**ĐẠI HỌC BÁCH KHOA HÀ NỘI**

**TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG**

**BÁO CÁO THỰC HÀNH**

**IT3103-744529-2024.1**

**BÀI THỰC HÀNH 5**

**Họ và tên sv: Nguyễn Trí Nam**

**MSSV 20226093**

Hà Nội 12/2024

# BÁO CÁO THỰC HÀNH LAB 5 LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG

Contents

1. [Swing components 4](#_bookmark0)
   1. AWTAccumulator 4
   2. SwingAccumulator 5
2. Organizing Swing components with Layout Managers 6
   1. Code 6
   2. Demo 8
3. Create a graphical user interface for AIMS with Swing 9
   1. Create class StoreScreen 9
   2. Create class MediaStore 13
   3. Demo 14
4. JavaFX API 16
   1. Create class Painter 16
   2. Create Painter.fxml 16
   3. Create class PainterController 17
5. View Cart Screen 19
   1. Create cart.fxml 19
   2. Create class CartScreen 20
   3. Create class CartScreenController 21
   4. Demo 22
6. Updating buttons based on selected item in TableView – ChangeListener 22
   1. Edit class CartScreenController 22
   2. Demo 23
7. Deleting a media 24
   1. Code 24
   2. Demo 25
8. Complete the Aims GUI application 26
9. Use case Diagram 30
10. Class Diagram 31

Figure 1.1: Source code of AWTAccumulator 4

Figure 1.2: Demo of AWTAccumulator 5

Figure 1.3: Source code of SwingAccumulator 5

Figure 1.4: Demo of SwingAccumulator 6

Figure 2.1: Source code of NumberGrid 1 6

Figure 2.2: Source code of NumberGrid 2 7

Figure 2.3: Demo buttons 0-9 8

Figure 2.4: Demo DEL button 8

Figure 2.5: Demo C button 8

Figure 3.1: Class StoreScreen 1 9

Figure 3.2: Class StoreScreen 2 10

Figure 3.3: Class StoreScreen 3 10

Figure 3.4: Class StoreScreen 4 11

Figure 3.5: Class StoreScreen 5 11

Figure 3.6: Class StoreScreen 6 12

Figure 3.7: Class MediaStore 1 13

Figure 3.8: Class MediaStore 2 13

Figure 3.9: Class MediaStore 3 14

Figure 3.10: StoreScreen 14

Figure 3.11 Demo Add to cart button 15

Figure 3.12 Demo Play button 15

Figure 3.13 Demo View cart button 15

Figure 4.1: Class Painter 16

Figure 4.2: Painter.fxml 1 16

Figure 4.3: Painter.fxml 2 17

Figure 4.4: PainterController 17

Figure 4.5: Use Pen 18

Figure 4.6: Use Eraser 18

Figure 4.7: Clear button 18

Figure 5.1: Cart.fxml 1 19

Figure 5.2: Cart.fxml 2 19

Figure 5.3: Cart.fxml 3 20

Figure 5.4: CartScreen class 20

Figure 5.5: CartScreenController 1 21

Figure 5.6: CartScreenController 2 21

Figure 5.7: Demo CartScreen 22

Figure 6.1: CartScreenController 1 22

Figure 6.2: CartScreenController 2 23

Figure 6.3: Demo media playable 23

Figure 6.4: Demo media unplayable 24

Figure 7.1: btnRemovePressed Method 24

Figure 7.2: button Remove 25

Figure 7.3: button Remove 25

Figure 8.1: Store before add book 26

Figure 8.2: Add book 26

Figure 8.3: Store after add book 27

Figure 8.4: Add CD 27

Figure 8.5: Store after add CD 28

Figure 8.6 Add DVD 28

Figure 8.7: Store after add DVD 29

Figure 8.8: Cart 29

Figure 8.9: Exception 30

# Swing component

* 1. Lớp AWTAccumulator

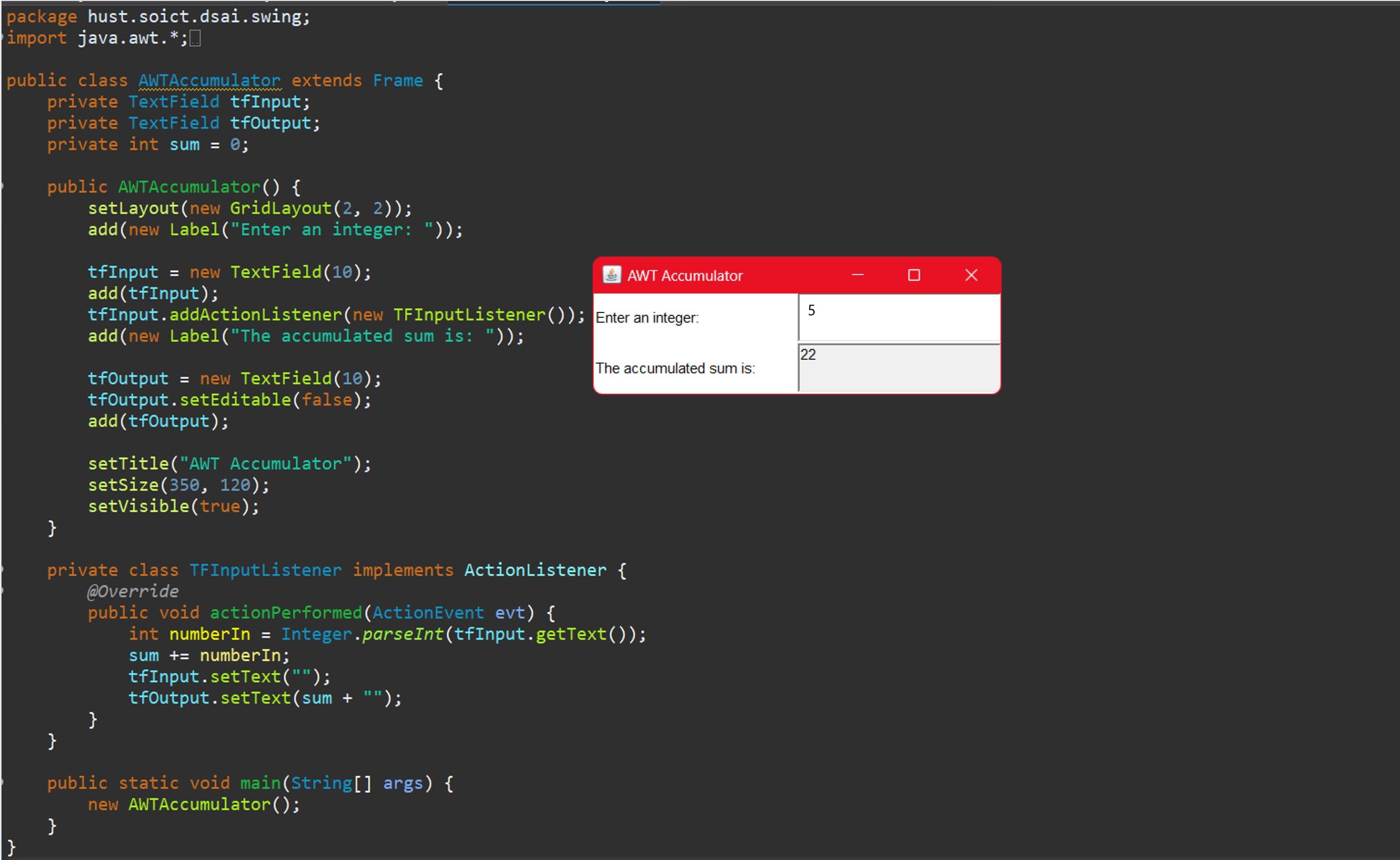


Figure 1: Mã nguồn và giao diện lớp AWTAccumulator

# Lớp SwingAccumulator

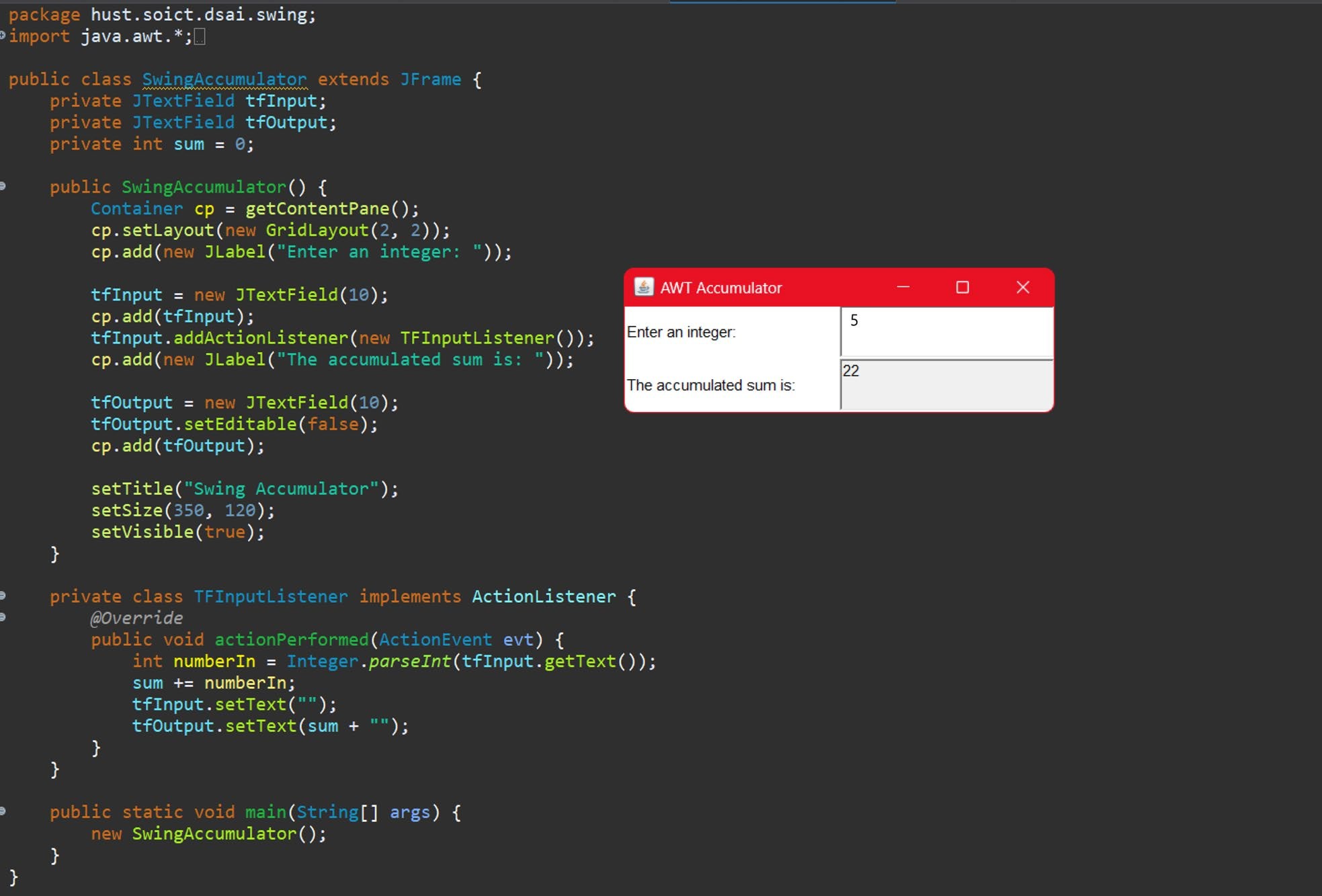


Figure 2: Giao diện và mã nguồn lớp SwingAccumulator

# 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.

# Organizing Swing components with Layout Managers

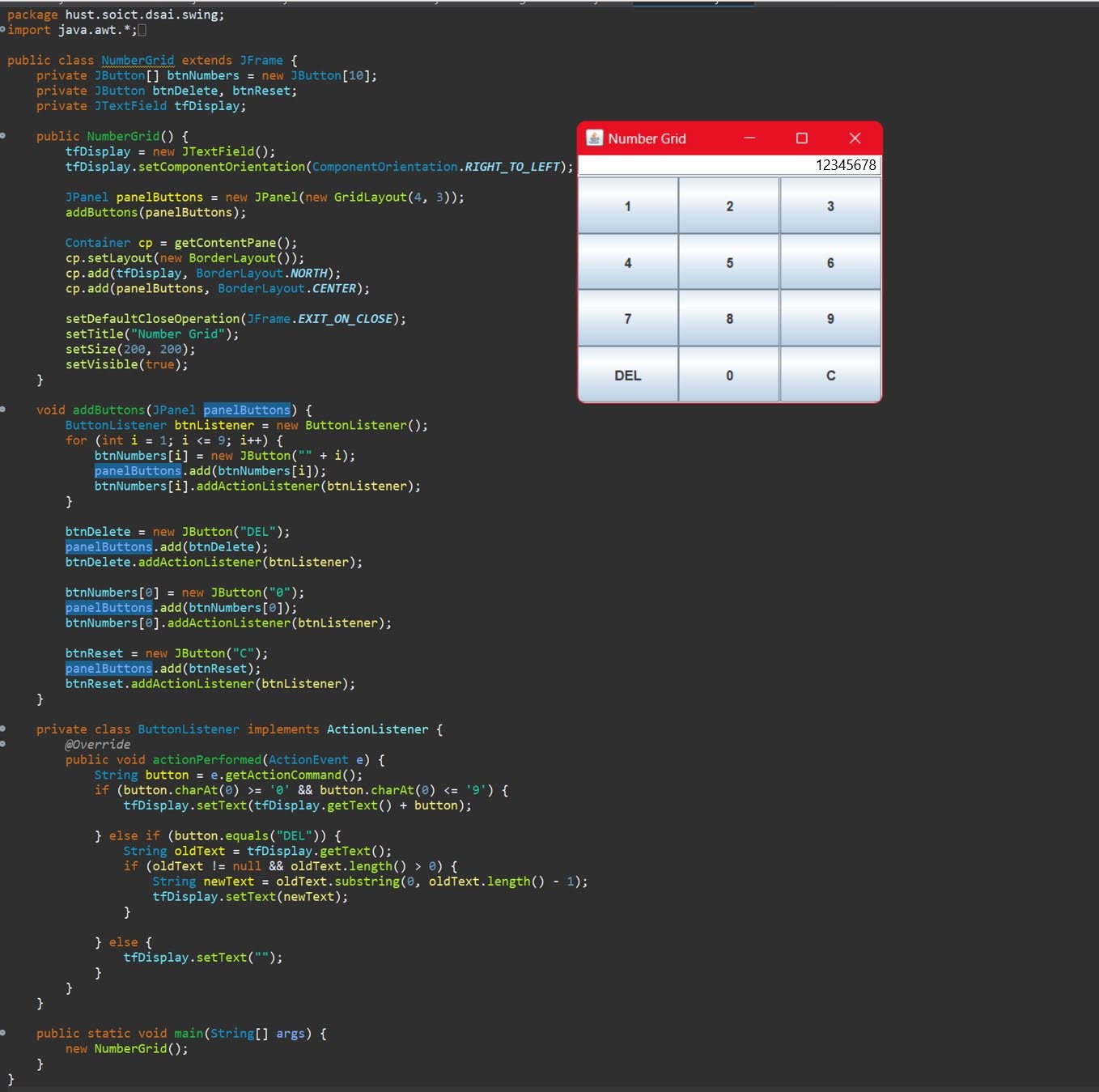
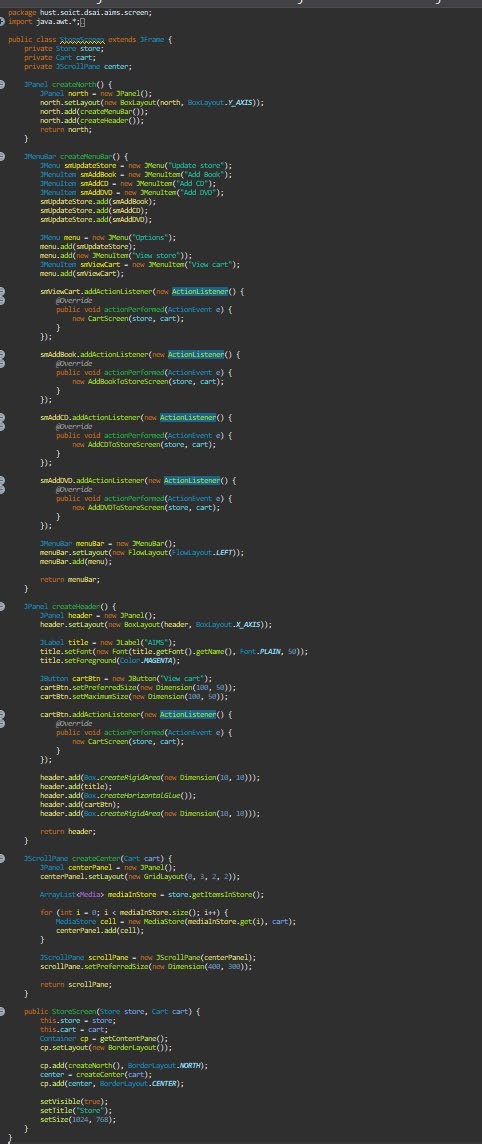


Figure 3: Mã nguồn và giao diện lớp NumberGrid

1. Create a graphical user interface for AIMS with Swing
   1. Lớp StoreScreen



* 1. Lớp MediaStore

Figure 4: Mã nguồn lớp StoreScreen



Figure 5: Mã nguồn lớp MediaStore

* 1. Chức năng của nút Play và nút Add to cart
     1. Nút Play



* + 1. Nút Add to cart

Figure 6: Mã nguồn nút Play của MediaStore



# Giao diện

Figure 7: Mã nguồn nút Add to cart của MediaStore

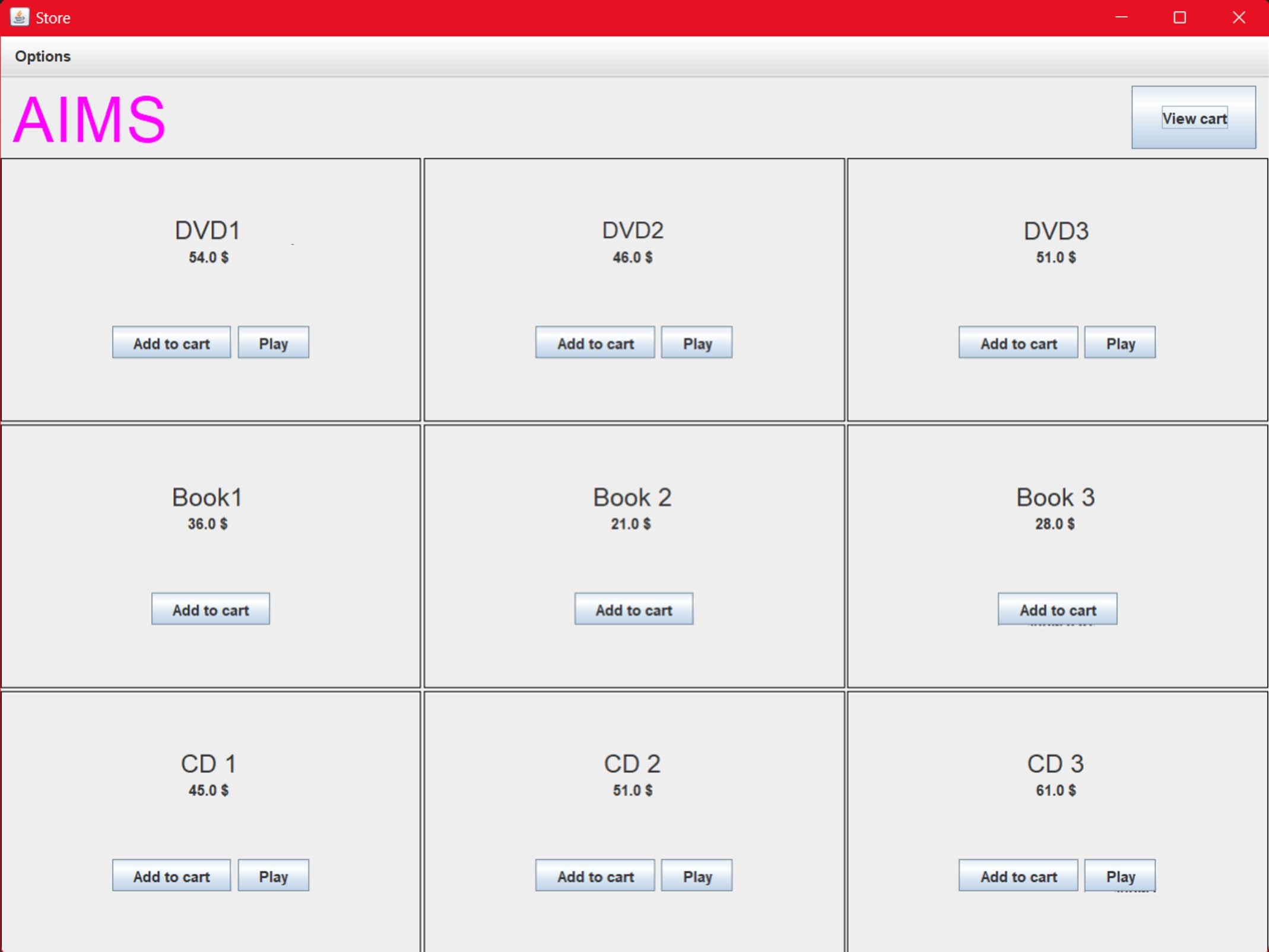


Figure 8: Giao diện Store



Figure 9: Giao diện Play DVD media trong

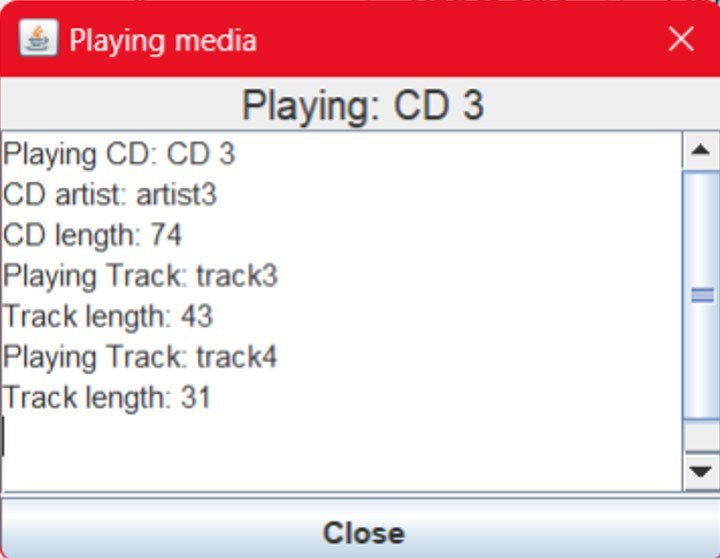
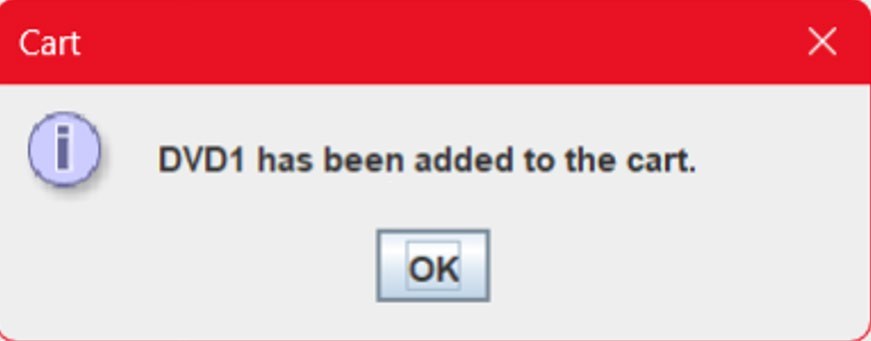
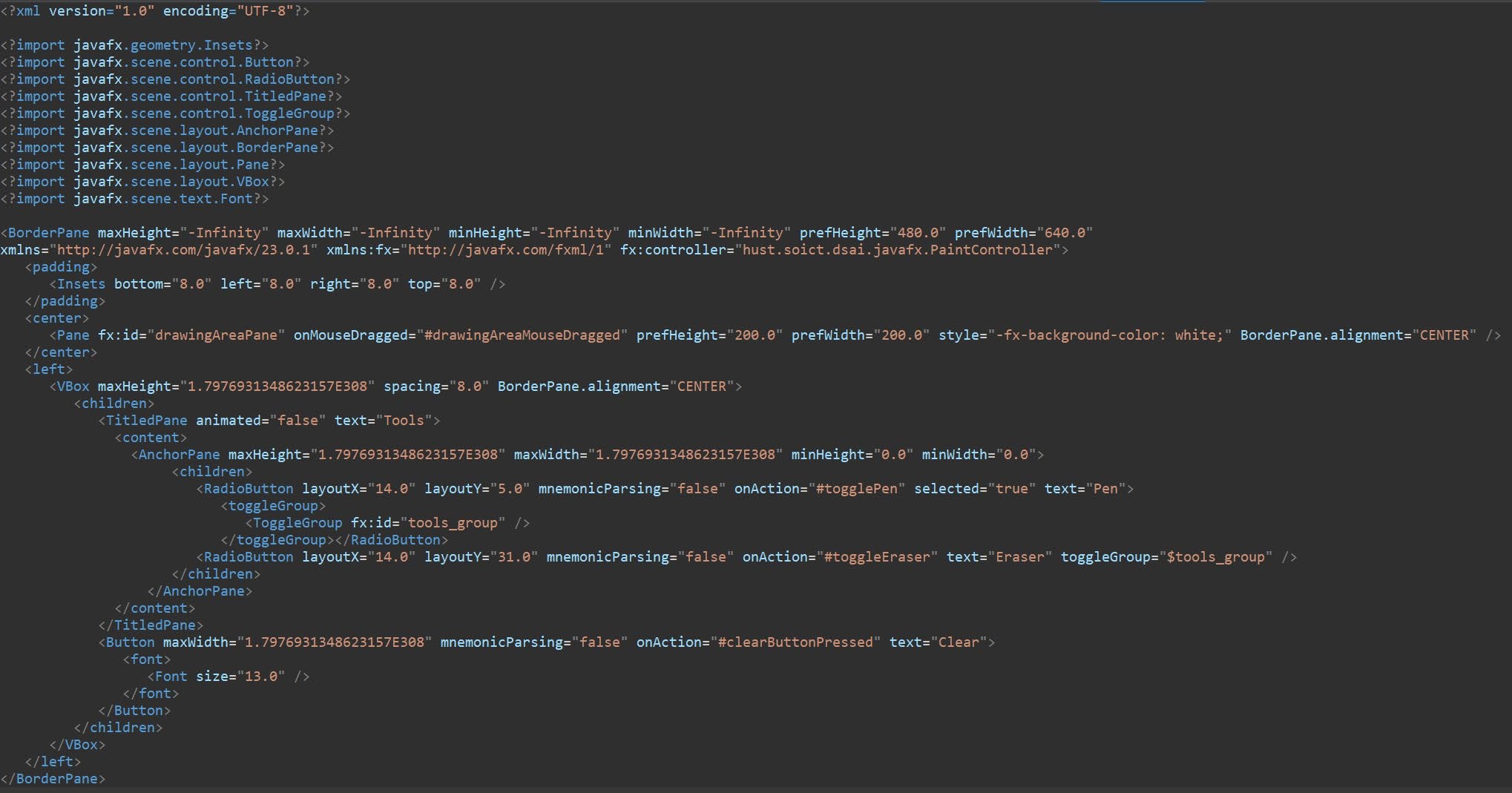


Figure 10: Giao diện Play CD media trong Store



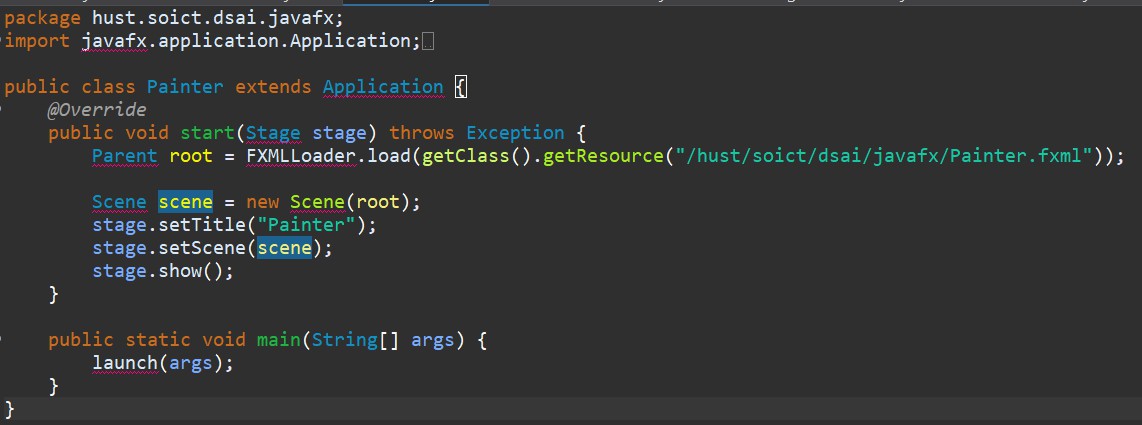
# JavaFX API

Figure 11: Thông báo thêm media vào cart



# Painter.java

Figure 12: Mã nguồn Painter.fxml



# PaintController.java

Figure 13: Mã nguồn Painter.java



# Giao diện

Figure 14: Mã nguồn PaintController.java

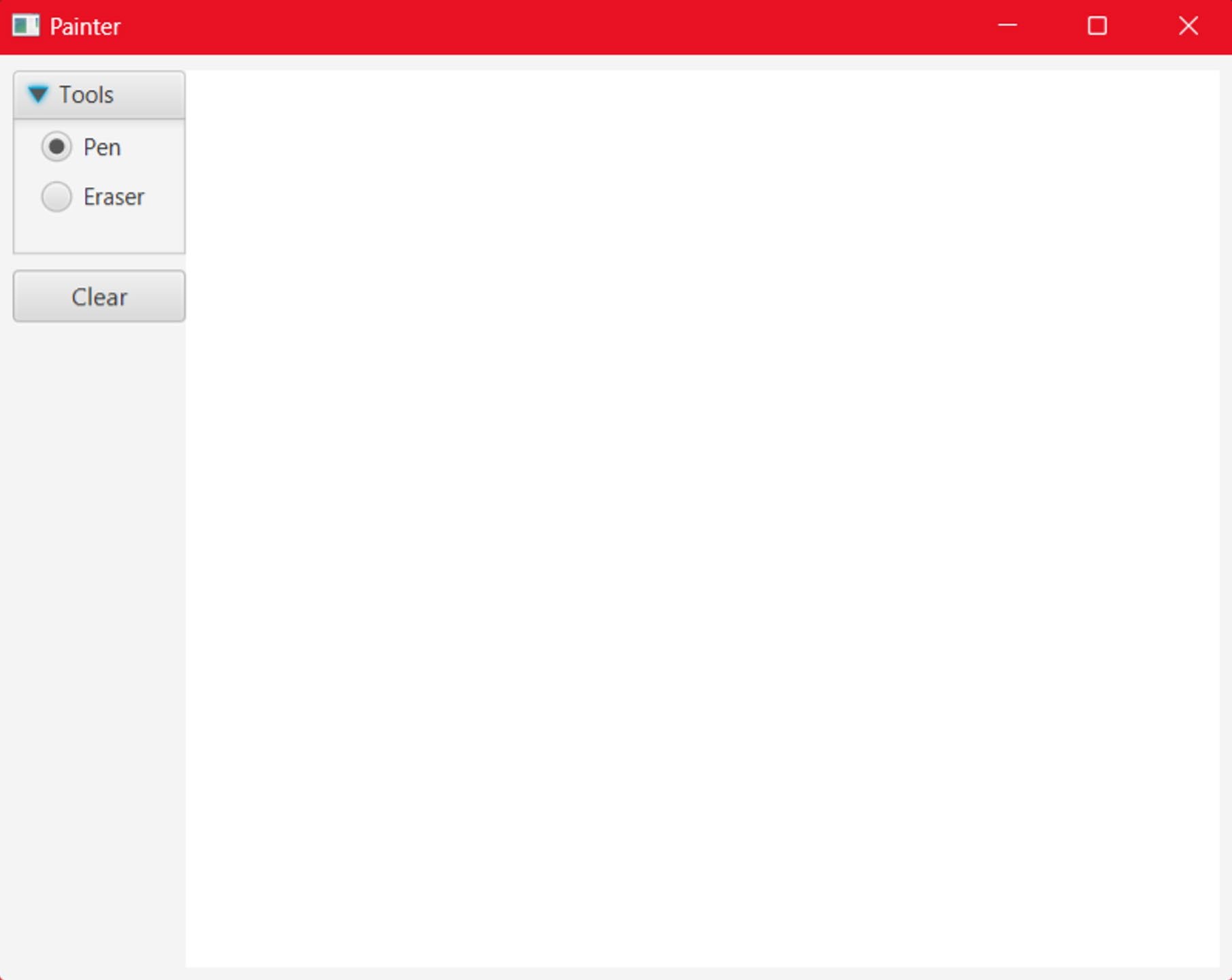


Figure 15: Giao diện Painter

# Setting up the View Cart Screen with ScreenBuilder

* 1. Cart.fxml



Figure 16: Mã nguồn Cart.fxml

# Giao diện Cart Screen

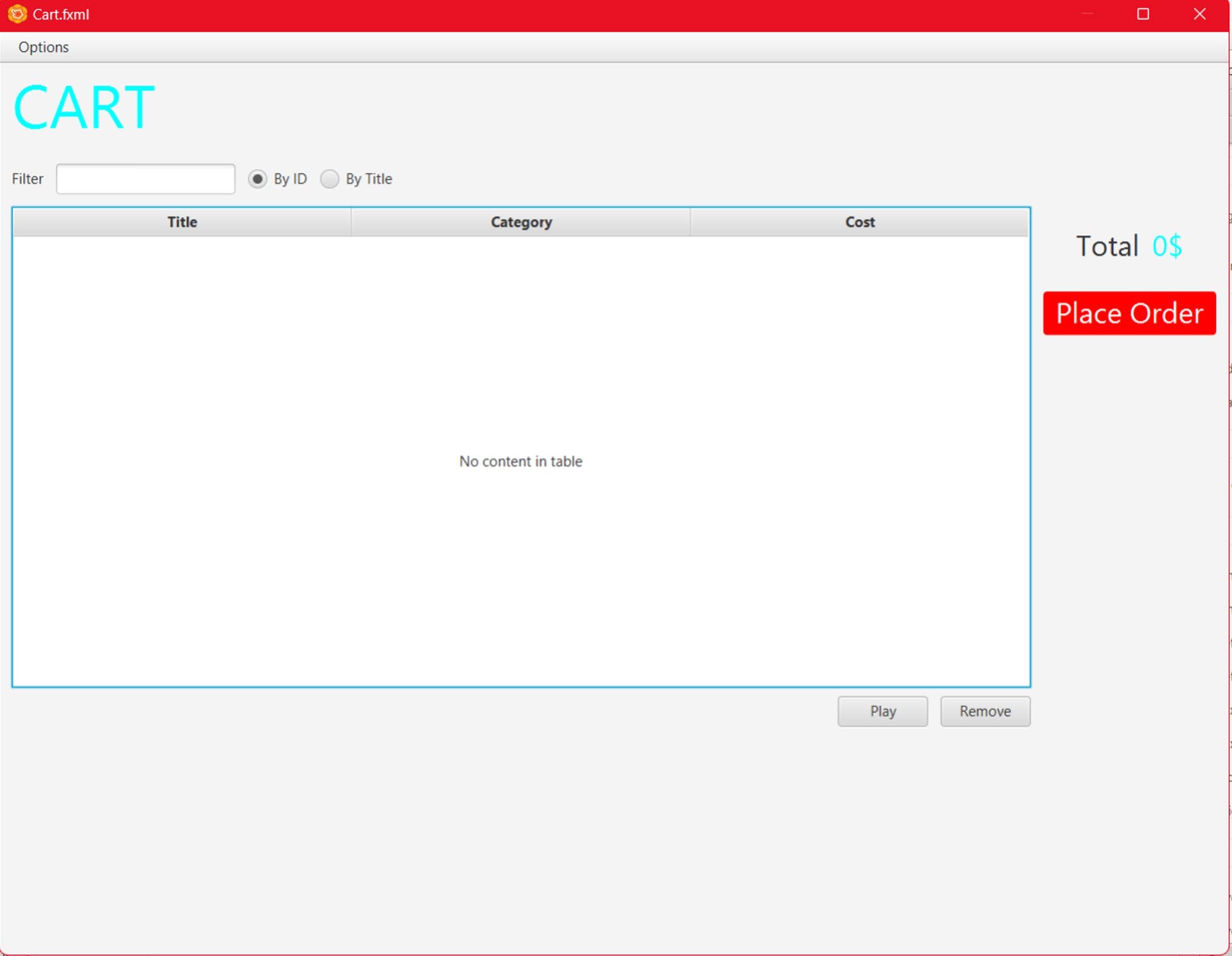


Figure 17: Giao diện Cart Screen

# Integrating JavaFX into Swing application – The JFXPanel class

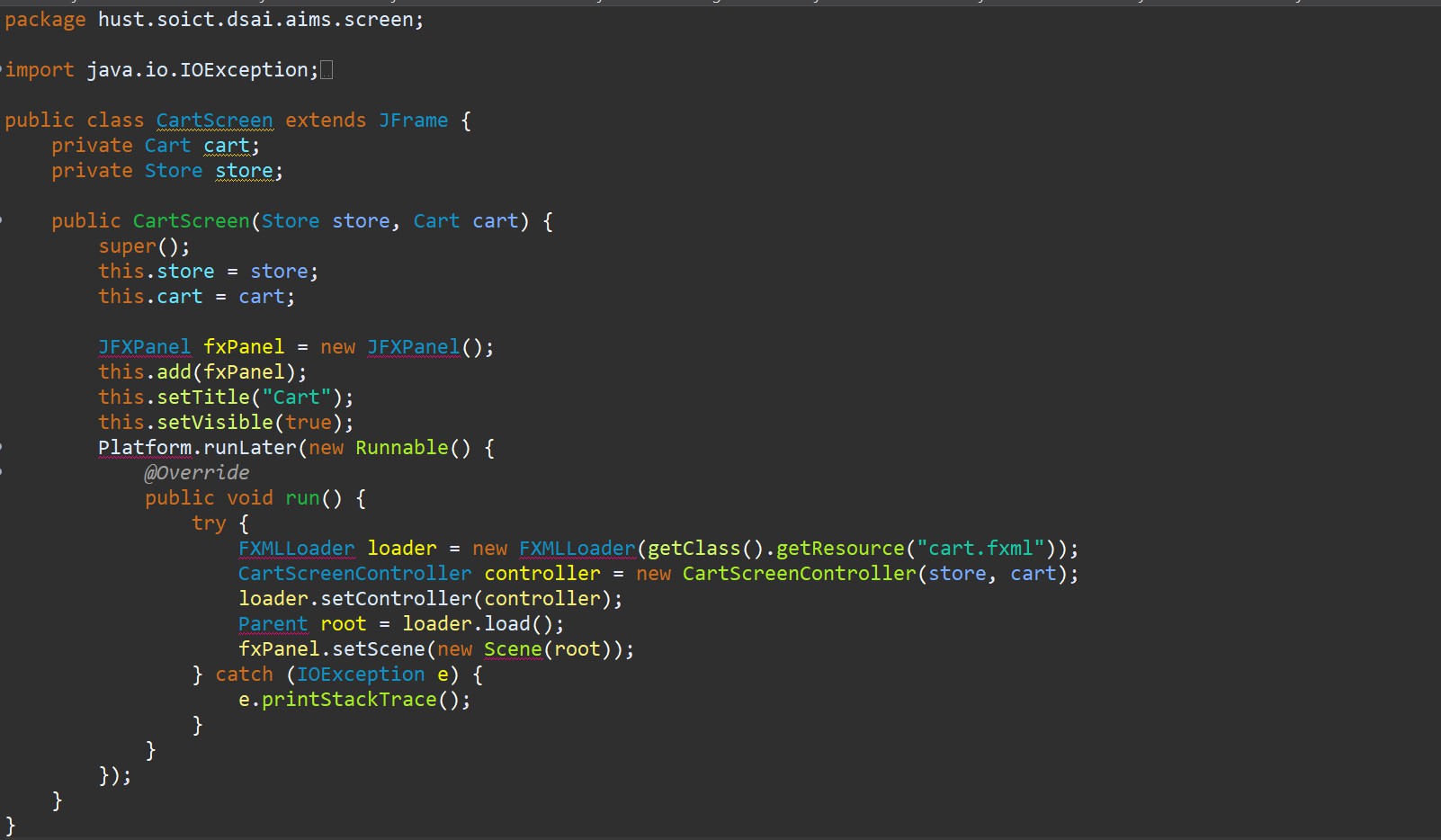


Figure 18: Mã nguồn Cart Screen (nhúng JavaFX vào Swing)

# View the items in cart – JavaFX’s data-driven UI

* 1. CartScreenController.java

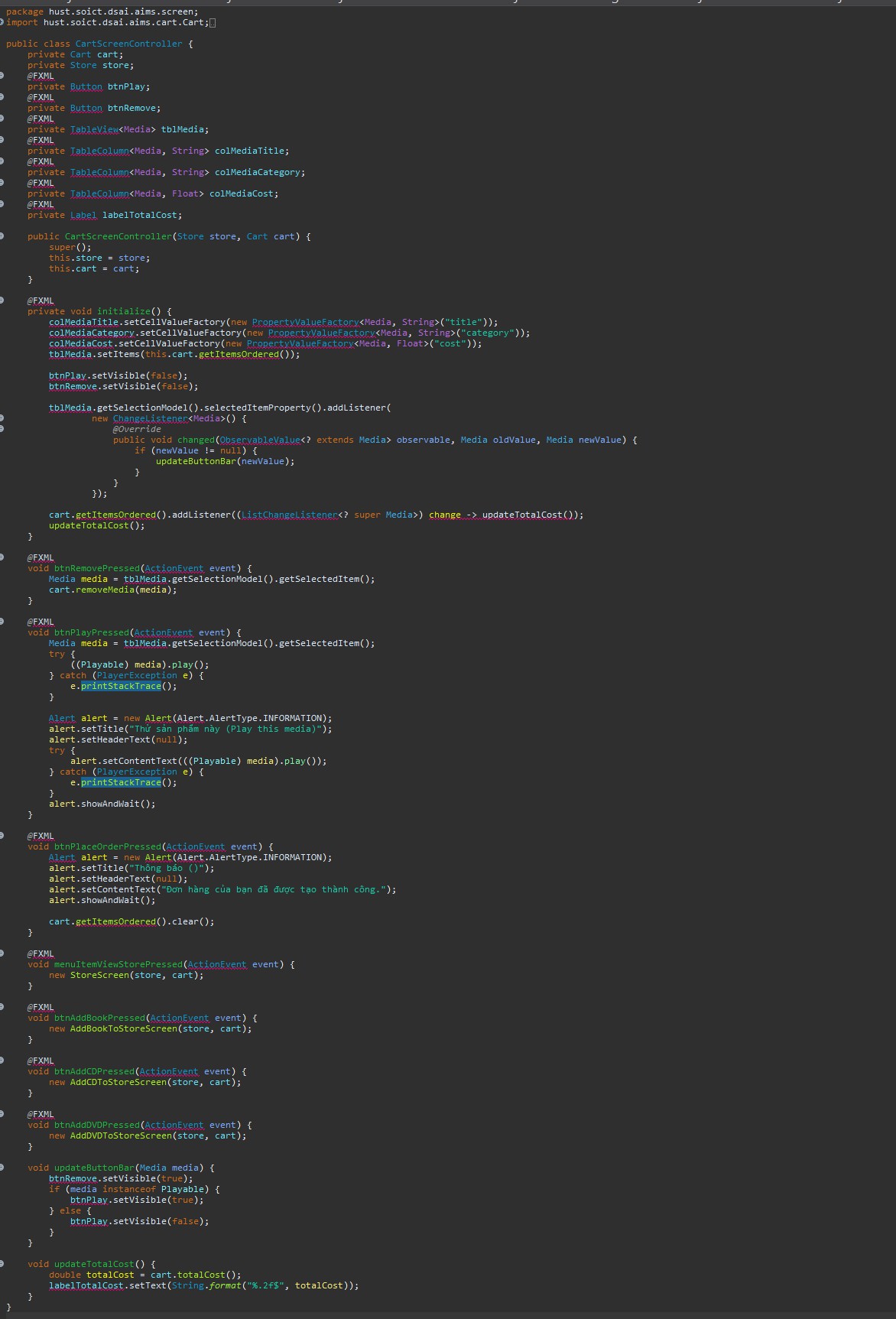


Figure 19: Mã nguồn CartScreenController.java

# Sửa lại Cart.java

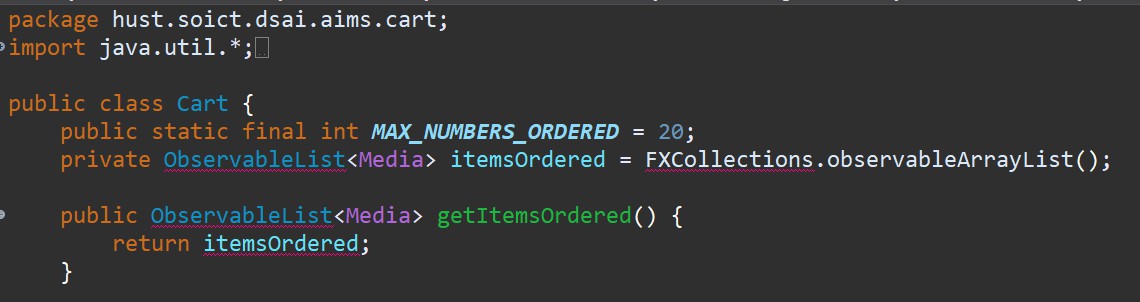


Figure 20: Chỉnh sửa Cart.java để phù hợp với JavaFX

# Giao diện CartScreen



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

1. Updating buttons based on selected item in TableView –

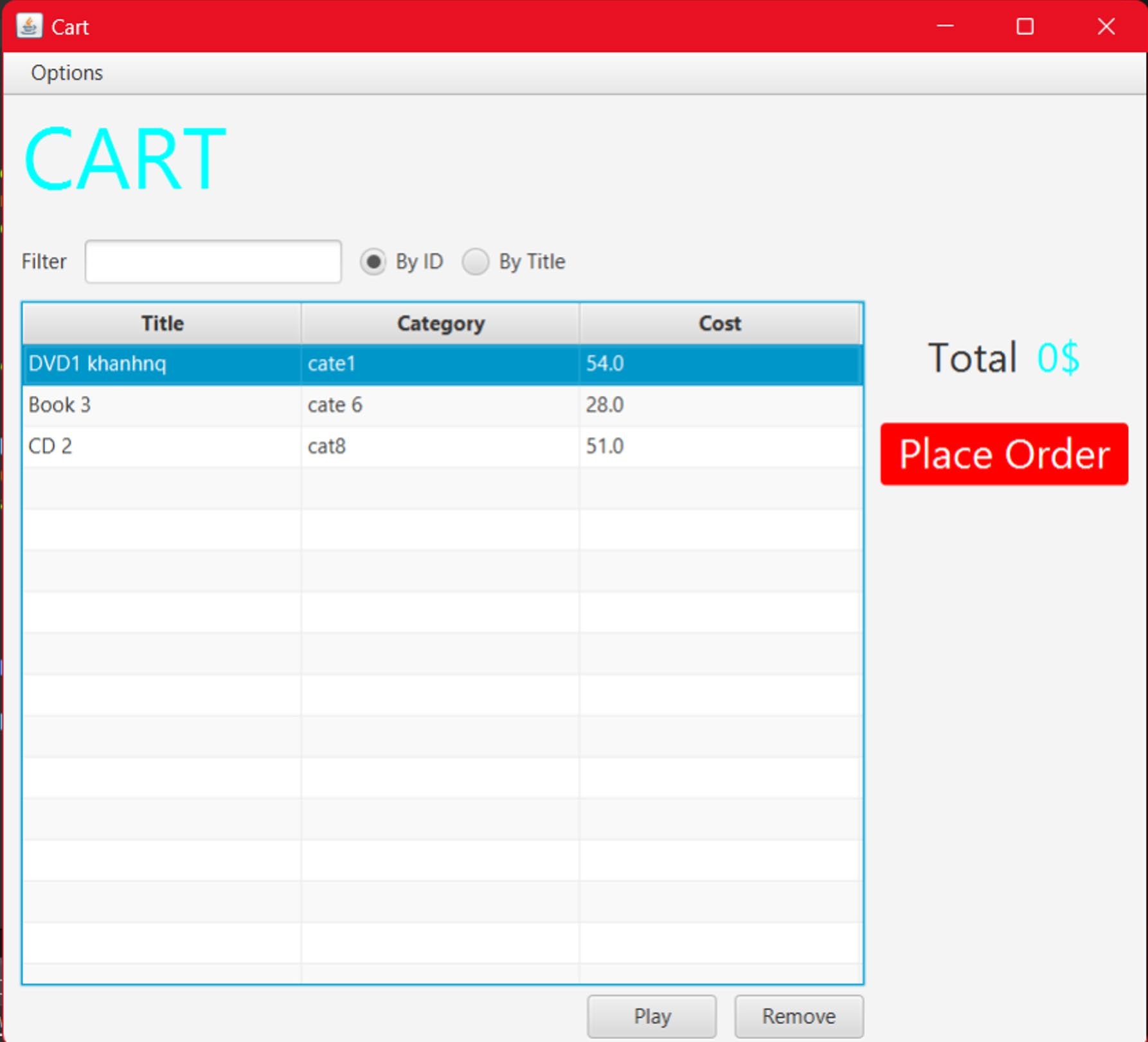
ChangeListener

* 1. Sửa lại CartScreenController.java



Figure 22: Chỉnh sửa CartScreenController.java

# Giao diện CartScreen



1. Deleting a media

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

# Thêm phương thức khi ấn nút Remove

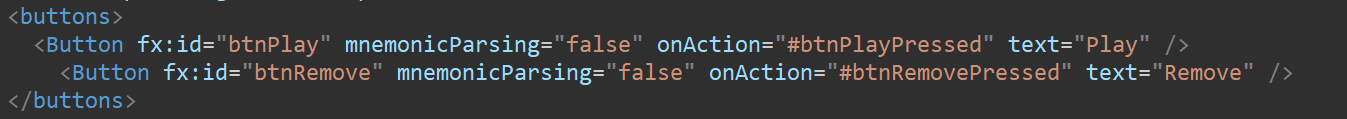


Figure 24: Chỉnh sửa Cart.fxml cho chức năng xóa sản phẩm khỏi cart

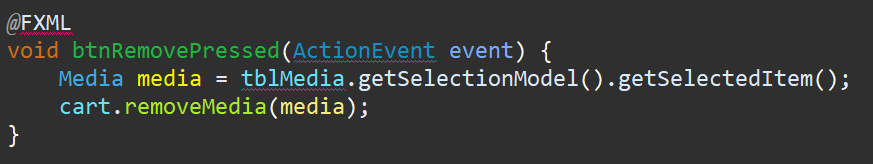


Figure 25: Thêm phương thức bắt sự kiện nút Remove được click

# Giao diện

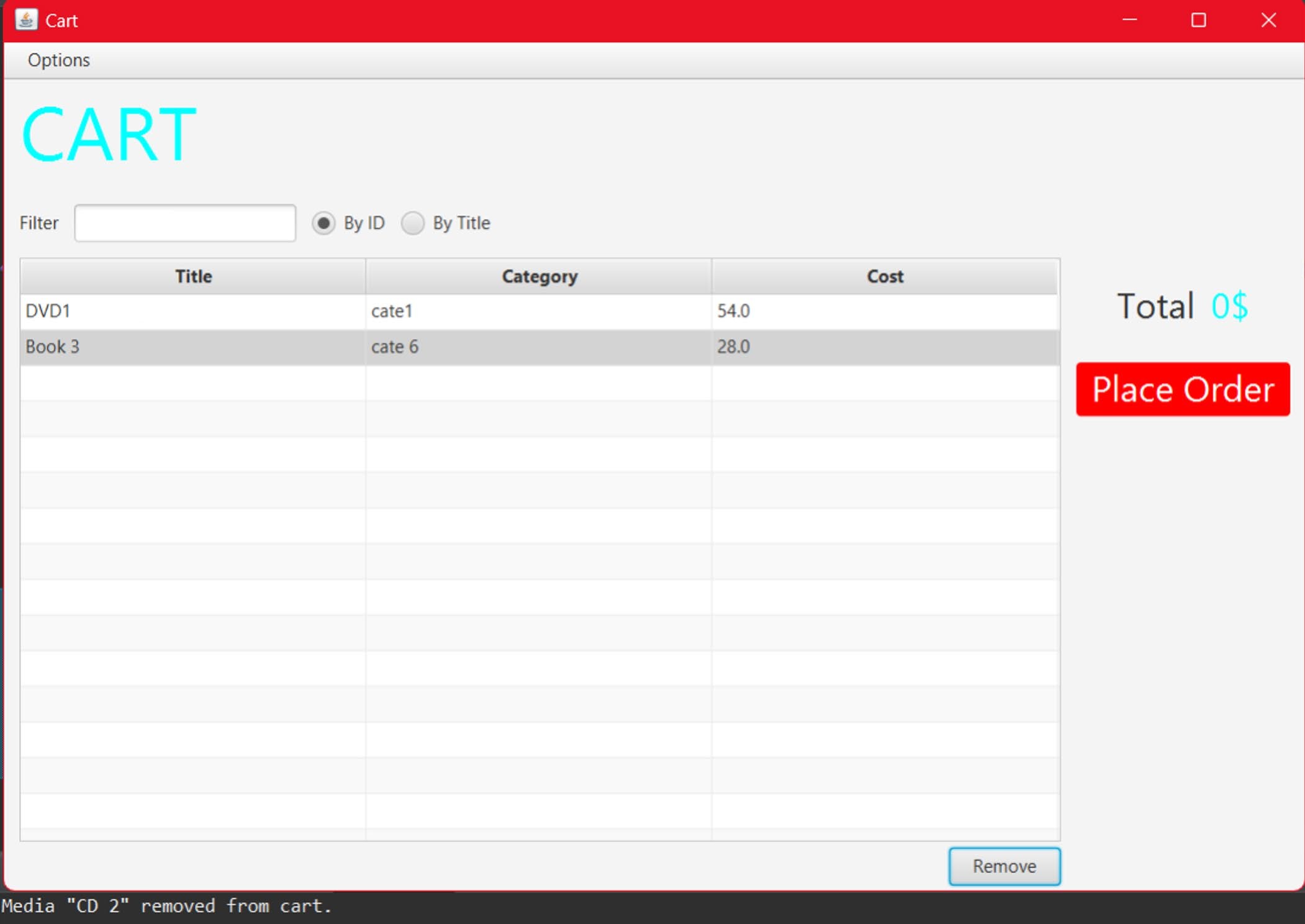
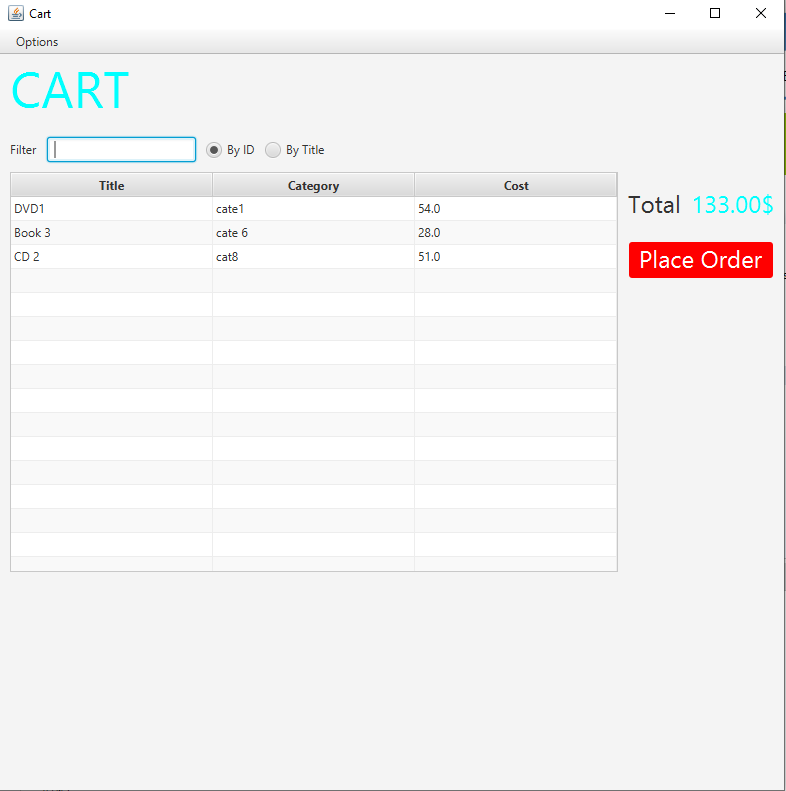


Figure 26: Giao diện Cart sau khi xóa CD2 khỏi cart

1. Filter items in cart – FilteredList (optional)



1. Complete the Aims GUI application
   1. Cart Screen
      1. “Place order” button

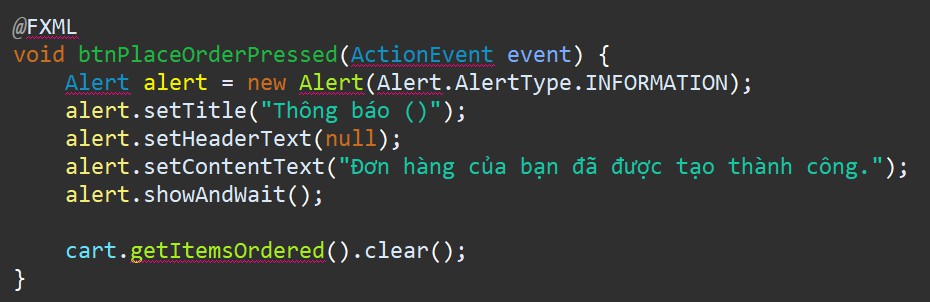


Figure 27: Thêm phương thức bắt sự kiện nút Place order được click



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

# “Play” button



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



Figure 30: Hộp thoại Play media trong Cart

# “Total cost” label

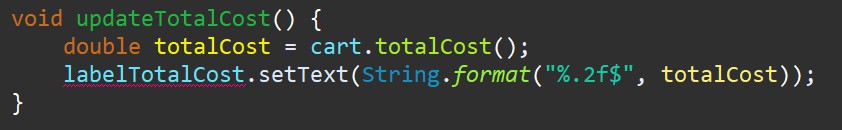


Figure 31: Mã nguồn cập nhật tổng giá tiền của đơn hàng

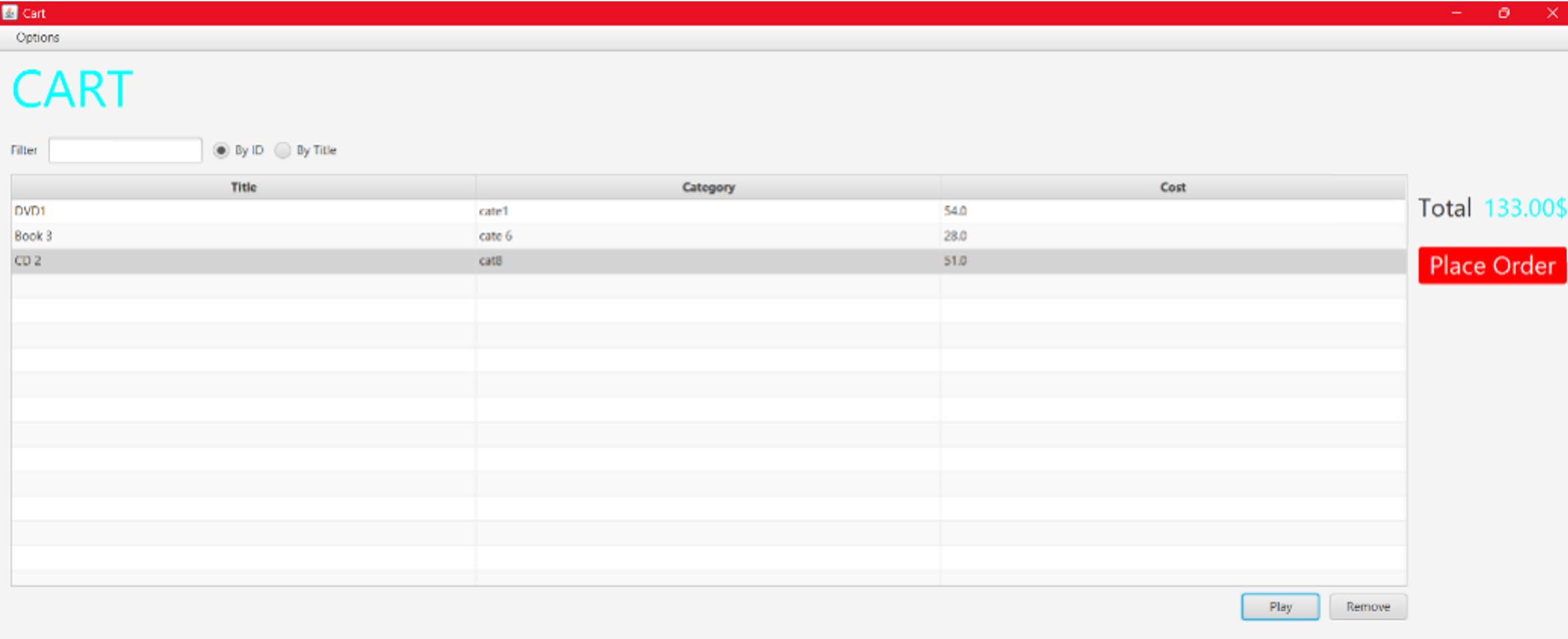
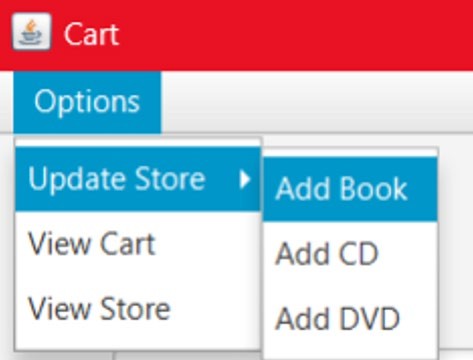


Figure 32: Giao diện Cart (hiển thị tổng giá tiền của đơn hàng)

# Menu bar



Figure 33: Mã nguồn MenuBar



# Store Screen

Figure 34: Giao diện MenuBar



Figure 35: Mã nguồn thêm sản phẩm vào cart



Figure 36: Thông báo sản phẩm được thêm vào cart

# Update Store Screen



Figure 37: Mã nguồn lớp AddItemToStoreScreen

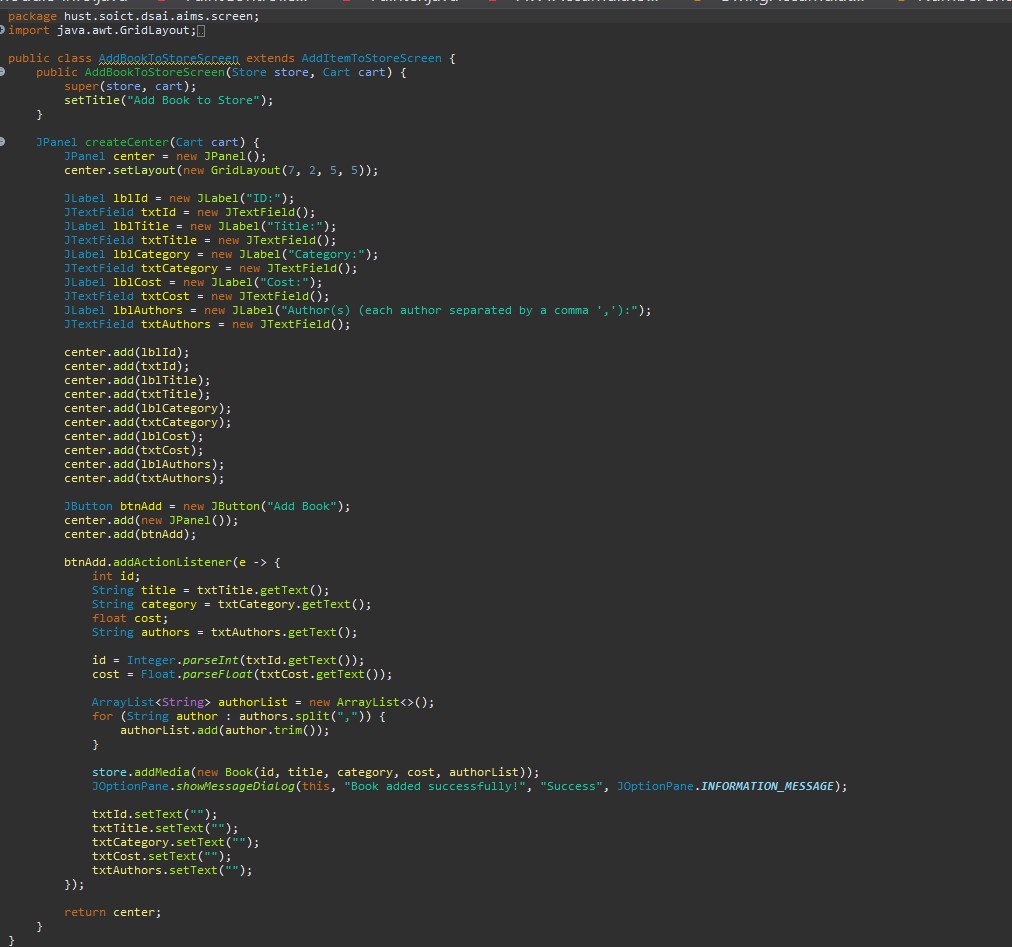


Figure 38: Mã nguồn lớp AddBookToStoreScreen

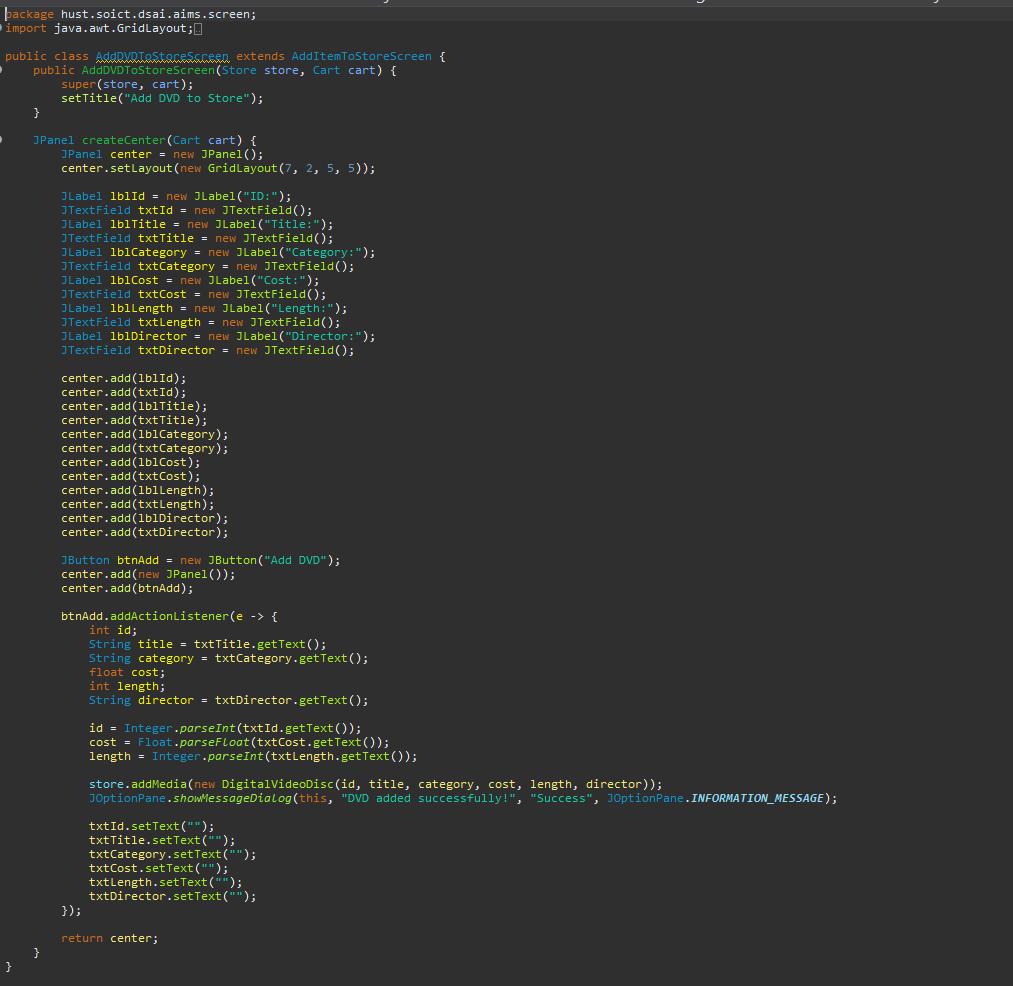


Figure 39: Mã nguồn lớp AddDVDToStoreScreen

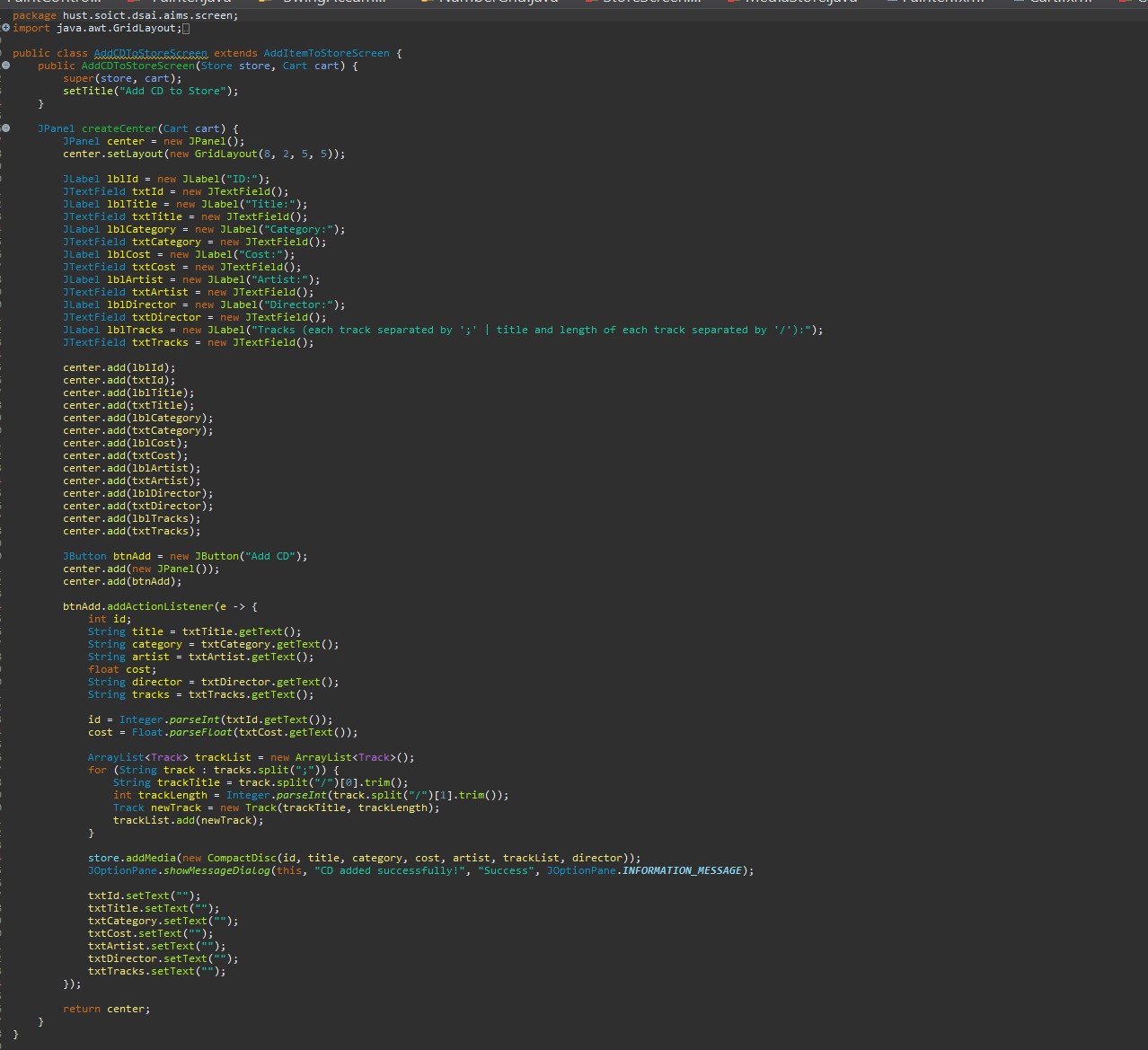


Figure 40: Mã nguồn lớp AddCDToStoreScreen



Figure 41: Giao diện AddBookToStore

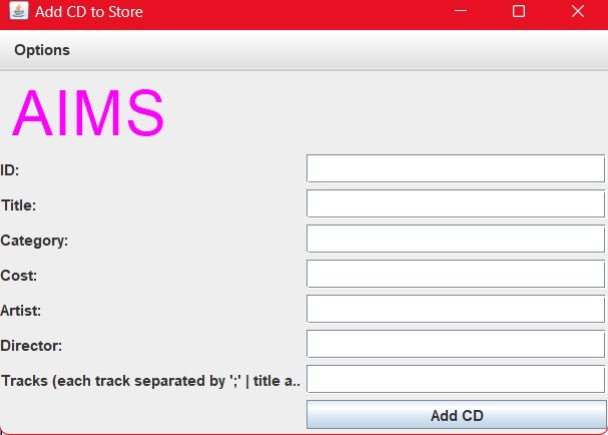


Figure 42: Giao diện AddCDToStore

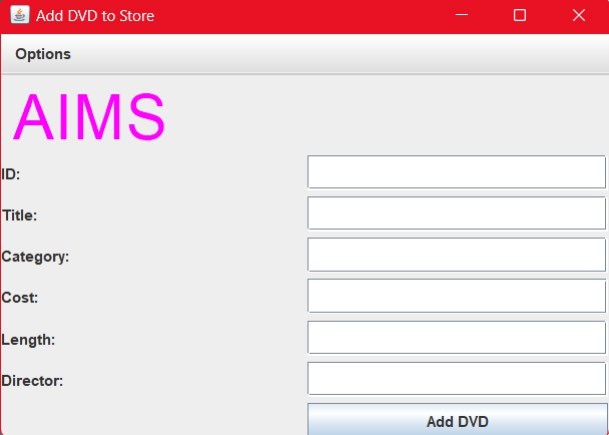


Figure 43: Giao diện AddDVDToStore

# 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



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

# Create a class which inherits from Exception

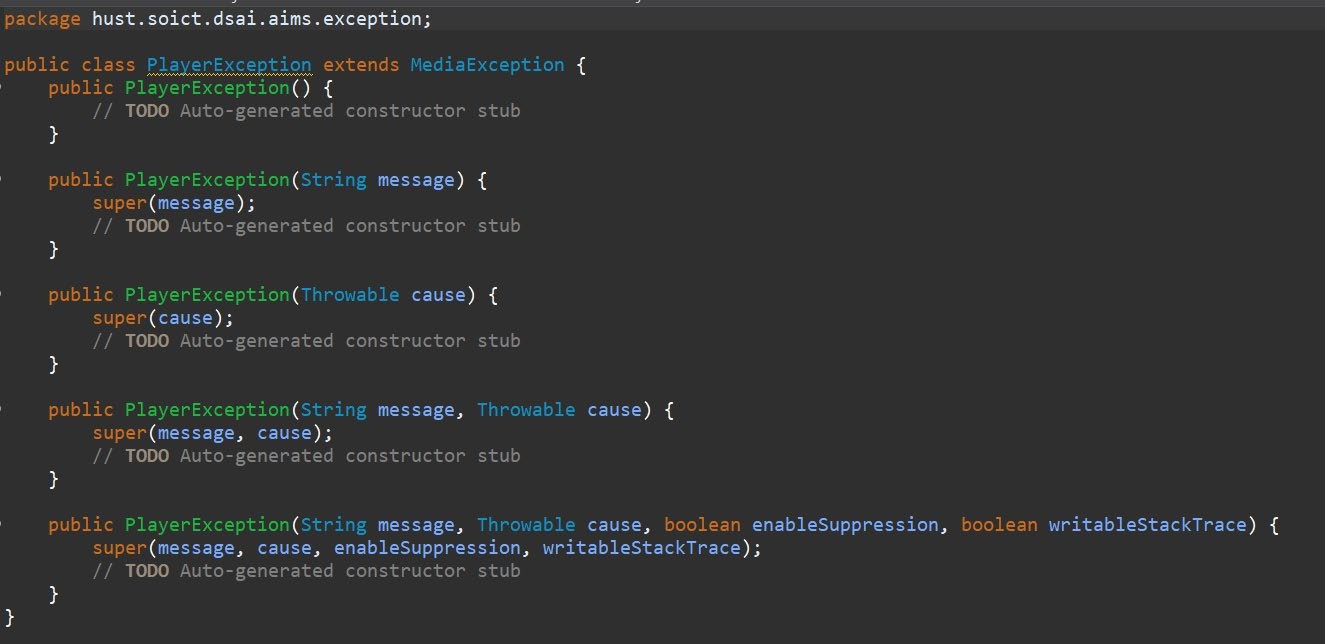


Figure 46: Chỉnh sửa interface Playable

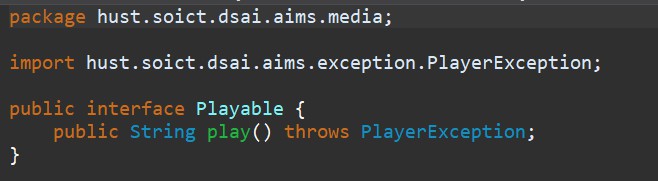


Figure 47: Chỉnh sửa phương thức play() của DigitalVideoDisc

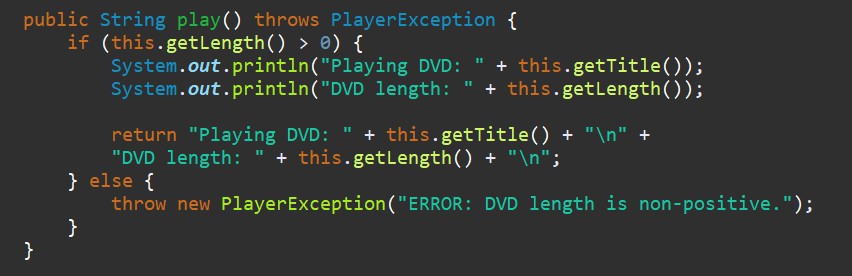


Figure 48: Chỉnh sửa phương thức play() của Track



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

# Update the Aims class

* 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.

# Update the Aims.java code





Figure 50: Chỉnh sửa lớp Aims

# Modify the equals() method of Media class

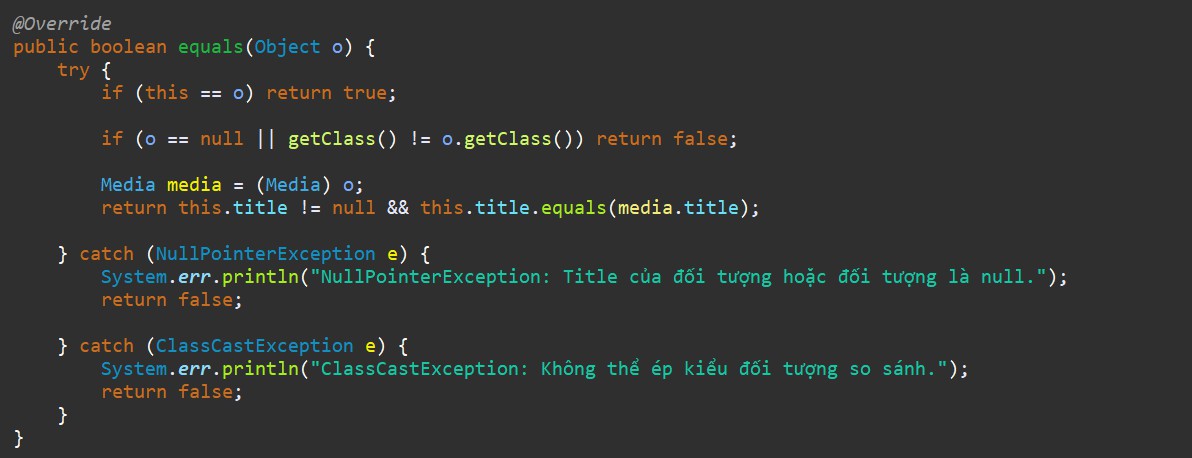


Figure 51: Chỉnh sửa phương thức equals() của lớp Media

# 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.

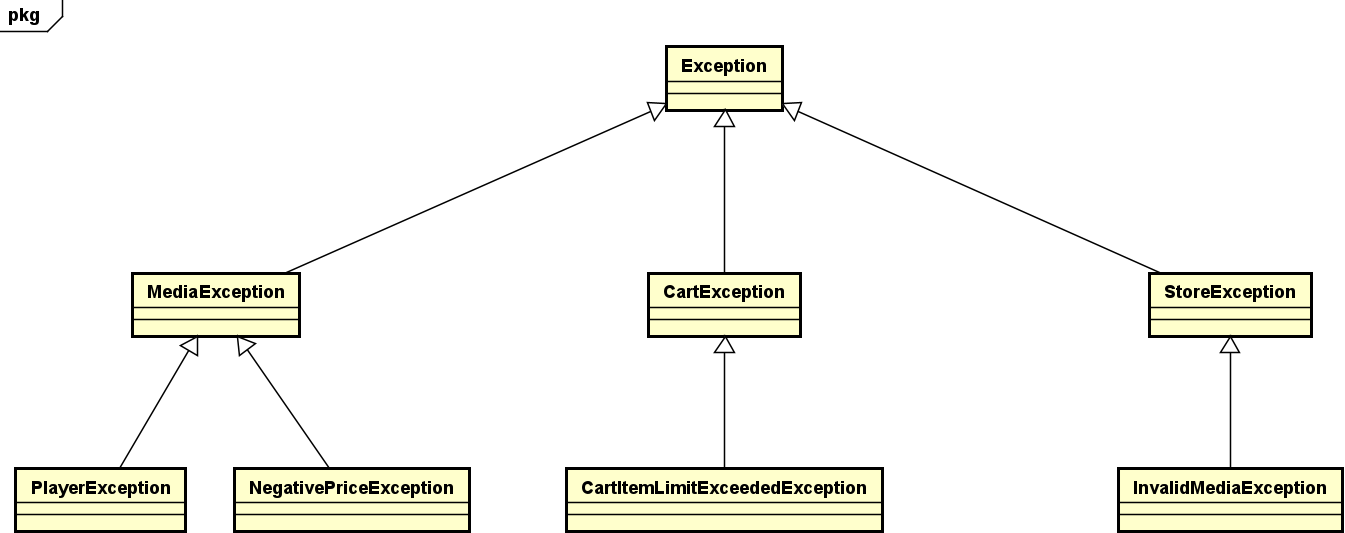


Figure 52: Cây phân cấp ngoại lệ