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

BÁO CÁO THỰC HÀNH
IT3103

BÀI THỰC HÀNH 5

Họ và tên sv: Nguyễn Minh Tuấn
Lớp: **K67-CNTT Việt Pháp 01**

Hà Nội 12/2024

BÁO CÁO THỰC HÀNH LAB 5

LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG

Contents

1. Swing component

1.1. Lớp AWTAccumulator

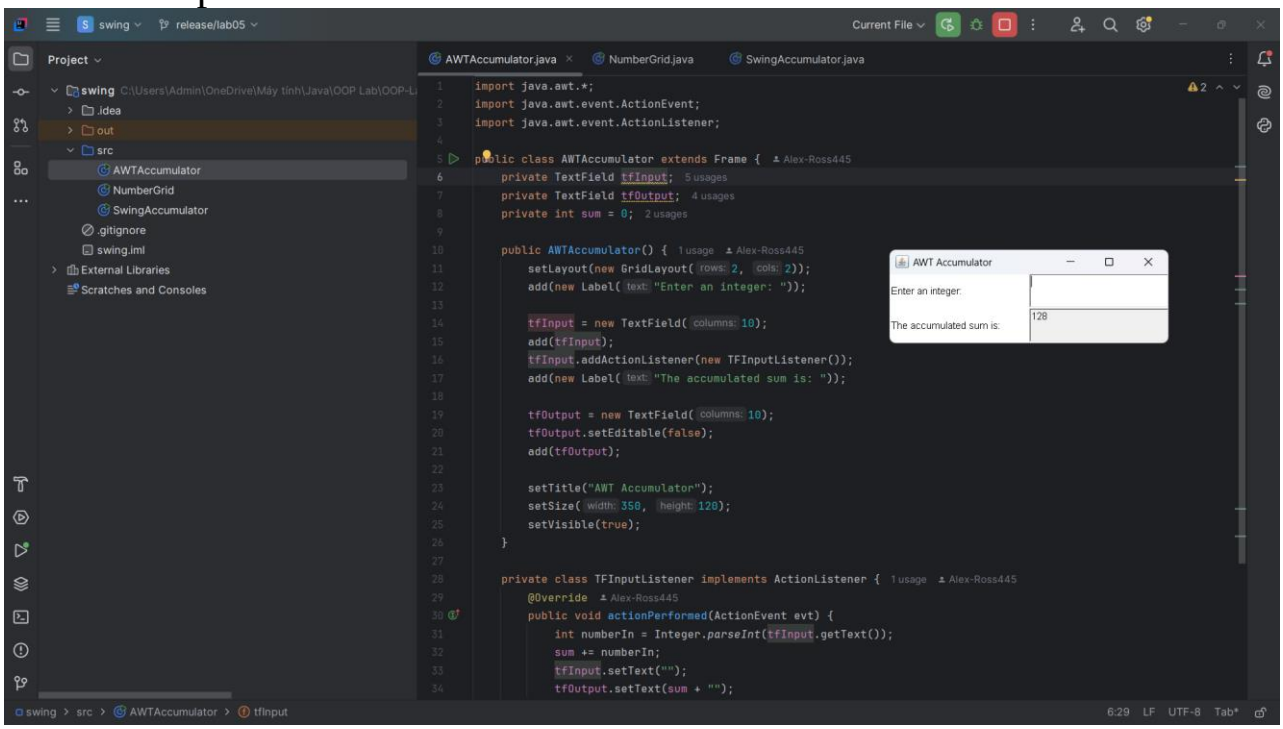

Figure 1: Mã nguồn và giao diện lớp AWTAccumulator

1.2. Lớp SwingAccumulator

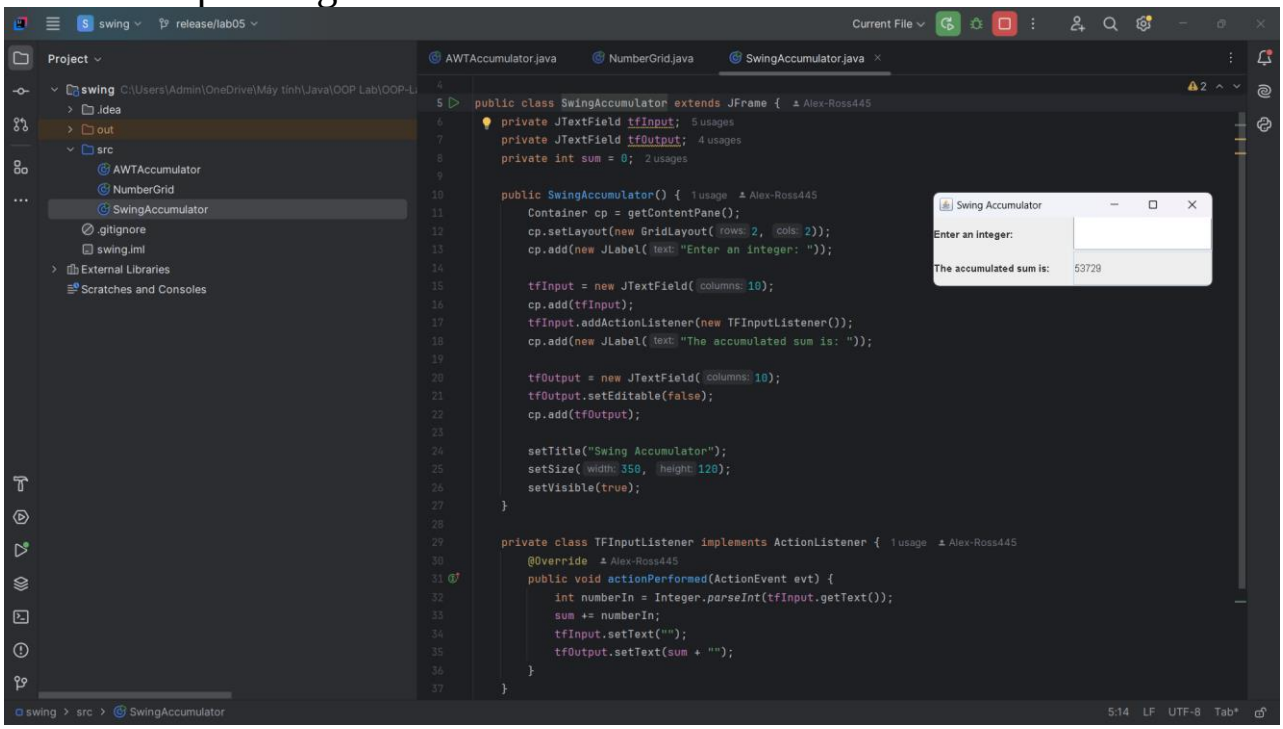

Figure 2: Giao diện và mã nguồn lớp SwingAccumulator

1.3: Compare Swing and AWT elements

- The top-level containers in Swing and AWT:
    - Swing: JFrame, JDialog, JApplet, Jwindow
    - AWT: Frame, Dialog, Applet
- The class name of components in AWT and corresponding class's name in Swing

AWT	Swing
Button	JButton
Label	JLabel
TextField	JTextField
TextArea	JTextArea
Checkbox	JCheckBox
Choice	JComboBox
List	JList
ScrollBar	JScrollBar
Panel	JPanel
Canvas	
MenuBar	JMenuBar
Menu	JMenu
MenuItem	JMenuItem

- The main differences between Swing and AWT:
    - AWT uses native components provided by the operating system (heavyweight), whereas Swing is entirely written in Java and rendered on a lightweight layer.
    - Swing offers many extended components that AWT does not, such as JTable, JTree, JTabbedPane, and more.

2. Organizing Swing components with Layout Managers

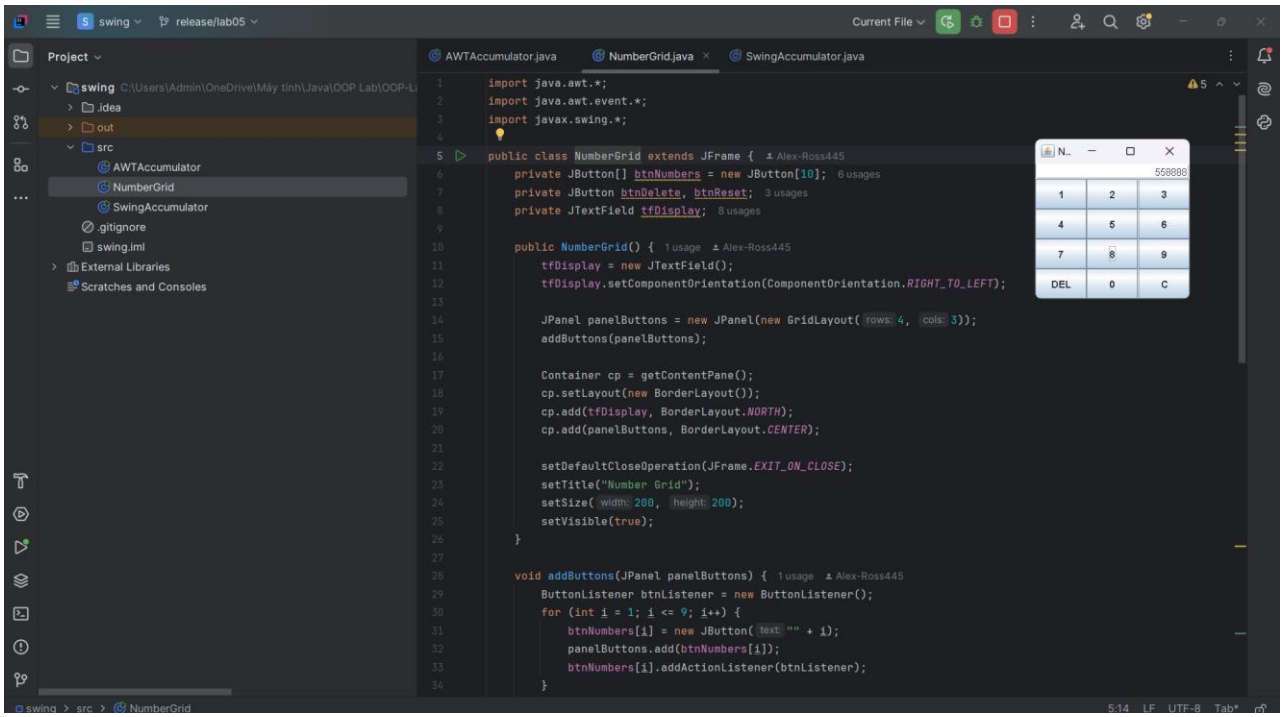

Figure 3: Mã nguồn và giao diện lớp NumberGrid

3. Create a graphical user interface for AIMS with Swing

3.1. Lớp StoreScreen

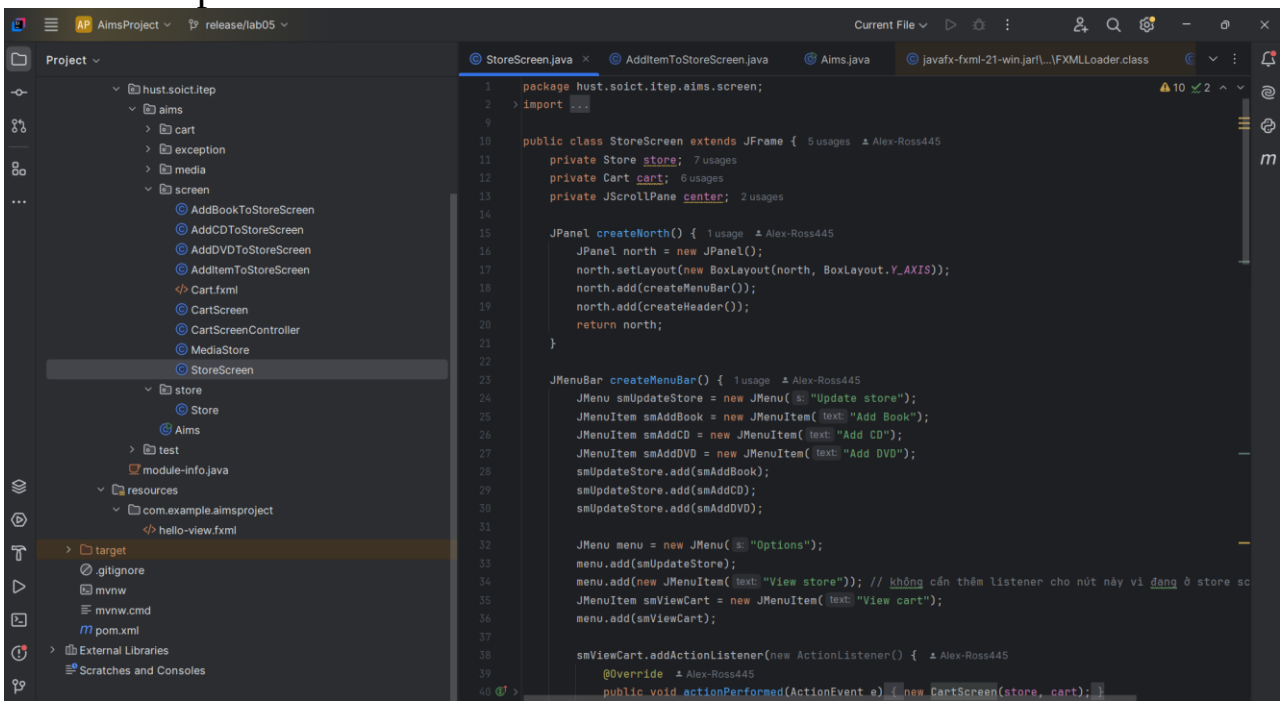

Figure 4: Mã nguồn lớp StoreScreen

3.2. Lớp MediaStore

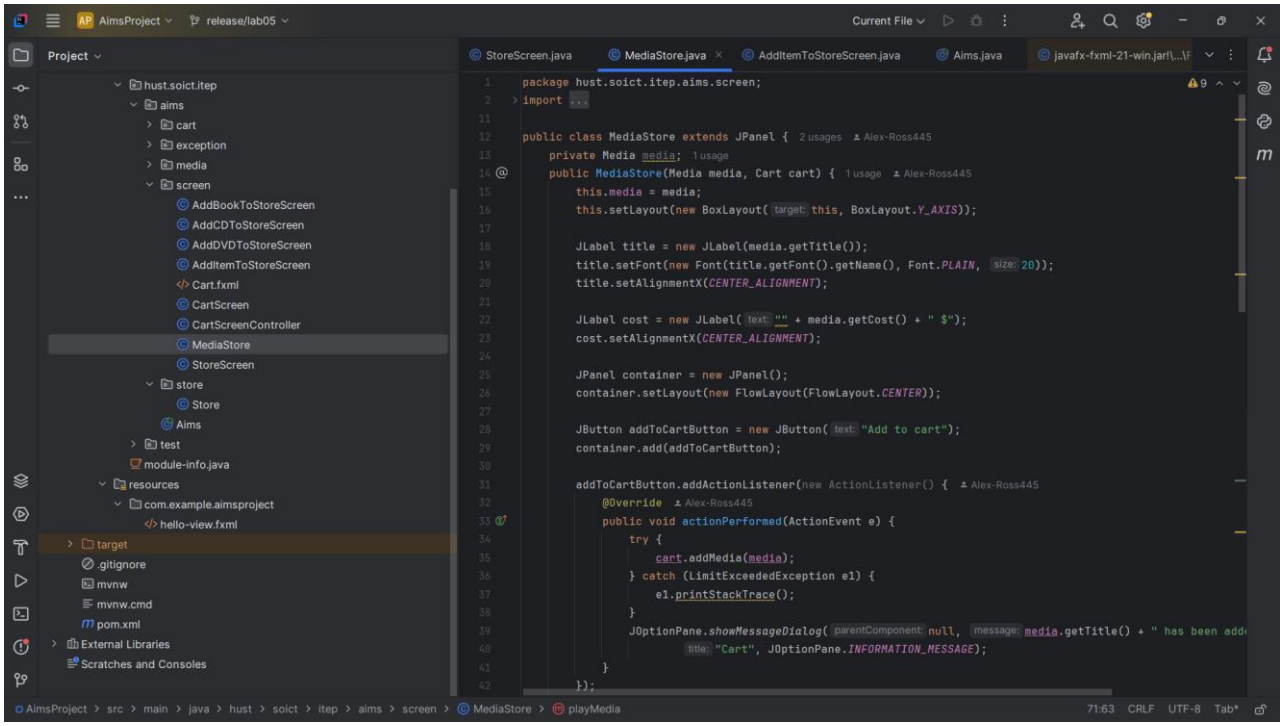

Figure 5: Mã nguồn lớp MediaStore

3.3. Chức năng của nút Play và nút Add to cart

3.3.1. Nút Play

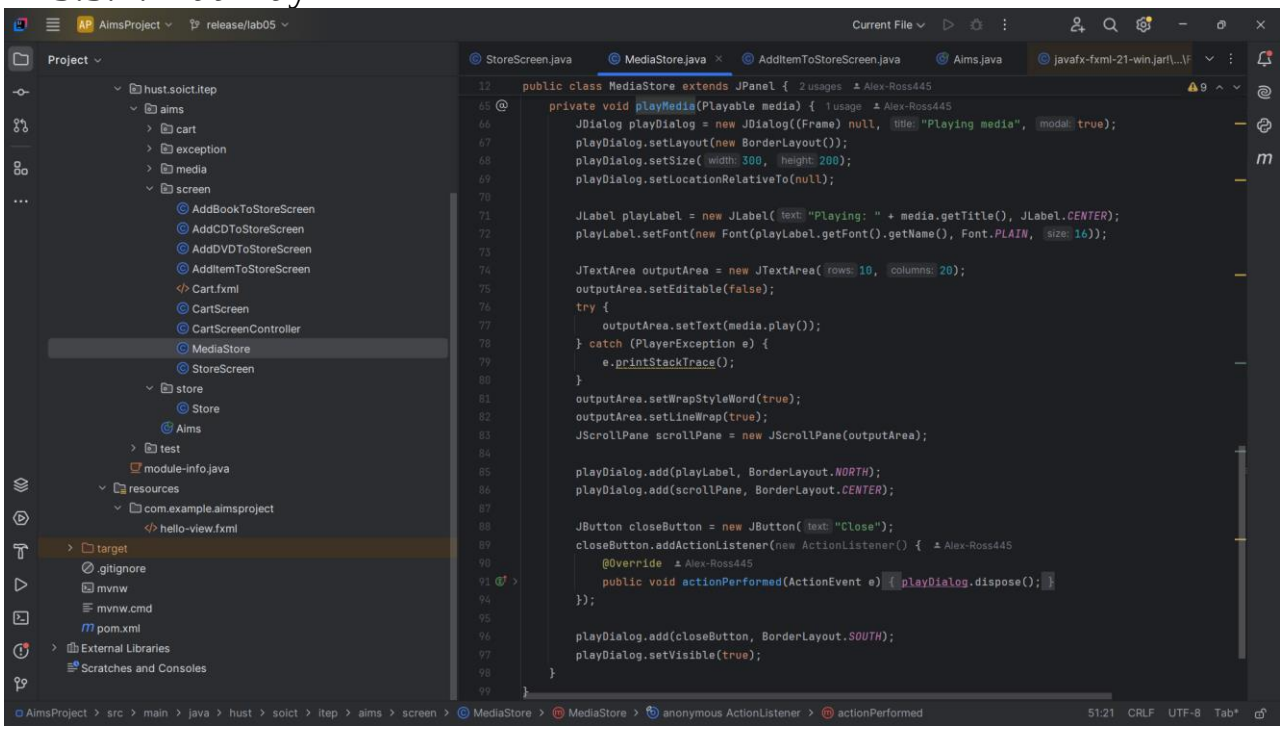

Figure 6: Mã nguồn nút Play của MediaStore

3.3.2.Nút Add to cart

```
JButton addToCartButton = new JButton( text: "Add to cart");
container.add(addToCartButton);

addToCartButton.addActionListener(new ActionListener() {  Alex-Ross445
    @Override  Alex-Ross445
    public void actionPerformed(ActionEvent e) {
        try {
            cart.addMedia(media);
        } catch (LimitExceededException e1) {
            e1.printStackTrace();
        }
        JOptionPane.showMessageDialog( parentComponent: null,  message: media.getTitle() + " has been add
                title: "Cart", JOptionPane.INFORMATION_MESSAGE);
    }
});
```

Figure 7: Mã nguồn nút Add to cart của MediaStore

3.4. Giao diện

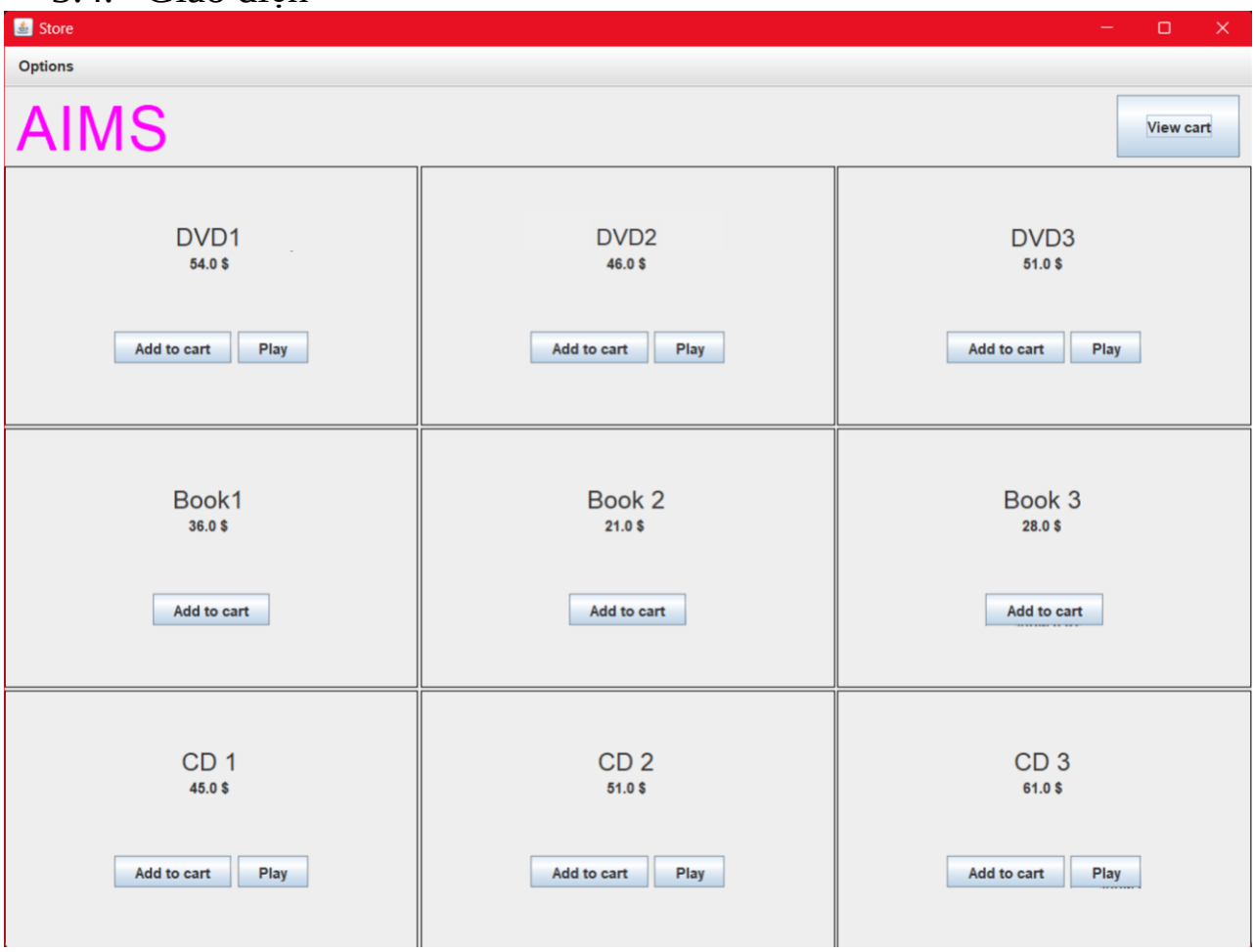

Figure 8: Giao diện Store

Figure 9: Giao diện Play DVD media trong

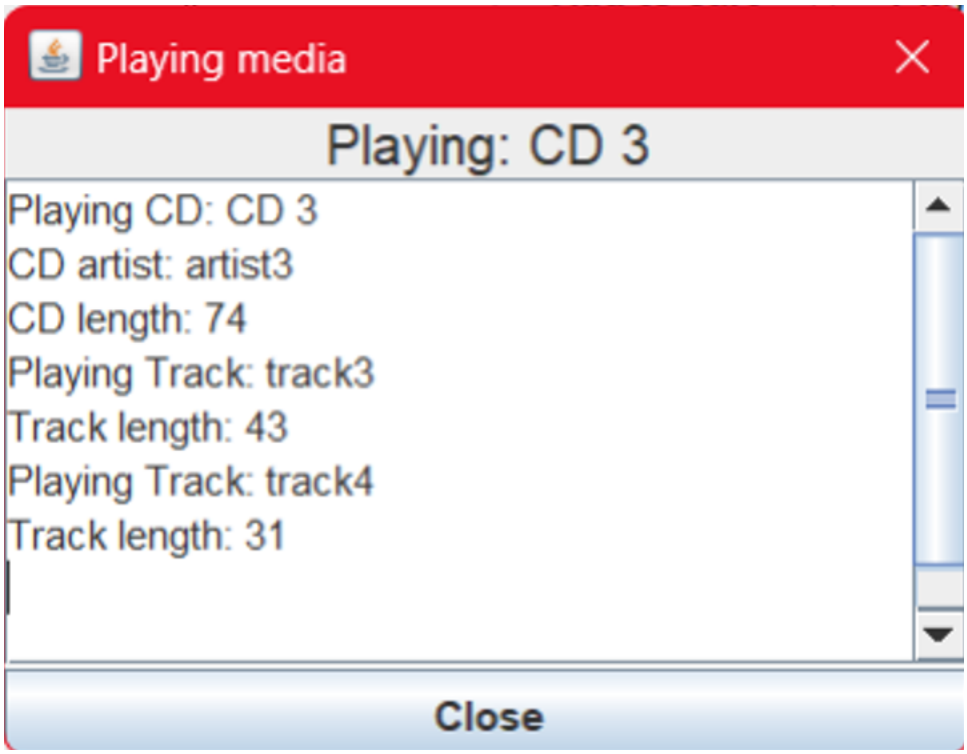

Figure 10: Giao diện Play CD media trong Store

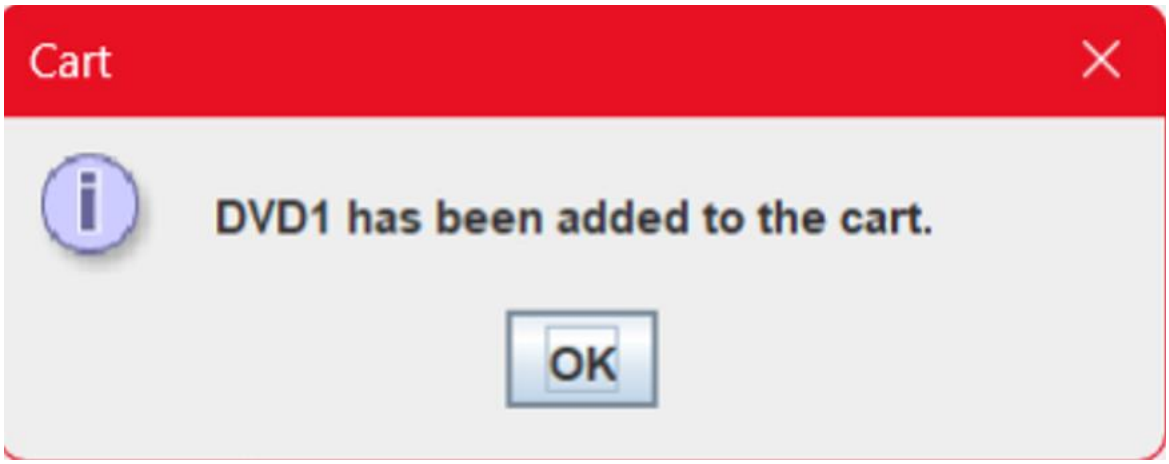

Figure 11: Thông báo thêm media vào cart

4. JavaFX API

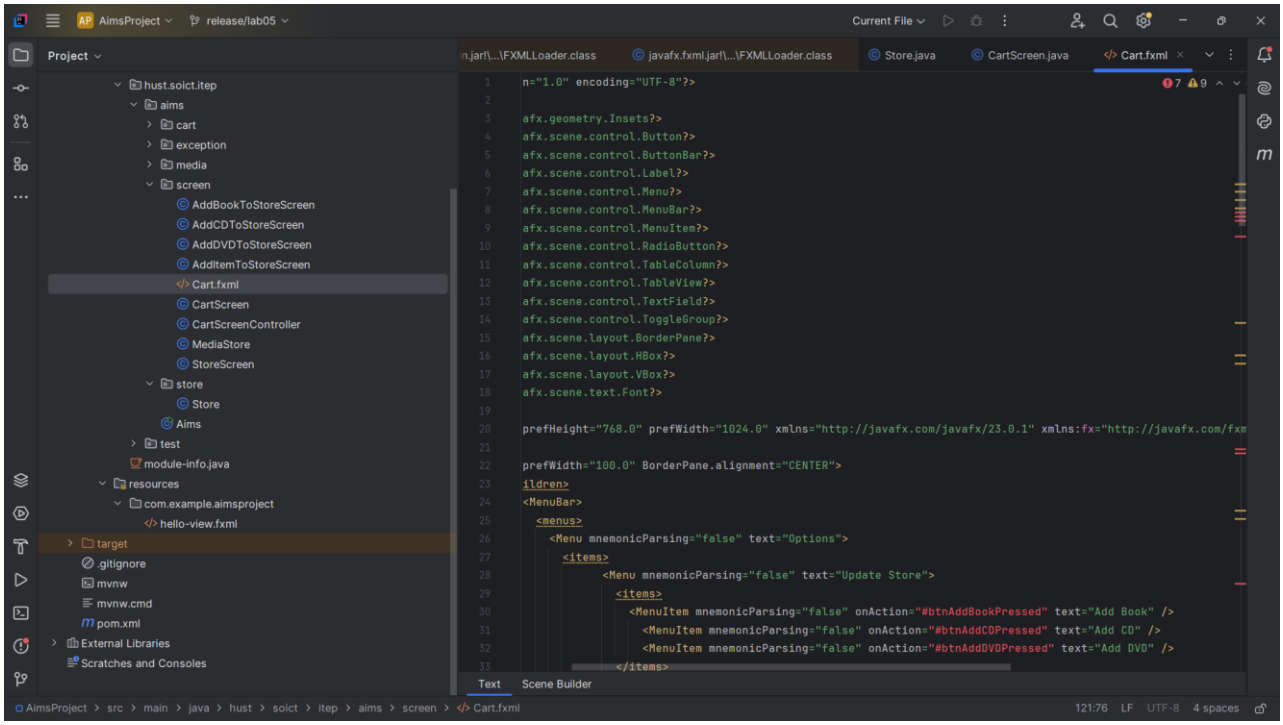

Figure 12: Mã nguồn Painter.fxml

4.2. Painter.java

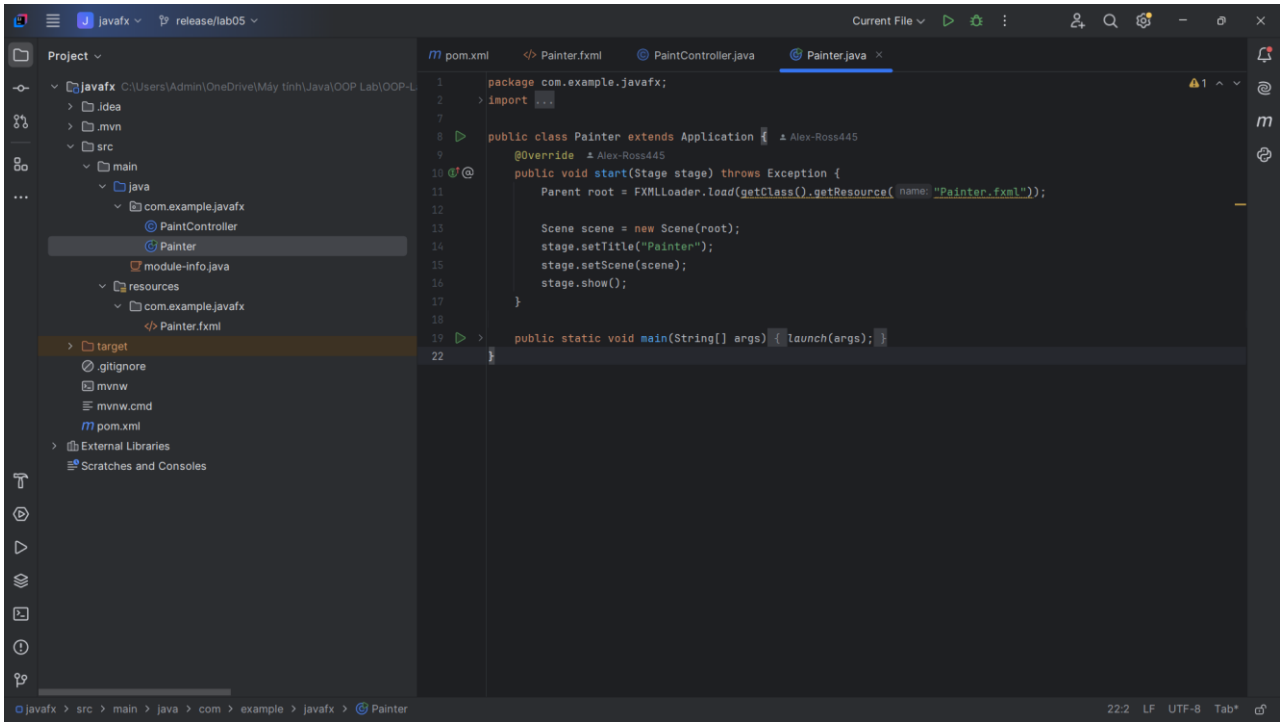

Figure 13: Mã nguồn Painter.java

4.3. PaintController.java

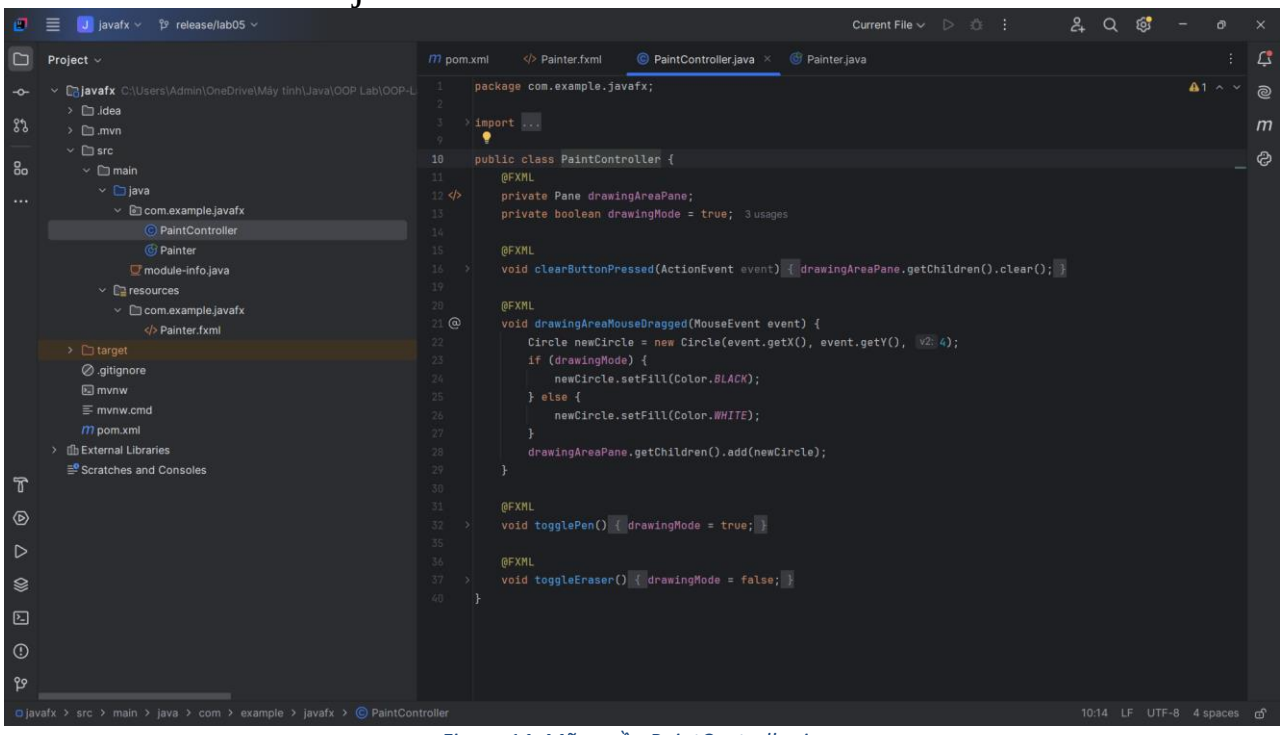

Figure 14: Mã nguồn PaintController.java

4.4. Giao diện

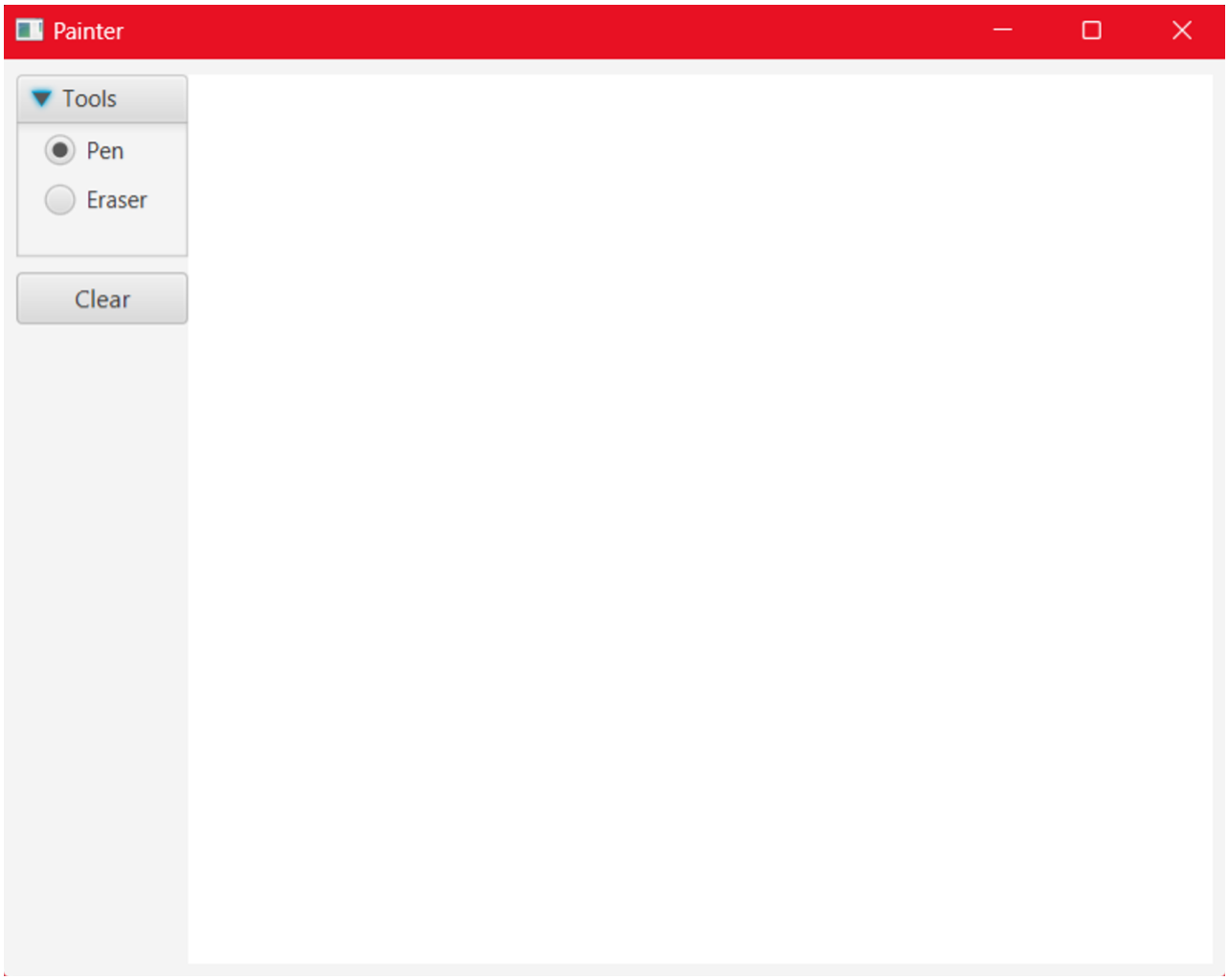

Figure 15: Giao diện Painter

5. Setting up the View Cart Screen with ScreenBuilder

5.1. Cart.fxml

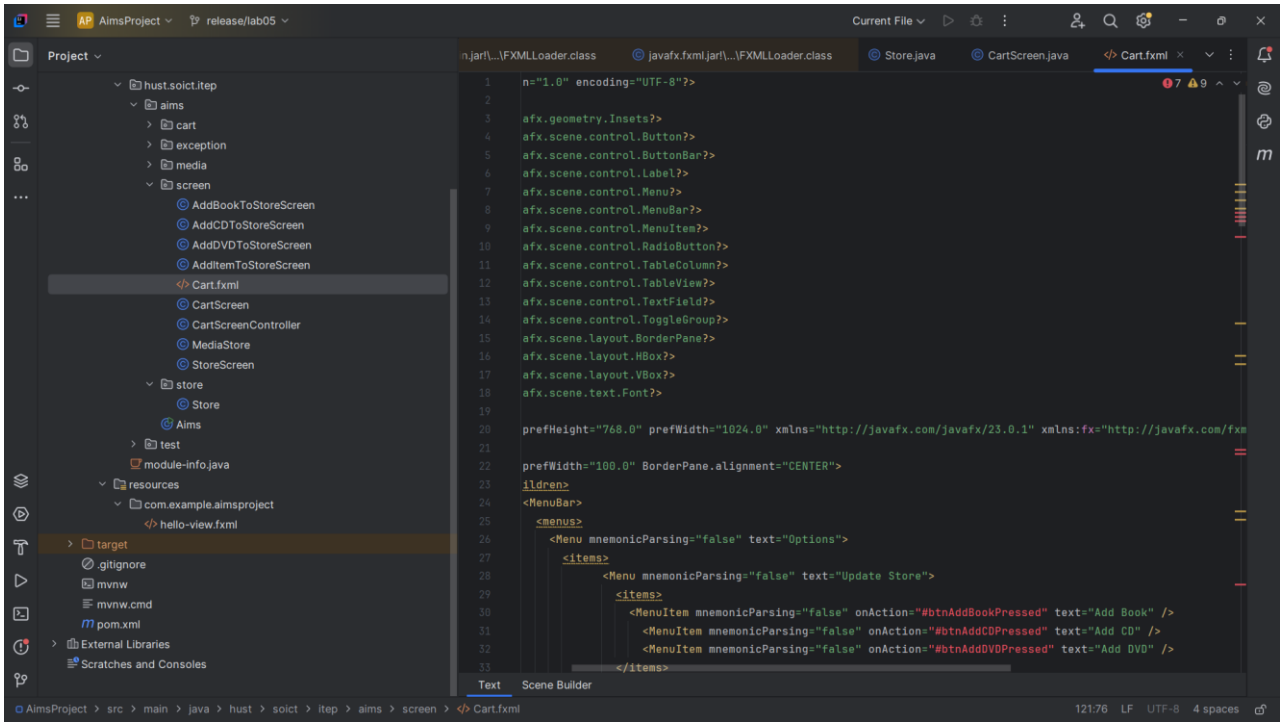

Figure 16: Mã nguồn Cart.fxml

5.2. Giao diện Cart Screen

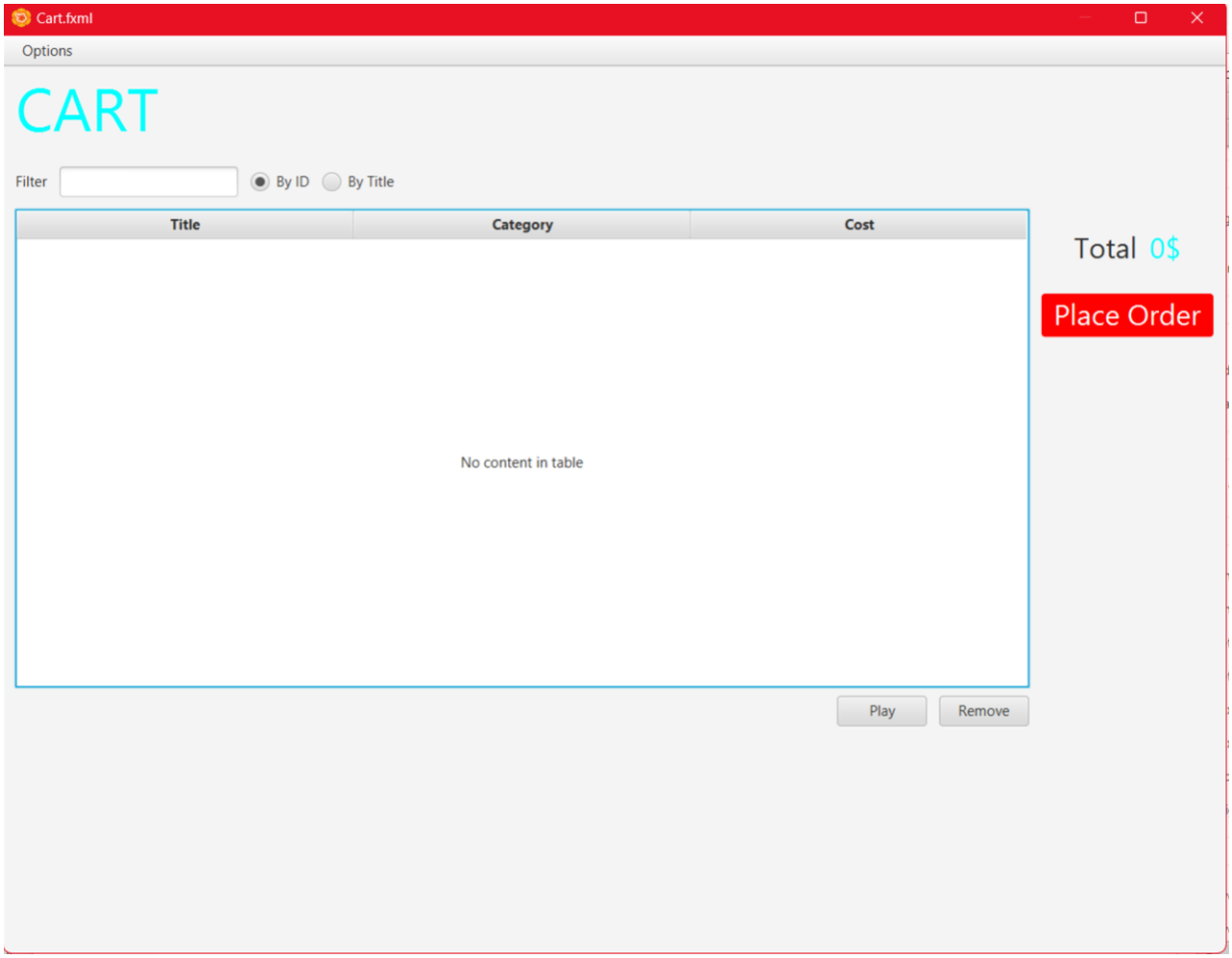

Figure 17: Giao diện Cart Screen

6. Integrating JavaFX into Swing application – The JFXPanel class

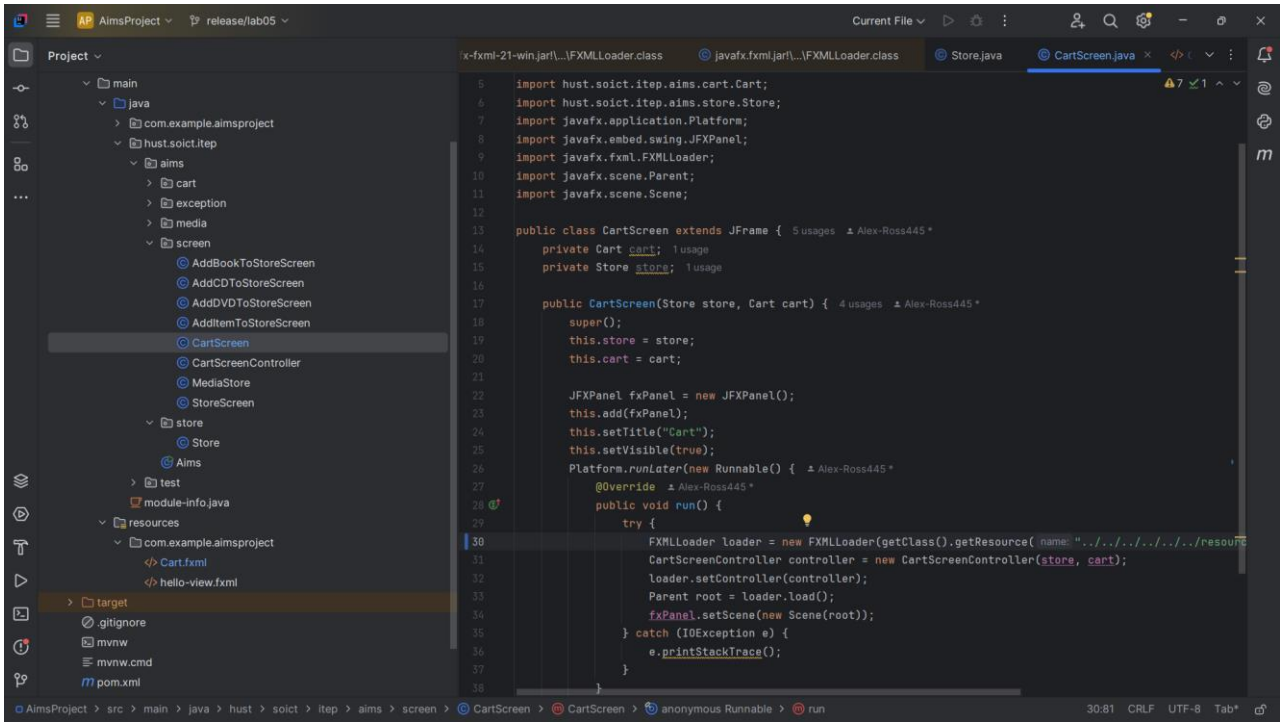

Figure 18: Mã nguồn Cart Screen (nhúng JavaFX vào Swing)

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

7.1. CartScreenController.java

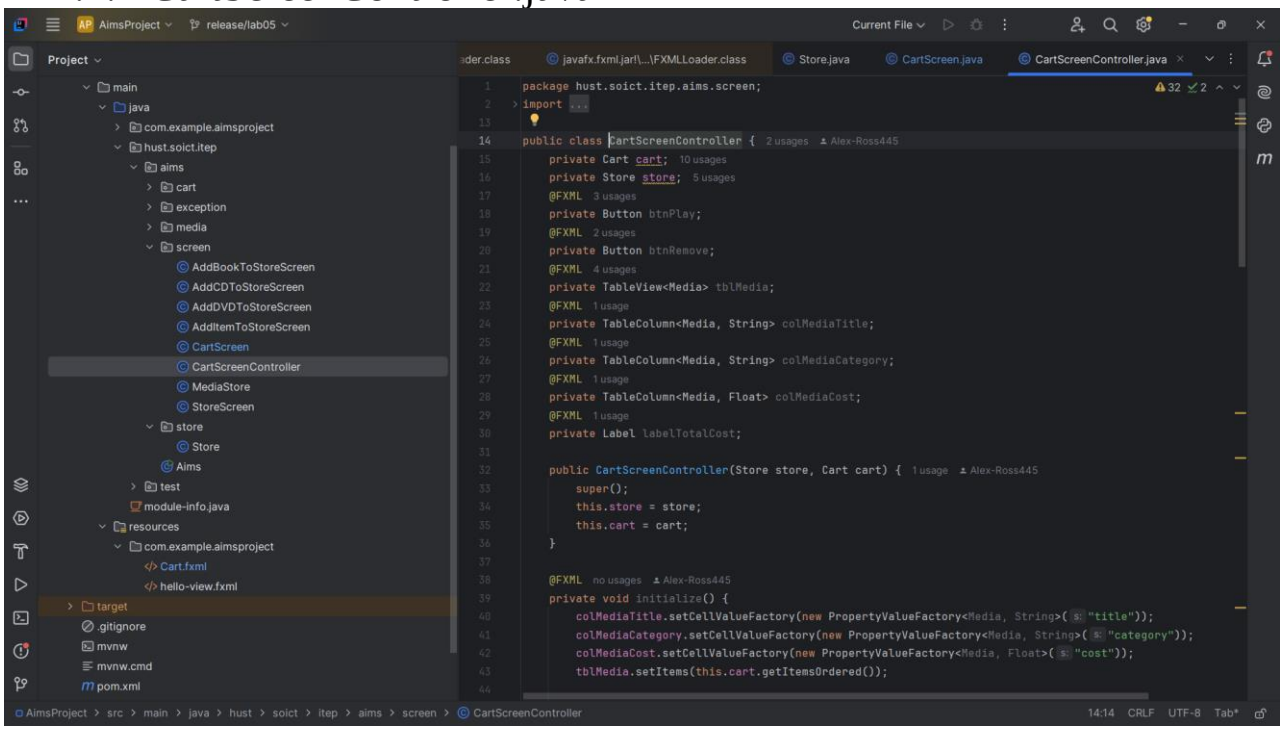

Figure 19: Mã nguồn CartScreenController.java

7.2. Sửa lại Cart.java

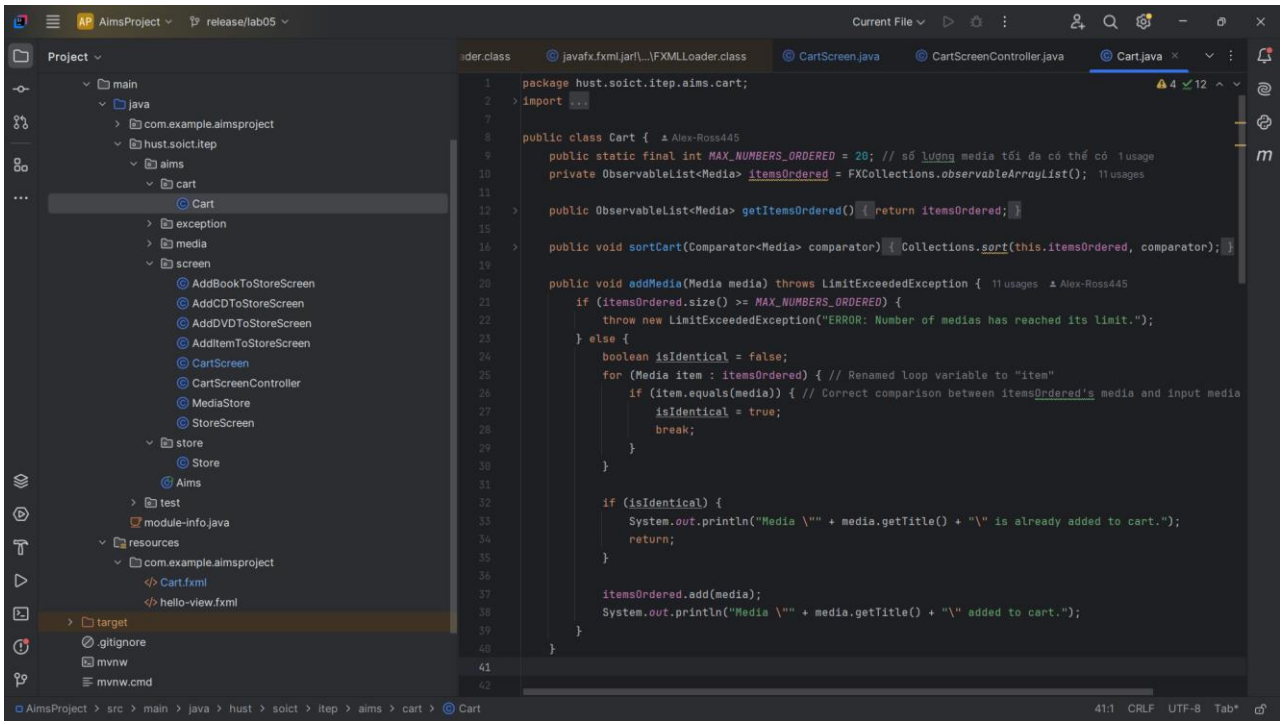

Figure 20: Chỉnh sửa Cart.java để phù hợp với JavaFX

7.3. Giao diện CartScreen

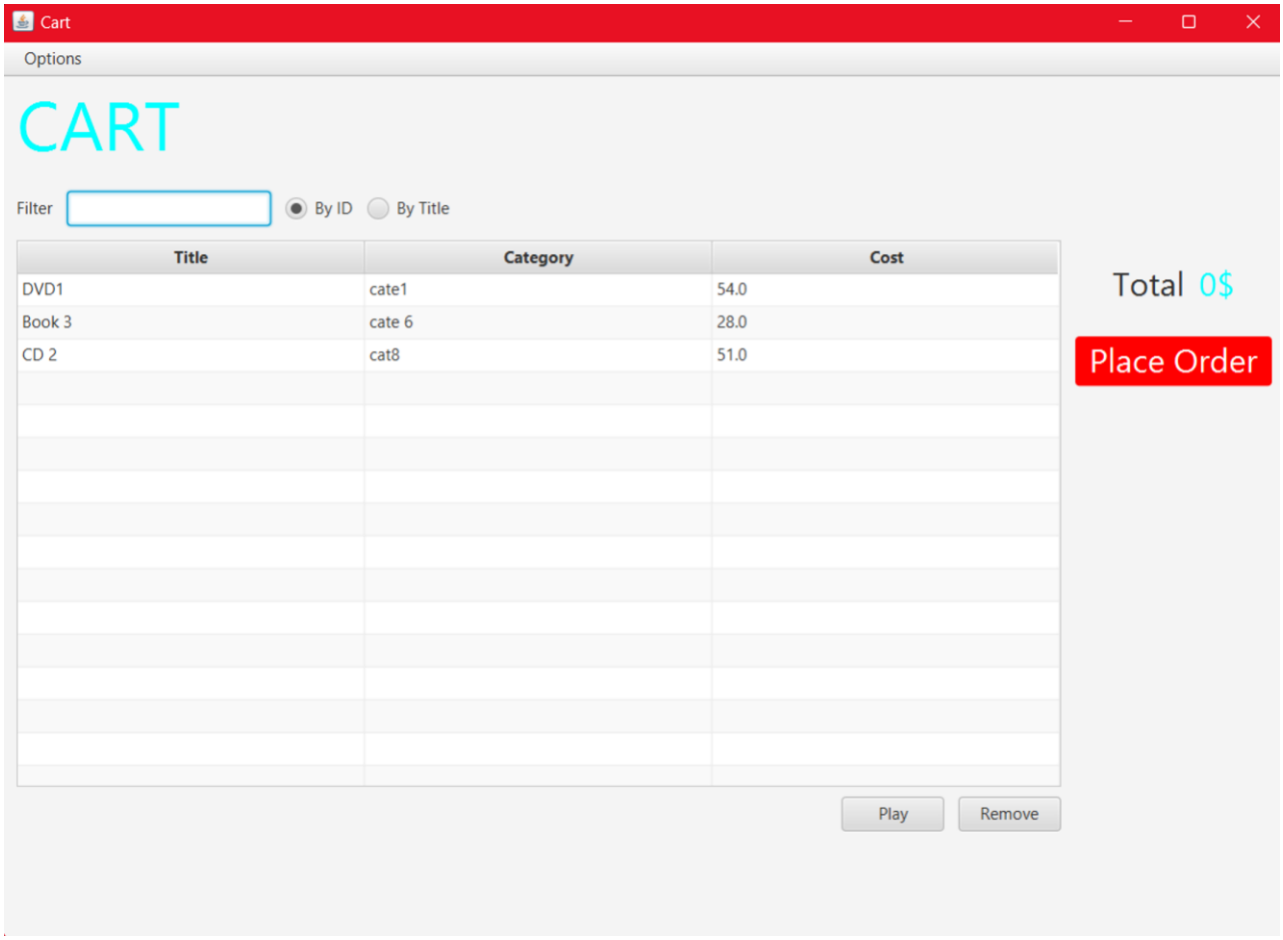

Figure 21: Giao diện Cart (hiển thị sản phẩm trong cart)

8. Updating buttons based on selected item in TableView – ChangeListener

8.1. Sửa lại CartScreenController.java

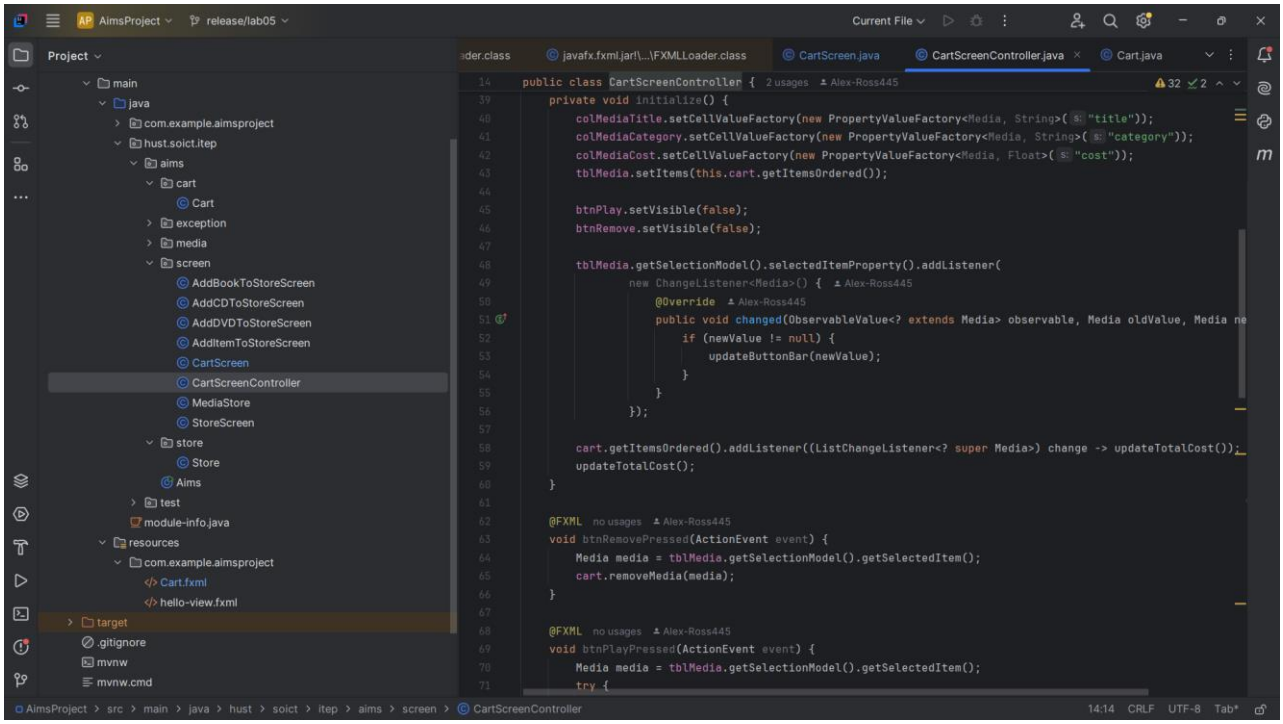

Figure 22: Chỉnh sửa CartScreenController.java

8.2. Giao diện CartScreen

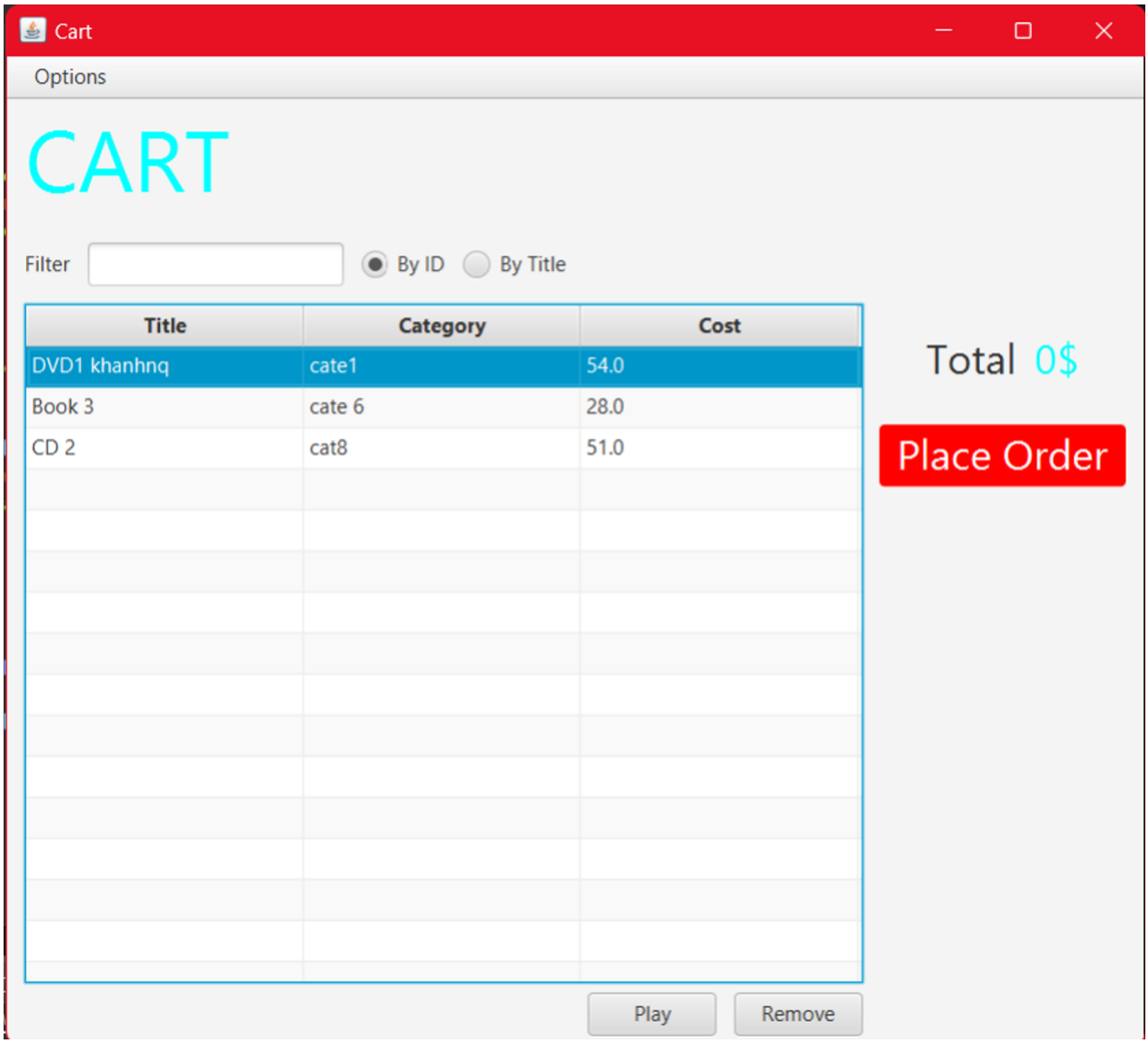

Figure 23: Giao diện Cart (cập nhật nút Play media)

9. Deleting a media

9.1. Thêm phương thức khi ấn nút Remove

```
<buttons>
  <Button fx:id="btnPlay" mnemonicParsing="false" onAction="#btnPlayPressed" text="Play" />
    <Button fx:id="btnRemove" mnemonicParsing="false" onAction="#btnRemovePressed" text="Remove" />
</buttons>
```

Figure 24: Chỉnh sửa Cart.fxml cho chức năng xóa sản phẩm khỏi cart

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

Figure 25: Thêm phương thức bắt sự kiện nút Remove được click

9.2. Giao diện

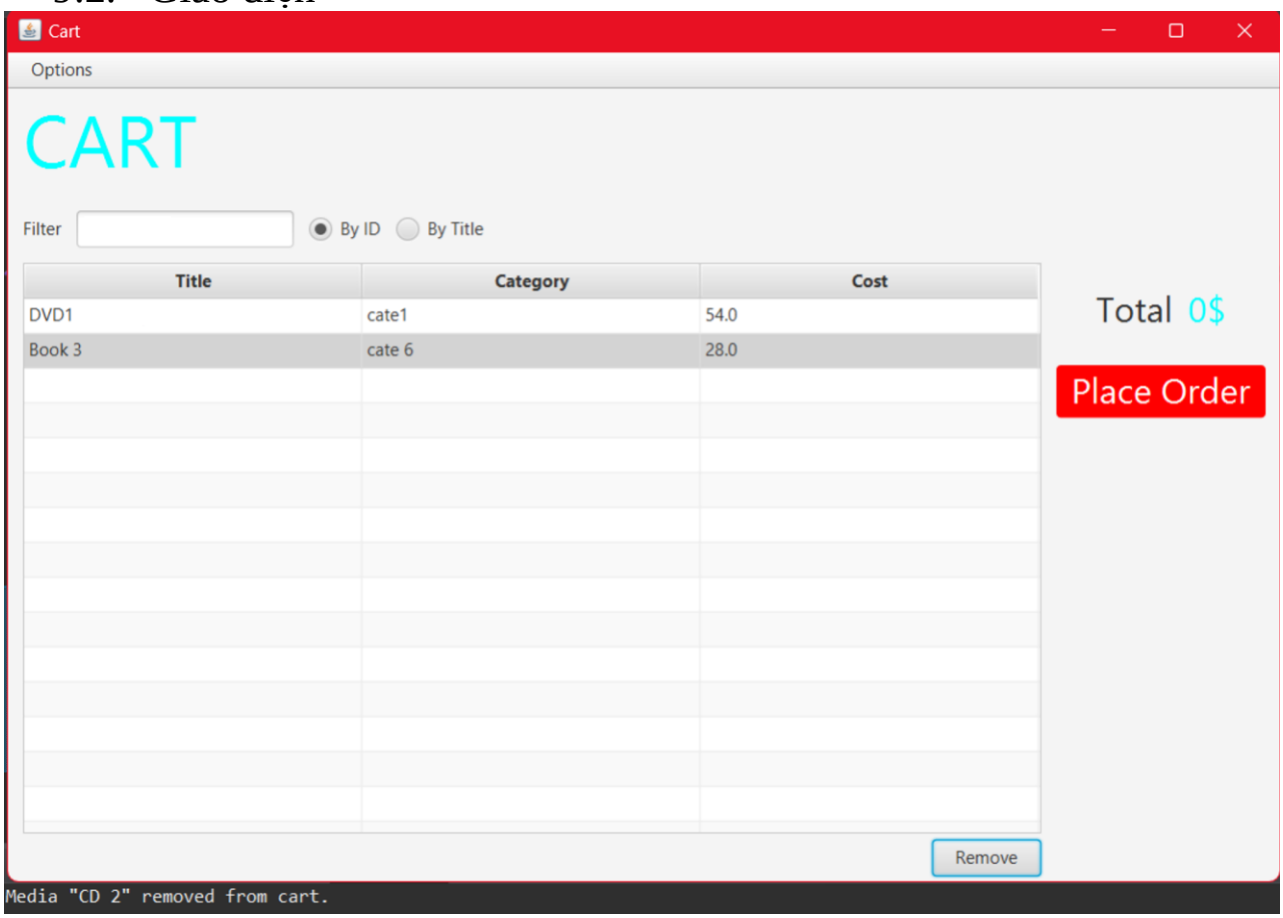

Figure 26: Giao diện Cart sau khi xóa CD2 khỏi cart

10. Filter items in cart – FilteredList (optional)

11. Complete the Aims GUI application

11.1. Cart Screen

11.1.1. "Place order" button

```
@FXML
void btnPlaceOrderPressed(ActionEvent event) {
    Alert alert = new Alert(Alert.AlertType.INFORMATION);
    alert.setTitle("Thông báo ()");
    alert.setHeaderText(null);
    alert.setContentText("Đơn hàng của bạn đã được tạo thành công.");
    alert.showAndWait();

    cart.getItemsOrdered().clear();
}
```

Figure 27: Thêm phương thức bắt sự kiện nút Place order được click

Figure 28: Thông báo đơn hàng đã được tạo thành công

11.1.2. "Play" button

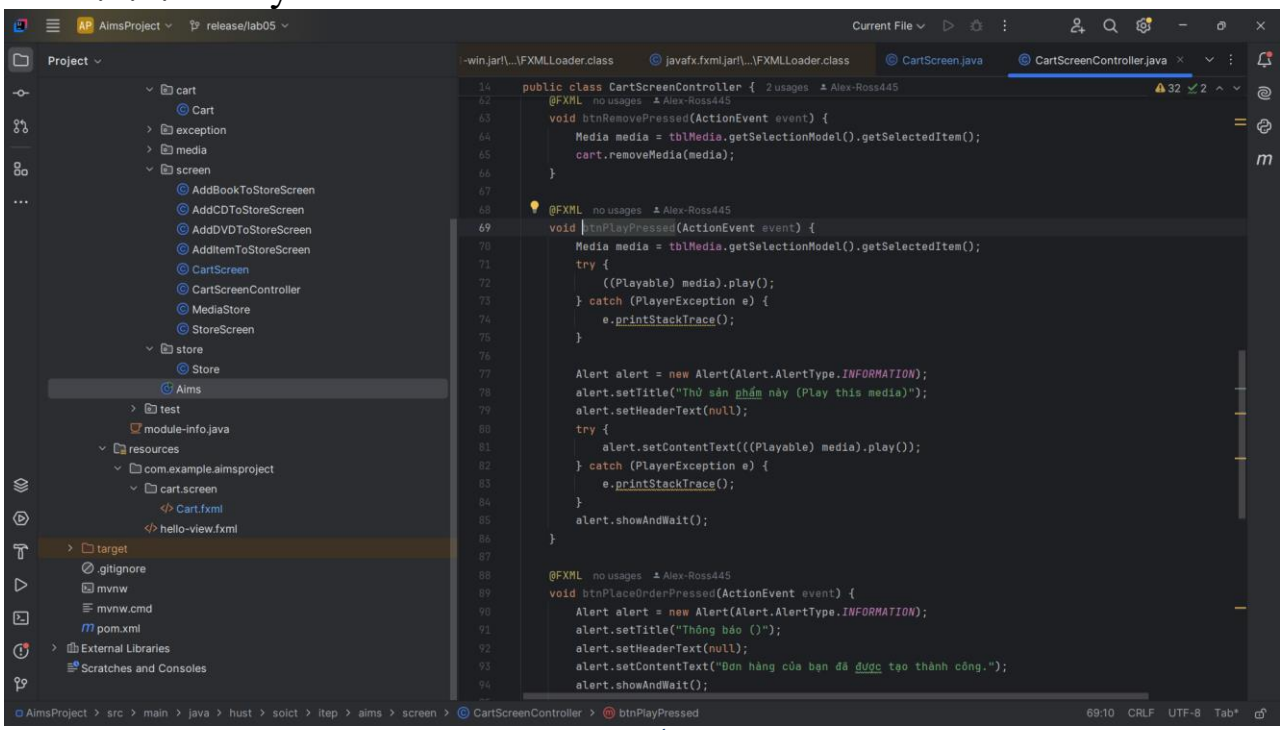

Figure 29: Thêm phương thức bắt sự kiện nút Play được click

Figure 30: Hộp thoại Play media trong Cart

11.1.3. "Total cost" label

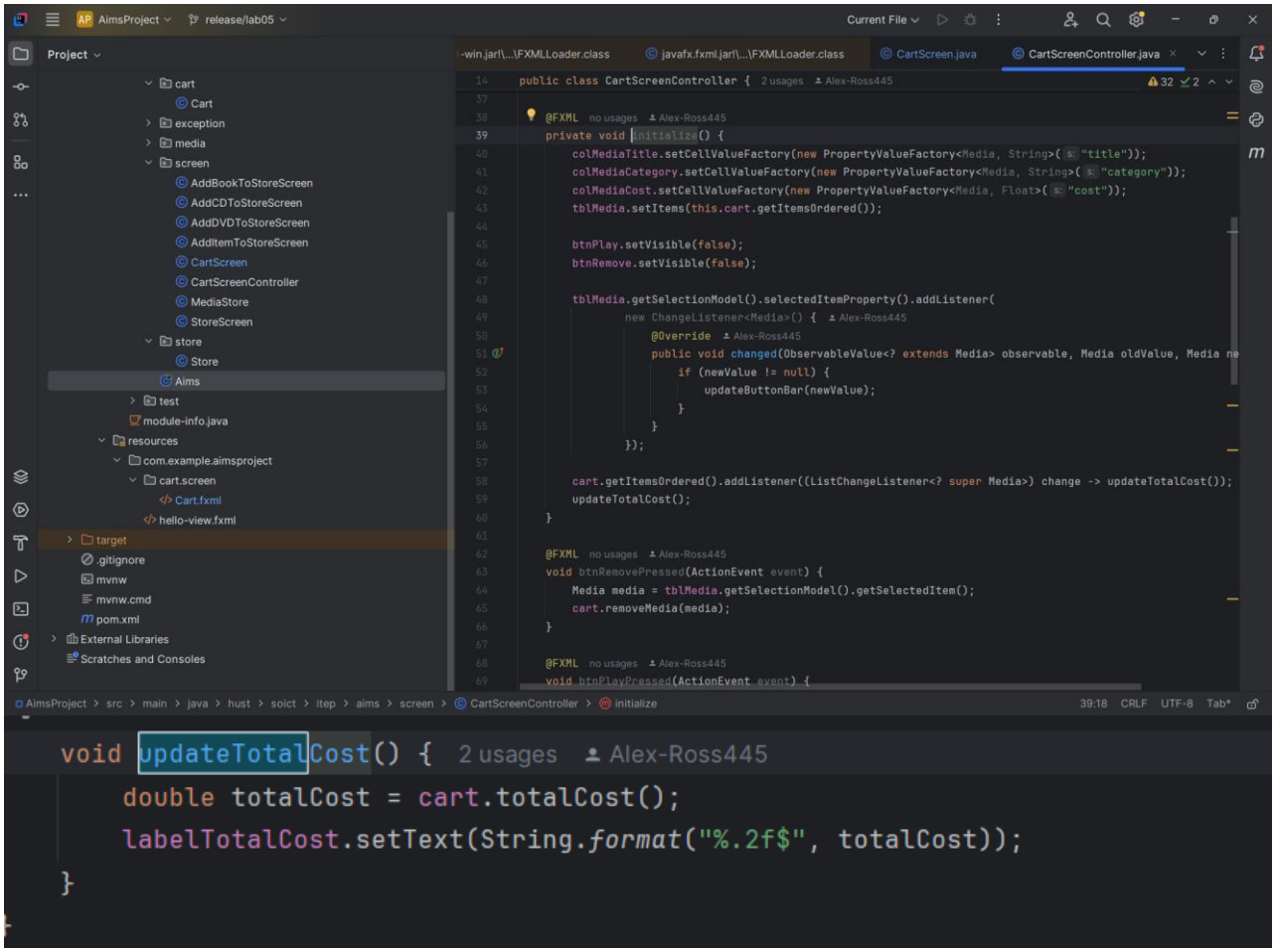

Figure 31: Mã nguồn cập nhật tổng giá tiền của đơn hàng

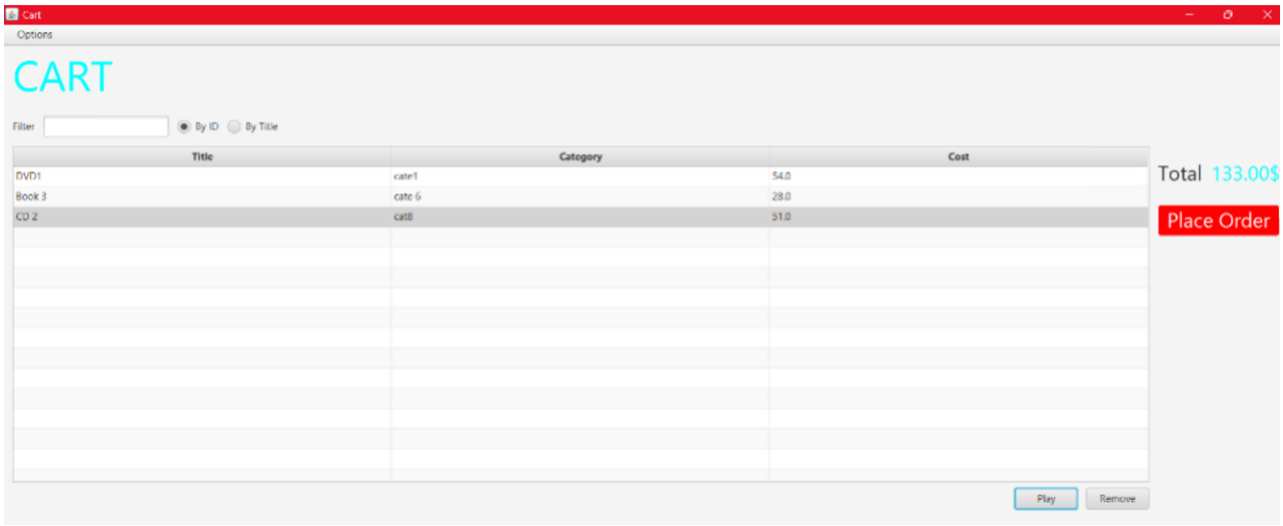

Figure 32: Giao diện Cart (hiển thị tổng giá tiền của đơn hàng)

11.1.4. Menu bar

```
@FXML  no usages  Alex-Ross445
void menuItemViewStorePressed(ActionEvent event) { new StoreScreen(store, cart); }

@FXML  no usages  Alex-Ross445
void btnAddBookPressed(ActionEvent event) { new AddBookToStoreScreen(store, cart); }

@FXML  no usages  Alex-Ross445
void btnAddCDPressed(ActionEvent event) { new AddCDToStoreScreen(store, cart); }

@FXML  no usages  Alex-Ross445
void btnAddDVDPressed(ActionEvent event) { new AddDVDToStoreScreen(store, cart); }
```

Figure 33: Mã nguồn MenuBar

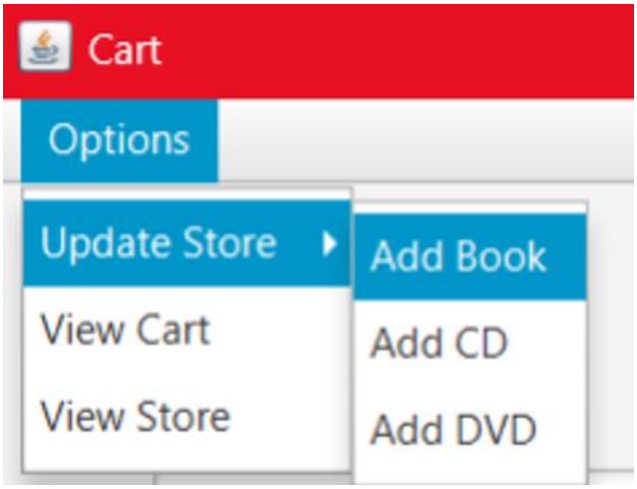

Figure 34: Giao diện MenuBar

11.2. Store Screen

```
addToCartButton.addActionListener(new ActionListener() {  Alex-Ross445
    @Override  Alex-Ross445
    public void actionPerformed(ActionEvent e) {
        try {
            cart.addMedia(media);
        } catch (LimitExceededException e1) {
            e1.printStackTrace();
        }
        JOptionPane.showMessageDialog( parentComponent: null,  message: media.getTitle() + " has been add
                title: "Cart", JOptionPane.INFORMATION_MESSAGE);
    }
});
```

Figure 35: Mã nguồn thêm sản phẩm vào cart

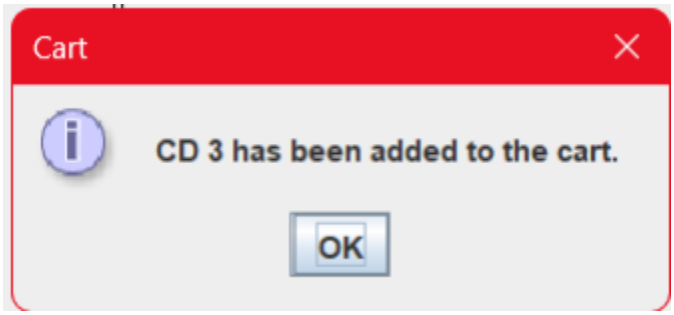

Figure 36: Thông báo sản phẩm được thêm vào cart

11.3. Update Store Screen

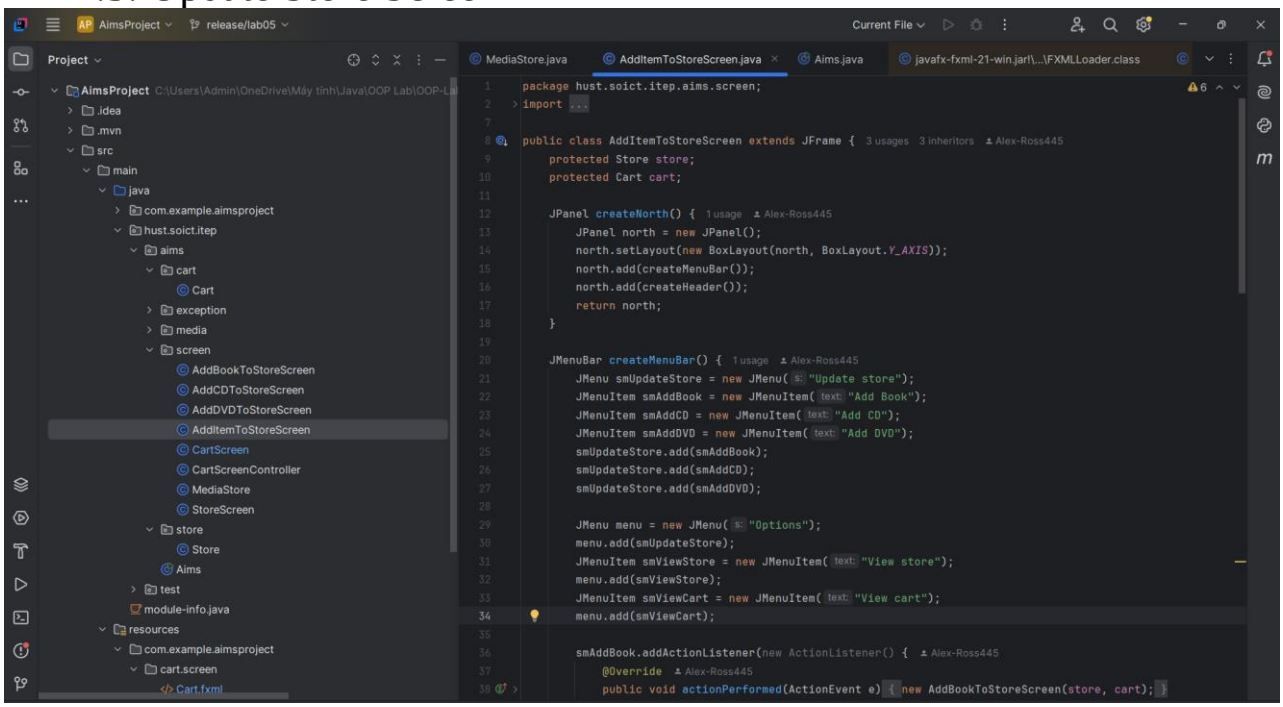

Figure 37: Mã nguồn lớp AddItemToStoreScreen

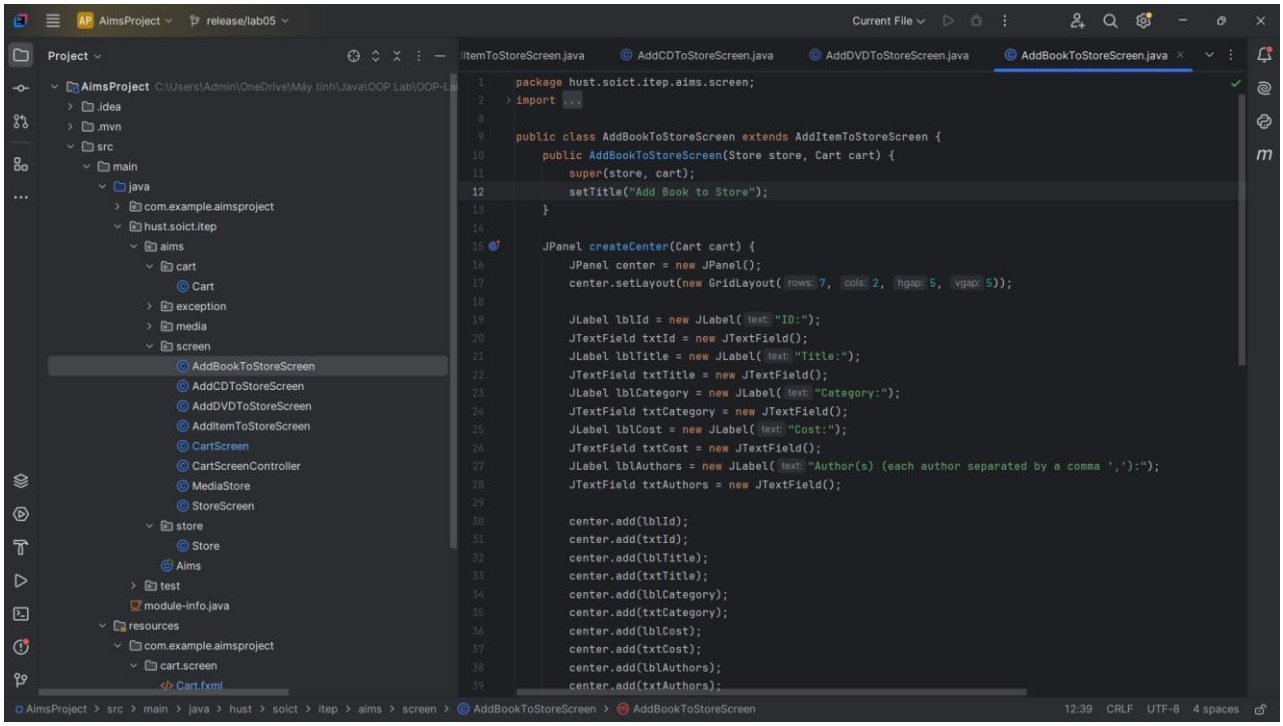

Figure 38: Mã nguồn lớp AddBookToStoreScreen

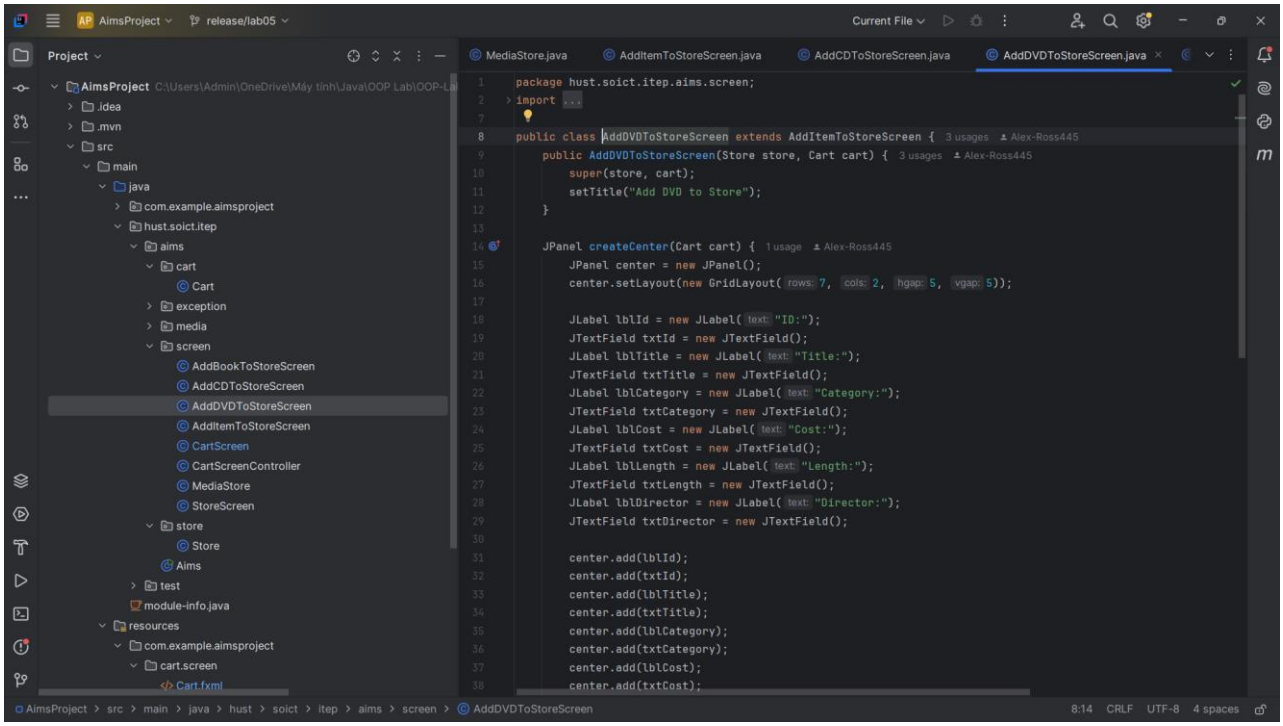

Figure 39: Mã nguồn lớp AddDVDToStoreScreen

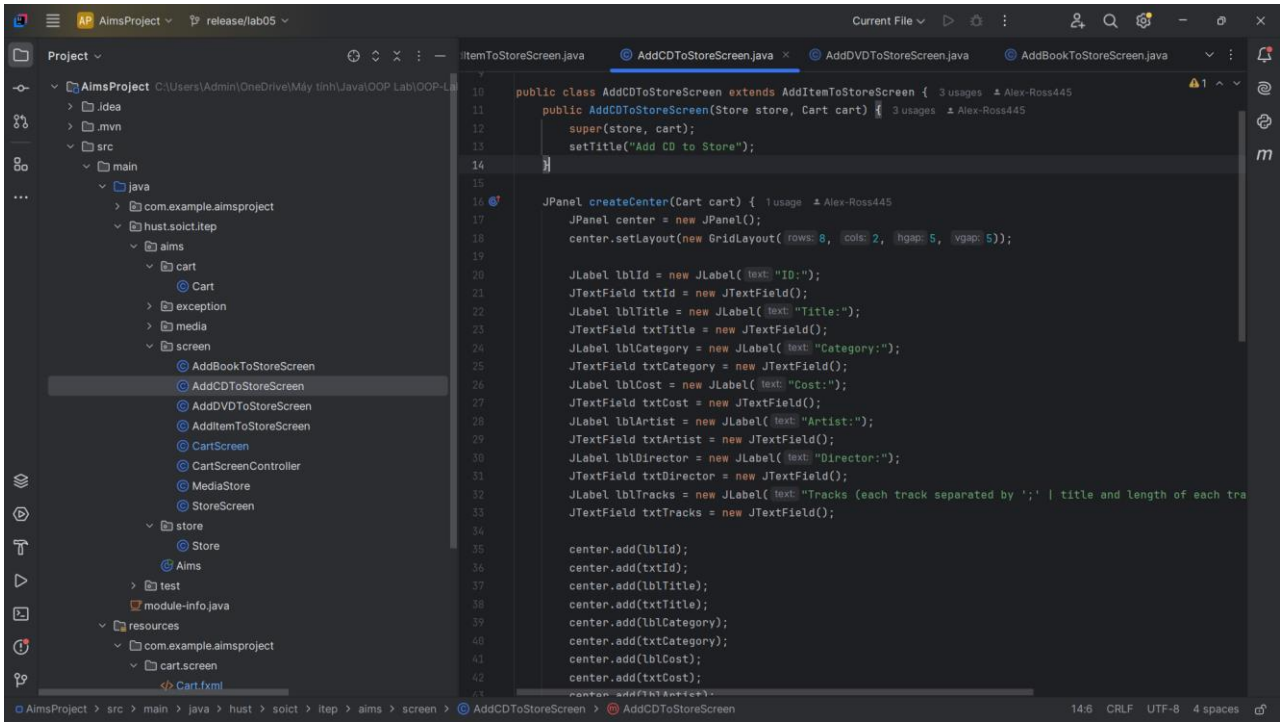

Figure 40: Mã nguồn lớp AddCDToStoreScreen

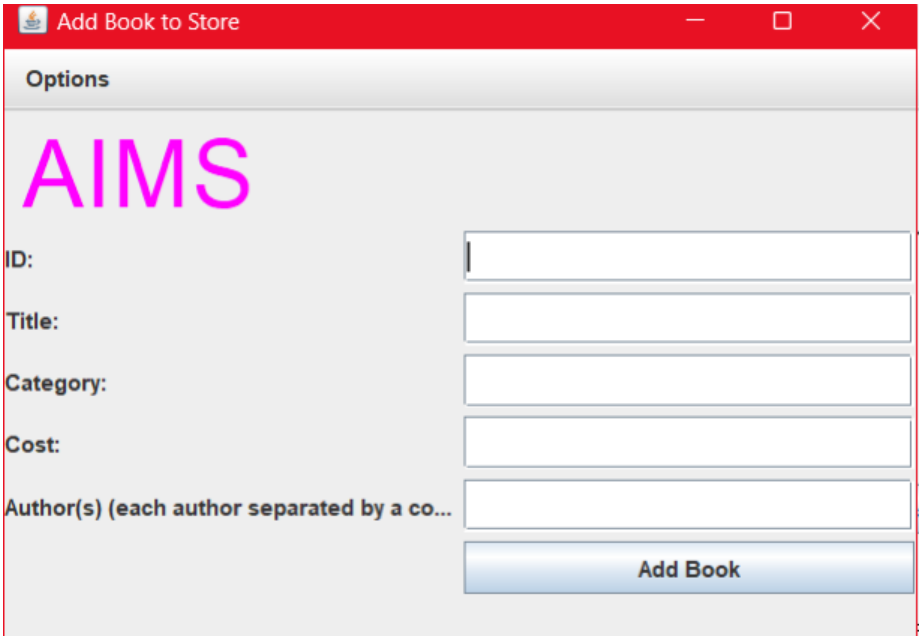

Figure 41: Giao diện AddBookToStore

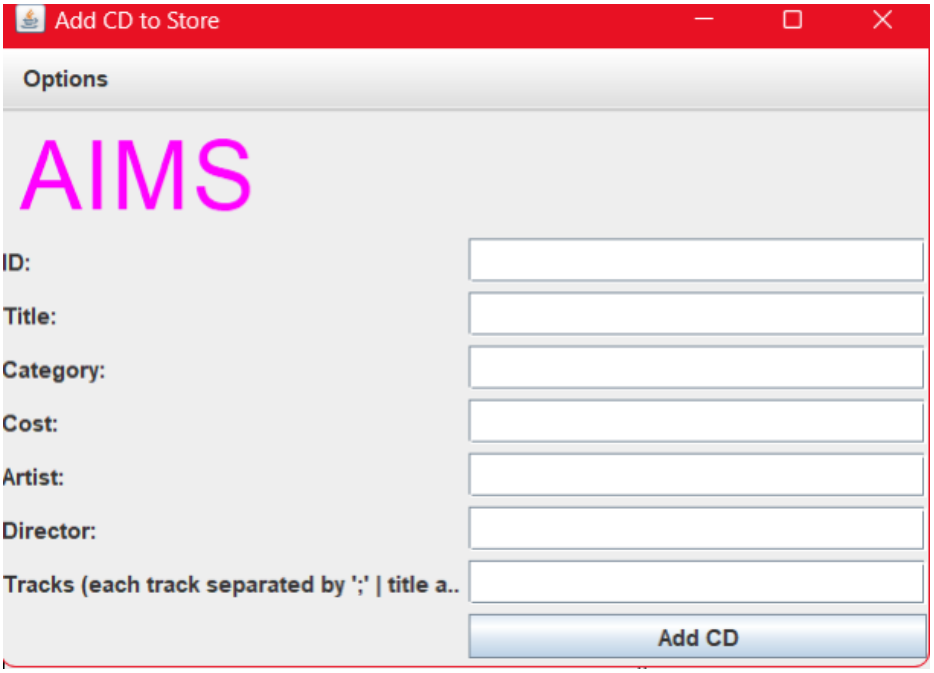

Figure 42: Giao diện AddCDToStore

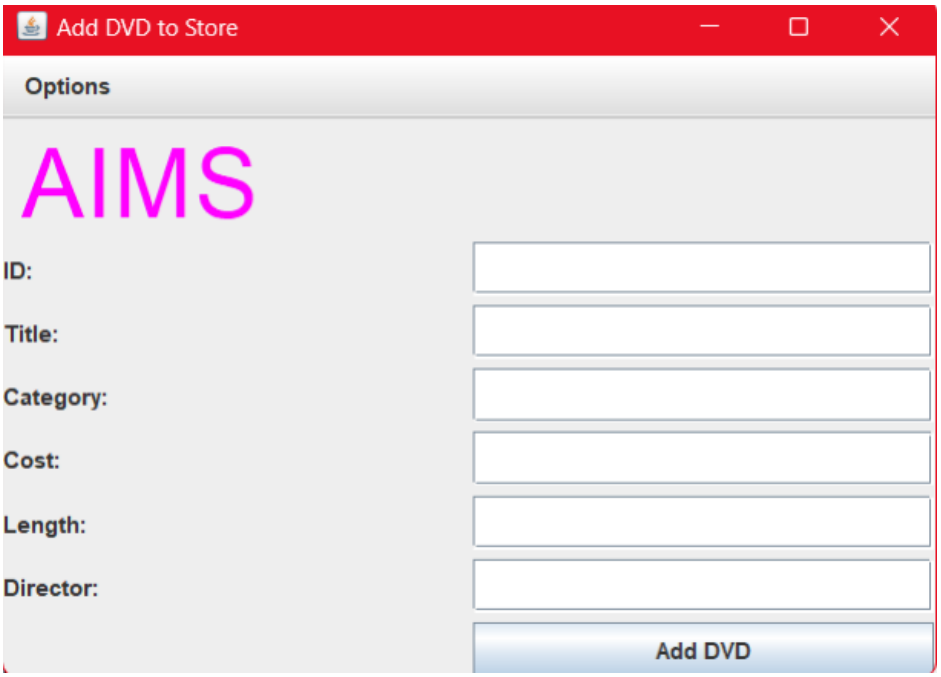

Figure 43: Giao diện AddDVDToStore

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

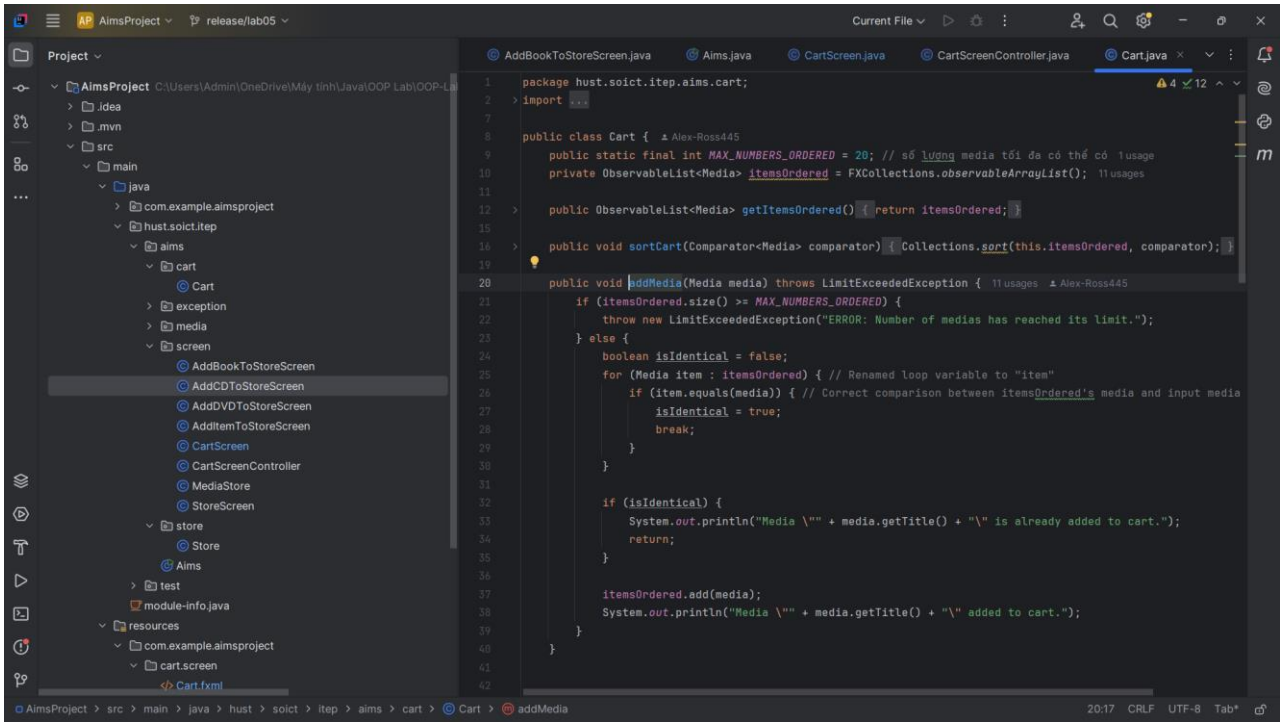

Figure 44: Mã nguồn ngoại lệ vượt quá số lượng sản phẩm tối đa trong cart

```
public void addMedia(Media media) throws IllegalArgumentException {  19 usages  Alex-Ross445
    if (media.getCost() < 0) {
        throw new IllegalArgumentException("ERROR: Cost cannot be negative.");
    } else {
        itemsInStore.add(media);
        System.out.println("Media \"" + media.getTitle() + "\" added to store.");
    }
}
```

Figure 45: Mã nguồn ngoại lệ giá sản phẩm nhỏ hơn 0

13. Create a class which inherits from Exception

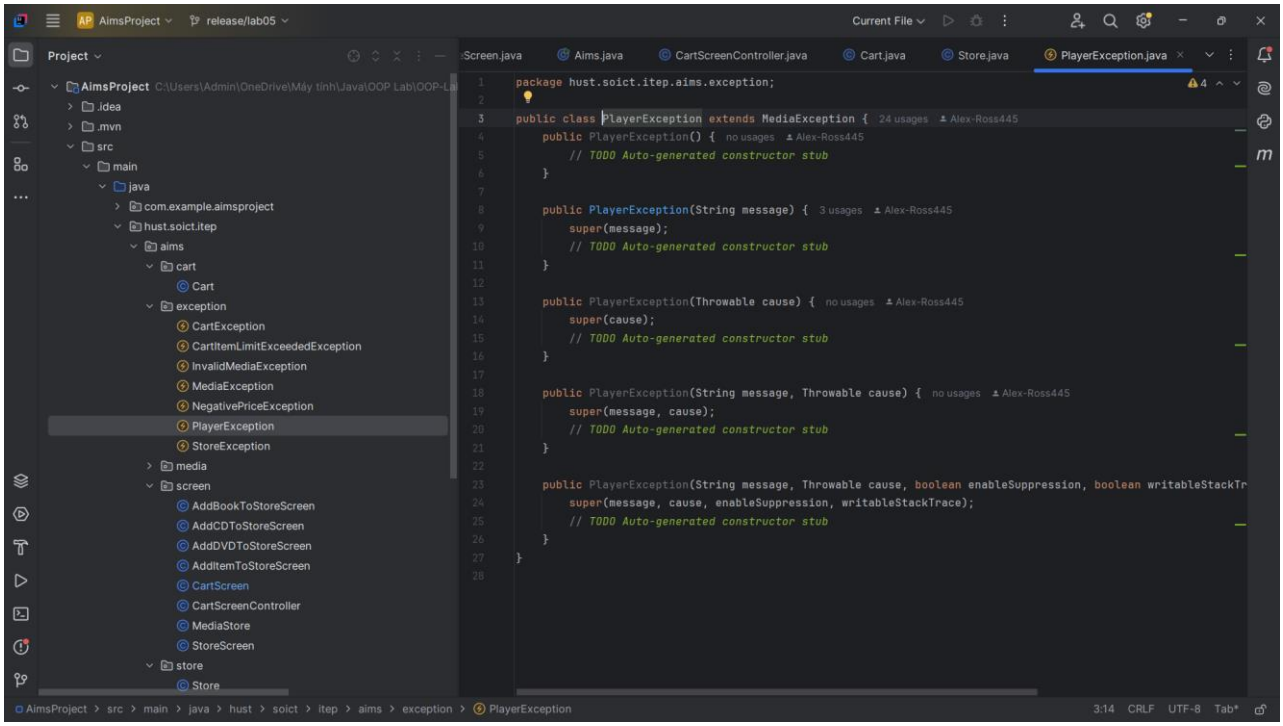

Figure 46: Chỉnh sửa interface Playable

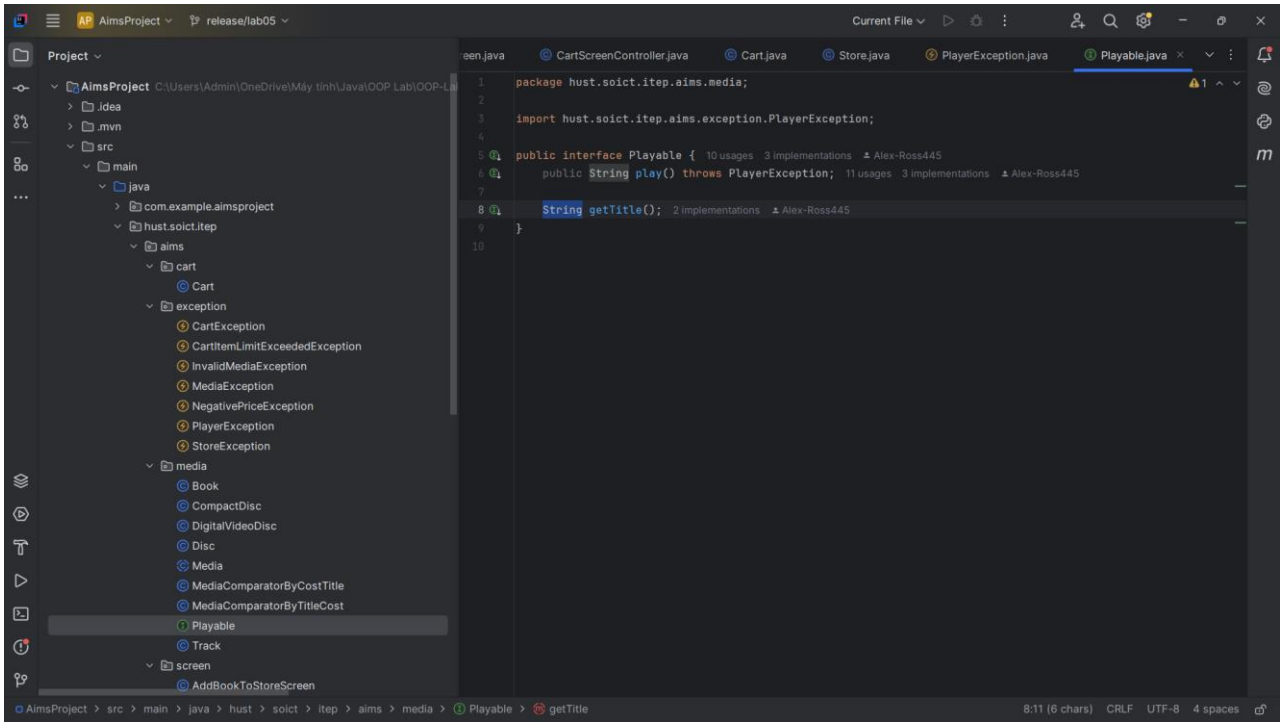

Figure 47: Chỉnh sửa phương thức play() của DigitalVideoDisc

```java
public String play() throws PlayerException {  11 usages  Alex-Ross445
    if (this.getLength() > 0) {
        System.out.println("Playing DVD: " + this.getTitle());
        System.out.println("DVD length: " + this.getLength());

        return "Playing DVD: " + this.getTitle() + "\n" +
        "DVD length: " + this.getLength() + "\n";
    } else {
        throw new PlayerException("ERROR: DVD length is non-positive.");
    }
}
```

Figure 48: Chỉnh sửa phương thức play() của Track

```java
public String play() throws PlayerException {  11 usages  Alex-Ross445
    if (this.getLength() > 0) {
        System.out.println("Playing CD: " + this.getTitle()); // thông tin về CD
        System.out.println("CD artist: " + this.getArtist());
        System.out.println("CD length: " + this.getLength());
        for (Track track : tracks) {
            try {
                track.play(); // play từng track trong CD
            } catch (PlayerException e) {
                throw e;
            }
        }
```

Figure 49: Chỉnh sửa phương thức play() của CompactDisc

14. Update the Aims class

14.1. Answer the question

Question: What happens if the Aims class is not updated to handle exceptions when the play() method is called?

Answer: The exception will propagate through the call stack until it encounters a try-catch block or reaches the main() method. If the main() method does not catch the exception, the program will terminate immediately, and the console will display the stack trace. This makes the program less robust and harder to use.

14.2. Update the Aims.java code

```
try {
    cart.addMedia(dvd1);
} catch (LimitExceededException e) {
    e.printStackTrace();
}
try {
    cart.addMedia(book3);
} catch (LimitExceededException e) {
    e.printStackTrace();
}
try {
    cart.addMedia(cd2);
} catch (LimitExceededException e) {
    e.printStackTrace();
}
```

```
if (media instanceof DigitalVideoDisc) {
    DigitalVideoDisc mediadvd = (DigitalVideoDisc) media;
    try {
        mediadvd.play();
    } catch (PlayerException e) {
        e.printStackTrace();
    }
} else if (media instanceof CompactDisc) {
    CompactDisc mediacd = (CompactDisc) media;
    try {
        mediacd.play();
    } catch (PlayerException e) {
        e.printStackTrace();
    }
} else {
    System.out.println("Media \"" + media.getTitle() + "\" is unplayable.");
}
```

Figure 50: Chỉnh sửa lớp Aims

15. Modify the equals() method of Media class

```java
@Override  Alex-Ross445
public boolean equals(Object o) {
    try {
        if (this == o) return true;

        if (o == null || getClass() != o.getClass()) return false;

        Media media = (Media) o;
        return this.title != null && this.title.equals(media.title);

    } catch (NullPointerException e) {
        System.err.println("NullPointerException: Title của đối tượng hoặc đối tượng là null.");
        return false;

    } catch (ClassCastException e) {
        System.err.println("ClassCastException: Không thể ép kiểu đối tượng so sánh.");
        return false;
    }
}
```

Microsoft Defender configuration

Figure 51: Chỉnh sửa phương thức equals() của lớp Media

16. Update Aims class diagram

The inheritance of exception classes is clearly demonstrated in the source code of the package hust.soict.dsai.aims.exception, as shown in the diagram below.

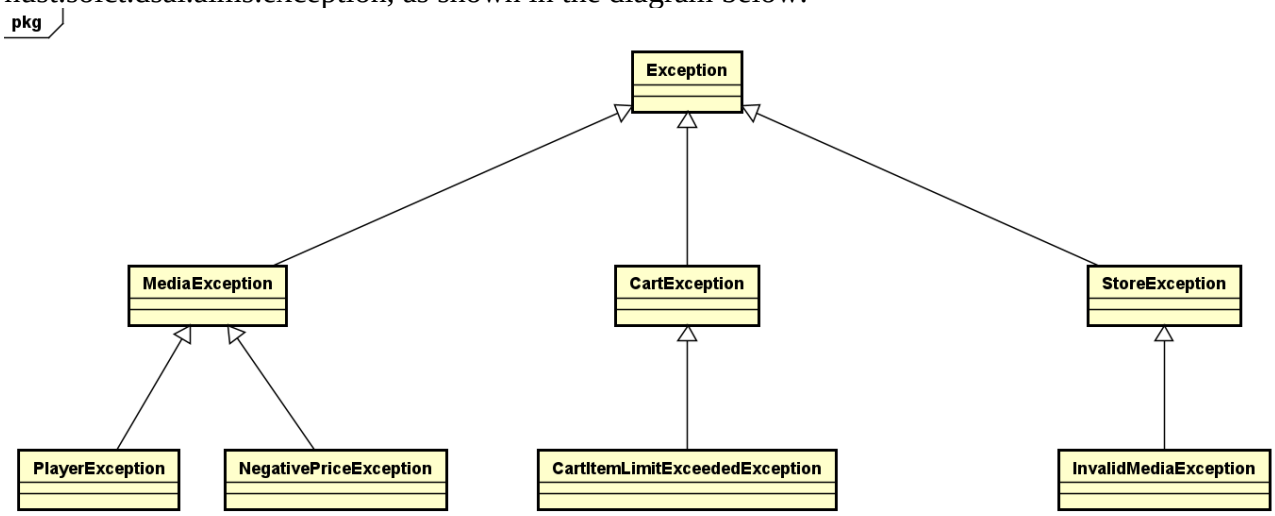

Figure 52: Cây phân cấp ngoại lệ