

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
1.1 Tạo class AWT Accumulator.....	3
1.2 Tạo class Swing Accumulator.....	4
2. Cách tổ chức Layout trong JavaSwing với Layout Managers.....	6
2.1 Tạo class Number Grid	7
3. Khởi tạo giao diện người dùng UI cho AIMS Project	8
3.1 View Store Screen	8
3.2 Triển khai xử lý sự kiện khi người dùng nhấn vào các button	15
4. JavaFX API.....	16
4.1 Xây dựng giao diện GUI sử dụng Scene Builder	16
4.2 Tính năng vẽ	19
4.3 Tính năng xóa	21
5. Setting up the View Cart Screen with ScreenBuilder	21
6. Sử dụng JavaFX trong ứng dụng Swing	22
7. Xem danh sách sản phẩm trong giỏ hàng – JavaFX’s data-driven UI	24
7.1 Tạo class CartScreenController	24
7.2 Hiện button play và remove khi 1 sản phẩm được select	28
8. Chức năng xóa sản phẩm trong giỏ hàng	30
9. Tính năng tìm kiếm sản phẩm trong giỏ hàng.....	32
10. Hoàn thiện chương trình Aims	34
10.1 Màn hình Cart Screen.....	34
10.2 Màn hình Store Screen.....	37
10.3 Màn hình Update Store Screen.....	38
11. Kiểm tra tất cả các mã nguồn trước đó để nắm bắt/xử lý/ủy nhiệm các ngoại lệ runtime Exception	48
11.1 Kiểm tra ngoại lệ khi nhập quá số lượng tối đa trong một giỏ hàng	48
11.2 Tạo 1 class xử lý ngoại lệ Exception.....	48
12. Cập nhật phương thức equals của class Media	49

1. Tạo một ứng dụng đơn giản với Swing

1.1 Tạo class AWT Accumulator

```

/*
Lê Thế Anh
MSSV: 20200018
Mã lớp thực hành: 721428
Ngày thực hành: Lap 05 - 29/12/2022
*/
package hust.soict.dsai.swing;

import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

public class AWTAccumulator extends Frame {
    private final TextField tfInput;
    private final TextField tfOutput;
    private int sum;           // Accumulate sum, init to 0

    //Constructor to set up the GUI components and event handlers
    public AWTAccumulator(){
        setLayout(new GridLayout(2,2)); //Đặt bố cục của khung là GridLayout
        add(new Label("Enter an Integer: "));

        tfInput = new TextField(10);    //Tạo thành phần TextField vào khung
        add(tfInput);    //Thêm một TextField vào khung để nhập số
        tfInput.addActionListener(new TFInputListener()); //Bắt các hành động hoặc sự kiện xảy ra

        add(new Label("The Accumulated Sum is: ")); // Thêm 1 nhãn

        tfOutput = new TextField(10);    //Tạo thành phần TextField vào khung
        tfOutput.setEditable(false);    //Đặt cờ xem phần văn bản trong TextField c được phép chỉnh
sửa không
        add(tfOutput);

        setTitle("AWT Accumulator"); //set tiêu đề cho class
        setSize(350, 120); //Đặt size
        setVisible(true);
    }

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

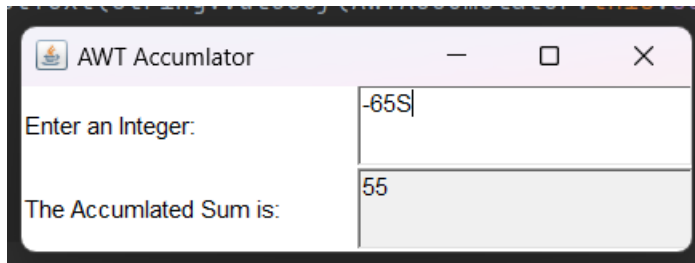
    private class TFInputListener implements ActionListener {
        // Ghi đè phương thức actionPerformed, xử lý sự kiện khi ng dùng nhấn enter trên TextField

        @Override
        public void actionPerformed(ActionEvent evt){
            int numberIn = Integer.parseInt(tfInput.getText()); // Lấy số nhập vào từ TextField

            sum += numberIn;
            tfInput.setText("");
            tfOutput.setText(String.valueOf(AWTAccumulator.this.sum)); //Hiển thị kết quả
        }
    }
}

```

Kết quả:



1.2 Tạo class Swing Accumulator

```
/*
Lê Thế Anh
MSSV: 20200018
Mã lớp thực hành: 721428
Ngày thực hành: Lap 05 - 29/12/2022
*/
package hust.soict.dsai.swing;

import hust.soict.dsai.swing.AWTAccumulator;

import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

public class SwingAccumulator extends JFrame {
    private final TextField tfInput;
    private final TextField tfOutput;
    private int sum; // Accumulate sum, init to 0

    //Trình xây dựng để thiết lập các thành phần GUI và trình xử lý sự kiện
    public SwingAccumulator(){
        Container cp = getContentPane(); // lấy contentPane
        cp.setLayout(new GridLayout(2,2)); // thiết lập layout cho contentPane
        cp.add(new Label("Enter an Integer: ")); // thêm Label vào contentPane

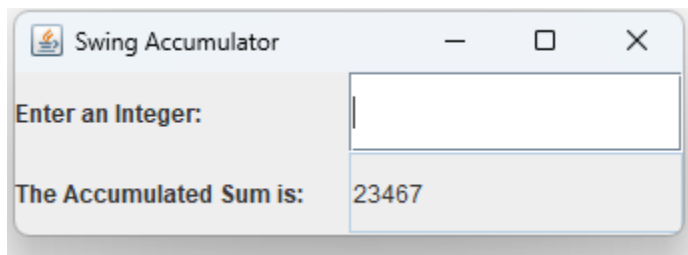
        tfInput = new TextField(10); // tạo TextField với 10 cột
        cp.add(tfInput);
        tfInput.addActionListener(new SwingAccumulator.TFInputListener()); // đăng ký sự kiện cho
        TextField
        cp.add(new Label("The Accumulate Sum is: "));

        tfOutput = new TextField(10); // tạo TextField với 10 cột
        tfOutput.setEditable(false); // không cho phép chỉnh sửa
        cp.add(tfOutput);

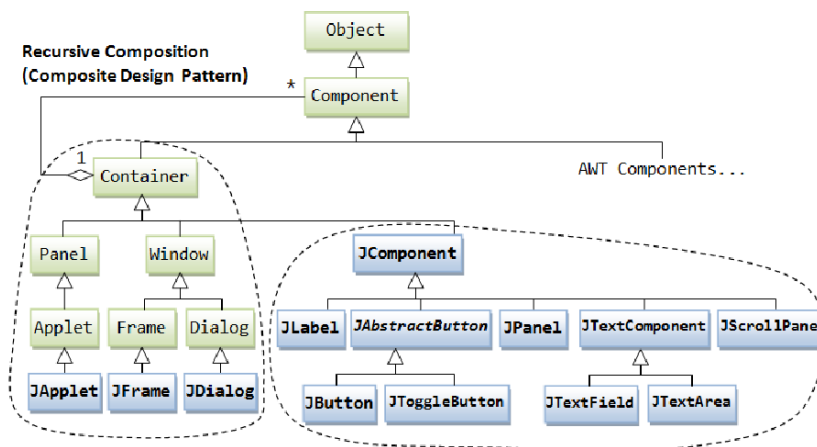
        setTitle("AWT Accumulator");
        setSize(350, 120); // thiết lập kích thước cho frame
        setVisible(true);
    }

    public static void main(String[] args){
        new AWTAccumulator();
    }
    private class TFInputListener implements ActionListener {
        // Trình xử lý sự kiện cho TextField tfInput
        @Override
        public void actionPerformed(ActionEvent evt){
            int numberIn = Integer.parseInt(tfInput.getText()); // lấy số nhập vào
            sum += numberIn;
            tfInput.setText("");
            tfOutput.setText(String.valueOf(SwingAccumulator.this.sum));
        }
    }
}
```

Kết quả



2. Cách tổ chức Layout trong JavaSwing với Layout Managers



2.1 Tạo class Number Grid

```

/*
Lê Thế Anh
MSSV: 20200018
Mã lớp thực hành: 721428
Ngày thực hành: Lap 05 - 29/12/2022
*/
package hust.soict.dsai.swing;

import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

public class NumberGrid extends JFrame {
    private final JButton[] btnNumbers = new JButton[10]; //Mảng chứa các nút số từ 0 đến 9
    private JButton btnDelete, btnReset; //Các nút xóa và reset
    private final JTextField tfDisplay;

    public NumberGrid(){

        tfDisplay = new JTextField(); //Tạo 1 TextField mới
        tfDisplay.setComponentOrientation(
            ComponentOrientation.RIGHT_TO_LEFT); //Đặt hướng của văn bản từ phải sang trái
        JPanel panelButtons = new JPanel(new GridLayout(4,3)); //Tạo 1 JPanel mới với bố cục là
        GridLayout 4 dòng 3 cột
        add(panelButtons); //Thêm panelButtons vào khung

        Container cp = getContentPane();

        cp.setLayout(new BorderLayout()); //Đặt bố cục của khung là BorderLayout
        cp.add(tfDisplay, BorderLayout.NORTH); //Thêm TextField vào phía bắc của khung
        cp.add(panelButtons, BorderLayout.CENTER); //Thêm panelButtons vào phía trung tâm

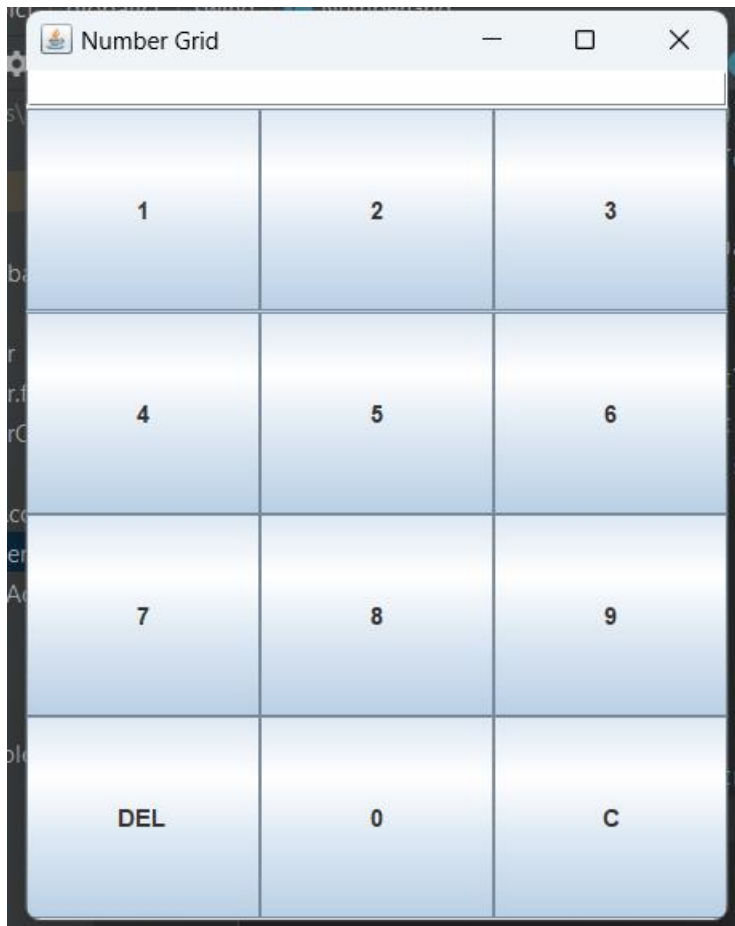
        setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE); //Đặt hành động khi nhấn nút
        setTitle("Number Grid"); //Đặt tiêu đề cho khung
        setSize(200,200); //Đặt kích thước cho khung
        setVisible(true); //Hiện thị khung
    }

    void addButtons(JPanel panelButtons){
        ButtonListener btnListener = new ButtonListener(); //Tạo 1 đối tượng ButtonListener
        for(int i = 0; i <= 9; i++){
            btnNumbers[i] = new JButton(String.valueOf(i)); //Tạo 1 nút số từ 0 đến 9
        }
    }

    private class ButtonListener implements ActionListener{
        //Ghi đè phương thức actionPerformed, xử lý sự kiện khi người dùng nhấn nút
        @Override
        public void actionPerformed (ActionEvent e){
            String button = e.getActionCommand(); //Lấy ra chuỗi mô tả của nút được nhấn
            if('0' <= button.charAt(0) && '9' >= button.charAt(0)){
                //Nếu nút được nhấn là số từ 0 đến 9
                tfDisplay.setText(tfDisplay.getText() + button); //Thêm số vào TextField
            }
            else if("DEL".equals(button)){
                //Nếu nút được nhấn là Delete
                String text = tfDisplay.getText(); //Lấy ra chuỗi hiện tại trong TextField
                if(0 < text.length()){
                    //Nếu chuỗi hiện tại không rỗng
                    tfDisplay.setText(text.substring(0, text.length() - 1)); //Xóa 1 kí tự
                }
            }
            else if("C".equals(button)){
                //Nếu nút được nhấn là c
                tfDisplay.setText(""); //Xóa hết chuỗi trong TextField
            }
        }
    }
}

```

Kết quả:



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

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

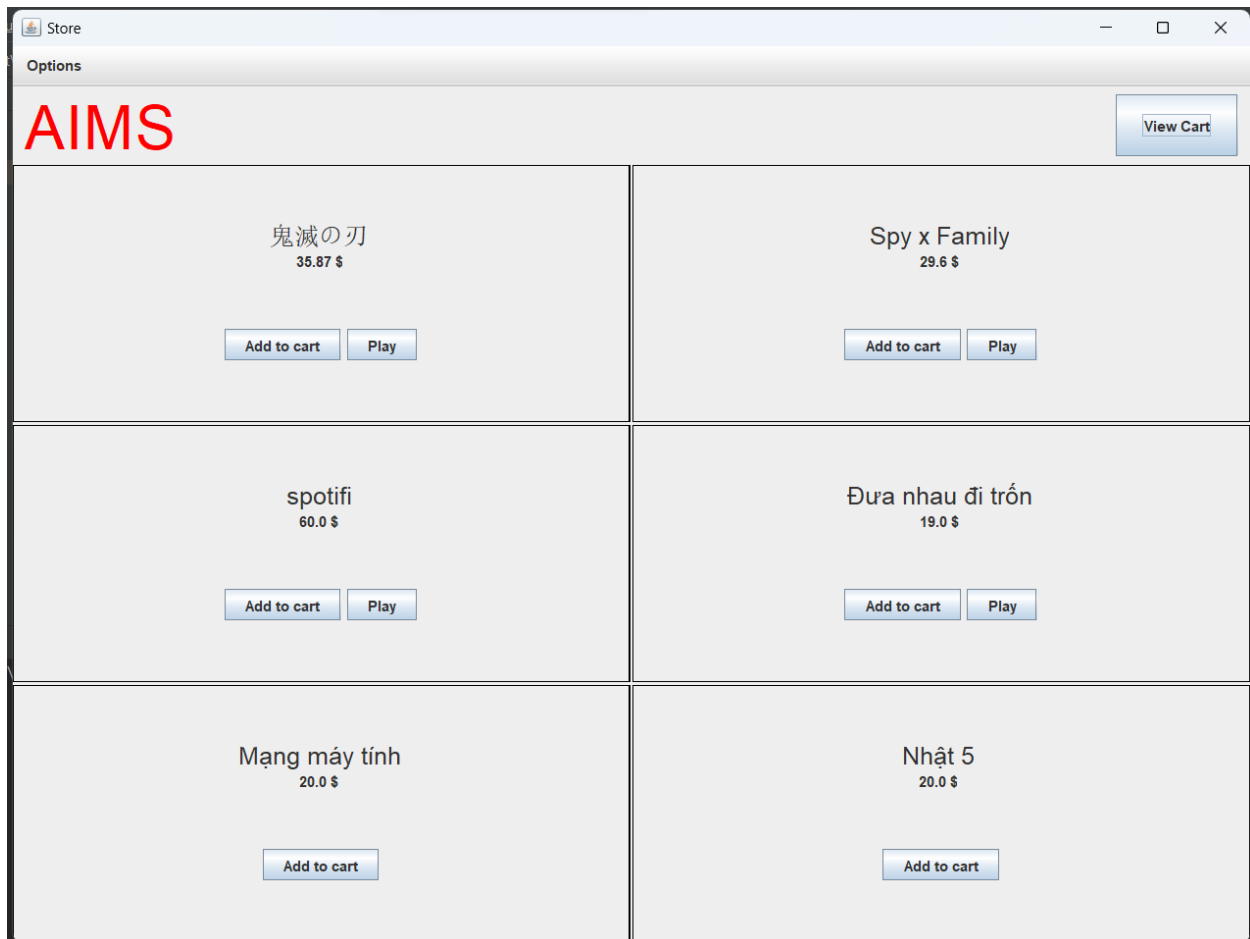
- View Store: Sử dụng Swing
- View Cart: Sử dụng JavaFX
- Update Store: Sử dụng Swing hoặc JavaFX

3.1 View Store Screen

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

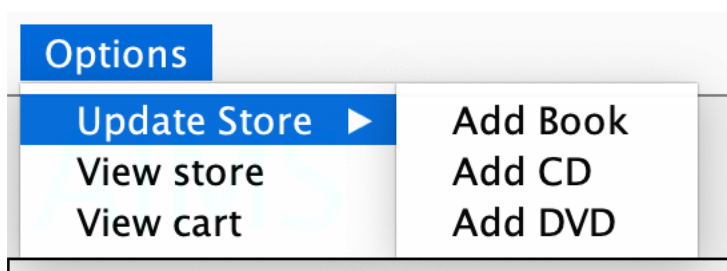
Trong thành phần NORTH sẽ có thanh menu và tiêu đề

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:



```
1 /*
2 Họ và tên: Lê Thế Anh
3 MSSV: 20200018
4 */
5 package hust.soict.dsai.aims.screen;
6 +
7 public class StoreScreen extends JFrame {
8     private Store store;
9     private Cart cart;
10    private JPanel center;
11    private static final String AddBook = "AddBook";
12    private static final String AddCD = "AddCD";
13    private static final String AddDVD = "AddDVD";
14
15    public StoreScreen(Store store, Cart cart) {
16        ...
17    }
18
19    JPanel createNorth(){ //tạo phương thức createNorth
20        JPanel north = new JPanel();
21
22        north.setLayout(new BorderLayout(north, BorderLayout.Y_AXIS)); // set layout cho north panel
23        north.add(createMenuBar()); // thêm menu bar vào north panel
24        north.add(createHeader()); // thêm header vào north panel
25        return north;
26    }
27
28    JMenuBar createMenuBar() {
29        // Tạo 1 menu bar mới
30        JMenu menu = new JMenu("Option");
31
32        JMenu smUpdateStore = new JMenu("Update Store"); // tạo 1 sub menu mới có tên là "Update Store"
33        smUpdateStore.add(new JMenuItem("Add Book")); // thêm 1 item mới có tên là "Add Book" vào sub menu
34        smUpdateStore.add(new JMenuItem("Add CD")); // thêm 1 item mới có tên là "Add CD" vào sub menu
35        smUpdateStore.add(new JMenuItem("Add DVD")); // thêm 1 item mới có tên là "Add DVD" vào sub menu
36
37        menu.add(smUpdateStore); // thêm sub menu vào menu
38        menu.add(new JMenuItem("View store")); // thêm 1 item mới có tên là "View store" vào menu
39        menu.add(new JMenuItem("View cart")); // thêm 1 item mới có tên là "View cart" vào menu
40
41        JMenuBar menuBar = new JMenuBar();
42        menuBar.setLayout(new FlowLayout(FlowLayout.LEFT)); // set layout cho menu bar
43        menuBar.add(menu); // thêm menu vào menu bar
44
45        return menuBar;
46    }
}
```

```
48  JPanel createHeader() {
49      JPanel header = new JPanel();
50      header.setLayout(new BoxLayout(header, BoxLayout.X_AXIS)); // set layout cho header panel
51
52      JLabel title = new JLabel("AIMS"); // tạo 1 label mới có tên là "AIMS"
53      title.setFont(new Font(title.getFont().getName(), Font.PLAIN, 50)); // set font plain cho label
54      title.setForeground(Color.CYAN); // set màu Cyan cho label
55
56      JButton cart = new JButton("View Cart"); // tạo 1 button mới có tên là "View Cart"
57      cart.setPreferredSize(new Dimension(100, 50)); // set kích thước cho button là 100x50
58      cart.setMaximumSize(new Dimension(100, 50)); // set kích thước tối đa cho button là 100x50
59
60      header.add(Box.createRigidArea(new Dimension(10, 10))); // tạo 1 khoảng trống 10x10
61      header.add(title); // thêm label vào header panel
62      header.add(Box.createHorizontalGlue()); // tạo 1 khoảng trống ở giữa
63      header.add(cart); // thêm button vào header panel
64      header.add(Box.createRigidArea(new Dimension(10, 10))); // tạo 1 khoảng trống 10x10
65
66      return header;
67  }
```

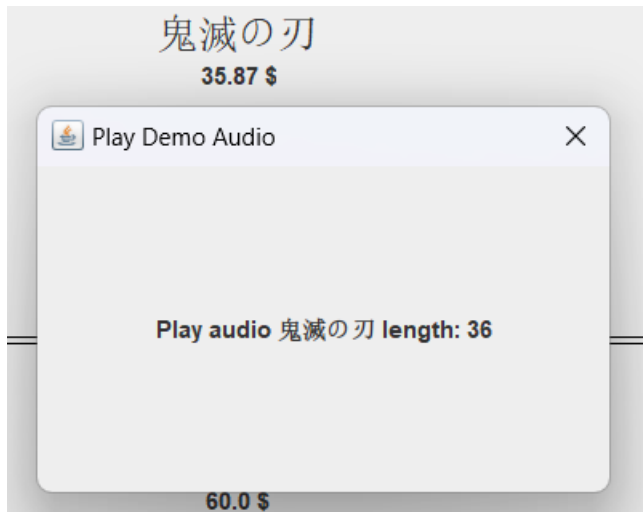
3.1.2 Tạo Center component Layout

```

15     public StoreScreen(Store store, Cart cart) {
16         this.store = store;
17         this.cart = cart;
18         Container cp = getContentPane();
19         cp.setLayout(new BorderLayout());
20         cp.add(createNorth(), BorderLayout.NORTH); // creates a panel
21         cp.add(createCenter(), BorderLayout.CENTER); // creates a panel
22
23
24         setVisible(true); // show the frame
25         setTitle("Store"); // set the title of the frame
26         setSize(1024, 768);
27         setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE); // close the frame when click
28
29         store.getItemsInStore().addListener(new ListChangeListener<Media>() {
30             @Override
31             public void onChanged(Change<? extends Media> c) {
32                 if (c.next()) {
33                     for (Media m : c.getAddedSubList()) {
34                         center.add(new MediaStore(m, cart));
35
36                     }
37                     center.revalidate();
38                 }
39             }
40         });
41     }
42 }
43
44 JPanel createCenter() {
45     center = new JPanel();
46     center.setLayout(new GridLayout(3, 3, 2, 2)); // set layout cho center panel
47
48     ObservableList<Media> mediaInStore = (ObservableList<Media>) store.getItemsInStore();
49
50     // thêm các media trong store vào center panel
51     for (int i = 0; i < 9; i++) {
52         try {
53             MediaStore cell = new MediaStore(mediaInStore.get(i), cart);
54             center.add(cell);
55         } catch (Exception ex) {
56
57         }
58     }
59     return center;
60 }
61

```

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



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

```

1 /*
2 Lê Thế Anh
3 Mssv: 20200018
4 Ma Lop TH: 721428
5 Lap 05: 29/12/2022
6 */
7 package hust.soict.dsai.aims.screen;
8
9
10 public class MediaStore extends JPanel {
11     private Media media;
12     private static final String ADD_TO_CART = "Add to cart";    //tạo một biến static final
13     private static final String PLAY = "Play"; //tạo một biến static final để lưu trữ chuỗi "Play"
14     private Cart cart;
15     public MediaStore(Media Store, Cart cart) {
16         this.media = media;
17
18         this.setLayout(new BoxLayout(this, BoxLayout.Y_AXIS));
19         // Tạo 1 title label mới
20         JLabel title = new JLabel(media.getTitle());
21         title.setFont(new Font(title.getFont().getName(), Font.PLAIN, 20)); // set font cho title
22         title.setAlignmentX(CENTER_ALIGNMENT); // căn giữa title
23
24         JLabel cost = new JLabel("" + media.getCost() + "$"); // tạo 1 label cost mới
25         cost.setAlignmentX(CENTER_ALIGNMENT); // căn giữa cost
26
27         JPanel container = new JPanel(); // tạo 1 container mới
28         container.setLayout(new FlowLayout(FlowLayout.CENTER)); // set layout cho container
29
30         JButton addToCartButton = new JButton(ADD_TO_CART); // tạo 1 button mới
31         BtnListener btnListener = new BtnListener(media);
32         addToCartButton.addActionListener(btnListener);
33         // Tạo 1 button mới có tên là "Add to cart"
34         container.add(new JButton("Add to cart"));
35         // Nếu media có thể play được thì tạo 1 button mới có tên là "Play"
36         if(media instanceof Playable) {
37             JButton playButton = new JButton(PLAY);
38             playButton.addActionListener(btnListener);
39             container.add(playButton);
40         }
41
42         this.add(Box.createVerticalGlue()); // tạo 1 khoảng trống giữa các thành phần
43         this.add(title); // thêm title vào container
44         this.add(cost); // thêm cost vào container
45         this.add(Box.createVerticalGlue()); // tạo 1 khoảng trống giữa các thành phần
46         this.add(container); // thêm container vào container
47
48         // set border cho container với màu đen và độ dày là 1
49         this.setBorder(BorderFactory.createLineBorder(Color.BLACK));
50     }

```

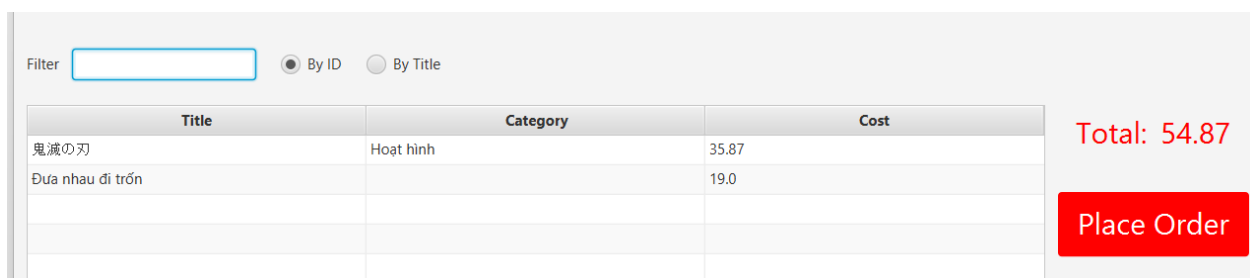
3.2 Triển khai xử lý sự kiện khi người dùng nhấn vào các button

3.2.1 Khi ấn vào nút play



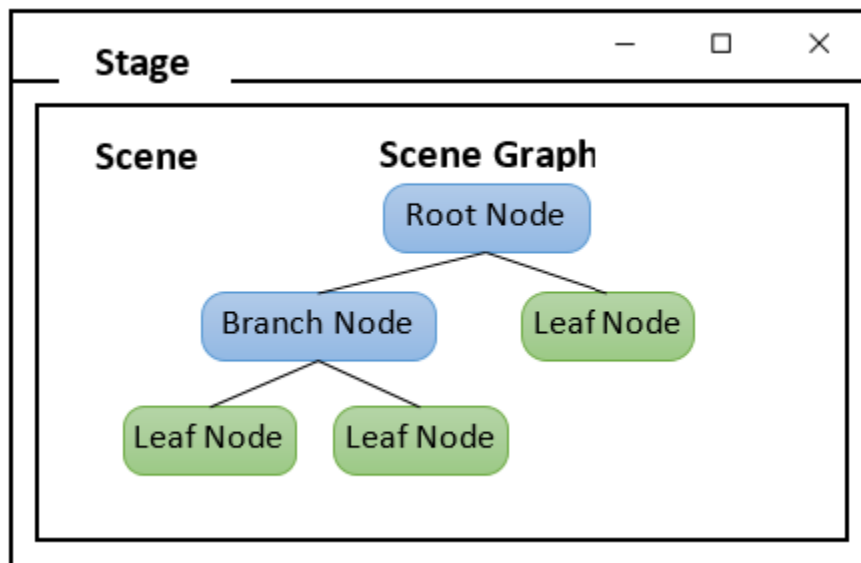
3.2.2 Khi ấn vào nút Add to cart

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



4. JavaFX API

Cấu trúc của JavaFX:

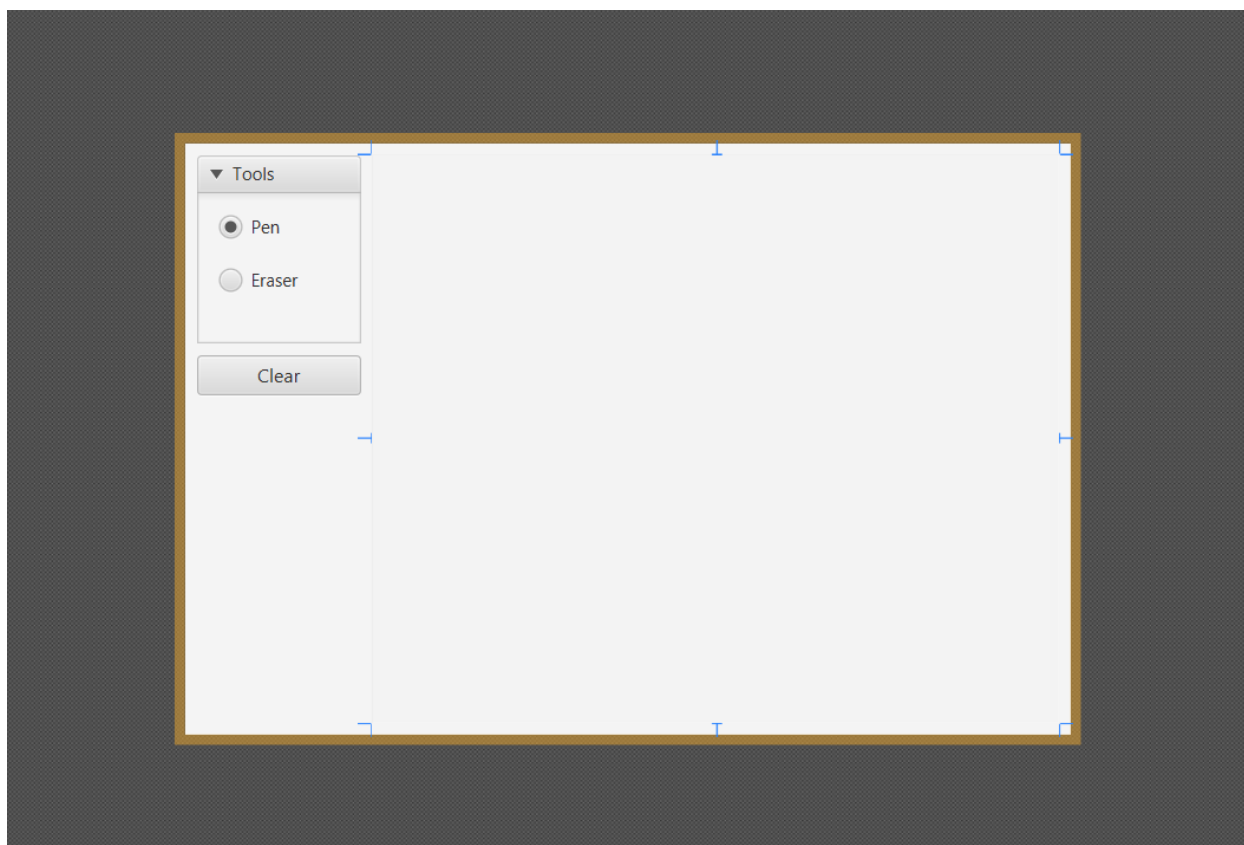


Ở phần này, chúng ta sẽ tạo một ứng dụng vẽ cơ bản (Painter):

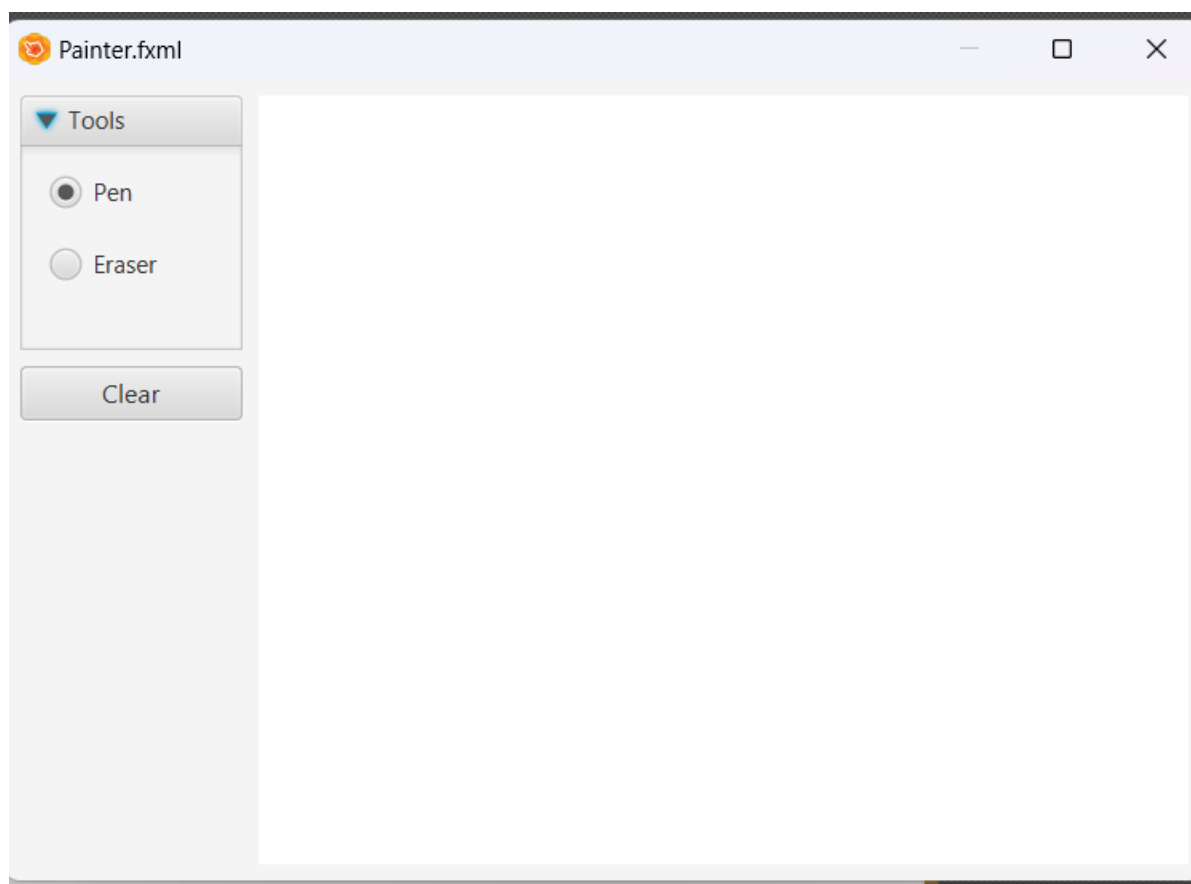
Các bước cần thực hiện như sau:

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

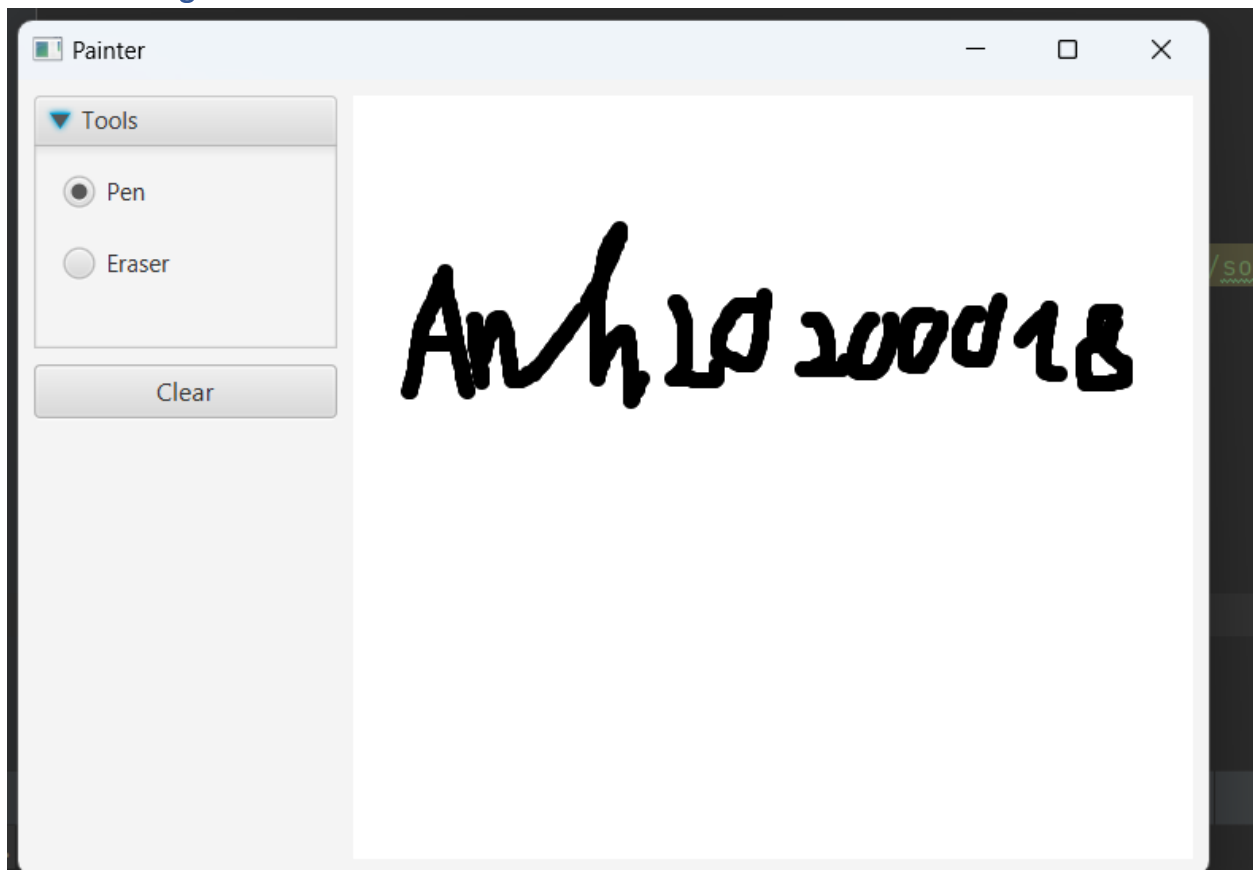
4.1 Xây dựng giao diện GUI sử dụng Scene Builder



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



4.2 Tính năng vẽ



```
1 /*
2 Lê Thế Anh
3 MSSV: 20200018
4 Mã lớp thực hành: 721428
5 Ngày thực hành: Lap 05 - 29/12/2022
6 */
7 package hust.soict.dsai.javaafx;
8
9 import javafx.event.ActionEvent;
10 import javafx.fxml.FXML;
11 import javafx.scene.control.RadioButton;
12 import javafx.scene.input.MouseEvent;
13 import javafx.scene.layout.Pane;
14 import javafx.scene.paint.Color;
15 import javafx.scene.shape.Circle;
16
17 public class PainterController {
18
19     // Khu vực vẽ
20     @FXML
21     private Pane drawingAreaPane;
22
23     // Chọn chế độ xóa
24     @FXML
25     private RadioButton radioEraserMode;
26
27     // Chọn chế độ vẽ
28     @FXML
29     private RadioButton radioPenDrawMode;
30
31     // Bút vẽ mặc định có màu đen
32     private Color currentColor = Color.BLACK;
33     @FXML
34     // Khi nhấn clear, xóa tất cả
35     void clearButtonPressed(ActionEvent event) {
36         drawingAreaPane.getChildren().clear();
37     }
38 }
```

```

39 // Sự kiện khi click chuột và di trong vùng vẽ
40 @FXML
41 void drawingAreaMouseDragged(MouseEvent event) {
42     Circle newCircle = new Circle(event.getX(), event.getY(), 4, currentColor);
43     drawingAreaPane.getChildren().add(newCircle);
44 }
45 // Đổi màu mực sang trắng nếu là xóa
46 @FXML
47 void eraserMode(ActionEvent event) {
48     currentColor = Color.WHITE;
49     System.out.println(event);
50 }
51
52 // Nếu vẽ thì đổi màu mực là đen
53 @FXML
54 void penMode(ActionEvent event) {
55     currentColor = Color.BLACK;
56     System.out.println(event);
57 }

```

4.3 Tính năng xóa

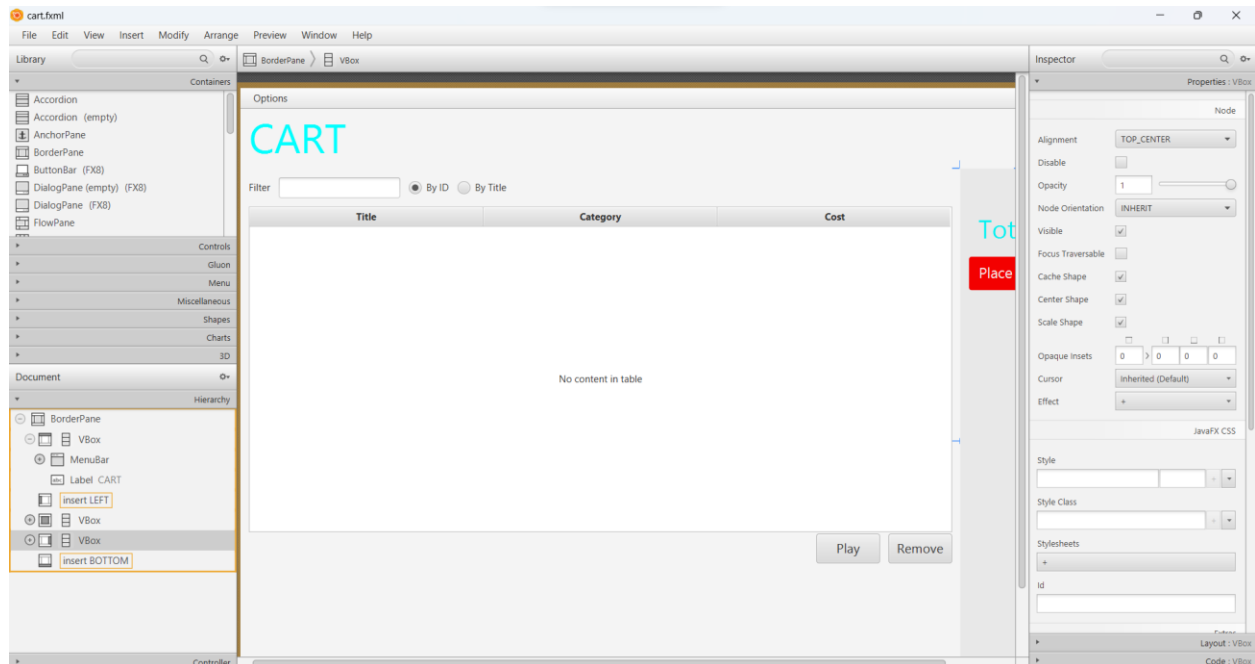
```

39 // Sự kiện khi click chuột và di trong vùng vẽ
40 @FXML
41 void drawingAreaMouseDragged(MouseEvent event) {
42     Circle newCircle = new Circle(event.getX(), event.getY(), 4, currentColor);
43     drawingAreaPane.getChildren().add(newCircle);
44 }
45 // Đổi màu mực sang trắng nếu là xóa
46 @FXML
47 void eraserMode(ActionEvent event) {
48     currentColor = Color.WHITE;
49     System.out.println(event);
50 }
51
52 // Nếu vẽ thì đổi màu mực là đen
53 @FXML
54 void penMode(ActionEvent event) {
55     currentColor = Color.BLACK;
56     System.out.println(event);
57 }

```

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:

```
1 /*
2 Lê Thế Anh
3 mssv: 20200018
4 Ma Lop TH: 721428
5 Lap 05: 29/12/2022
6 */
7 package hust.soict.dsai.aims.screen;
8 import hust.soict.dsai.aims.cart.Cart.Cart;
9 import javafx.application.Platform;
10 import javafx.embed.swing.JFXPanel;
11 import javafx.fxml.FXMLLoader;
12 import javafx.scene.Parent;
13 import javafx.scene.Scene;
14
15 import javax.swing.*;
16 import java.io.IOException;
17
18 public class CartScreen extends JFrame {
19     private JFrame frame;
20     private Cart cart;
21     public CartScreen(Cart cart, JFrame frame) {
22         super();
23
24         this.cart = cart;
25         this.frame = frame;
26         JFXPanel fxPanel = new JFXPanel(); //set up a JFXPanel in our JFrame.
27         this.add(fxPanel); //add the JFXPanel to the JFrame
28
29         this.setTitle("Cart");
30         this.setVisible(true);
31
32
33         Platform.runLater(new Runnable(){ //runLater() is a method of the Platform class. It is used to run
34             a Runnable on the JavaFX Application Thread.
35             @Override
36             public void run() {
37                 try {
38                     //load the root Node from the FXML file and create its controller object
39                     FXMLLoader loader = new FXMLLoader(getClass().getResource("cart.fxml")); //load the
40                     FXML file
41                     CartScreenController controller = new CartScreenController(cart); //create the
42                     controller object
43                     loader.setController(controller); //set the controller object to the FXML loader
44                     Parent root = loader.load(); //load the root Node from the FXML file
45                     fxPanel.setScene(new Scene(root)); //create a new Scene with the root Node and add the
46                     Scene to the JFXPanel.
47                 }catch(IOException e) { //if the FXML file cannot be loaded, print the stack trace
48                     e.printStackTrace();
49                 }
50             }
51         });
52     }
53 }
```

Kết quả

Title	Category	Cost
鬼滅の刃	Hoạt hình	35.87
Đưa nhau đi trốn		19.0

Total: 54.87

Place Order

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

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

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

7.1 Tạo class CartScreenController

Kết quả:

Title	Category	Cost
鬼滅の刃	Hoạt hình	35.87
Đưa nhau đi trốn		19.0

Total: 54.87

Place Order

Mã nguồn:


```

1 /*
2 Lê Thế Anh
3 Mssv: 20200018
4 Ma Lop TH: 721428
5 Lap 05: 29/12/2022
6 */
7 package hust.soict.dsai.aims.screen;
8
9 import hust.soict.dsai.aims.cart.Cart.Cart;
10 import hust.soict.dsai.aims.media.Media;
11 import hust.soict.dsai.aims.media.Playable;
12 import javafx.beans.value.ChangeListener;
13 import javafx.beans.value.ObservableValue;
14 import javafx.collections.FXCollections;
15 import javafx.collections.ObservableList;
16 import javafx.event.ActionEvent;
17 import javafx.fxml.FXML;
18 import javafx.scene.control.*;
19 import javafx.scene.control.cell.PropertyValueFactory;
20
21 import javax.swing.*;
22 import java.util.stream.Collectors;
23
24 public class CartScreenController {
25
26     @FXML
27     private Button btnPlay;
28
29     @FXML
30     private Button btnRemove;
31
32     @FXML
33     private TableColumn<Media, String> colMediaCost;
34
35     @FXML
36     private TableColumn<Media, String> colMediaTitle;
37
38     @FXML
39     private TableColumn<Media, String> colMediacategory;
40
41     @FXML
42     private ToggleGroup filterCategory;
43
44     @FXML
45     private TableView<Media> tblMedia;
46
47     // hàm btnRemovePressed được gọi khi nút remove được nhấn
48     @FXML
49     private void btnRemovePressed(ActionEvent event) {
50         Media media = tblMedia.getSelectionModel().getSelectedItem();
51         cart.removeMedia(media);
52         setTotalCost();
53         playingMedia.setText("Removed " + media.toString());
54     }
55
56     @FXML
57     private Label playingMedia;
58
59     @FXML
60     private RadioButton radioBtnFilterId;

```

```

61  @FXML
62  private RadioButton radioBtnFilterTitle;
63
64  private Cart cart;
65
66  private JFrame storeScreen;
67
68  public CartScreenController(Cart cart) {
69      super();
70      this.cart = cart;
71      this.storeScreen = storeScreen;
72  }
73
74  @FXML
75  private TextField tfFilter;
76  @FXML
77  private Button btnPlaceOrder;
78  @FXML
79  private Label totalCost;
80
81  @FXML
82  void showStoreScreen(ActionEvent event) {
83      storeScreen.setVisible(true);
84  }
85
86  @FXML
87  public void initialize() {
88      //tạo cột title
89      colMediaTitle.setCellValueFactory(
90          new PropertyValueFactory<Media, String>("title")
91      );
92      //tạo cột category
93      colMediacategory.setCellValueFactory(
94          new PropertyValueFactory<Media, String>("category")
95      );
96      //tạo cột cost
97      colMediaCost.setCellValueFactory(
98          new PropertyValueFactory<Media, String>("cost")
99      );
100     //tạo bảng
101     tblMedia.setItems((ObservableList<Media>) this.cart.getItemsOrdered());

```



```

103     btnPlay.setVisible(false); // ẩn nút play
104     btnRemove.setVisible(false); // ẩn nút remove
105     setTotalCost();
106
107     // lắng nghe sự kiện khi chọn 1 item trong bảng
108     tblMedia.getSelectionModel().selectedItemProperty().addListener(
109         new ChangeListener<Media>() {
110             @Override
111             public void changed(ObservableValue<? extends Media> observable, Media oldValue, Media newValue) {
112                 if (newValue != null) {
113                     updateButtonBar(newValue);
114                 }
115                 setTotalCost();
116             }
117         }
118     );

```

```

120     // lắng nghe sự kiện khi nhập vào ô filter
121     tfFilter.textProperty().addListener(new ChangeListener<String>() {
122         @Override
123         public void changed(ObservableValue<? extends String> observable, String oldValue, String newValue) {
124             showFilteredMedia(newValue);
125         }
126     });
127
128 }
129
130 // Sự kiện khi ấn vào nút Place Order
131 @FXML
132 void btnPlaceOrderPressed(ActionEvent event) {
133     cart.emptyCart();
134     tblMedia.setItems((ObservableList<Media>) cart.getItemsOrdered());
135     setTotalCost();
136     playingMedia.setText("Placed order");
137 }
138
139
140 private void setTotalCost() { // hiển thị tổng tiền
141     float cost = cart.totalCost();
142     totalCost.setText(cost + "$");
143 }
144
145 // hàm showFilterMedia giúp lọc media theo tiêu chí
146 private void showFilteredMedia(String filter) {
147     ObservableList<Media> keep = (ObservableList<Media>) this.cart.getItemsOrdered();
148     if (filterType == 0) {
149         // lọc theo id
150         tblMedia.setItems(FXCollections.observableArrayList(keep.stream().filter(m -> {
151             String id = Integer.toString(m.getId());
152             return id.contains(filter);
153         })).collect(Collectors.toList()));
154     } else {
155         // lọc theo title
156         tblMedia.setItems(FXCollections.observableArrayList(keep.stream().filter(m -> {
157             String title = m.getTitle();
158             return title.contains(filter);
159         })).collect(Collectors.toList()));
160     }
161 }

```

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.

```

1 //HỌ VÀ TÊN: Lê Thế Anh
2 //MSSV: 20200018
3 //Mã Lớp TH: 721428
4 // Lap 05: 29/12/2022
5 package hust.soict.dsai.aims.cart.Cart;
6 import ...
7
8 public class Cart {
9
10     public static final int MAX_NUMBERS_ORDERED = 20;
11
12     private ObservableList<Media> itemsOrdered = FXCollections.observableArrayList();
13     public Object getItemsOrdered();
14     public int nbItemsOrdered = itemsOrdered.size();
15
16     ...

```

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”

Filter
☒ By ID
☐ By Title

Title	Category	Cost
鬼滅の刃	Hoạt hình	35.87
Đua nhau đi trốn		19.0

Total: 54.87
Place Order

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

Filter
☒ By ID
☐ By Title

Title	Category	Cost
鬼滅の刃	Hoạt hình	35.87
Đua nhau đi trốn		19.0
Mạng máy tính	Tiếng Lòng	20.0

Total: 74.869995
Place Order

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:

```

43 //tạo bảng
44 tblMedia.setItems((ObservableList<Media>) this.cart.getItemsOrdered());
45
46 btnPlay.setVisible(false); // ẩn nút play
47 btnRemove.setVisible(false); // ẩn nút remove
48 setTotalCost();
49
50 // lắng nghe sự kiện khi chọn 1 item trong bảng
51 tblMedia.getSelectionModel().selectedItemProperty().addListener(
52     new ChangeListener<Media>() {
53         // hàm này được gọi khi chọn 1 item trong bảng
54         @Override
55         public void changed(ObservableValue<? extends Media> observable, Media oldValue, Media newValue) {
56             if (newValue != null) {
57                 updateButtonBar(newValue);
58             }
59             setTotalCost();
60         }
61     }
62 );
63
64 // lắng nghe sự kiện khi nhập vào ô filter
65 tfFilter.textProperty().addListener(new ChangeListener<String>() {
66     // hàm này được gọi khi nhập vào ô filter
67     @Override
68     public void changed(ObservableValue<? extends String> observable, String oldValue, String newValue) {
69         showFilteredMedia(newValue);
70     }
71 });

```

```

136 // hàm updateButtonBar giúp hiển thị nút play và remove
137 private void updateButtonBar(Media media) {
138     btnRemove.setVisible(true);
139     if (media instanceof Playable) {
140         btnPlay.setVisible(true);
141     } else {
142         btnPlay.setVisible(false);
143     }
144 }

```

Kết quả

Khi không có item nào được select

Filter

☒ By ID ☐ By Title

Title	Category	Cost
鬼滅の刃	Hoạt hình	35.87
Đưa nhau đi trốn		19.0
Mạng máy tính	Tiếng Lòng	20.0

Total: 74.869995

Place Order

Khi có item được Select

Filter

☒ By ID ☐ By Title

Title	Category	Cost
鬼滅の刃	Hoạt hình	35.87
Đưa nhau đi trốn		19.0
Mạng máy tính	Tiếng Lòng	20.0

Total: 74.869995

Place Order

Play

Remove

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

Filter

☒ By ID ☐ By Title

Title	Category	Cost
鬼滅の刃	Hoạt hình	35.87
Đua nhau đi trốn		19.0
Mạng máy tính	Tiếng Lòng	20.0

Total: 74.869995

Place Order

Filter

☒ By ID ☐ By Title

Title	Category	Cost
鬼滅の刃	Hoạt hình	35.87
Mạng máy tính	Tiếng Lòng	20.0

Total: 55.87 \$

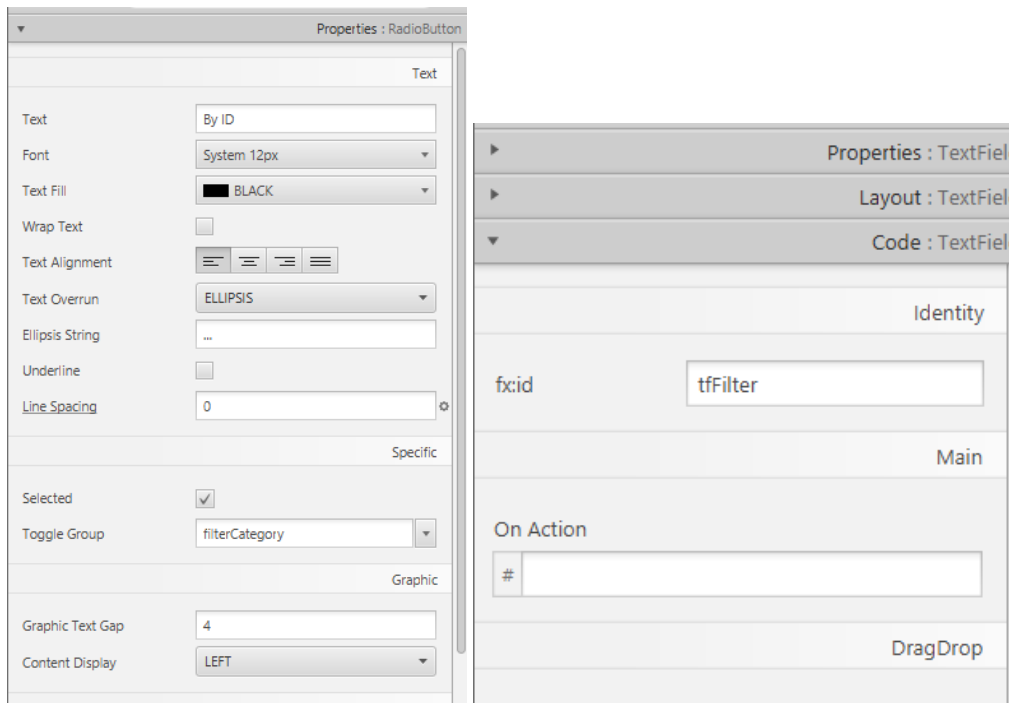
Place Order

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.


```

107 // hàm showFilterMedia giúp lọc media theo tiêu chí
108 private void showFilteredMedia(String filter) {
109     ObservableList<Media> keep = (ObservableList<Media>) this.cart.getItemsOrdered();
110     if (filterType == 0) {
111         // lọc theo id
112         tblMedia.setItems(FXCollections.observableArrayList(keep.stream().filter(m -> {
113             String id = Integer.toString(m.getId());
114             return id.contains(filter);
115         })).collect(Collectors.toList()));
116     } else {
117         // lọc theo title
118         tblMedia.setItems(FXCollections.observableArrayList(keep.stream().filter(m -> {
119             String title = m.getTitle();
120             return title.contains(filter);
121         })).collect(Collectors.toList()));
122     }
123 }
124

```

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

```

46 @FXML
47 public void initialize() {
48     ...
49
50     // lắng nghe sự kiện khi nhập vào ô filter
51     tfFilter.textProperty().addListener(new ChangeListener<String>() {
52         // hàm này được gọi khi nhập vào ô filter
53         @Override
54         public void changed(ObservableValue<? extends String> observable, String oldValue, String newValue) {
55             showFilteredMedia(newValue);
56         }
57     });
58 }
59 }
60

```

Kết quả

Danh sách sản phẩm ban đầu

Filter
☐ By ID
☒ By Title

Title	Category	Cost
鬼滅の刃	Hoạt hình	35.87
Mạng máy tính	Tiếng Lòng	20.0
Spy x Family	Hoạt hình	29.6
spotify		60.0
Nhật 5	Tiếng Nhật	20.0

Total: 165.47

Place Order

Lọc theo ID:

Filter ☒ By ID ☐ By Title

Title	Category	Cost
spotifi		60.0

Total: 165.47

Place Order

Lọc theo title:

Filter ☐ By ID ☒ By Title

Title	Category	Cost
Mạng máy tính	Tiếng Lòng	20.0

10. Hoàn thiện chương trình Aims

Danh sách các tính năng của chương trình

10.1 Màn hình Cart Screen

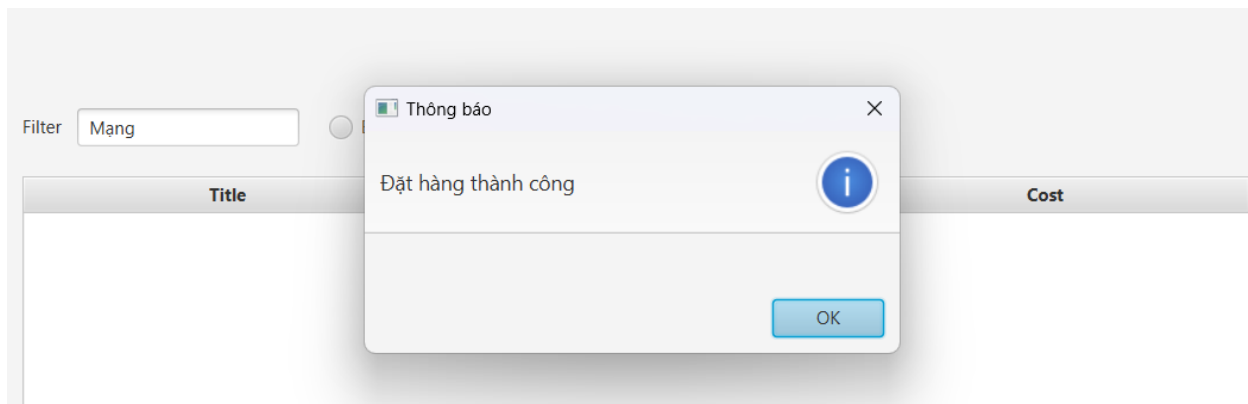
10.1.1 Create place order button

```

61 // Sự kiện khi ấn vào nút Place Order
62 @FXML
63 void btnPlaceOrderPressed(ActionEvent event) {
64     cart.emptyCart();
65     tblMedia.setItems((ObservableList<Media>) cart.getItemsOrdered());
66     setTotalCost();
67     playingMedia.setText("Placed order");
68 }
69

```

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



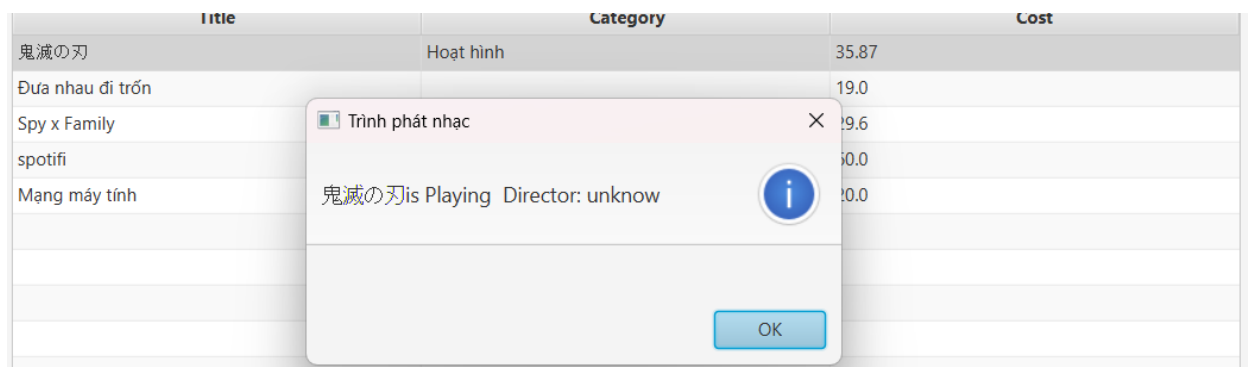
10.1.2 Create play Button

```

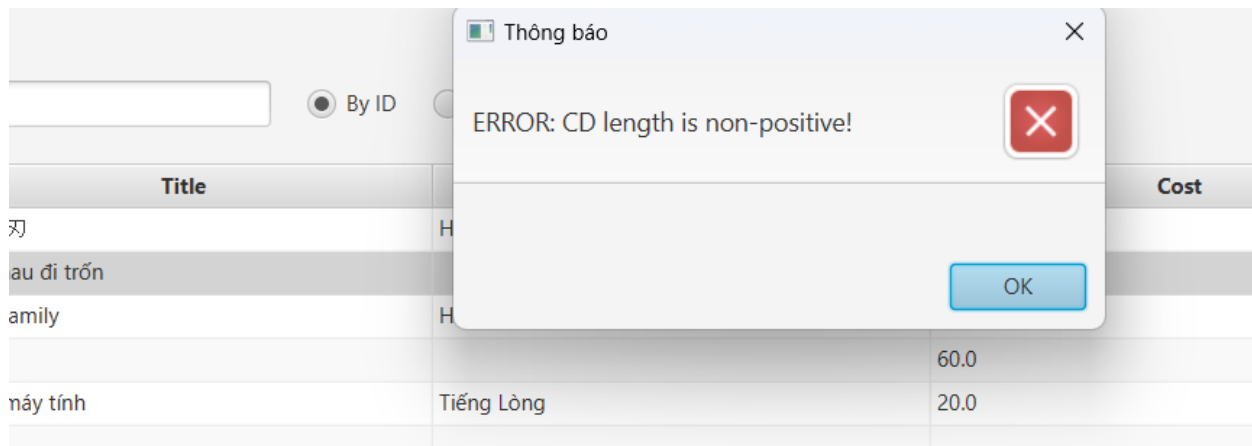
94
95 @FXML
96 private Button btnPlay;
97 // hàm btnPlayPressed được gọi khi nút play được nhấn
98 @FXML
99 void btnPlayPressed(ActionEvent event) {
100     Media media = tblMedia.getSelectionModel().getSelectedItem();
101     if (media instanceof Playable) {
102         // nếu media có thể play
103         try {
104             ((Playable) media).play();
105             playingMedia.setText("Playing " + media.toString());
106         } catch (Exception e) { // nếu có lỗi thì hiện lên thông báo
107             playingMedia.setText(e.getMessage());
108         }
109     }
110 }
111
112

```

Kết quả



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



10.1.3: Tổng tiền sản phẩm

```

69 //Tính lại tổng tiền sau khi thêm hoặc xóa media
70 private void setTotalCost() {
71     float cost = cart.totalCost();
72     totalCost.setText(cost + "$");
73 }
74

```

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.

```

86 @FXML
87 public void initialize() {
88     ...
89     tblMedia.setItems((ObservableList<Media>) this.cart.getItemsOrdered());
90
91     btnPlay.setVisible(false); // ẩn nút play
92     btnRemove.setVisible(false); // ẩn nút remove
93     setTotalCost();
94
95     // lắng nghe sự kiện khi chọn 1 item trong bảng
96     tblMedia.getSelectionModel().selectedItemProperty().addListener(
97         new ChangeListener<Media>() {
98             // hàm này được gọi khi chọn 1 item trong bảng
99             @Override
100             public void changed(ObservableValue<? extends Media> observable, Media oldValue, Media newValue) {
101                 if (newValue != null) {
102                     updateButtonBar(newValue);
103                 }
104                 setTotalCost();
105             }
106         }
107     );
108     ...
109 }
110

```

Kết quả

Trước khi thêm 1 sản phẩm:

Filter ☒ By ID ☐ By Title

Title	Category	Cost
鬼滅の刃	Hoạt hình	35.87
Đưa nhau đi trốn		19.0

Total: 54.87

Place Order

Sau khi thêm 1 sản phẩm

Filter ☒ By ID ☐ By Title

Title	Category	Cost
鬼滅の刃	Hoạt hình	35.87
Đưa nhau đi trốn		19.0
Mạng máy tính	Tiếng Lòng	20.0

Total: 74.869995

Place Order

10.2 Màn hình Store Screen

Tính năng: Add to cart

Kết quả:

Store

Options

AIMS

鬼滅の刃
35.87 \$

Add to cart

spotifi
60.0 \$

Add to cart

Mạng máy tính
20.0 \$

Add to cart

Cart

Options

CART

Filter ☒ By ID ☐ By Title

Title	Category	Cost
鬼滅の刃	Hoạt hình	35.87
Đưa nhau đi trốn		19.0
spotifi		60.0

Total: 114.869

Place Order

Source code:

```

1 // Class handle event Add to Cart
2 private class AddToCartButtonListener implements ActionListener {
3     @Override
4     public void actionPerformed(ActionEvent e) {
5         JButton button = (JButton) e.getSource();
6         String type = (String) button.getClientProperty("type"); // Get type of button
7         Media selectedMedia = (Media)button.getClientProperty("media"); // Get media of button
8         try {
9             // Thêm sản phẩm vào cart
10            InitData.myCart.addMedia(selectedMedia);
11        } catch (LimitExceededException ex) {
12            // Ném ra ngoại lệ khi số lượng sản phẩm trong cart vượt quá 10
13            JDialog dialog = new JDialog(parentFrame, "Thông báo");
14            JLabel label = new JLabel(ex.getLocalizedMessage(), SwingConstants.CENTER);
15            dialog.add(label);
16            dialog.setLocationRelativeTo(parentFrame);
17            dialog.setSize(300, 200);
18            dialog.setVisible(true);
19        }
20    }
21 }

```

10.3 Màn hình Update Store Screen

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

```

1 JMenuBar createMenuBar() {
2     // Tạo 1 menu bar mới
3     JMenu menu = new JMenu("Options");
4     JMenu smUpdateStore = new JMenu("Update Store");
5     JMenuItem addBook = new JMenuItem("Add book");
6     //Thêm addBook vào smUpdateStore
7     addBook.addActionListener(new MenuItemListener(AddBook));
8     smUpdateStore.add(addBook); // add menu item vào submenu
9     JMenuItem addCD = new JMenuItem("Add CD");
10    // thêm addCD cho menu item
11    addCD.addActionListener(new MenuItemListener(AddCD));
12    smUpdateStore.add(addCD);
13    JMenuItem addDVD = new JMenuItem("Add DVD");
14    // thêm addDVD cho menu item
15    addDVD.addActionListener(new MenuItemListener(AddDVD));
16    smUpdateStore.add(addDVD);
17
18    menu.add(smUpdateStore);
19    menu.add(new JMenuItem("View store"));
20    menu.add(new JMenuItem("View cart"));
21
22    JMenuBar menuBar = new JMenuBar();
23    menuBar.setLayout(new FlowLayout(FlowLayout.LEFT));
24    menuBar.add(menu);
25
26    return menuBar;
27 }
28

```

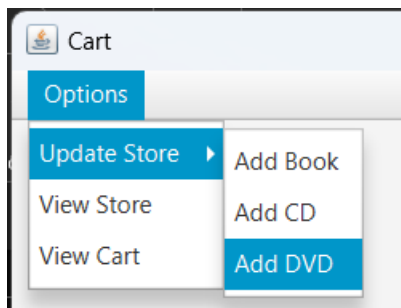
```
29 JPanel createHeader() {
30     JPanel header = new JPanel();
31     header.setLayout(new BoxLayout(header, BoxLayout.X_AXIS)); // set layout cho header panel
32
33     JLabel title = new JLabel("AIMS"); // tạo 1 label mới có tên là "AIMS"
34     title.setFont(new Font(title.getFont().getName(), Font.PLAIN, 50)); // set font plain cho label với size là 50
35     title.setForeground(Color.CYAN); // set màu Cyan cho label
36
37     JButton cart = new JButton("View Cart"); // tạo 1 button mới có tên là "View Cart"
38     cart.setPreferredSize(new Dimension(100, 50)); // set kích thước cho button là 100x50
39     cart.setMaximumSize(new Dimension(100, 50)); // set kích thước tối đa cho button là 100x50
40
41     header.add(Box.createRigidArea(new Dimension(10, 10))); // tạo 1 khoảng trống 10x10
42     header.add(title); // thêm label vào header panel
43     header.add(Box.createHorizontalGlue()); // tạo 1 khoảng trống ở giữa
44     header.add(cart); // thêm button vào header panel
45     header.add(Box.createRigidArea(new Dimension(10, 10))); // tạo 1 khoảng trống 10x10
46
47     return header;
48 }
49
50 JPanel createCenter() {
51     center = new JPanel();
52     center.setLayout(new GridLayout(3, 3, 2, 2)); // set layout cho center panel
53
54     ObservableList<Media> mediaInStore = (ObservableList<Media>) store.getItemsInStore();
55
56     // thêm các media trong store vào center panel
57     for (int i = 0; i < 9; i++) {
58         try {
59             MediaStore cell = new MediaStore(mediaInStore.get(i), cart);
60             center.add(cell);
61         } catch (Exception ex) {
62
63         }
64     }
65     return center;
66 }
```

10.3.1 Menu lựa chọn

```

1  JMenuBar createMenuBar() {
2      // Tạo 1 menu bar mới
3      JMenu menu = new JMenu("Options");
4      JMenu smUpdateStore = new JMenu("Update Store");
5      JMenuItem addBook = new JMenuItem("Add book");
6      //Thêm addBook vào smUpdateStore
7      addBook.addActionListener(new JMenuItemListener(AddBook));
8      smUpdateStore.add(addBook); // add menu item vào submenu
9      JMenuItem addCD = new JMenuItem("Add CD");
10     // thêm addCD cho menu item
11     addCD.addActionListener(new JMenuItemListener(AddCD));
12     smUpdateStore.add(addCD);
13     JMenuItem addDVD = new JMenuItem("Add DVD");
14     // thêm addDVD cho menu item
15     addDVD.addActionListener(new JMenuItemListener(AddDVD));
16     smUpdateStore.add(addDVD);
17
18     menu.add(smUpdateStore);
19     menu.add(new JMenuItem("View store"));
20     menu.add(new JMenuItem("View cart"));
21
22     JMenuBar menuBar = new JMenuBar();
23     menuBar.setLayout(new FlowLayout(FlowLayout.LEFT));
24     menuBar.add(menu);
25
26     return menuBar;
27 }
28

```



10.3.2 Màn hình thêm DVD

```

1  /*
2  Lê Thế Anh
3  MSSV: 20200018
4  Ma Lop TH: 721428
5  Lap 05: 29/12/2022
6  */
7  package hust.soict.dsai.aims.screen;
8  import ...
9
10 public class AddDigitalVideoDiscToStoreScreen extends AddItemToStoreScreen{
11     private Store myStore = InitData.myStore;
12     // Render UI
13     public AddDigitalVideoDiscToStoreScreen() {
14         // Thiết lập vị trí và kích thước
15         Container cp = getContentPane();
16         cp.setLayout(new BorderLayout());
17         cp.add(createNorth(), BorderLayout.NORTH);
18         cp.add(renderAddDVDScreen(), BorderLayout.CENTER);
19
20         setVisible(true);
21         setTitle("Store");
22         setSize(1024, 768);
23     }
24
25
26
27     JPanel renderAddDVDScreen() {
28         JPanel layout = new JPanel();
29         layout.setLayout(new GridLayout(3, 3, 2, 2));
30         // Input Title
31         JPanel inputTitleGroup = this.inputGroup("Title", 20);
32         layout.add(inputTitleGroup);
33         // Input category
34         JPanel inputCategoryGroup = this.inputGroup("Category", 20);
35         layout.add(inputCategoryGroup);
36         // Input director
37         JPanel inputDirectorGroup = this.inputGroup("Director", 20);
38         layout.add(inputDirectorGroup);
39         // Input Length
40         JPanel inputLengthGroup = this.inputGroup("Length", 20);
41         layout.add(inputLengthGroup);
42         // Input Cost
43         JPanel inputCostGroup = this.inputGroup("Cost", 20);
44         layout.add(inputCostGroup);
45
46         // Submit button
47         layout.add(submitButton(new ActionListener() {
48             @Override
49             public void actionPerformed(ActionEvent e) {
50                 // Tạo media mới
51                 DigitalVideoDisc dvd = new DigitalVideoDisc(getValueTextField(inputTitleGroup),
52                     getValueTextField(inputCategoryGroup), getValueTextField(inputDirectorGroup),
53                     Integer.valueOf(getValueTextField(inputLengthGroup)), Float.valueOf(getValueTextField(inputCostGroup)));
54                 System.out.println(dvd);
55                 //Thêm media
56                 myStore.addMedia(dvd);
57                 System.out.println(myStore.getItemsInStore());
58             }
59         }));
60         return layout;
61     }
62 }

```

Kết quả:

MS

Title	<input type="text" value="Hust one love one tin chỉ"/>	Category	<input type="text" value="music"/>
Director	<input type="text" value="Anh Thế Lê"/>	Length	<input type="text" value="99.99"/>
Cost	<input type="text" value="999"/>	<input type="button" value="Add Media"/>	

AIMS			View Cart
<div>鬼滅の刃 35.87 \$</div> <div>Add to cartPlay</div>	<div>Spy x Family 29.6 \$</div> <div>Add to cartPlay</div>	<div>spotifi 60.0 \$</div> <div>Add to cartPlay</div>	
<div>Đưa nhau đi trốn 19.0 \$</div> <div>Add to cartPlay</div>	<div>Mạng máy tính 20.0 \$</div> <div>Add to cart</div>	<div>Nhật 5 20.0 \$</div> <div>Add to cart</div>	
<div>Abc 23.0 \$</div> <div>Add to cartPlay</div>	<div>Hust one love one tín chỉ 43.0 \$</div> <div>Add to cartPlay</div>		

10.3.3 Màn hình thêm CD

Mã nguồn

```
1 /*
2 Lê Thế Anh
3 MSSV: 20200018
4 Mã lớp TH: 721428
5 Lap 05: 29/12/2022
6 */
7 package hust.soict.dsai.aims;
8 import ...
9
10 public class Aims {
11     private static JFrame currentView;
12     private static Store myStore;
13     private static Cart myCart;
14     public static void main(String[] args) throws LimitExceededException {
15         // Khởi tạo data dùng chung
16         InitData.init();
17         myStore = InitData.myStore;
18         myCart = InitData.myCart;
19         // In menu ra màn hình
20         currentView = new StoreScreen(myStore);
21     }
22
23     // Khi click vào menu, cần hiển thị những chức năng tương ứng
24     public static class MenuItemListener implements ActionListener {
25         public void actionPerformed(ActionEvent e) {
26             String command = e.getActionCommand();
27             if(command.equals("view_store")) {
28                 currentView = new StoreScreen(myStore);
29             } else if(command.equals("view_cart")) {
30                 currentView = new CartScreen(myCart);
31             } else if(command.equals("add_dvd")) {
32                 new AddDigitalVideoDiscToStoreScreen();
33             } else if(command.equals("add_cd")) {
34                 new AddCompactDiscToStoreScreen();
35             } else if(command.equals("add_book")) {
36                 new AddBookToStoreScreen();
37             }
38             System.out.println(e.getActionCommand() + " JMenuItem clicked.");
39         }
40     }
41 }
```

Kết quả

AIMS

Title Artist Director Cost Add Media

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

AIMS

View Cart鬼滅の刃
35.87 \$Add to cart PlaySpy x Family
29.6 \$Add to cart Playspotifi
60.0 \$Add to cart PlayĐưa nhau đi trốn
19.0 \$Add to cart PlayMạng máy tính
20.0 \$Add to cartNhật 5
20.0 \$Add to cartAbc
23.0 \$Add to cart PlayHust one love one tin chỉ
43.0 \$Add to cart Play

10.3.4 Màn hình thêm Book

Mã nguồn:

```
1 /*
2 Lê Thế Anh
3 MSSV: 20200018
4 Ma Lop TH: 721428
5 Lap 05: 29/12/2022
6 */
7 package hust.soict.dsai.aims.screen;
8
9 import ...
10
11 public class AddBookToStoreScreen extends AddItemToStoreScreen {
12     private Store myStore = InitData.myStore;
13     // Render UI
14     public AddBookToStoreScreen() {
15         Container cp = getContentPane();
16         cp.setLayout(new BorderLayout());
17         cp.add(createNorth(), BorderLayout.NORTH);
18         cp.add(renderAddBookScreen(), BorderLayout.CENTER);
19
20         setVisible(true);
21         setTitle("Store");
22         setSize(1024, 768);
23     }
24
25     /**
26      * Render giao diện form điền thông tin
27      * @return
28      */
29     JPanel renderAddBookScreen() {
30         JPanel layout = new JPanel();
31         layout.setLayout(new GridLayout(3, 3, 2, 2));
32         // Input Title
33         JPanel inputTitleGroup = this.inputGroup("Title", 20);
34         layout.add(inputTitleGroup);
35         // Input category
36         JPanel inputCategoryGroup = this.inputGroup("Category", 20);
37         layout.add(inputCategoryGroup);
38
39         // Input author
40         JPanel inputAuthorGroup = this.inputGroup("Author", 20);
41         layout.add(inputAuthorGroup);
42     }
```

AIMS

Title

Category

Author

Cost

Store		
Options		
AIMS		
鬼滅の刃 35.87 \$ <input type="button" value="Add to cart"/> <input type="button" value="Play"/>	Spy x Family 29.6 \$ <input type="button" value="Add to cart"/> <input type="button" value="Play"/>	spotifi 60.0 \$ <input type="button" value="Add to cart"/> <input type="button" value="Play"/>
Đưa nhau đi trốn 19.0 \$ <input type="button" value="Add to cart"/> <input type="button" value="Play"/>	Mạng máy tính 20.0 \$ <input type="button" value="Add to cart"/>	Nhật 5 20.0 \$ <input type="button" value="Add to cart"/>
Abc 23.0 \$ <input type="button" value="Add to cart"/> <input type="button" value="Play"/>	Hust one love one tín chỉ 43.0 \$ <input type="button" value="Add to cart"/> <input type="button" value="Play"/>	Pay Back Time - Ngày đòi nợ 200.0 \$ <input type="button" value="Add to cart"/>

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

```
1 // Thêm nhiều media vào giỏ hàng, đồng thời kiểm tra ngoại lệ xảy ra
2 public boolean addMedia(Media ... medias) throws LimitExceededException {
3     // Nếu vượt quá số lượng sản phẩm tối đa thì throw lỗi
4     if(itemsOrdered.size() + medias.length > MAX_NUMBERS_ORDERD) {
5         throw new LimitExceededException("Giỏ hàng đã đầy.Tối đa " + (MAX_NUMBERS_ORDERD) + " sản phẩm");
6     }
7     for(Media media: medias) {
8         if(!itemsOrdered.contains(media)) {
9             itemsOrdered.add(media);
10        } else {
11            System.out.println("Sản phẩm " + media.getTitle() + " đã tồn tại");
12        }
13    }
14
15    return true;
16 }
```

11.2 Tạo 1 class xử lý ngoại lệ Exception

Class PlayerException

```
1 /*
2 Lê Thế Anh
3 Mssv: 20200018
4 Ma Lop TH: 721428
5 Lap 05: 29/12/2022
6 */
7 package hust.soict.dsai.aims.exception;
8 public class PlayerException extends Exception{ //Tạo lớp PlayException kế thừa Exception
9     public PlayerException(String message) {
10         super(message);
11     }
12 }
13
```


12. Cập nhật phương thức equals của class Media

Mã nguồn:



```
1  @Override
2  // Ghi đè phương thức equals trong lớp Objects
3  public boolean equals(Object o) {
4      // TODO Auto-generated method stub
5      if (this == o) return true;
6      if (o == null || getClass() != o.getClass()) return false;
7      Media temp = (Media) o;
8      return title == temp.title;
9  }
10
```