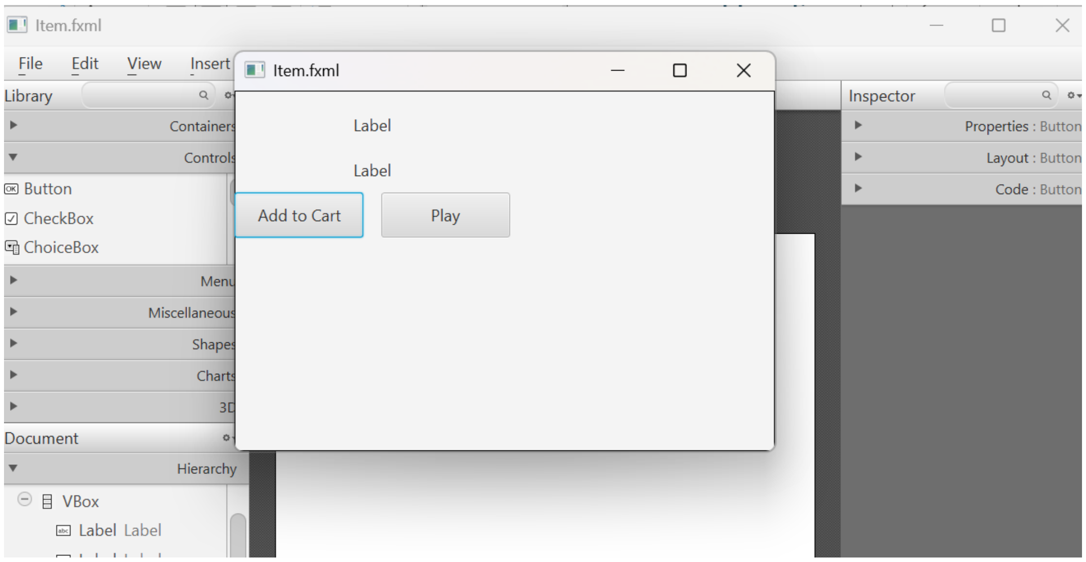


5.2 Set up Item in the Store

In the View Store Screen, we already created an empty GridPane. In this section, we will create an FXML component to display the information of media and dynamically add it to the GridPane.

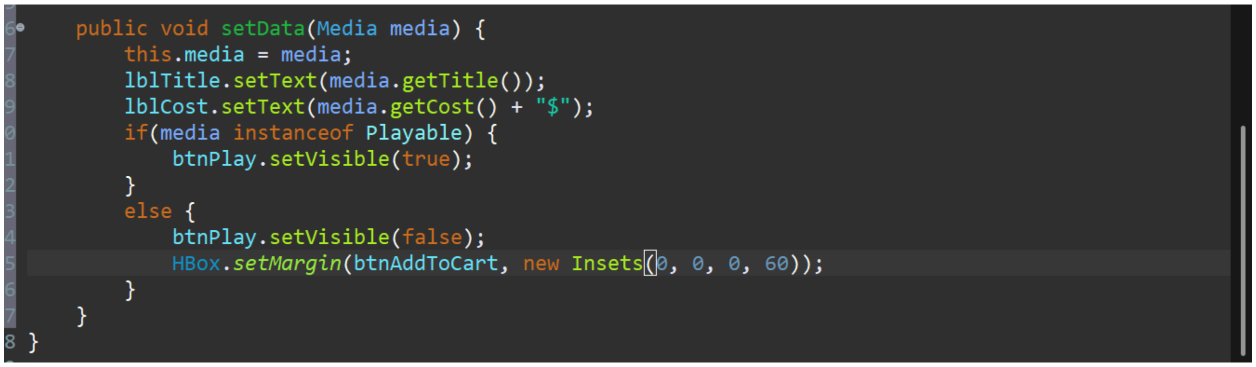
5.2.1 Create Item.fxml file

In package `screen.customer.view`, create the Item.fxml file, choose the root node to be AnchorPane.



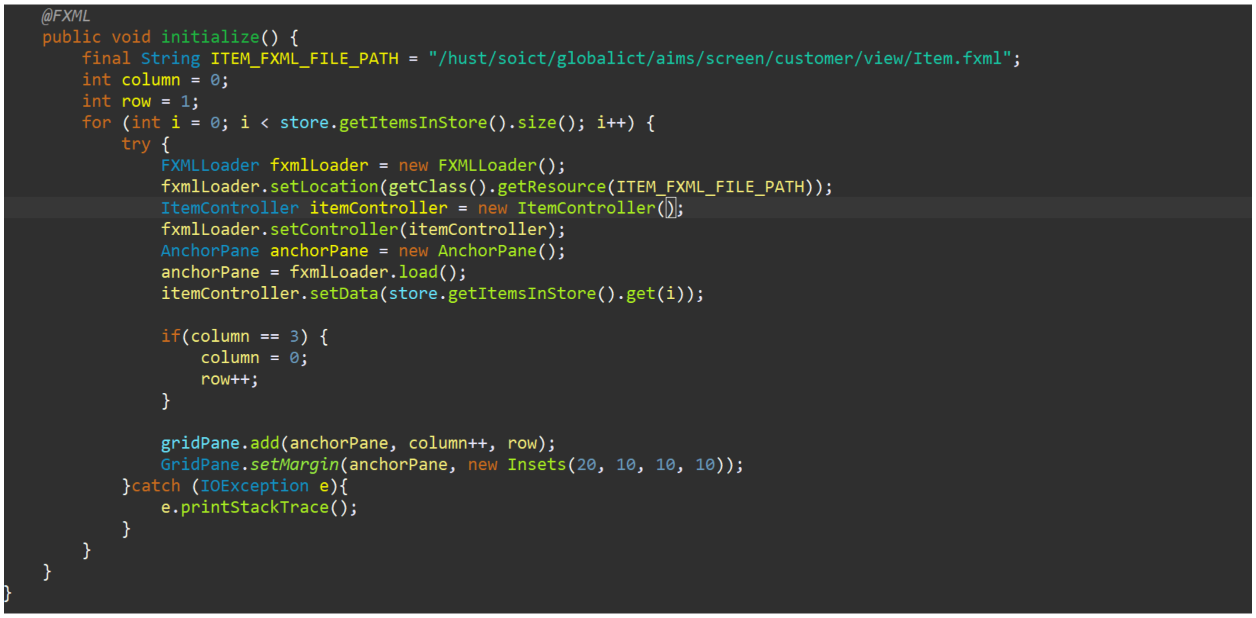
5.2.2 Create ItemController class

The fxml file just contains the GUI, not the logic. You need to create a controller to implements the behaviors of the FXML components.



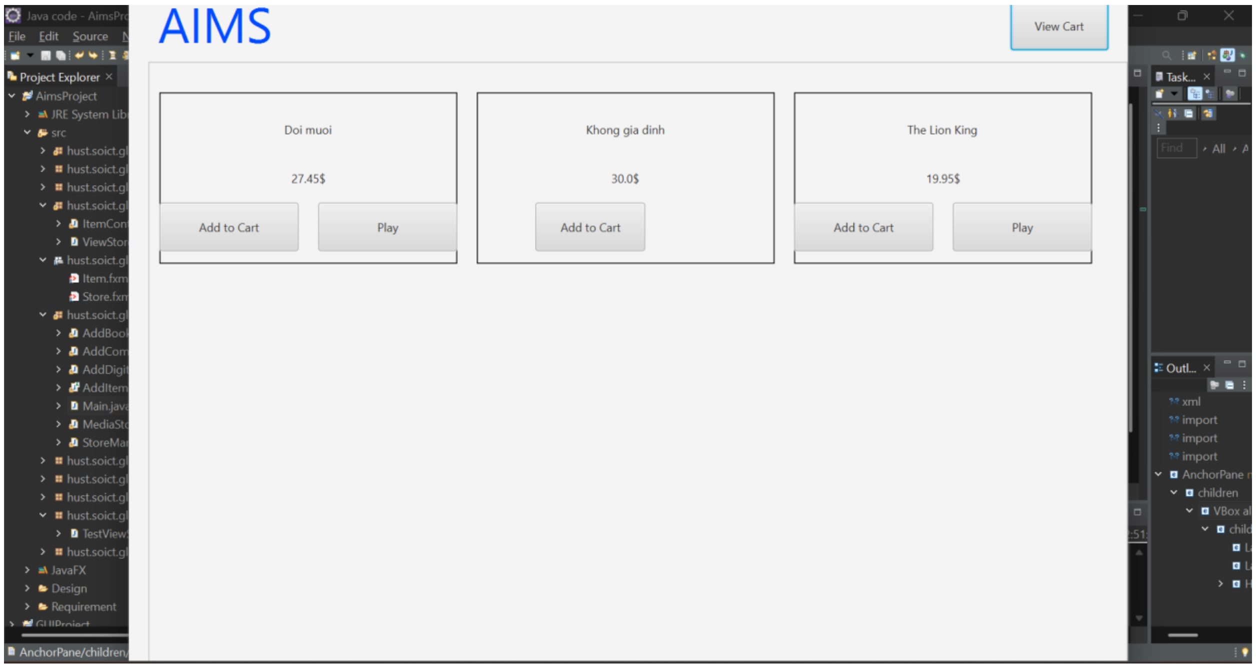
```
    public void setData(Media media) {
        this.media = media;
        lblTitle.setText(media.getTitle());
        lblCost.setText(media.getCost() + "$");
        if(media instanceof Playable) {
            btnPlay.setVisible(true);
        }
        else {
            btnPlay.setVisible(false);
            HBox.setMargin(btnAddToCart, new Insets(0, 0, 0, 60));
        }
    }
}
```

5.3 Create ViewStoreController class



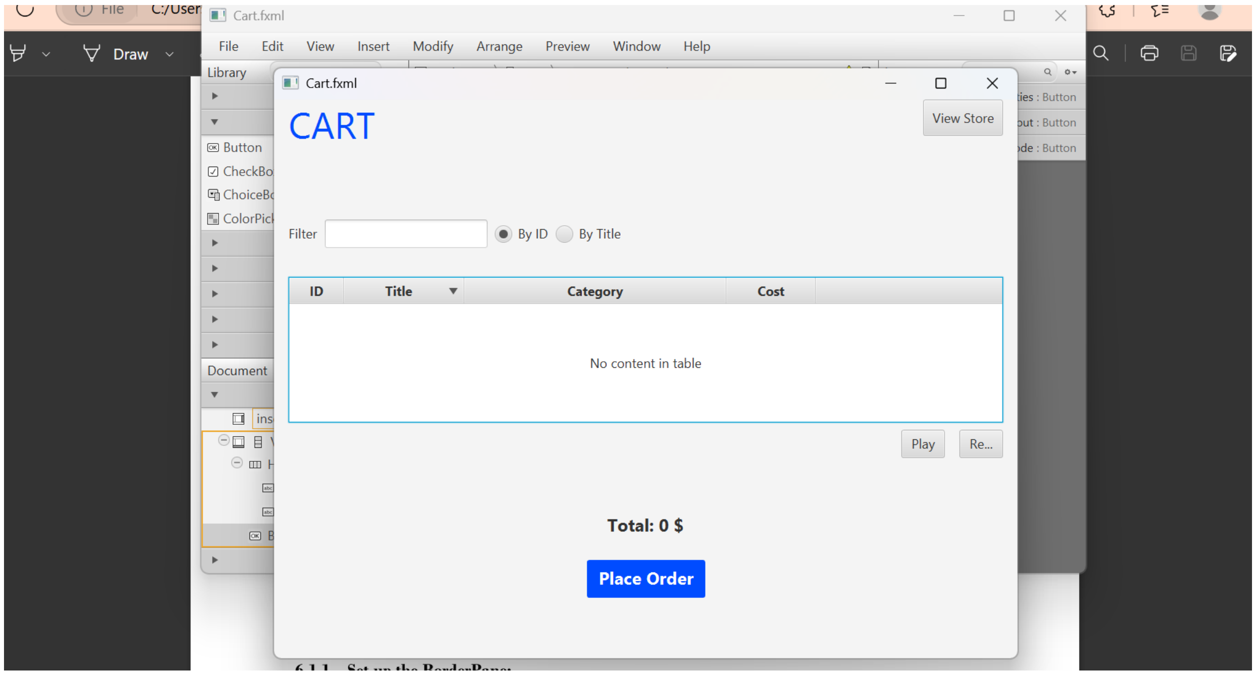
```
    @FXML
    public void initialize() {
        final String ITEM_FXML_FILE_PATH = "/hust/soict/globalict/aims/screen/customer/view/Item.fxml";
        int column = 0;
        int row = 1;
        for (int i = 0; i < store.getItemsInStore().size(); i++) {
            try {
                FXMLLoader fxmlLoader = new FXMLLoader();
                fxmlLoader.setLocation(getClass().getResource(ITEM_FXML_FILE_PATH));
                ItemController itemController = new ItemController();
                fxmlLoader.setController(itemController);
                AnchorPane anchorPane = new AnchorPane();
                anchorPane = fxmlLoader.load();
                itemController.setData(store.getItemsInStore().get(i));

                if(column == 3) {
                    column = 0;
                    row++;
                }

                gridPane.add(anchorPane, column++, row);
                GridPane.setMargin(anchorPane, new Insets(20, 10, 10, 10));
            }catch (IOException e){
                e.printStackTrace();
            }
        }
    }
}
```

5.4 *Test View Store Screen*



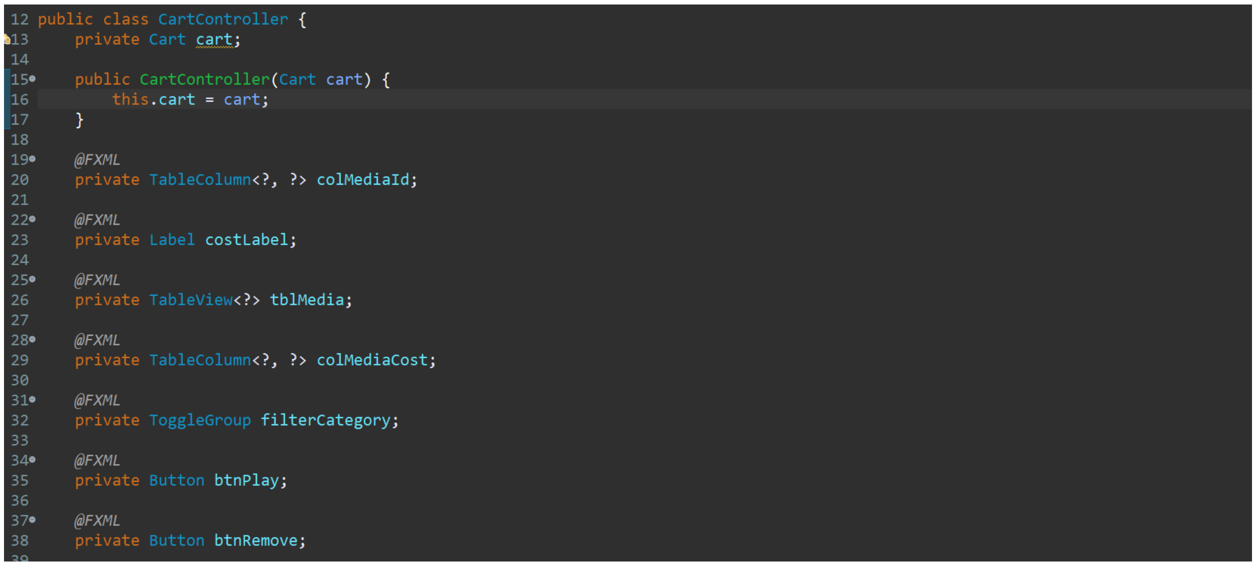
6 Build Cart for AIMS Customer Application

6.1 *Set up Cart Screen with Scene Builder*

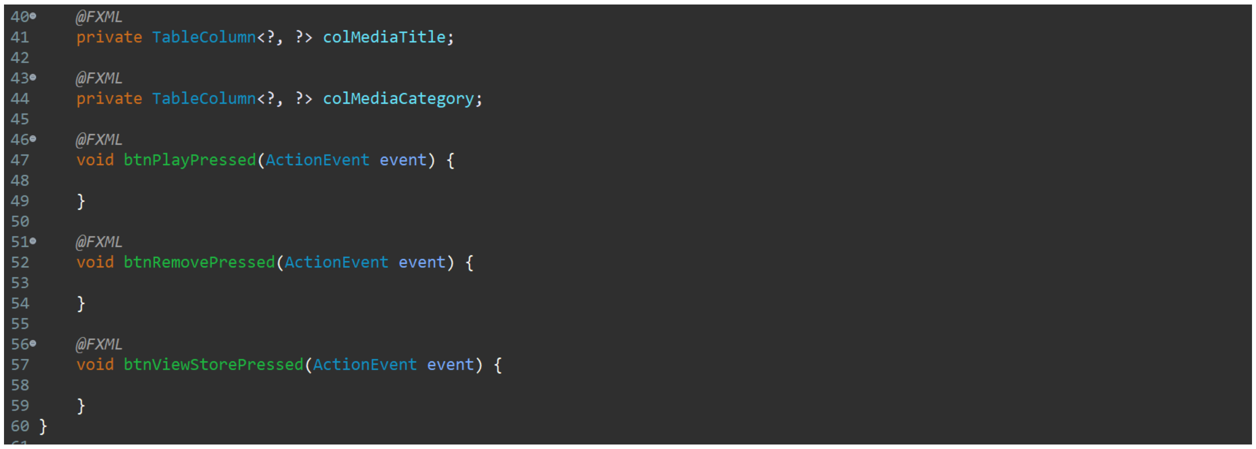


6.2 *Create CartController class*

In the package `customer.screen.controller`, create the CartController class. Add all attributes and methods defined in the *Cart.fxml* file to the Controller. You can copy the Controller Skeleton from Scene Builder by select View > Show Sample Controller Skeleton.



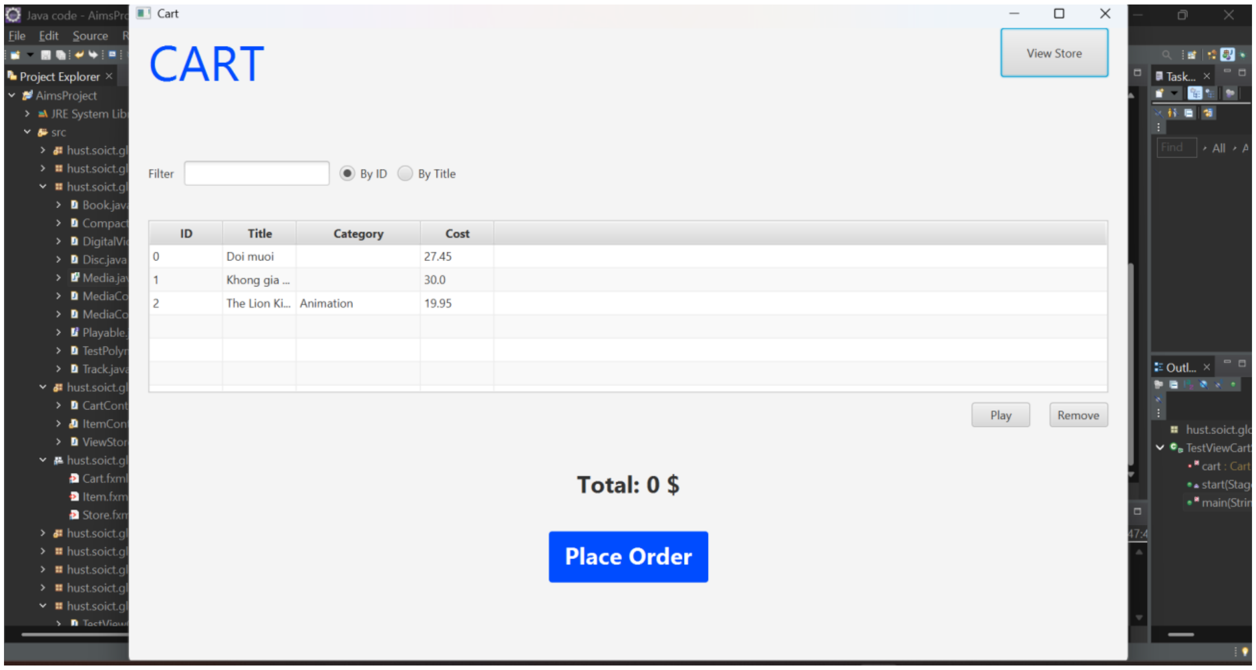
```java
public class CartController {
    private Cart cart;

    public CartController(Cart cart) {
        this.cart = cart;
    }

    @FXML
    private TableColumn<?, ?> colMediaId;

    @FXML
    private Label costLabel;

    @FXML
    private TableView<?> tblMedia;

    @FXML
    private TableColumn<?, ?> colMediaCost;

    @FXML
    private ToggleGroup filterCategory;

    @FXML
    private Button btnPlay;

    @FXML
    private Button btnRemove;
```



```java
    @FXML
    private TableColumn<?, ?> colMediaTitle;

    @FXML
    private TableColumn<?, ?> colMediaCategory;

    @FXML
    void btnPlayPressed(ActionEvent event) {

    }

    @FXML
    void btnRemovePressed(ActionEvent event) {

    }

    @FXML
    void btnViewStorePressed(ActionEvent event) {

    }
}
```

6.3 *View the items in the cart – JavaFX's data-driven UI*

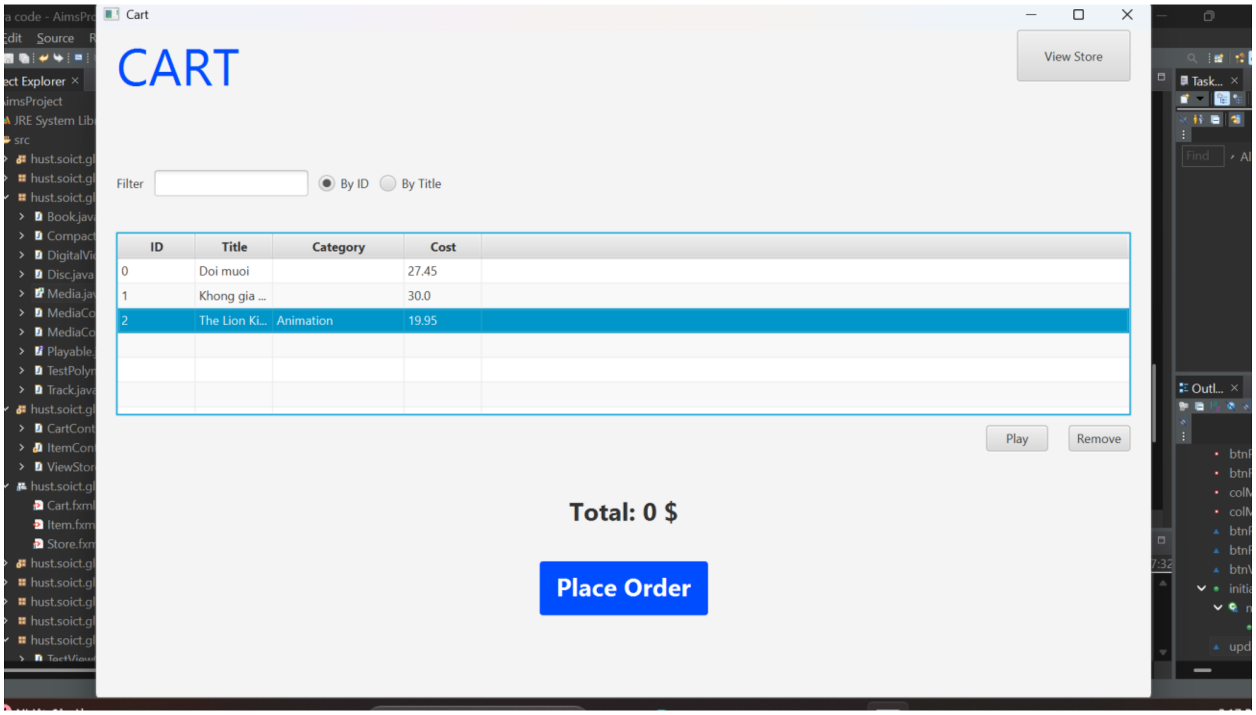
The `TableView` we created in the earlier is currently empty, we need to fill it with data of media items in our cart. Similar to the View Store Controller, we will fill the data in `initialize()` method.



ID	Title	Category	Cost
0	Doi muoi		27.45
1	Khong gia ...		30.0
2	The Lion Ki...	Animation	19.95

Total: 0 $

6.4 *Updating buttons based on the selected item in* `TableView` *–* `ChangeListener`



ID	Title	Category	Cost
0	Doi muoi		27.45
1	Khong gia ...		30.0
2	The Lion Ki...	Animation	19.95

6.5 *Deleting a media*

Next, we will implement the event handling for the "Remove" button. Please add a method name to the onAction property of the button in Scene Builder. You can refer to the event-handling code below:

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

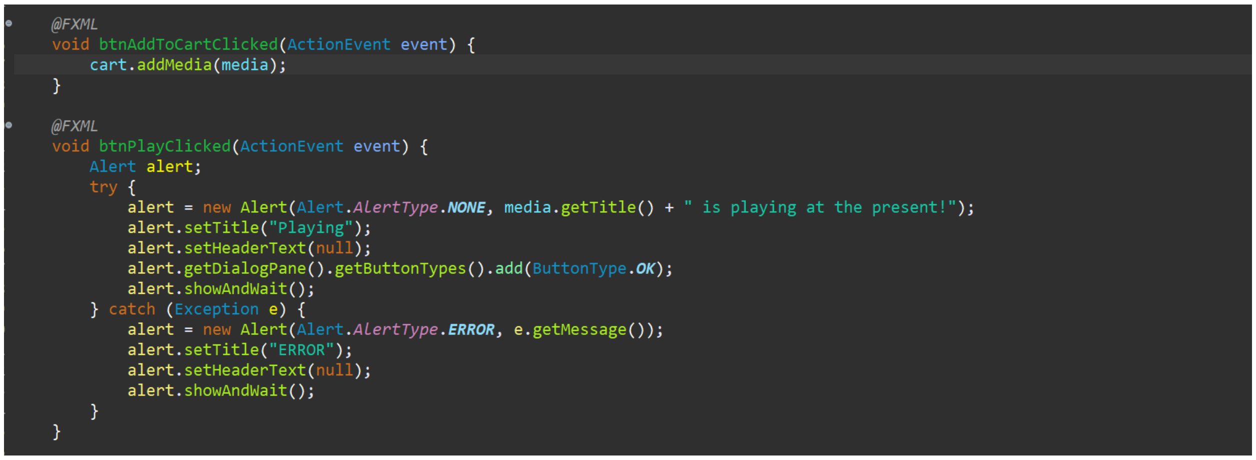
7 Switch Screen between Store and Cart

```java
@FXML
void btnViewCartPressed(ActionEvent event) {
    try {
        final String CART_FXML_FILE_PATH = "/hust/soict/globalict/aims/screen/customer/view/Cart.fxml";
        FXMLLoader fxmlLoader = new FXMLLoader(getClass().getResource(CART_FXML_FILE_PATH));
        fxmlLoader.setController(new CartController(store, cart));
        Parent root = fxmlLoader.load();
        Stage stage = (Stage)((Node)event.getSource()).getScene().getWindow();
        stage.setTitle("Cart");
        stage.setScene(new Scene(root));
        stage.show();

    }catch(IOException e) {
        e.printStackTrace();
    }
}
```

8 Complete the Aims GUI application

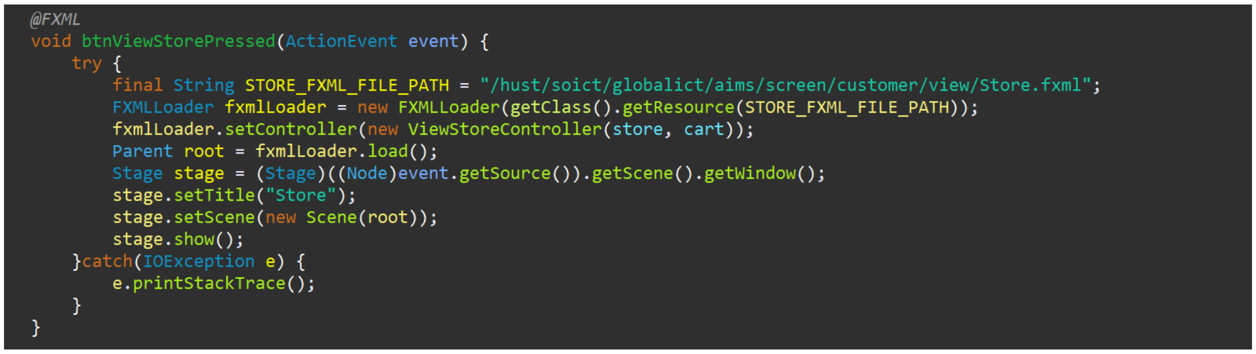
Complete the remaining UI of Aims to make a functioning GUI application

- Store Screen:
  - "Play" Button
  - "Add to cart" Button



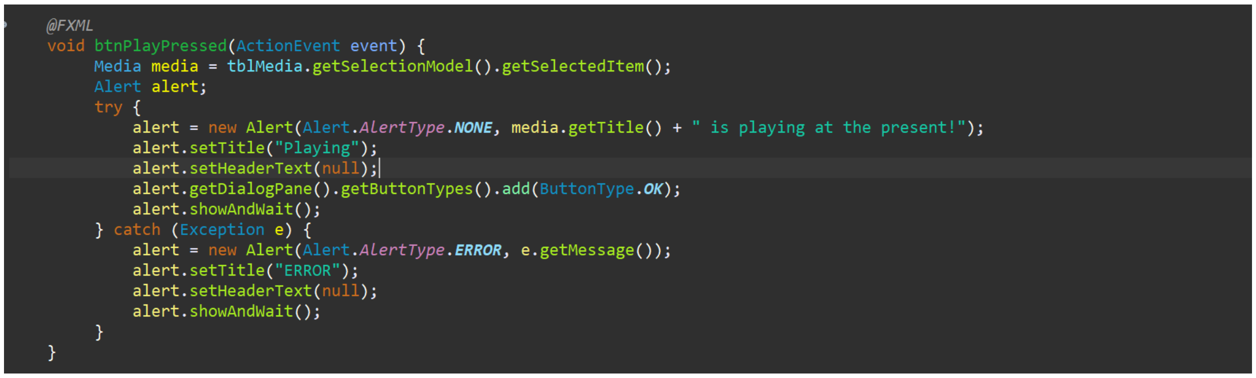
```java
@FXML
void btnAddToCartClicked(ActionEvent event) {
    cart.addMedia(media);
}

@FXML
void btnPlayClicked(ActionEvent event) {
    Alert alert;
    try {
        alert = new Alert(Alert.AlertType.NONE, media.getTitle() + " is playing at the present!");
        alert.setTitle("Playing");
        alert.setHeaderText(null);
        alert.getDialogPane().getButtonTypes().add(ButtonType.OK);
        alert.showAndWait();
    } catch (Exception e) {
        alert = new Alert(Alert.AlertType.ERROR, e.getMessage());
        alert.setTitle("ERROR");
        alert.setHeaderText(null);
        alert.showAndWait();
    }
}
```

- Cart Screen:
  - "View Store" button



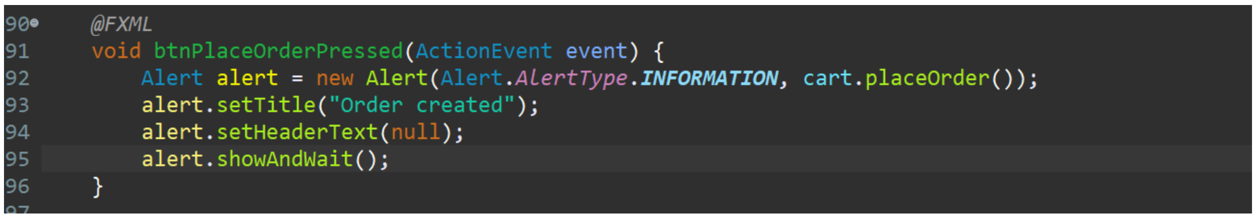
```java
@FXML
void btnViewStorePressed(ActionEvent event) {
    try {
        final String STORE_FXML_FILE_PATH = "/hust/soict/globalict/aims/screen/customer/view/Store.fxml";
        FXMLLoader fxmlLoader = new FXMLLoader(getClass().getResource(STORE_FXML_FILE_PATH));
        fxmlLoader.setController(new ViewStoreController(store, cart));
        Parent root = fxmlLoader.load();
        Stage stage = (Stage)((Node)event.getSource()).getScene().getWindow();
        stage.setTitle("Store");
        stage.setScene(new Scene(root));
        stage.show();
    }catch(IOException e) {
        e.printStackTrace();
    }
}
```

  - "Play" Button



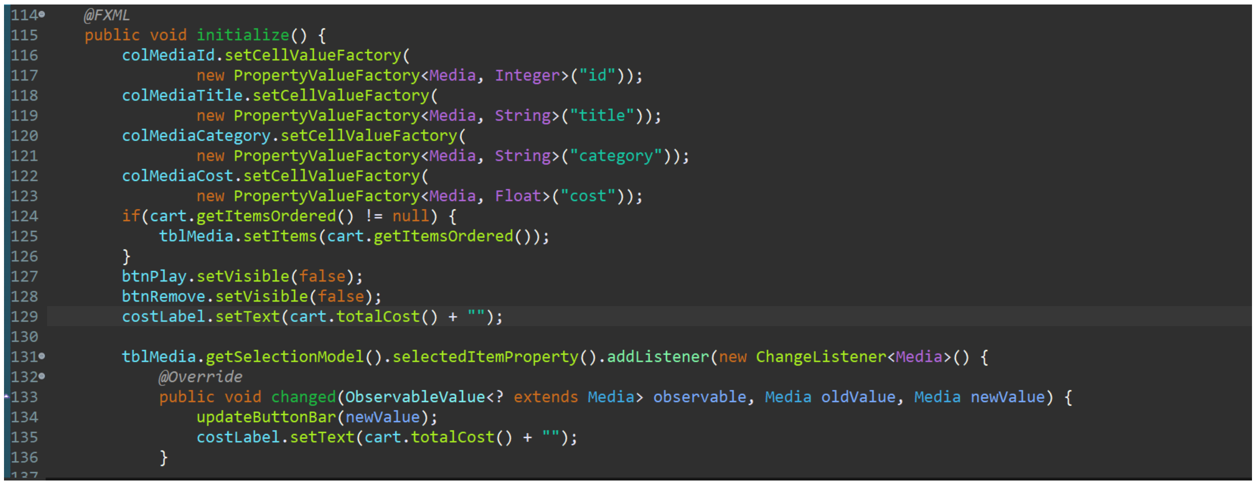
```java
@FXML
void btnPlayPressed(ActionEvent event) {
    Media media = tblMedia.getSelectionModel().getSelectedItem();
    Alert alert;
    try {
        alert = new Alert(Alert.AlertType.NONE, media.getTitle() + " is playing at the present!");
        alert.setTitle("Playing");
        alert.setHeaderText(null);
        alert.getDialogPane().getButtonTypes().add(ButtonType.OK);
        alert.showAndWait();
    } catch (Exception e) {
        alert = new Alert(Alert.AlertType.ERROR, e.getMessage());
        alert.setTitle("ERROR");
        alert.setHeaderText(null);
        alert.showAndWait();
    }
}
```

  - "Place order" Button



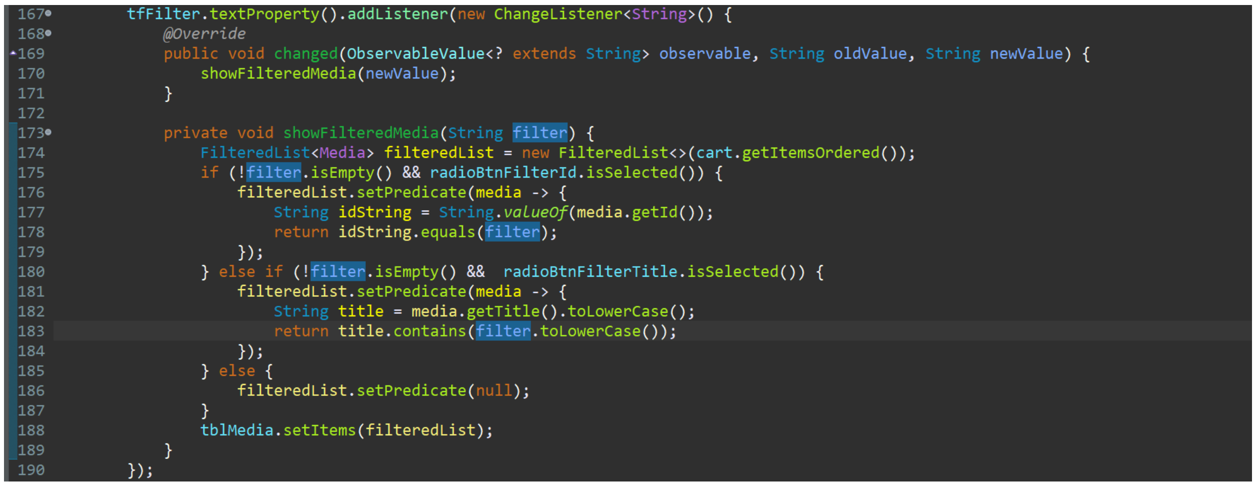
```
    @FXML
    void btnPlaceOrderPressed(ActionEvent event) {
        Alert alert = new Alert(Alert.AlertType.INFORMATION, cart.placeOrder());
        alert.setTitle("Order created");
        alert.setHeaderText(null);
        alert.showAndWait();
    }
```

- The total cost Label - should update along with changes in the current cart



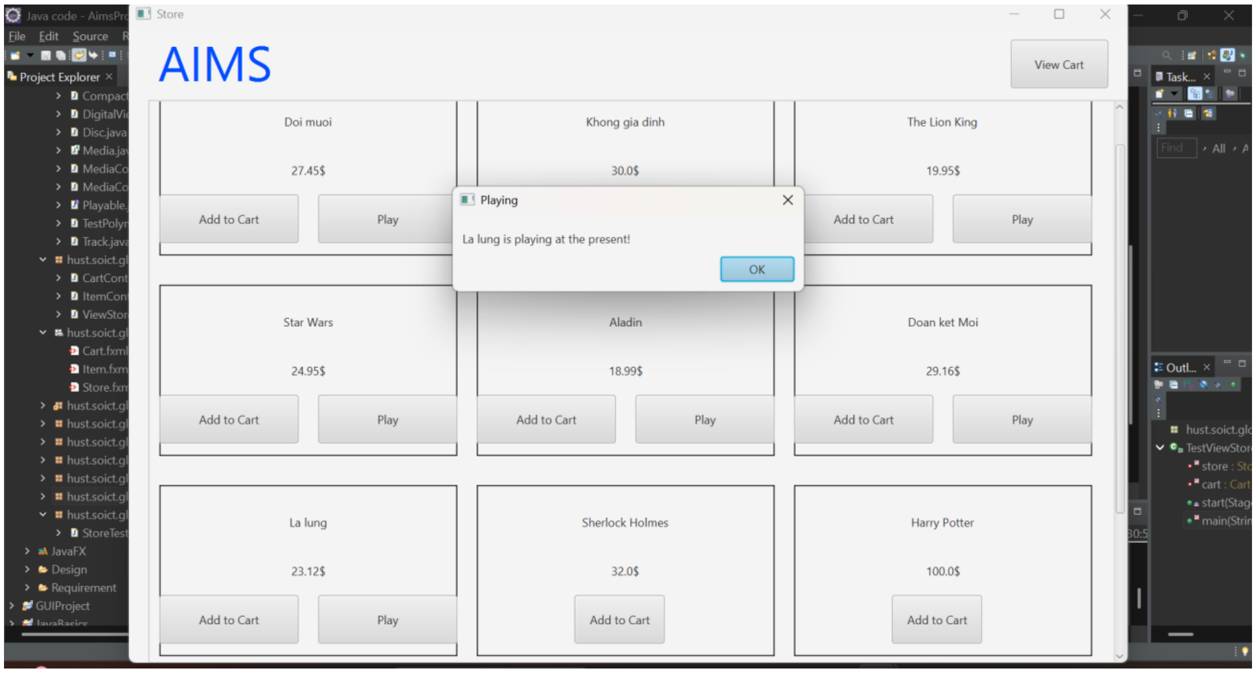
```
    @FXML
    public void initialize() {
        colMediaId.setCellValueFactory(
                new PropertyValueFactory<Media, Integer>("id"));
        colMediaTitle.setCellValueFactory(
                new PropertyValueFactory<Media, String>("title"));
        colMediaCategory.setCellValueFactory(
                new PropertyValueFactory<Media, String>("category"));
        colMediaCost.setCellValueFactory(
                new PropertyValueFactory<Media, Float>("cost"));
        if(cart.getItemsOrdered() != null) {
            tblMedia.setItems(cart.getItemsOrdered());
        }
        btnPlay.setVisible(false);
        btnRemove.setVisible(false);
        costLabel.setText(cart.totalCost() + "");

        tblMedia.getSelectionModel().selectedItemProperty().addListener(new ChangeListener<Media>() {
            @Override
            public void changed(ObservableValue<? extends Media> observable, Media oldValue, Media newValue) {
                updateButtonBar(newValue);
                costLabel.setText(cart.totalCost() + "");
            }
```

- Filter (Search)

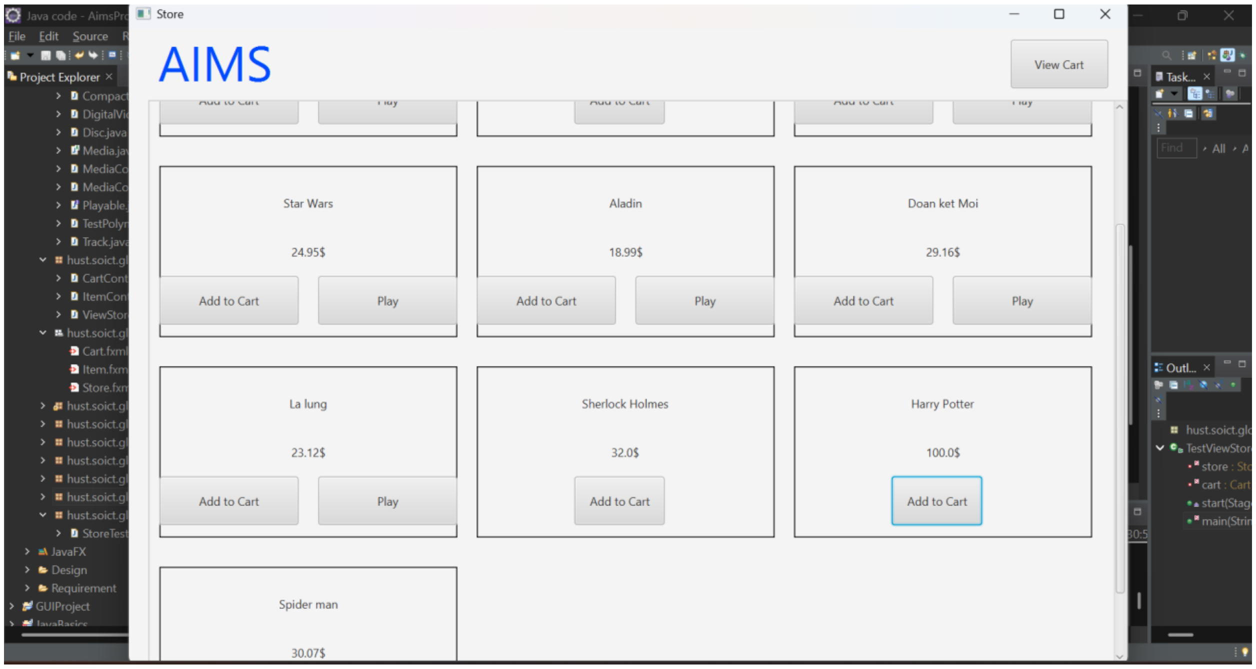


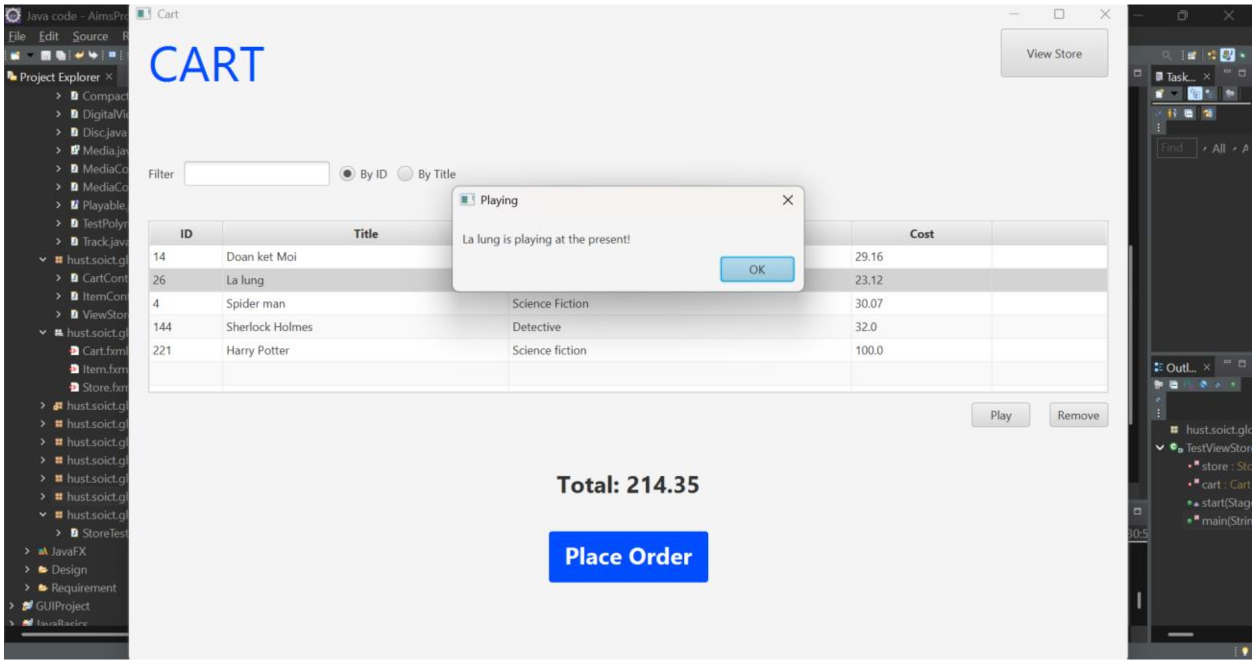
```
        tfFilter.textProperty().addListener(new ChangeListener<String>() {
            @Override
            public void changed(ObservableValue<? extends String> observable, String oldValue, String newValue) {
                showFilteredMedia(newValue);
            }

            private void showFilteredMedia(String filter) {
                FilteredList<Media> filteredList = new FilteredList<>(cart.getItemsOrdered());
                if (!filter.isEmpty() && radioBtnFilterId.isSelected()) {
                    filteredList.setPredicate(media -> {
                        String idString = String.valueOf(media.getId());
                        return idString.equals(filter);
                    });
                } else if (!filter.isEmpty() &&  radioBtnFilterTitle.isSelected()) {
                    filteredList.setPredicate(media -> {
                        String title = media.getTitle().toLowerCase();
                        return title.contains(filter.toLowerCase());
                    });
                } else {
                    filteredList.setPredicate(null);
                }
                tblMedia.setItems(filteredList);
            }
        });
```

DEMO:



La lung is playing at the present!



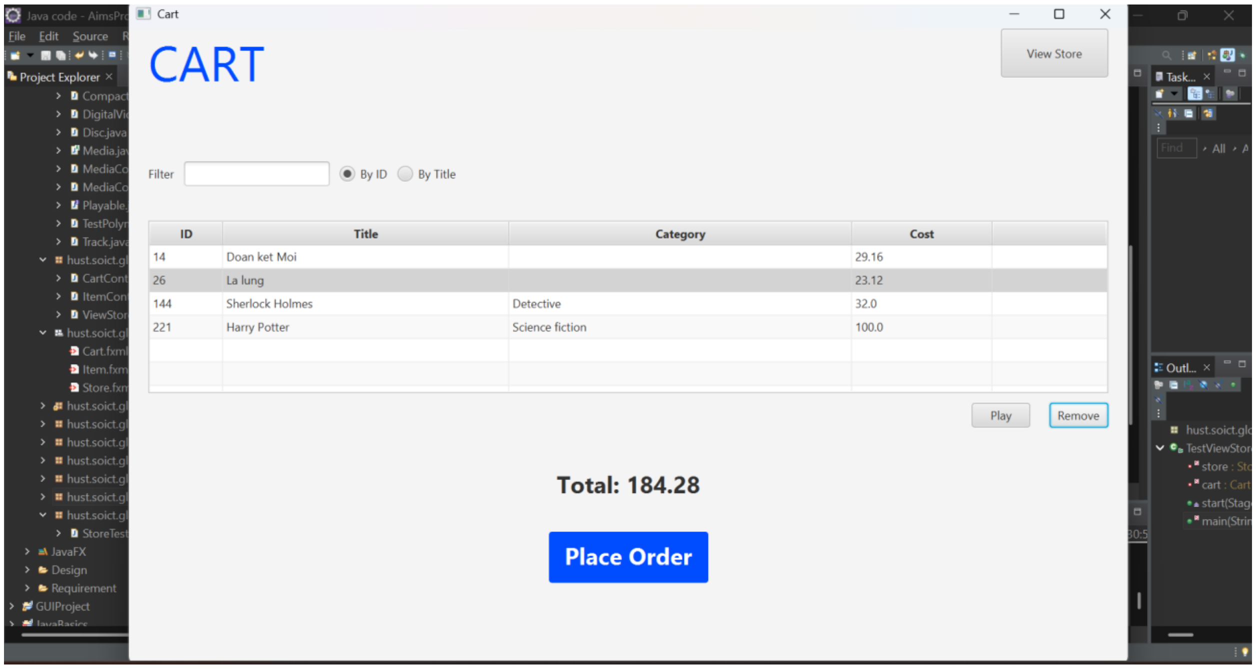


Playing

La lung is playing at the present!

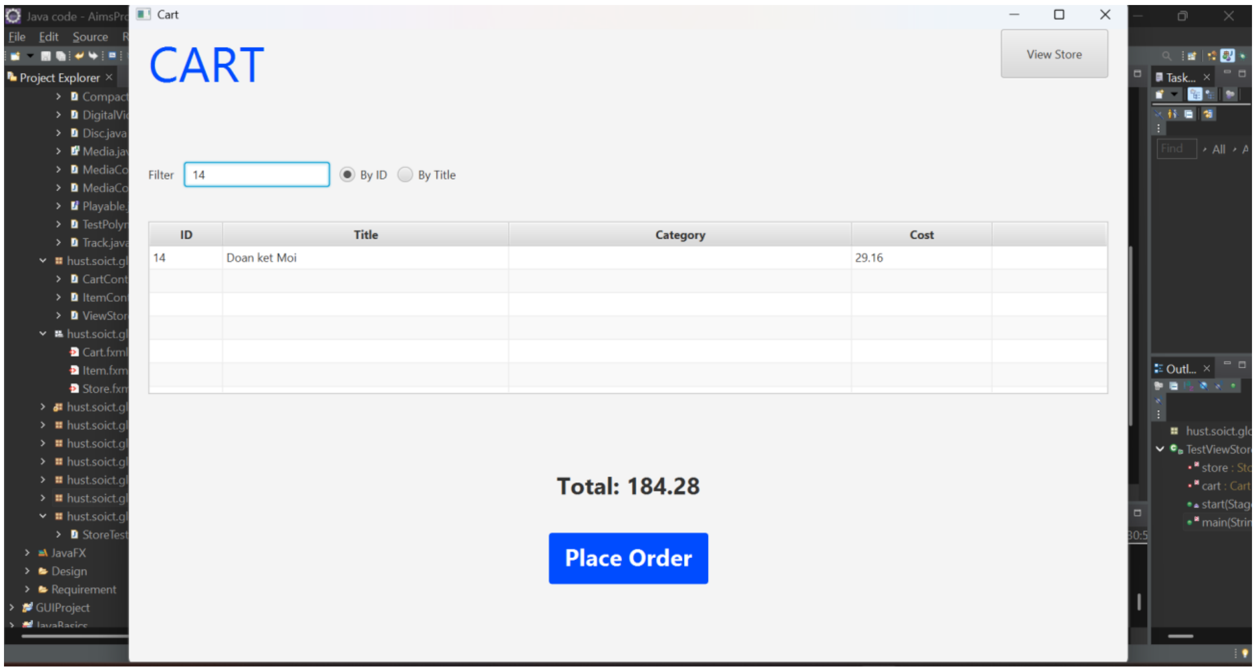
ID	Title	Category	Cost
14	Doan ket Moi		29.16
26	La lung		23.12
4	Spider man	Science Fiction	30.07
144	Sherlock Holmes	Detective	32.0
221	Harry Potter	Science fiction	100.0

Total: 214.35



ID	Title	Category	Cost
14	Doan ket Moi		29.16
26	La lung		23.12
144	Sherlock Holmes	Detective	32.0
221	Harry Potter	Science fiction	100.0

Total: 184.28

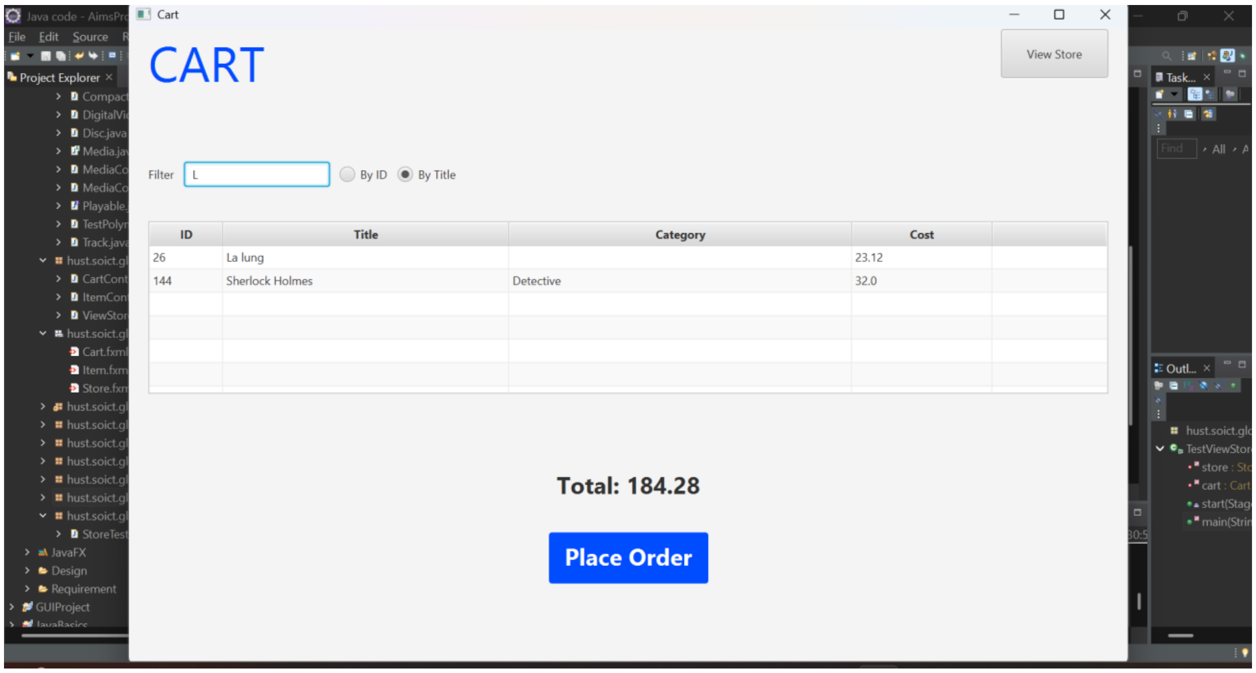


CART

Filter 14 [✓ By ID] By Title

ID	Title	Category	Cost
14	Doan ket Moi		29.16

Total: 184.28

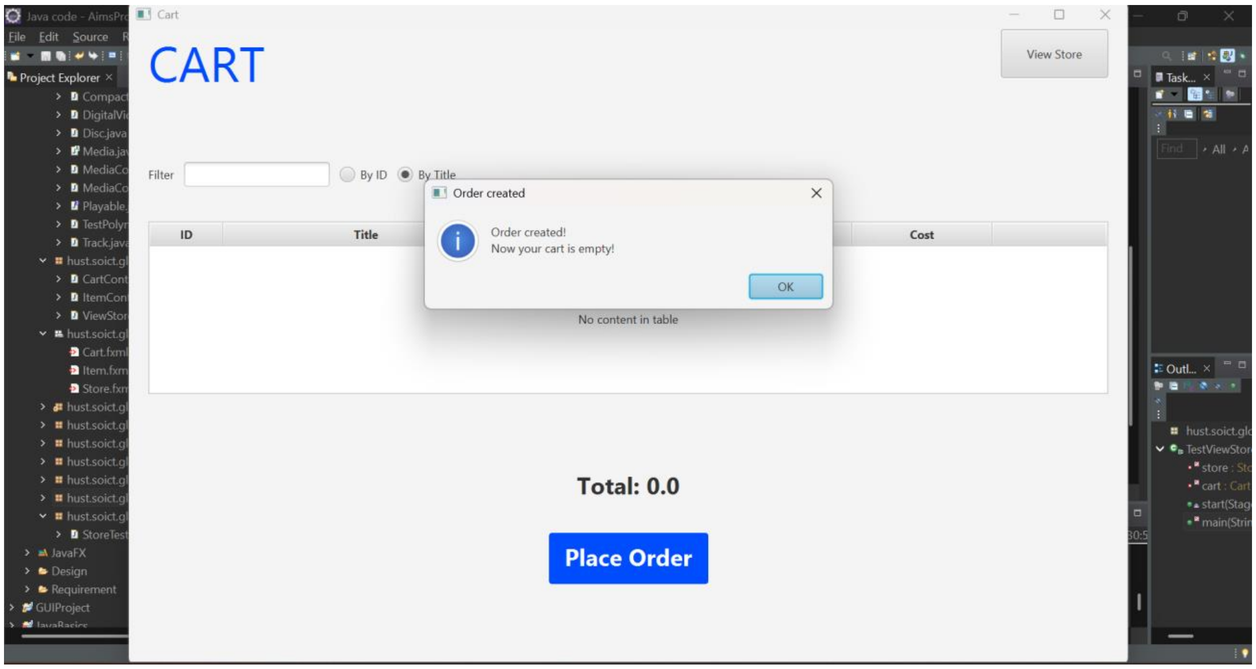


CART

Filter L By ID [✓ By Title]

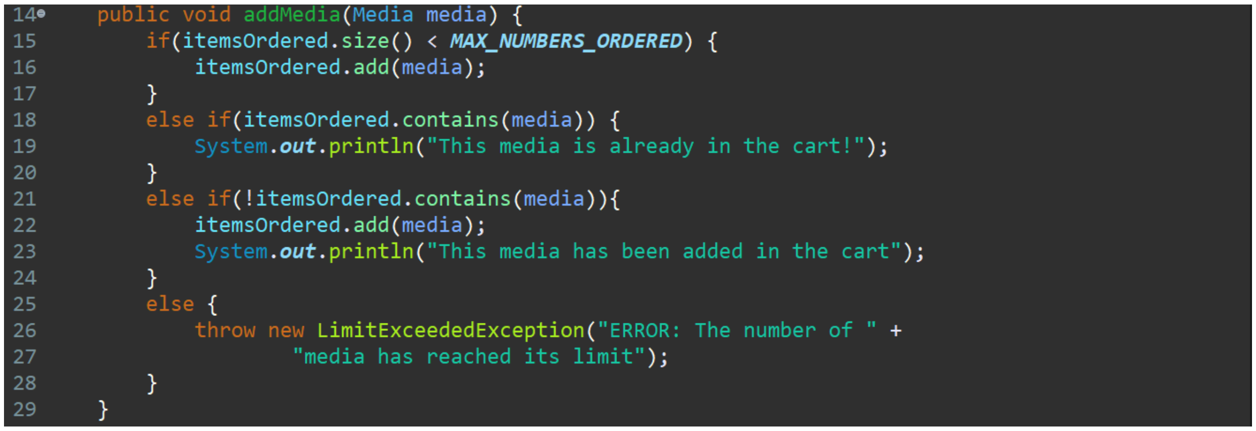
ID	Title	Category	Cost
26	La lung		23.12
144	Sherlock Holmes	Detective	32.0

Total: 184.28

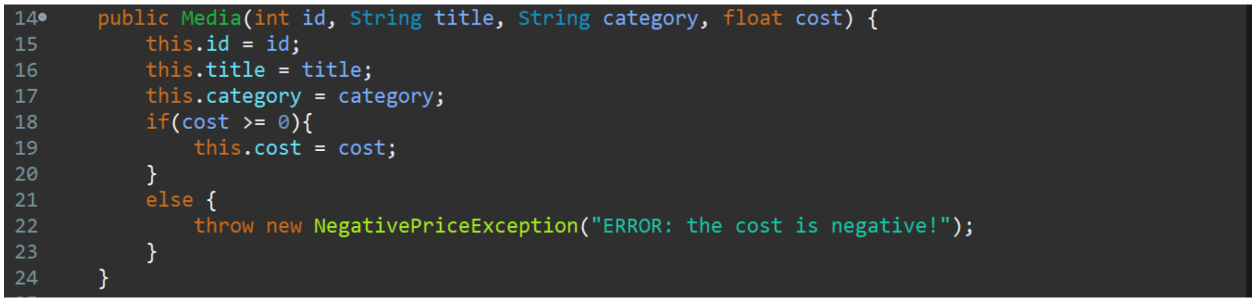


Order created!
Now your cart is empty!

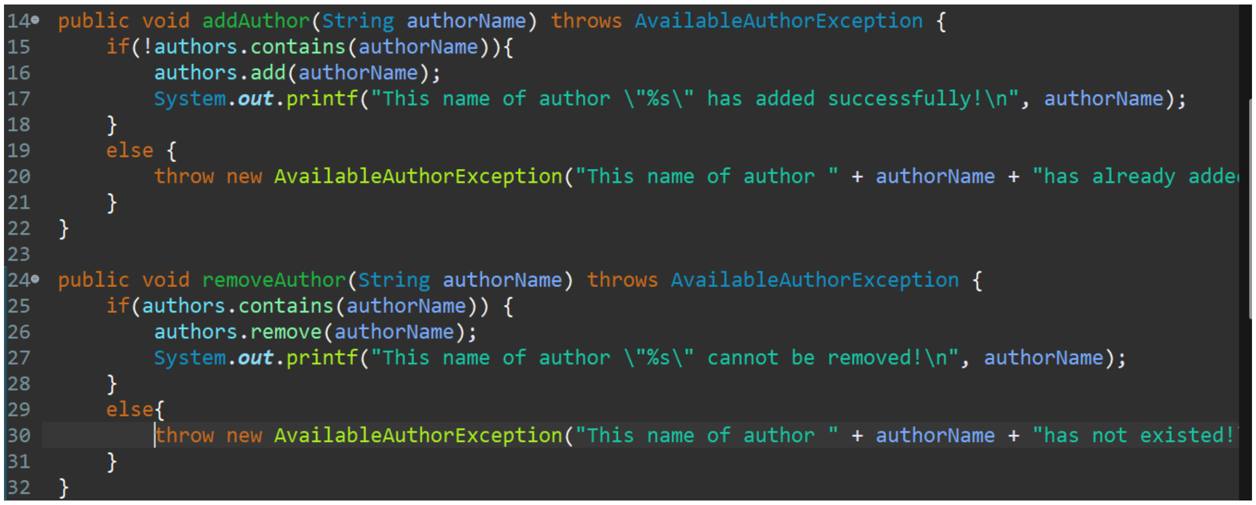
9 Check all the previous source codes to catch/handle/delegate runtime exceptions



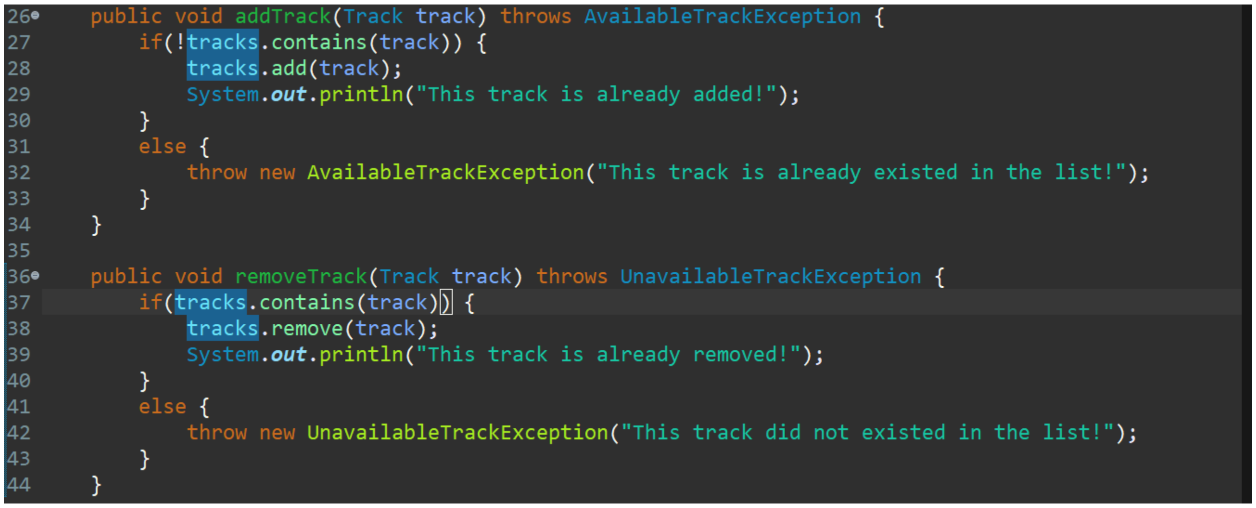
```java
    public void addMedia(Media media) {
        if(itemsOrdered.size() < MAX_NUMBERS_ORDERED) {
            itemsOrdered.add(media);
        }
        else if(itemsOrdered.contains(media)) {
            System.out.println("This media is already in the cart!");
        }
        else if(!itemsOrdered.contains(media)){
            itemsOrdered.add(media);
            System.out.println("This media has been added in the cart");
        }
        else {
            throw new LimitExceededException("ERROR: The number of " +
                    "media has reached its limit");
        }
    }
```



```java
    public Media(int id, String title, String category, float cost) {
        this.id = id;
        this.title = title;
        this.category = category;
        if(cost >= 0){
            this.cost = cost;
        }
        else {
            throw new NegativePriceException("ERROR: the cost is negative!");
        }
    }
```



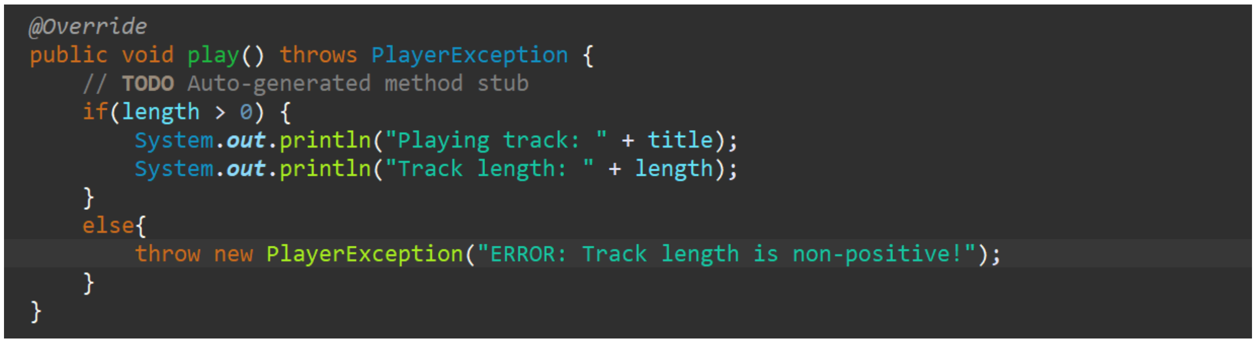
```java
public void addAuthor(String authorName) throws AvailableAuthorException {
    if(!authors.contains(authorName)){
        authors.add(authorName);
        System.out.printf("This name of author \"%s\" has added successfully!\n", authorName);
    }
    else {
        throw new AvailableAuthorException("This name of author " + authorName + "has already adde
    }
}

public void removeAuthor(String authorName) throws AvailableAuthorException {
    if(authors.contains(authorName)) {
        authors.remove(authorName);
        System.out.printf("This name of author \"%s\" cannot be removed!\n", authorName);
    }
    else{
        throw new AvailableAuthorException("This name of author " + authorName + "has not existed!
    }
}
```



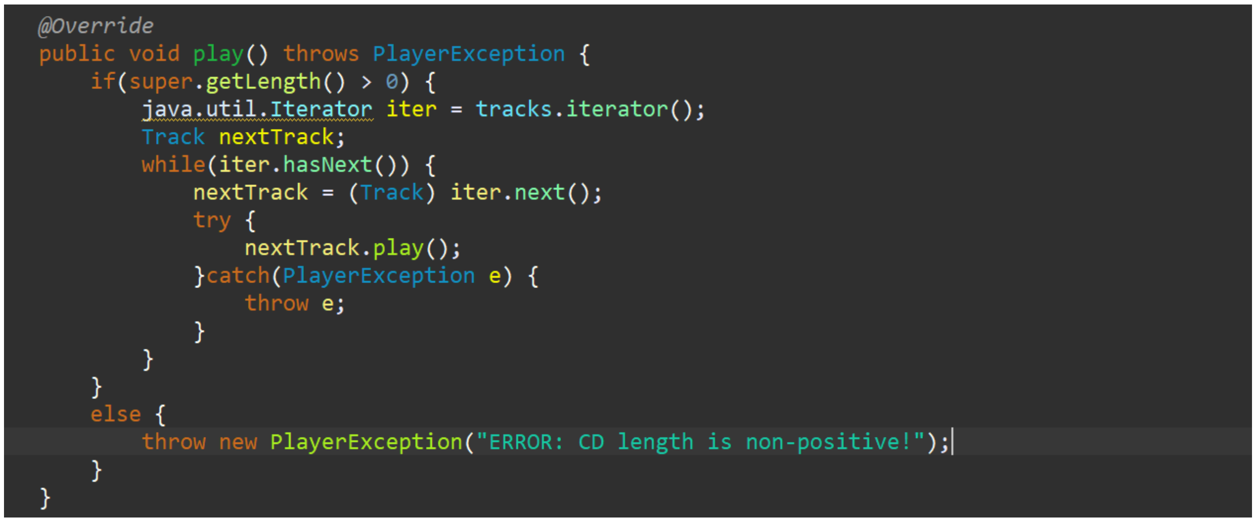
```java
    public void addTrack(Track track) throws AvailableTrackException {
        if(!tracks.contains(track)) {
            tracks.add(track);
            System.out.println("This track is already added!");
        }
        else {
            throw new AvailableTrackException("This track is already existed in the list!");
        }
    }

    public void removeTrack(Track track) throws UnavailableTrackException {
        if(tracks.contains(track)) {
            tracks.remove(track);
            System.out.println("This track is already removed!");
        }
        else {
            throw new UnavailableTrackException("This track did not existed in the list!");
        }
    }
```

10 Create a class which inherits from **Exception**

The **PlayerException** class represents an exception that will be thrown when an exceptional condition occurs during the playing of a media in your **AimsProject**.

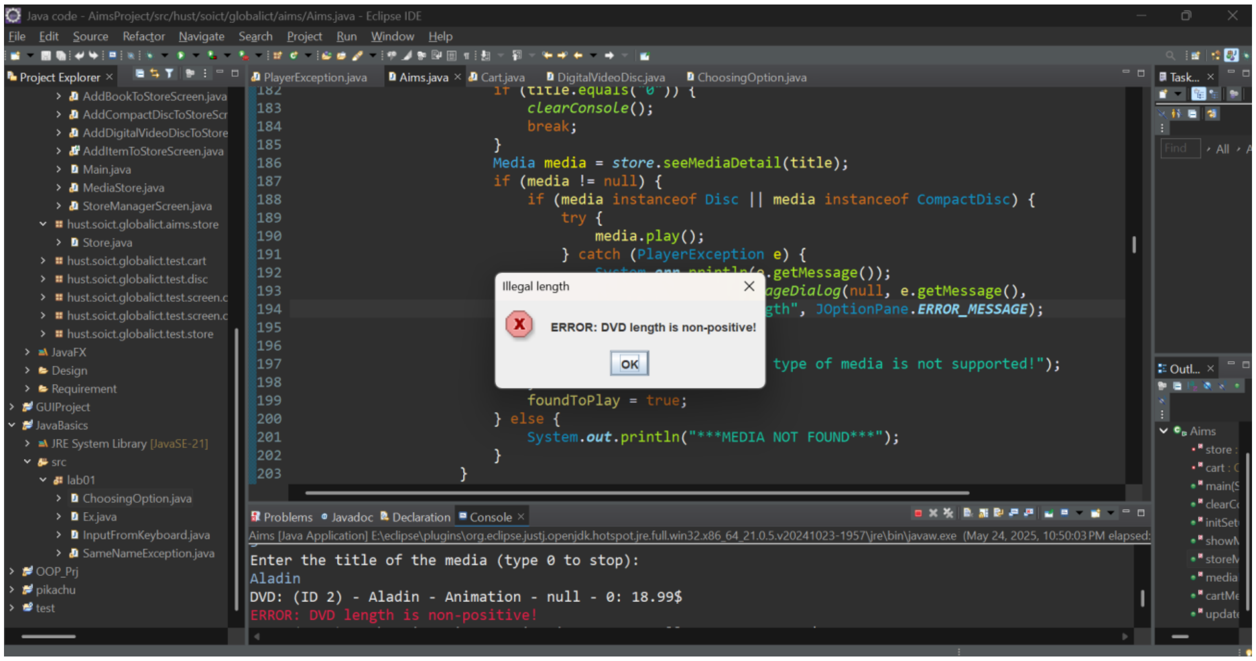


```java
@Override
public void play() throws PlayerException {
    // TODO Auto-generated method stub
    if(length > 0) {
        System.out.println("Playing track: " + title);
        System.out.println("Track length: " + length);
    }
    else{
        throw new PlayerException("ERROR: Track length is non-positive!");
    }
}
```



```java
@Override
public void play() throws PlayerException {
    if(super.getLength() > 0) {
        java.util.Iterator iter = tracks.iterator();
        Track nextTrack;
        while(iter.hasNext()) {
            nextTrack = (Track) iter.next();
            try {
                nextTrack.play();
            }catch(PlayerException e) {
                throw e;
            }
        }
    }
    else {
        throw new PlayerException("ERROR: CD length is non-positive!");
    }
}
```

11 Update the **Aims** class



```java
                    if (title.equals("0")) {
                        clearConsole();
                        break;
                    }
                    Media media = store.seeMediaDetail(title);
                    if (media != null) {
                        if (media instanceof Disc || media instanceof CompactDisc) {
                            try {
                                media.play();
                            } catch (PlayerException e) {
                                                   .getMessage());
                                               ageDialog(null, e.getMessage(),
                                               gth", JOptionPane.ERROR_MESSAGE);


                                               type of media is not supported!");

                        foundToPlay = true;
                    } else {
                        System.out.println("***MEDIA NOT FOUND***");
                    }
                }
```

```
Illegal length
ERROR: DVD length is non-positive!
OK
```

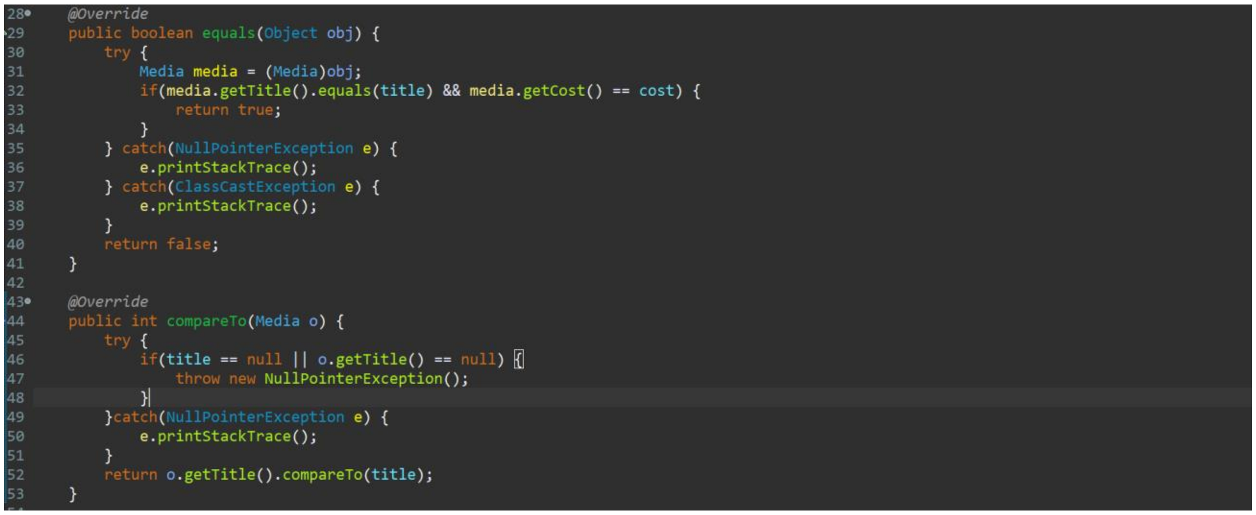
```
Aims [Java Application] E:\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_21.0.5.v20241023-1957\jre\bin\javaw.exe (May 24, 2025, 10:50:03 PM elapsed:
Enter the title of the media (type 0 to stop):
Aladin
DVD: (ID 2) - Aladin - Animation - null - 0: 18.99$
ERROR: DVD length is non-positive!
```

12 Modify the **equals()** method and **compareTo()** method of Comparable for Media class

- Two medias are equals if they have the same **title** and **cost**

- Please remember to check for **NullPointerException** and **ClassCastException** if applicable.

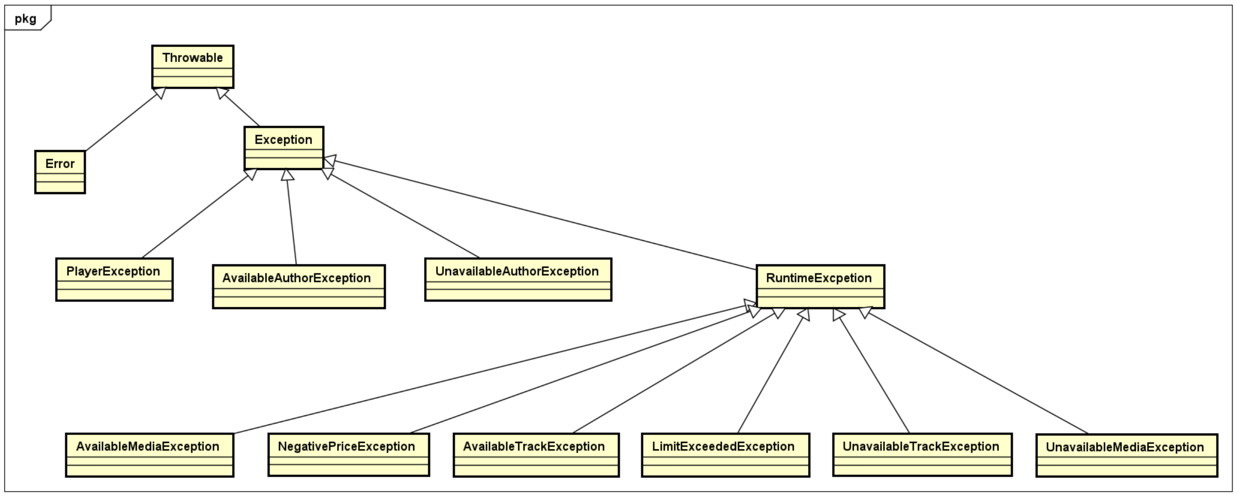
You may use **instanceof** operator to check if an object is an instance of a **ClassType**.



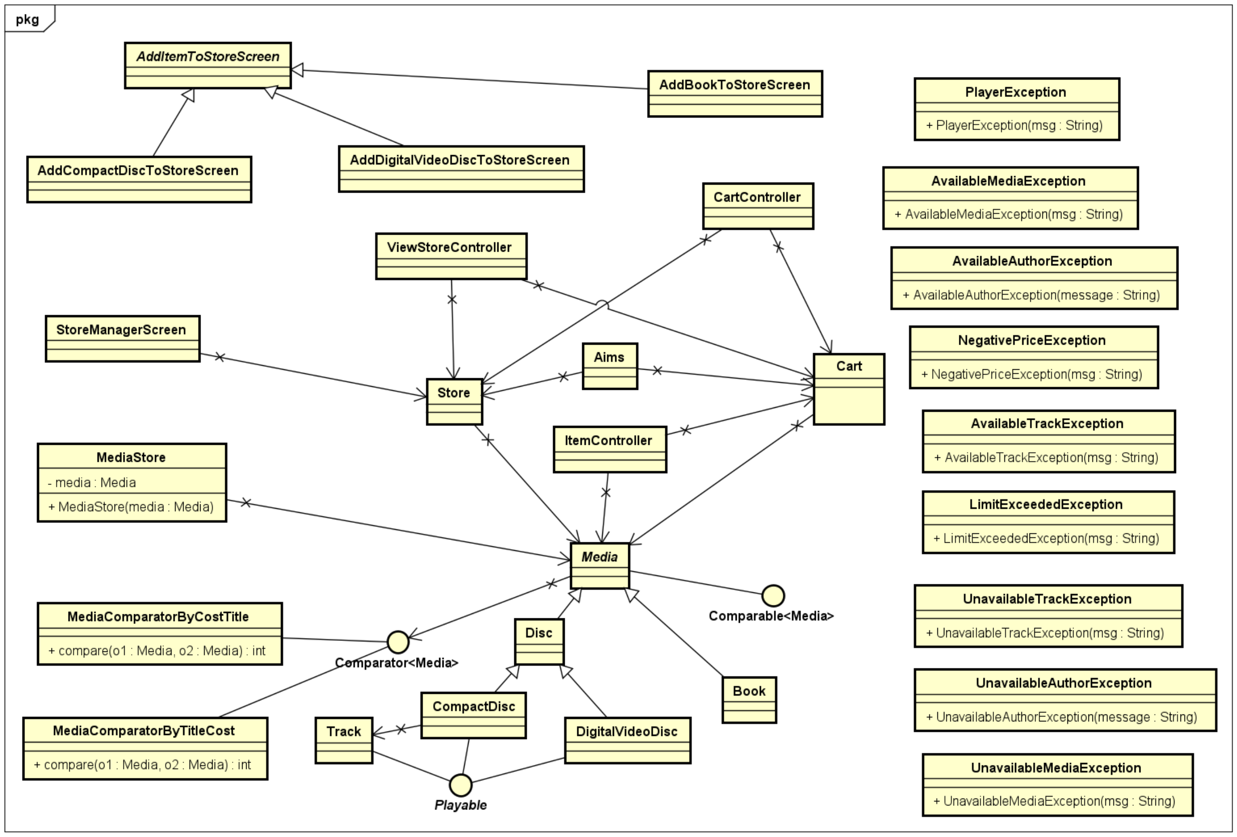
```
    @Override
    public boolean equals(Object obj) {
        try {
            Media media = (Media)obj;
            if(media.getTitle().equals(title) && media.getCost() == cost) {
                return true;
            }
        } catch(NullPointerException e) {
            e.printStackTrace();
        } catch(ClassCastException e) {
            e.printStackTrace();
        }
        return false;
    }

    @Override
    public int compareTo(Media o) {
        try {
            if(title == null || o.getTitle() == null) {
                throw new NullPointerException();
            }
        }catch(NullPointerException e) {
            e.printStackTrace();
        }
        return o.getTitle().compareTo(title);
    }
```

14 Exercises: Hierarchical tree diagram

- Make an exception hierarchical tree for all self-defined exceptions in Aims Project. Use the class diagram in Astah to draw this tree, export it as a png file, and save them in the design directory.



Update class diagram:


pkg
AddItemToStoreScreen
AddBookToStoreScreen
AddCompactDiscToStoreScreen
AddDigitalVideoDiscToStoreScreen
PlayerException
+ PlayerException(msg : String)
AvailableMediaException
+ AvailableMediaException(msg : String)
AvailableAuthorException
+ AvailableAuthorException(message : String)
NegativePriceException
+ NegativePriceException(msg : String)
AvailableTrackException
+ AvailableTrackException(msg : String)
LimitExceededException
+ LimitExceededException(msg : String)
UnavailableTrackException
+ UnavailableTrackException(msg : String)
UnavailableAuthorException
+ UnavailableAuthorException(message : String)
UnavailableMediaException
+ UnavailableMediaException(msg : String)
CartController
ViewStoreController
StoreManagerScreen
Store
Aims
Cart
ItemController
MediaStore
- media : Media
+ MediaStore(media : Media)
Media
Comparable<Media>
MediaComparatorByCostTitle
+ compare(o1 : Media, o2 : Media) : int
Comparator<Media>
Disc
Book
MediaComparatorByTitleCost
+ compare(o1 : Media, o2 : Media) : int
Track
CompactDisc
DigitalVideoDisc
Playable