**Đại học Bách khoa Hà Nội**

**Trường Công nghệ thông tin và truyền thông**

**-----o0o-----**

****

**BÁO CÁO THỰC HÀNH**

**Môn: Thực hành lập trình hướng đối tượng**

**Học phần: IT3103**

**Mã lớp: 721428**

**LAP 05 : 29/12/2022**

**Giảng viên hướng dẫn: Lê thị Hoa**

**Sinh viên thực hiện: Lê Thế Anh**

**MSSV: 20200018**

**Mục lục**

[***1.* Tạo một ứng dụng đơn giản với Swing** 3](#_Toc123764737)

[**1.1 Tạo class AWT Accumulator** 3](#_Toc123764738)

[**1.2 Tạo class Swing Accumulator** 4](#_Toc123764739)

[**2. Cách tổ chức Layout trong JavaSwing với Layout Managers** 6](#_Toc123764740)

[**2.1 Tạo class Number Grid** 7](#_Toc123764741)

[**3. Khởi tạo giao diện người dùng UI cho AIMS Project** 8](#_Toc123764742)

[**3.1 View Store Screen** 8](#_Toc123764743)

[**3.2 Triển khai xử lý sự kiện khi người dùng nhấn vào các button** 15](#_Toc123764744)

[**4. JavaFX API** 16](#_Toc123764745)

[**4.1 Xây dựng giao diện GUI sử dụng Scene Builder** 16](#_Toc123764746)

[**4.2 Tính năng vẽ** 19](#_Toc123764747)

[**4.3 Tính năng xóa** 21](#_Toc123764748)

[**5. Setting up the View Cart Screen with ScreenBuilder** 21](#_Toc123764749)

[**6. Sử dụng JavaFX trong ứng dụng Swing** 22](#_Toc123764750)

[**7. Xem danh sách sản phẩm trong giỏ hàng – JavaFX’s data-driven UI** 24](#_Toc123764751)

[**7.1 Tạo class CartScreenController** 24](#_Toc123764752)

[**7.2 Hiện button play và remove khi 1 sản phẩm được select** 28](#_Toc123764753)

[**8. Chức năng xóa sản phẩm trong giỏ hàng** 30](#_Toc123764754)

[**9. Tính năng tìm kiếm sản phẩm trong giỏ hàng** 32](#_Toc123764755)

[**10. Hoàn thiện chương trình Aims** 34](#_Toc123764756)

[**10.1 Màn hình Cart Screen** 34](#_Toc123764757)

[**10.2 Màn hình Store Screen** 37](#_Toc123764758)

[**10.3 Màn hình Update Store Screen** 38](#_Toc123764759)

[**11. Kiểm tra tất cả các mã nguồn trước đó để nắm bắt/xử lý/ủy nhiệm các ngoại lệ runtime Exception** 48](#_Toc123764760)

[**11.1 Kiểm tra ngoại lệ khi nhập quá số lượng tối đa trong một giỏ hàng** 48](#_Toc123764761)

[**11.2 Tạo 1 class xử lý ngoại lệ Exception** 48](#_Toc123764762)

[**12. Cập nhật phương thức equals của class Media** 49](#_Toc123764763)

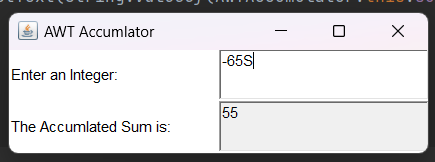
# ***1.* Tạo một ứng dụng đơn giản với Swing**

## **1.1 Tạo class AWT Accumulator**

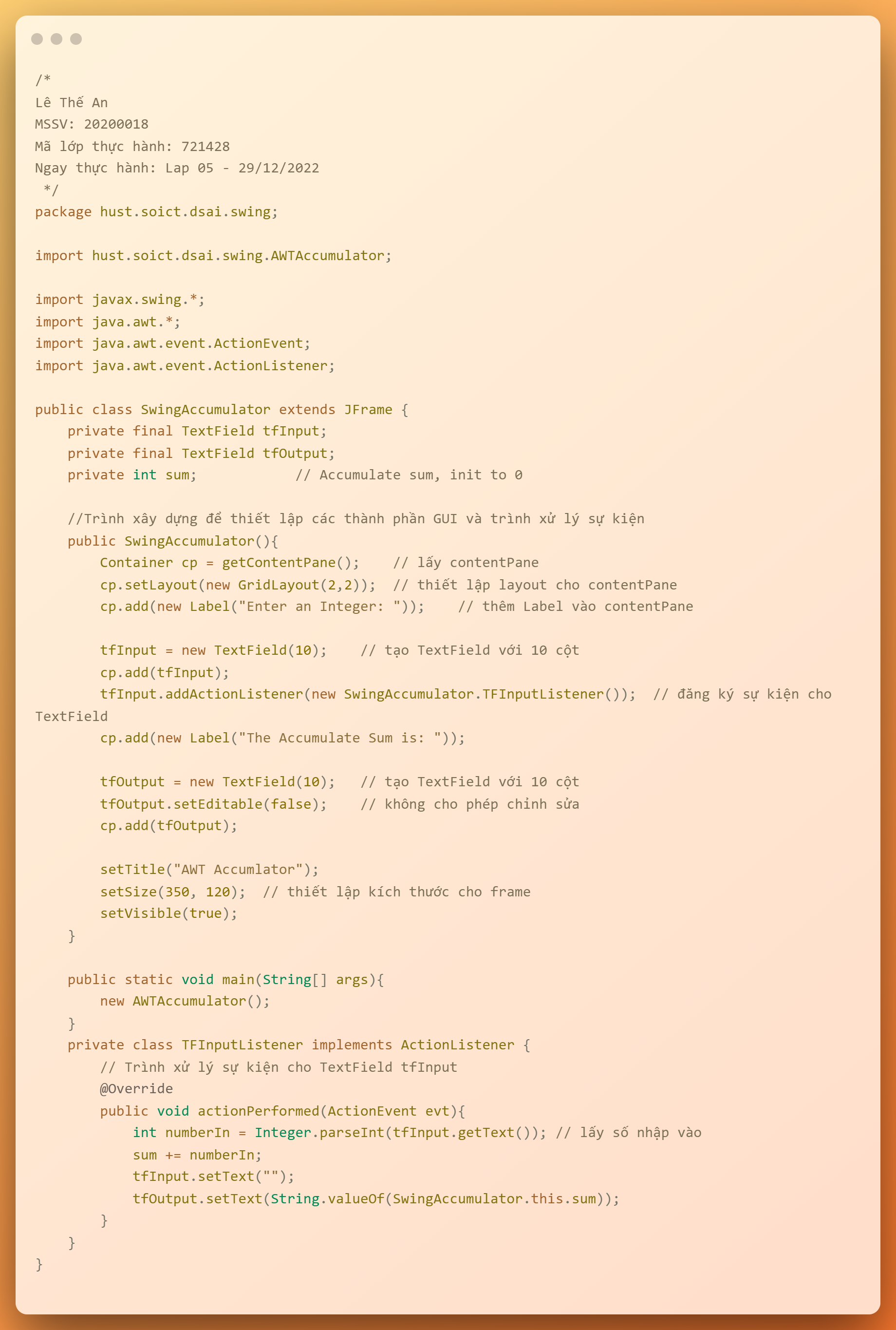
Ảnh có chứa văn bản

Mô tả được tạo tự động

Kết quả:



## **1.2 Tạo class Swing Accumulator**



Kết quả

Graphical user interface, application

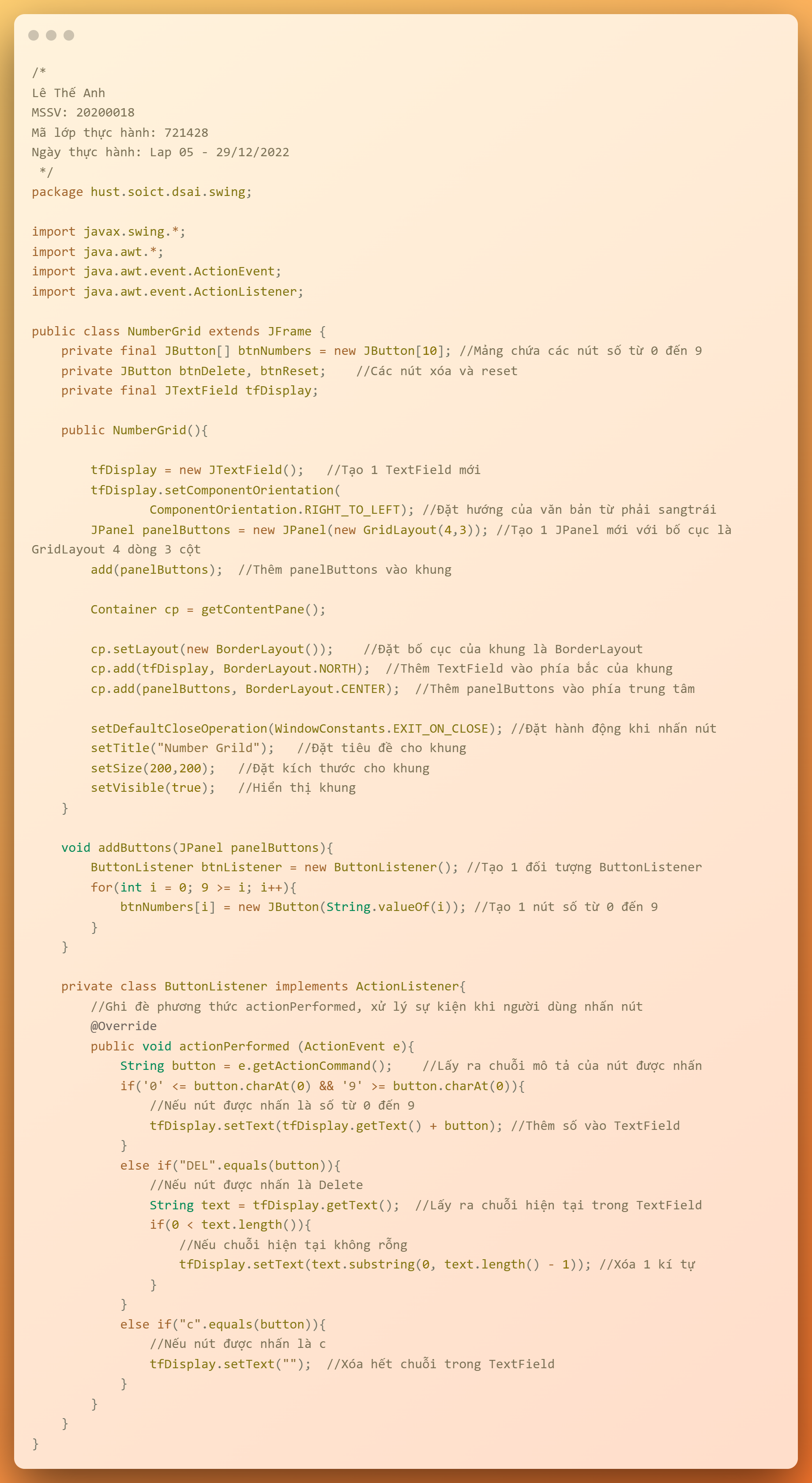
Description automatically generated

# **2. Cách tổ chức Layout trong JavaSwing với Layout Managers**

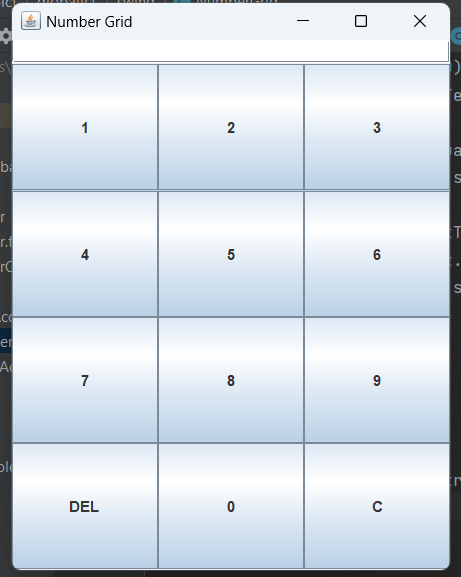
Graphical user interface, diagram

Description automatically generated

## **2.1 Tạo class Number Grid**



Kết quả:



# **3. Khởi tạo giao diện người dùng UI cho AIMS Project**

Đối với màn hình ứng dựng Aims, chúng ta sẽ triển khai 3 màn hình chính:

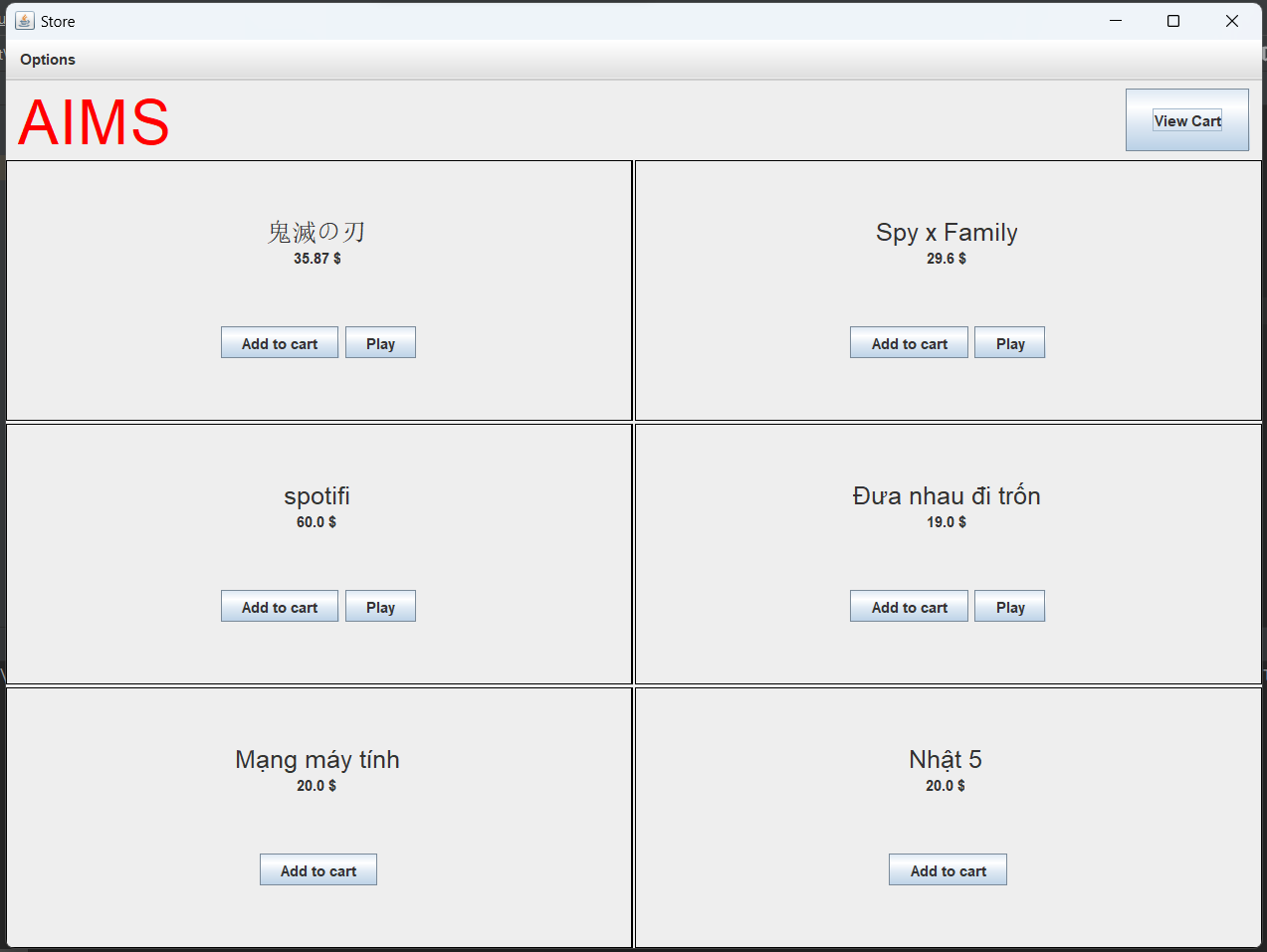
* View Store: Sử dụng Swing
* View Cart: Sử dụng JavaFX
* Update Store: Sử dụng Swing hoặc JavaFX

## **3.1 View Store Screen**

Ở View Store Screen, chúng ta sẽ sử dụng BorderLayout.

Trong thành phần NORTH sẽ có thanh menu và tiêu đề

Trong thành phần CENTER sẽ có một ô sử dụng GridLayout, mỗi ô là một mặt hàng trong cửa hàng.



#### **3.1.1 Tạo class StoreScreen**

Tạo các phương thức createNorth(), createMenuBar() và createHeader(). Thanh menu bar sẽ được hiển thị như sau:

Ảnh có chứa văn bản

Mô tả được tạo tự động

Ảnh có chứa văn bản

Mô tả được tạo tự động

Ảnh có chứa văn bản

Mô tả được tạo tự động

#### **3.1.2 Tạo Center component Layout**

Ảnh có chứa văn bản

Mô tả được tạo tự động

Khi nhấn chuột vào nút Button Play, ta được kết quả như sau:

Ảnh có chứa văn bản

Mô tả được tạo tự động

#### **3.1.3 The MediaStore Class**

Tạo một lớp GUI là MediaStore kế thừa lớp Jpanel; nó sẽ có 1 thuộc tính là Media media



Ảnh có chứa văn bản

Mô tả được tạo tự động

## **3.2 Triển khai xử lý sự kiện khi người dùng nhấn vào các button**

#### **3.2.1 Khi ấn vào nút play**

Ảnh có chứa văn bản

Mô tả được tạo tự động

#### **3.2.2 Khi ấn vào nút Add to cart**

Màn hình khi chọn “view cart” ở phần “Option” sẽ hiện ra như sau

Ảnh có chứa bàn

Mô tả được tạo tự động

# **4. JavaFX API**

Cấu trúc của JavaFX:

Graphical user interface, text, application, chat or text message

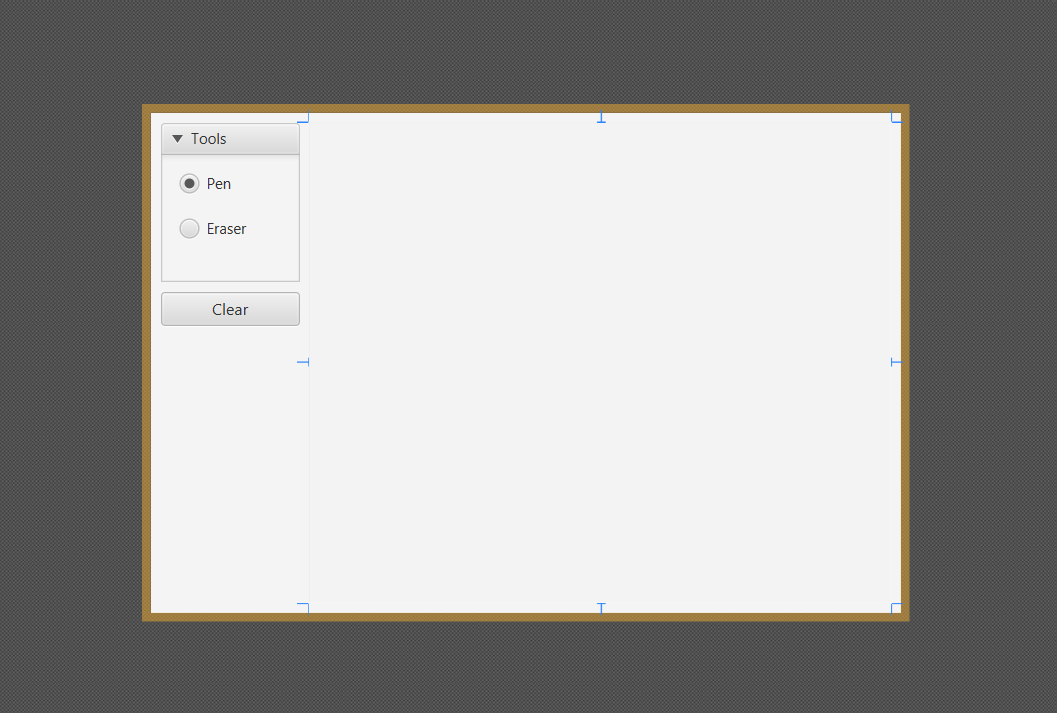
Description automatically generated

Ở phần này, chúng ta sẽ tạo một ứng dụng vẽ cơ bản (Painer):

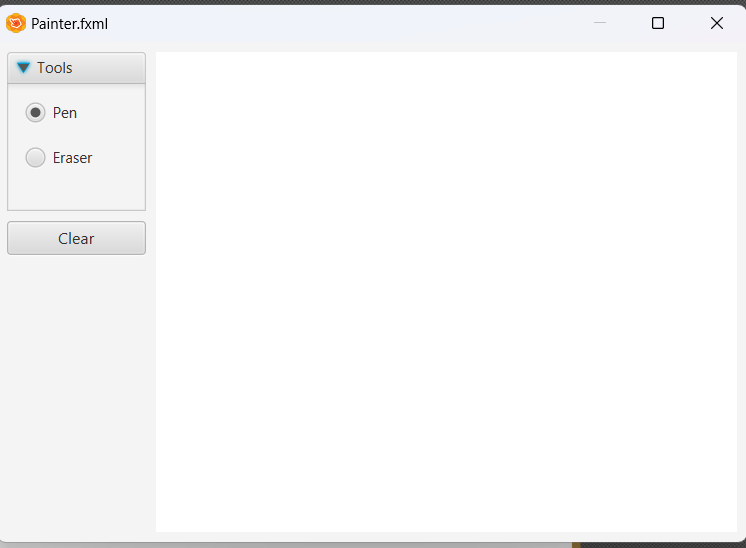
Các bước cần thực hiện như sau:

1. Tạo một FXML file “Painter.fxml”. Chúng ta sẽ sử dụng file này trong ứng dựng Scence Builder để xây dựng giao diện.
2. Tạo Class PainterController
3. Tạo Class Painter

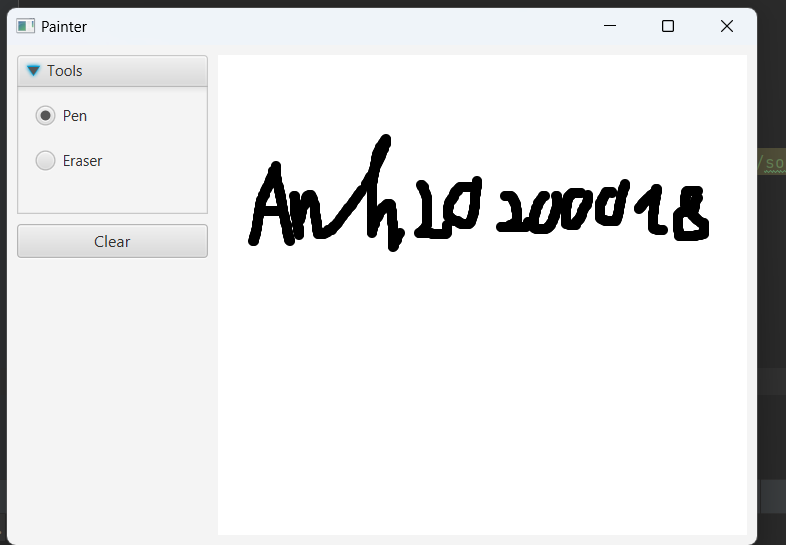
## **4.1 Xây dựng giao diện GUI sử dụng Scene Builder**



Sau khi xuất ra màn hình ta sẽ được kết quả như sau:

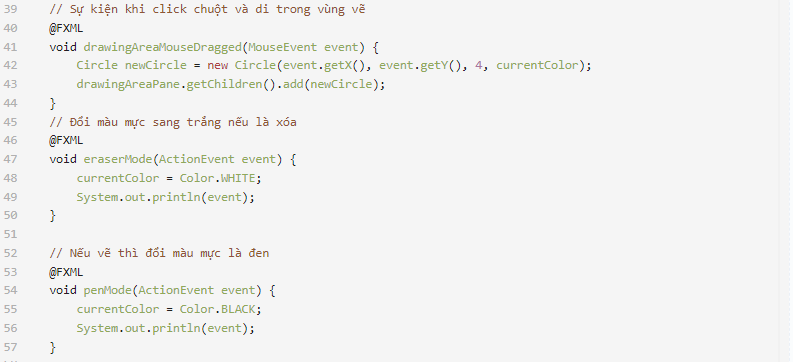


## **4.2 Tính năng vẽ**

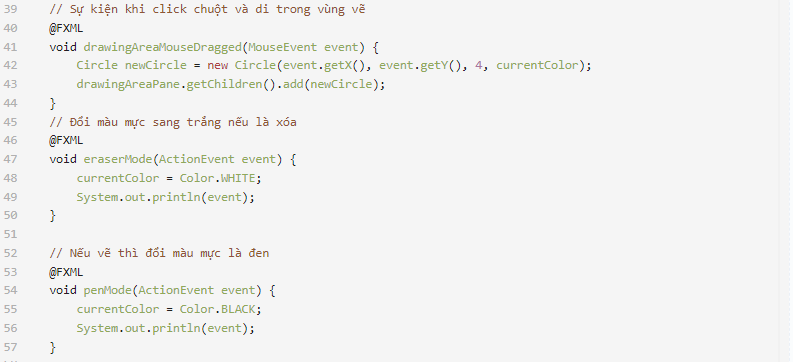


Ảnh có chứa văn bản

Mô tả được tạo tự động

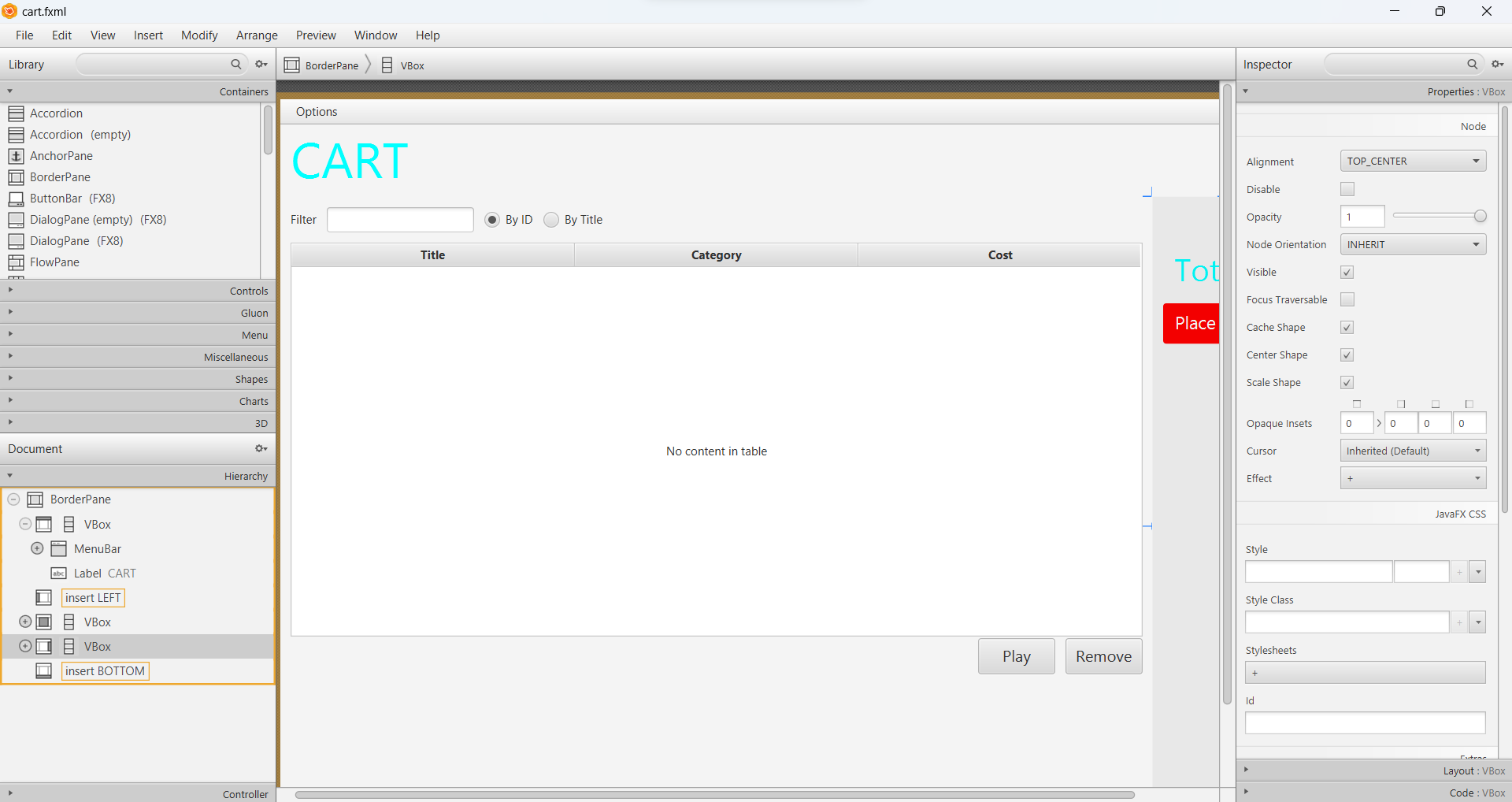


## **4.3 Tính năng xóa**



# **5. Setting up the View Cart Screen with ScreenBuilder**

Layout sử dụng ở đây là BorderPane.Dạng layout này cũng tương tự như BorderLayout trong Swing



Vị trí các component

* Top Area: Menu, Label(Logo)
* Center Area: Filter, Table, Button bar(Play, Remove)
* Right Area: Total, Place Order

Các event cần handle

* Input filter
* Play
* Remove
* Place Order
* Khi có sự kiện add sản phẩm vào giỏ hàng

# **6. Sử dụng JavaFX trong ứng dụng Swing**

Sử dụng phương thức load() của lớp FXMLLoader để tải một tệp FXML và tạo một đối tượng bộ điều khiển cho nó và trả về một đối tượng Node kết quả.

Mã Nguồn:



Kết quả

Ảnh có chứa văn bản

Mô tả được tạo tự động

# **7. Xem danh sách sản phẩm trong giỏ hàng – JavaFX’s data-driven UI**

Đặt thuộc tính fx:id cho cá tiêu đề trong cột table view

* Cột Title: colMediaTitle
* Cột Category: colMediacategory
* Cột Cost: colMediaCost

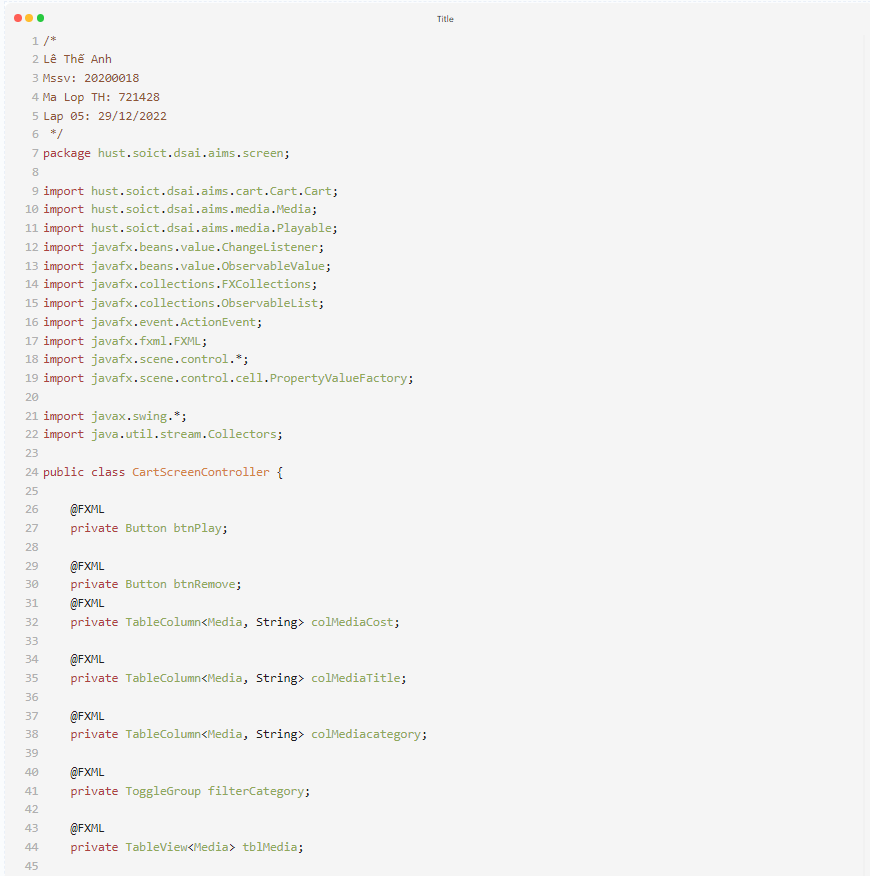
## **7.1 Tạo class CartScreenController**

Kết quả:

Ảnh có chứa bàn

Mô tả được tạo tự động

Mã nguồn:



Ảnh có chứa văn bản

Mô tả được tạo tự động



Ảnh có chứa văn bản

Mô tả được tạo tự động

Ảnh có chứa văn bản

Mô tả được tạo tự động

Bên cạnh đó, chúng ta cần sử dụng một ObservableList để TableView có thể quan sát và phản ánh bất kỳ thay đổi nào trên dữ liệu. Ảnh có chứa văn bản

Mô tả được tạo tự động

Khi có sự thay đổi các sản phẩm trong Cart thì dữ liệu trong table được tự động update lại

Trước khi click “Add to cart”

Ảnh có chứa bàn

Mô tả được tạo tự động

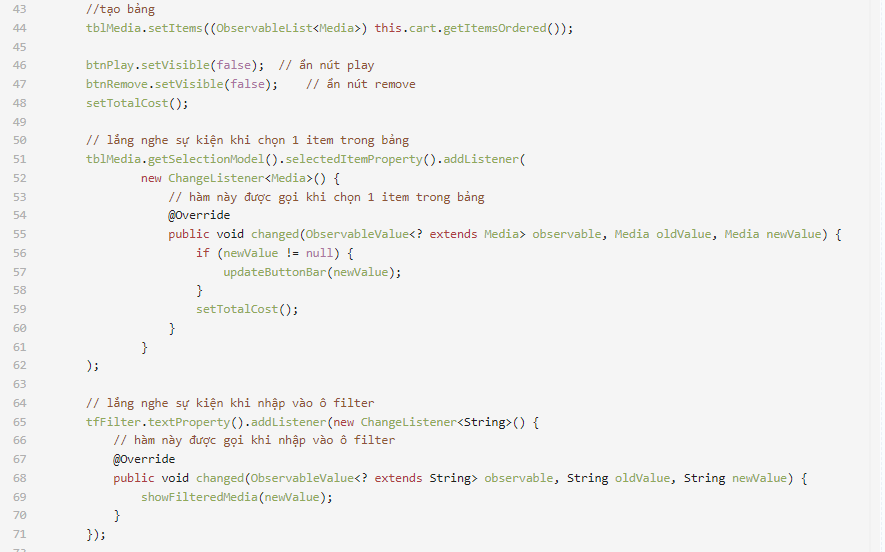
Khi click add to cart, danh sách sản phẩm trong bảng được update tự động

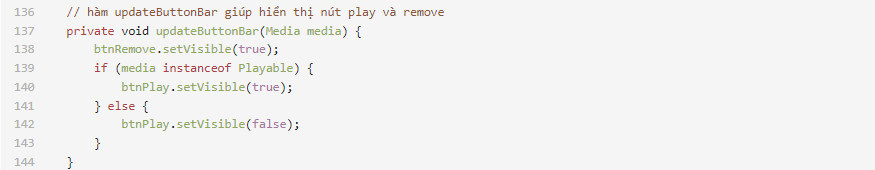
Ảnh có chứa bàn

Mô tả được tạo tự động

## **7.2 Hiện button play và remove khi 1 sản phẩm được select**

Ta cần thêm một EventListener sau để kích hoạt chế độ ẩn hay hiển thị nút play và remove:





Kết quả

Khi không có item nào được select

Ảnh có chứa văn bản

Mô tả được tạo tự động

Khi có item được Select

Ảnh có chứa bàn

Mô tả được tạo tự động

# **8. Chức năng xóa sản phẩm trong giỏ hàng**

Trước ta ta cần set fx:id cho button Remove và thêm ActionEvent cho nút này để bắt sự kiện khi nó được nhấn



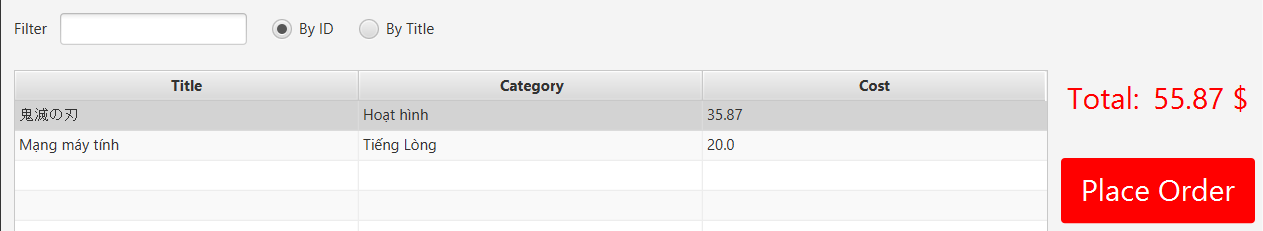
Kết quả

Trước khi xóa

Ảnh có chứa bàn

Mô tả được tạo tự động

Sau khi click vào button Remove thì sản phẩm đó lập tức biến mất trong danh sách sản phẩm

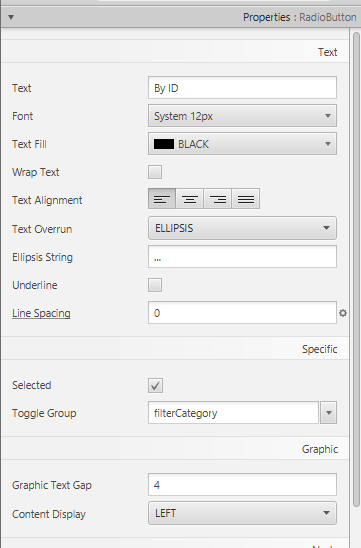
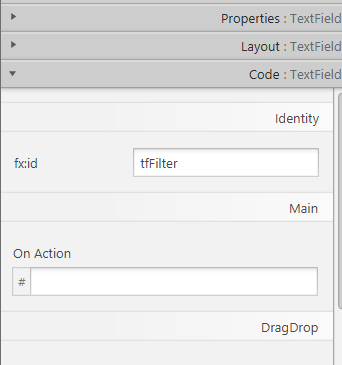


# **9. Tính năng tìm kiếm sản phẩm trong giỏ hàng**

Chúng ta cần tạo ba fx:id cho ba thuộc tính tương ứng trong tableView để điều khiển

* TextField: Có fx:id là tfFilter
* RadioButton: Có text là “By ID”, fx:id là “radioBtnFilterId”
* RadioButton: Có text là “By Title”, fx:id là “radioBtnFilterTitle”

Hai RadioButton trên có cùng thuộc tính ToggleGroup là filterCategory

Chúng ta cần khai báo thêm thuộc tính FilteredList để chứa danh sách các sản phẩm đã được lọc.Nếu không có điều kiện thì thuộc tính này chứa toàn bộ sản phẩm trong giỏ hàng

// Danh sách các List filter được lọc theo điều kiện  
private FilteredList<Media> filteredList;

Ngoài ra để lọc theo điều kiện thì trước tiên ta cần lấy được điều kiện lọc: Lọc theo ID hay theo Title. Đồng thời sử dụng Predicate để lọc dữ liệu.

Ảnh có chứa văn bản

Mô tả được tạo tự động

Sau đó, Chúng ta cần sử lý sự kiện khi người dùng nhập từ vào ô Filter

Ảnh có chứa văn bản

Mô tả được tạo tự động

Kết quả

Danh sách sản phẩm ban đầu

Ảnh có chứa bàn

Mô tả được tạo tự động

Lọc theo ID:

Ảnh có chứa bàn

Mô tả được tạo tự động

Lọc theo title:

Ảnh có chứa bàn

Mô tả được tạo tự động

# **10. Hoàn thiện chương trình Aims**

Danh sách các tính năng của chương trình

## **10.1 Màn hình Cart Screen**

#### **10.1.1 Create place order button**



Kết quả: Khi nhấn vào button Place Order thì toàn bộ sản phẩm trong giỏ hàng bị xóa và thông báo đặt hàng thành công

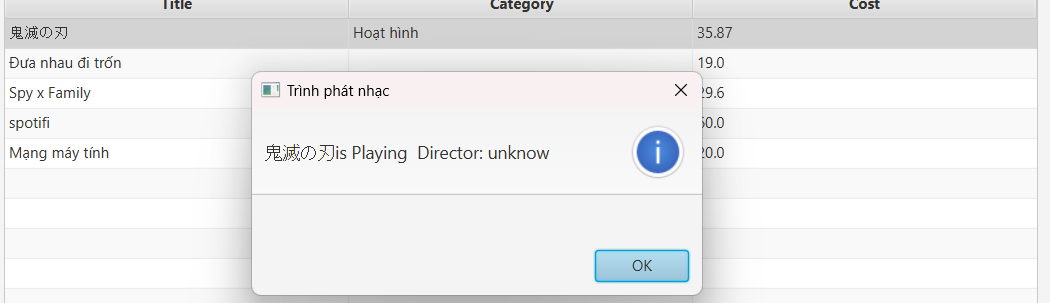
Ảnh có chứa văn bản

Mô tả được tạo tự động

#### **10.1.2 Create play Button**



Kết quả



Ví dụ khi xuất hiện lỗi khi play một media nào đó, chương trình sẽ hiển thị kết quả như sau:

Ảnh có chứa văn bản

Mô tả được tạo tự động

#### **10.1.3: Tổng tiền sản phẩm**

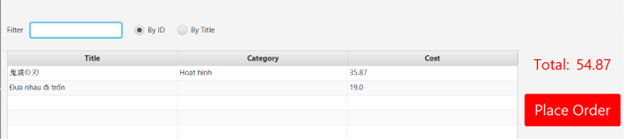


Bằng cách sử dụng hàm setTotalCost(), tổng tiền sẽ được tính lại sau mỗi lần thêm hoặc xóa media khỏi giỏ hàng một cách tự động.



Kết quả

Trước khi thêm 1 sản phẩm:



Sau khi thêm 1 sản phẩm

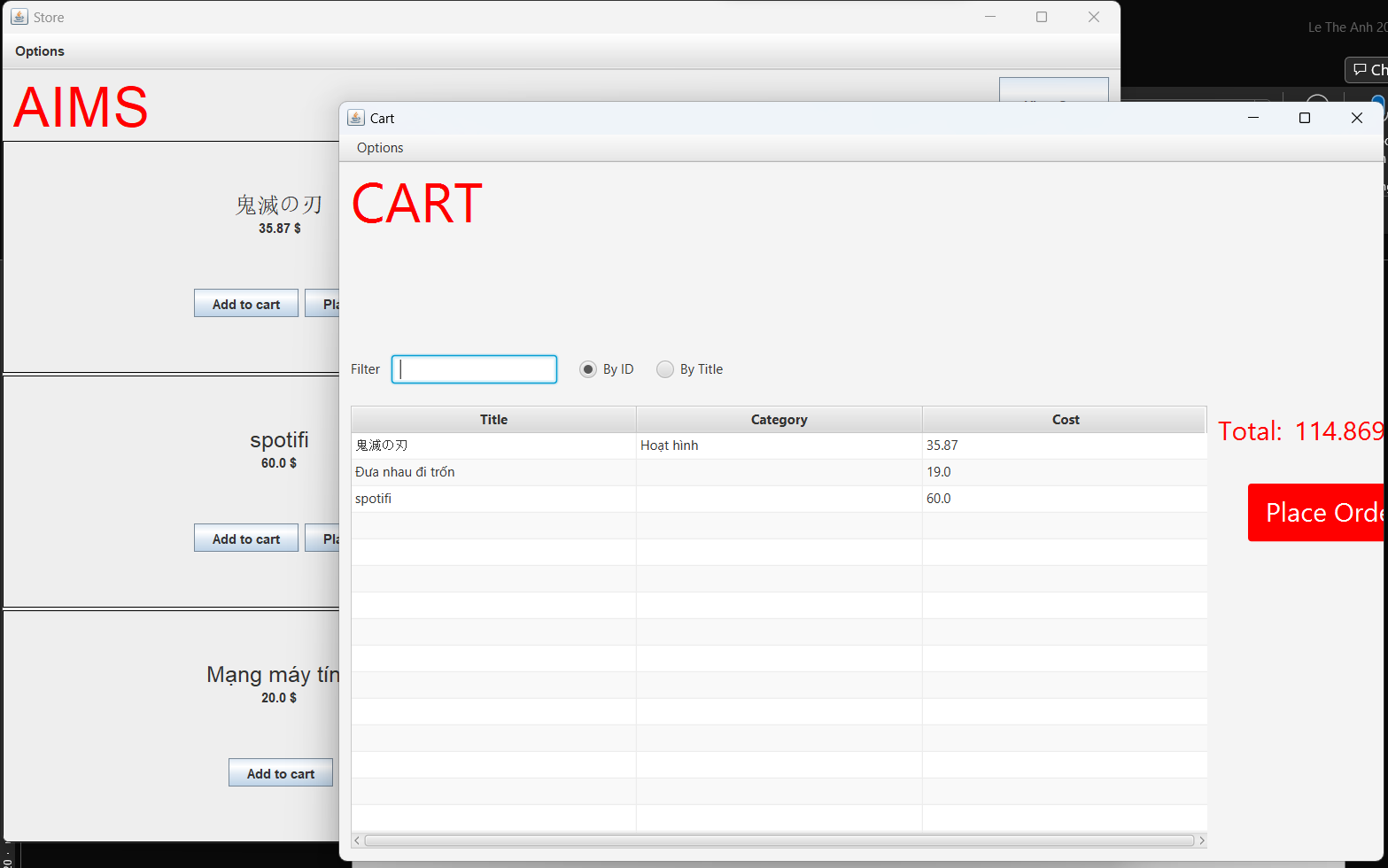
Ảnh có chứa bàn

Mô tả được tạo tự động

## **10.2 Màn hình Store Screen**

Tính năng: Add to cart

Kết quả:



Source code:

Ảnh có chứa văn bản

Mô tả được tạo tự động

## **10.3 Màn hình Update Store Screen**

Ta sẽ làm các tính năng: Thêm DVD mới, thêm CD mới và thêm Book mới

Ảnh có chứa văn bản

Mô tả được tạo tự động

Ảnh có chứa văn bản

Mô tả được tạo tự động

#### **10.3.1 Menu lựa chọn**

Ảnh có chứa văn bản

Mô tả được tạo tự động

Ảnh có chứa văn bản

Mô tả được tạo tự động

#### **10.3.2 Màn hình thêm DVD**

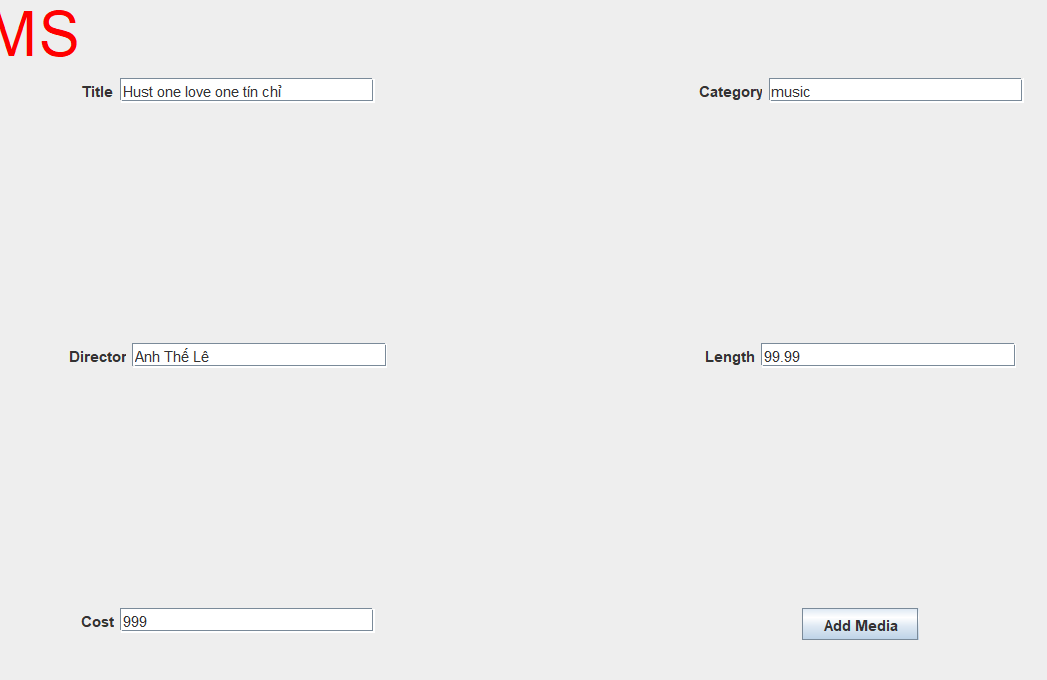
Ảnh có chứa văn bản

Mô tả được tạo tự động

Ảnh có chứa văn bản

Mô tả được tạo tự động

Kết quả:



Ảnh có chứa bàn

Mô tả được tạo tự động

#### **10.3.3 Màn hình thêm CD**

Mã nguồn

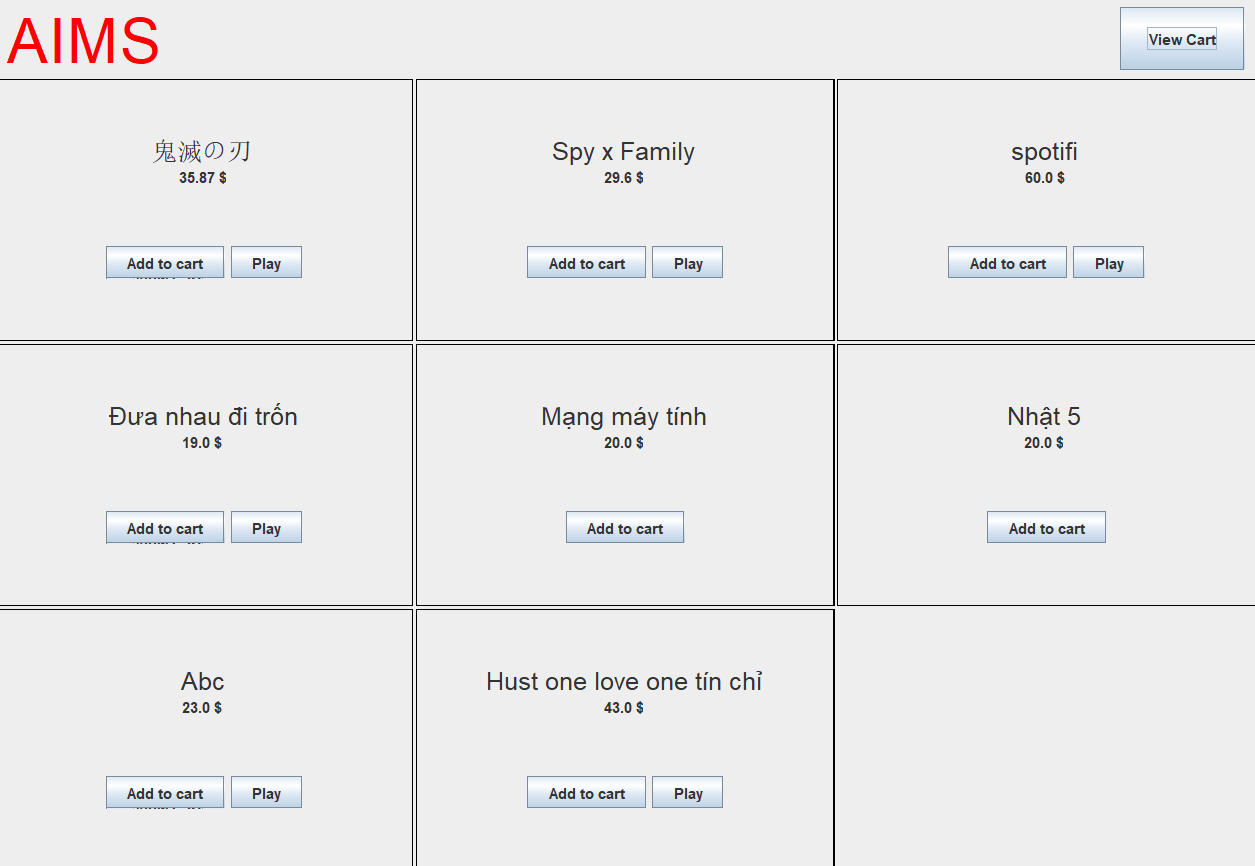
Ảnh có chứa văn bản

Mô tả được tạo tự động

Kết quả

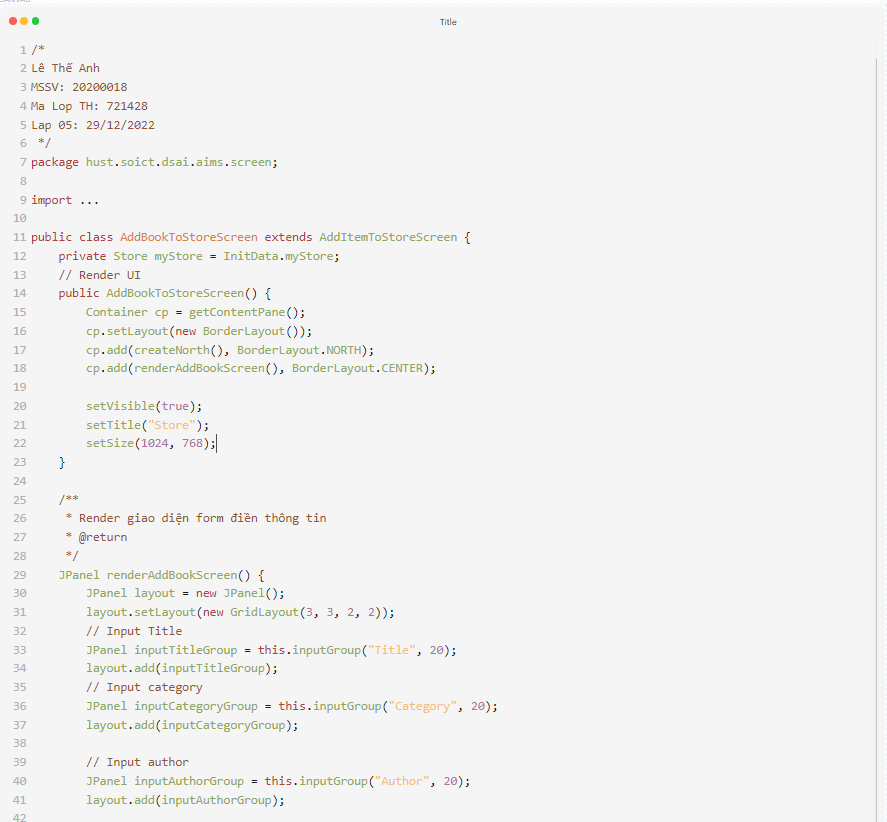


Sau khi click Add Media, sản phẩm được thêm ngay vào cửa hàng

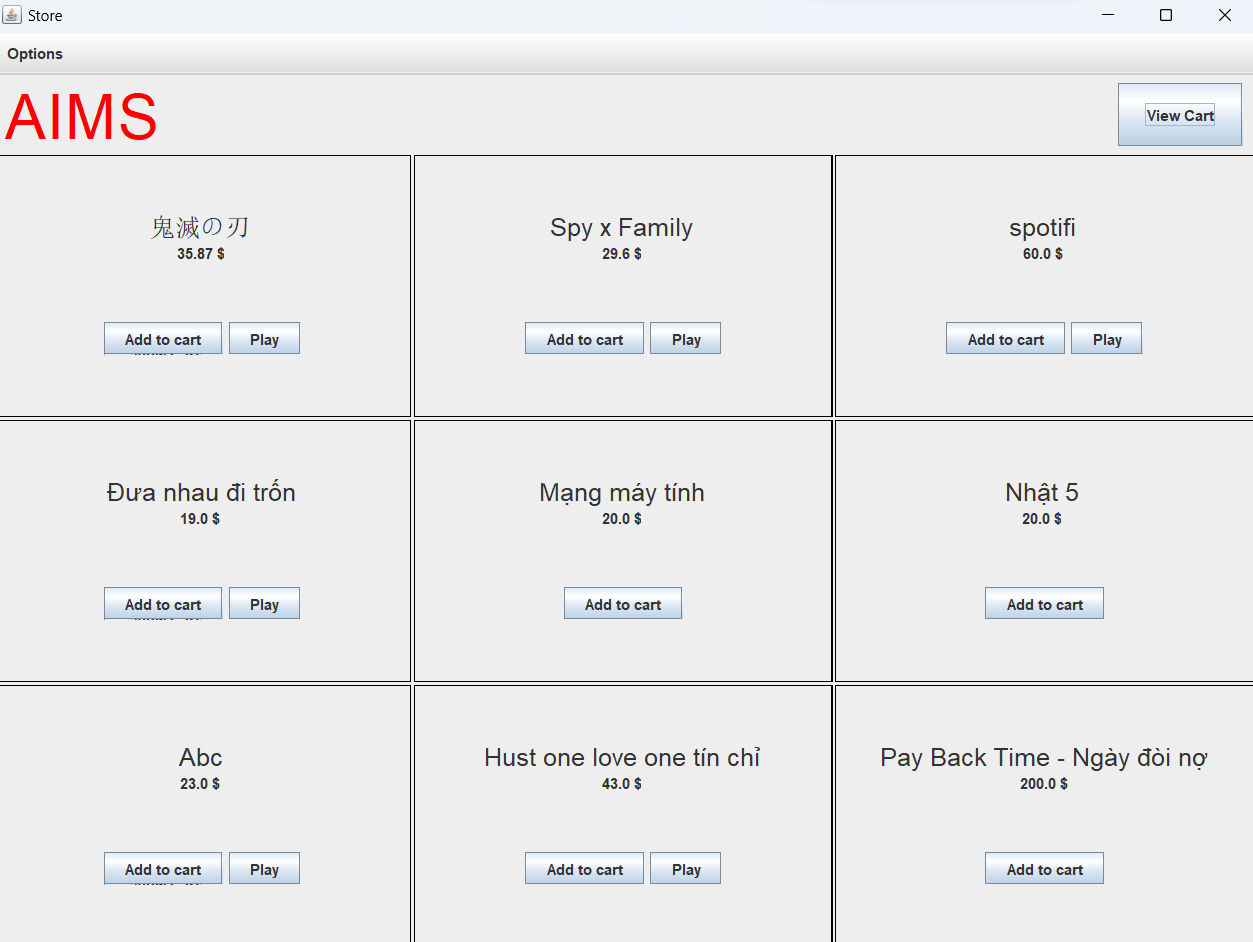


#### **10.3.4 Màn hình thêm Book**

Mã nguồn:







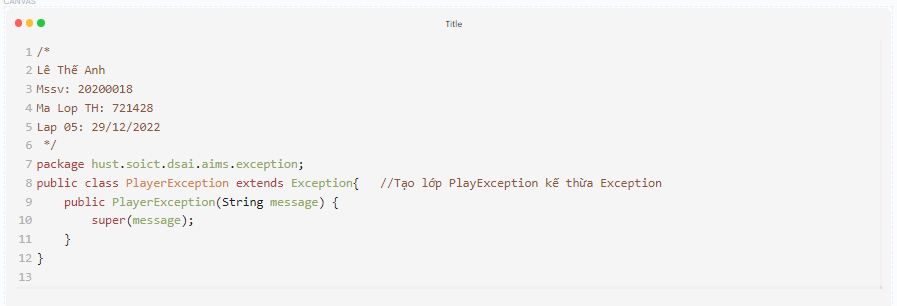
# **11. Kiểm tra tất cả các mã nguồn trước đó để nắm bắt/xử lý/ủy nhiệm các ngoại lệ runtime Exception**

## **11.1 Kiểm tra ngoại lệ khi nhập quá số lượng tối đa trong một giỏ hàng**



## **11.2 Tạo 1 class xử lý ngoại lệ Exception**

Class PlayerException



# **12. Cập nhật lại phương thức equals của class Media**

Mã nguồn:

