

BÁO CÁO THỰC HÀNH LAB 5 LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG

Mục lục

1. Tạo ứng dụng với Swing.....	2
1.1 AWT Accumulator	2
1.2 Swing Accumulator	3
2. Tổ chức Layout.....	4
2.1 Number Grid Layout.....	4
3. Tạo UI cho AIMS Project	5
3.1 View Store Screen	6
4. JavaFX API	11
4.1 Tạo UI sử dụng Scene Builder	11
4.2 Tính năng vẽ.....	12
4.3 Tính năng xóa.....	14
5. Setting up the View Cart Screen with ScreenBuilder.....	15
6. Sử dụng JavaFX trong ứng dụng Swing	16
7. Xem danh sách sản phẩm trong giỏ hàng.....	18
7.1 Tạo class CartScreenController	18
7.2 Hiện button play và remove khi 1 sản phẩm được select	20
8. Chức năng xóa sản phẩm trong giỏ hàng.....	22
9. Tính năng lọc sản phẩm trong giỏ hàng.....	24
10. Hoàn thiện chương trình	27
10.1 Màn hình Cart Screen	27
10.2 Màn hình Store Screen.....	31
10.3 Màn hình Update Store Screen.....	32
11. Thêm Exception, hiển thị thông báo lỗi bằng giao diện	42
11.1 Hiển thị lỗi quá số lượng sản phẩm trong giỏ hàng.....	42
11.2 Hiển thị lỗi nếu độ dài media không hợp lệ khi Play	43
12. Cập nhật phương thức equals của class Media	44

1. Tạo ứng dụng với Swing

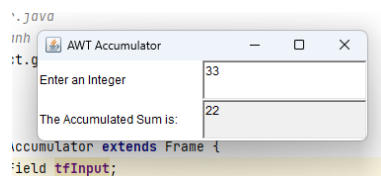
1.1 AWT Accumulator

```

4
5 import java.awt.*;
6 import java.awt.event.*;
7
8 public class AWTAccumulator extends Frame {
9     private TextField tfInput;
10    private TextField tfOutput;
11    private int sum = 0;
12    // Render UI
13    public AWTAccumulator() {
14        // Layout Grid
15        setLayout(new GridLayout(2,2));
16        // Input number
17        add(new Label("Enter an Integer"));
18        tfInput = new TextField(10);
19        add(tfInput);
20        tfInput.addActionListener(new TFInputListener());
21        // Total Sum
22        add(new Label("The Accumulated Sum is: "));
23        tfOutput = new TextField(10);
24        tfOutput.setEditable(false);
25        add(tfOutput);
26        setTitle("AWT Accumulator");
27        setSize(350, 120);
28        setVisible(true);
29    }
30
31    public static void main(String[] args) {
32        new AWTAccumulator();
33    }
34
35    // Event Listener
36    private class TFInputListener implements ActionListener {
37        @Override
38        public void actionPerformed(ActionEvent e) {
39            int numberIn = Integer.parseInt(tfInput.getText());
40            sum += numberIn;
41            tfInput.setText("");
42            tfOutput.setText(sum + "");
43        }
44    }
45
46 }

```

Kết quả



1.2 Swing Accumulator

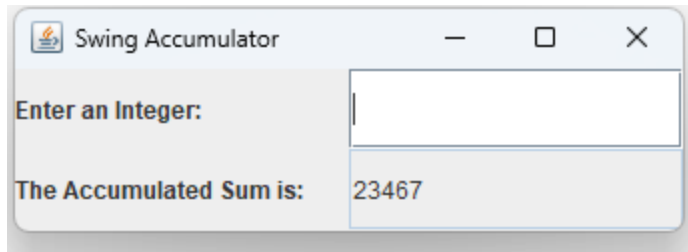
```

4
5 import javax.swing.*;
6 import java.awt.*;
7 import java.awt.event.ActionEvent;
8 import java.awt.event.ActionListener;
9
10 public class SwingAccumulator extends JFrame {
11     private JTextField tfInput;
12     private JTextField tfOutput;
13     private int sum = 0;
14
15     // Render UI
16     public SwingAccumulator() {
17         Container cp = getContentPane();
18         // Layout Grid
19         cp.setLayout(new GridLayout(2, 2));
20
21         // Input number
22         cp.add(new JLabel("Enter an Integer: "));
23         tfInput = new JTextField(10);
24         cp.add(tfInput);
25         tfInput.addActionListener(new TFInputListener());
26
27         // Total
28         cp.add(new JLabel("The Accumulated Sum is: "));
29         tfOutput = new JTextField(10);
30         cp.add(tfOutput);
31         tfOutput.setEditable(false);
32         setTitle("Swing Accumulator");
33         setSize(350, 120);
34         setVisible(true);
35     }
36
37     public static void main(String[] args) {
38         new SwingAccumulator();
39     }
40
41     // event listener
42     private class TFInputListener implements ActionListener {
43         @Override
44         public void actionPerformed(ActionEvent e) {
45             int numberIn = Integer.parseInt(tfInput.getText());
46             sum += numberIn;
47             tfInput.setText("");
48             tfOutput.setText(sum + "");
49         }
50     }
51 }

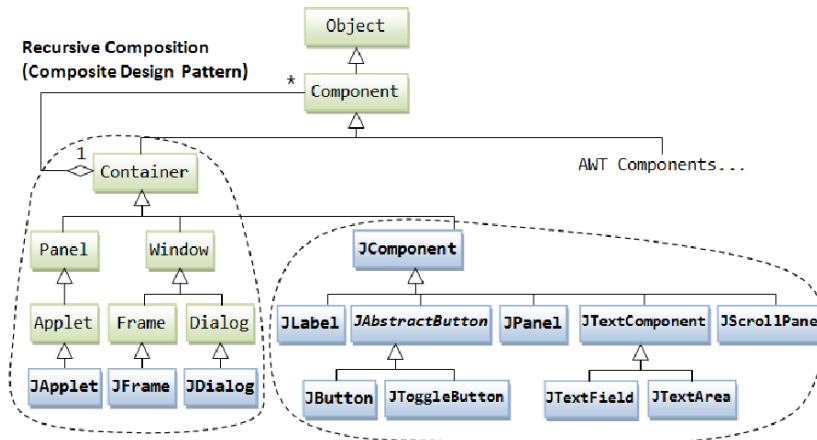
```

Kết quả

Thực hành lập trình hướng đối tượng



2. Tổ chức Layout



2.1 Number Grid Layout

Kết quả:



Mã nguồn:

Thực hành lập trình hướng đối tượng

```

4
5 import javax.swing.*;
6 import java.awt.*;
7 import java.awt.event.ActionEvent;
8 import java.awt.event.ActionListener;
9
10 public class NumberGrid extends JFrame {
11     private JButton[] btnNumbers = new JButton[10];
12     private JButton btnDelete, btnReset;
13     private JTextField tfDisplay;
14     // Render UI
15     public NumberGrid() {
16         tfDisplay = new JTextField();
17         tfDisplay.setComponentOrientation(ComponentOrientation.RIGHT_TO_LEFT);
18         // Tạo Layout Grid 4 hàng 3 cột
19         JPanel panelButtons = new JPanel(new GridLayout(4, 3));
20         addButtons(panelButtons);
21
22         Container cp = getContentPane();
23         cp.setLayout(new BorderLayout());
24         cp.add(tfDisplay, BorderLayout.NORTH);
25         cp.add(panelButtons, BorderLayout.CENTER);
26
27         setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
28         setTitle("Number Grid");
29         setSize(200, 200);
30         setVisible(true);
31     }
32
33     // Tạo các số
34     void addButtons(JPanel panelButtons) {
35         ButtonListener btnListener = new ButtonListener();
36         for(int i = 1; i <= 9; i++) {
37             btnNumbers[i] = new JButton("" + i);
38             panelButtons.add(btnNumbers[i]);
39             btnNumbers[i].addActionListener(btnListener);
40         }
41
42         // Tạo event khi click vào nút DEL
43         btnDelete = new JButton("DEL");
44         panelButtons.add(btnDelete);
45         btnDelete.addActionListener(btnListener);
46
47         // Tạo event khi click số 0
48         btnNumbers[0] = new JButton("0");
49         panelButtons.add(btnNumbers[0]);
50         btnNumbers[0].addActionListener(btnListener);
51
52         // Tạo event khi click vào nút xóa
53         btnReset = new JButton("C");
54         panelButtons.add(btnReset);
55         btnReset.addActionListener(btnListener);
56     }
57
58     // Event listener
59     private class ButtonListener implements ActionListener {
60         @Override
61         public void actionPerformed(ActionEvent e) {
62             String button = e.getActionCommand();
63             if(button.charAt(0) >= '0' && button.charAt(0) <= '9') {
64                 tfDisplay.setText(tfDisplay.getText() + button);
65             }
66             else if(button.equals("DEL")) {
67                 tfDisplay.setText("");
68             }
69             else {
70                 String currentText = tfDisplay.getText();
71                 if(currentText.length() > 0) {
72                     tfDisplay.setText(currentText.substring(0, currentText.length() - 1));
73                 }
74             }
75         }
76     }
77
78     public static void main(String[] args) {
79         new NumberGrid();
80     }
81 }
82

```

3. Tạo UI cho AIMS Project

Chúng ta sẽ có 3 màn hình

- View Store: Sử dụng Swing

Thực hành lập trình hướng đối tượng

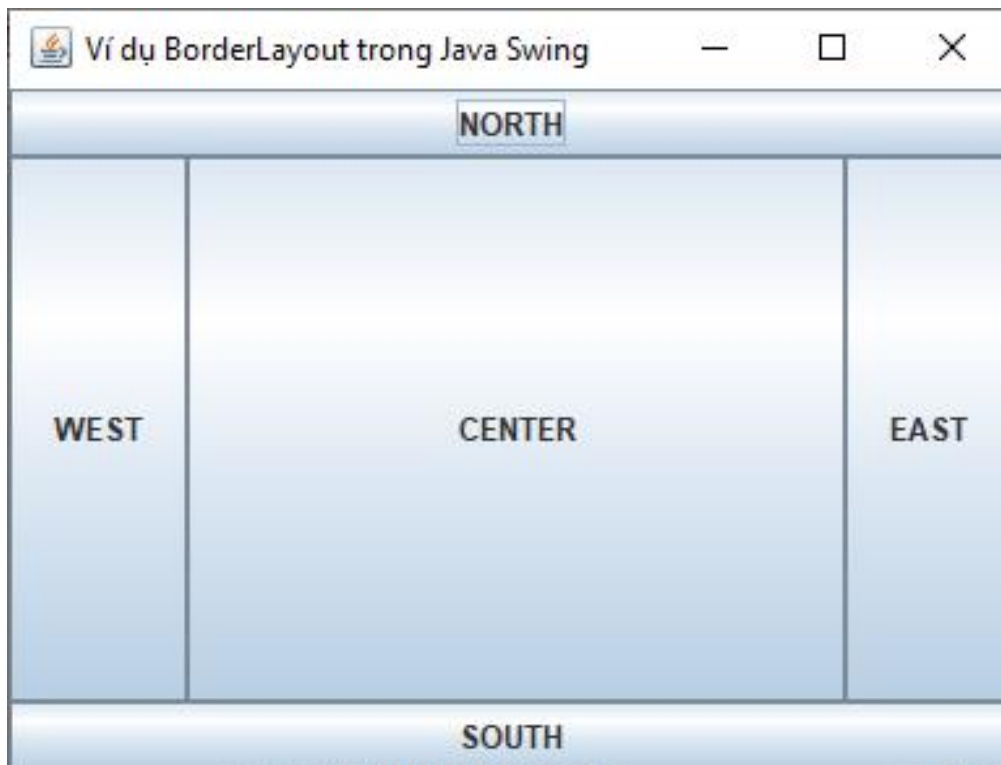
- View Cart: Sử dụng JavaFX
- Update Store: Sử dụng Swing

3.1 View Store Screen

Ở Screen này, ta sẽ sử dụng layout BorderLayout. Layout này chia cửa sổ của chúng ta ra 5 phần theo các hướng đông, tây, nam, bắc, chính giữa

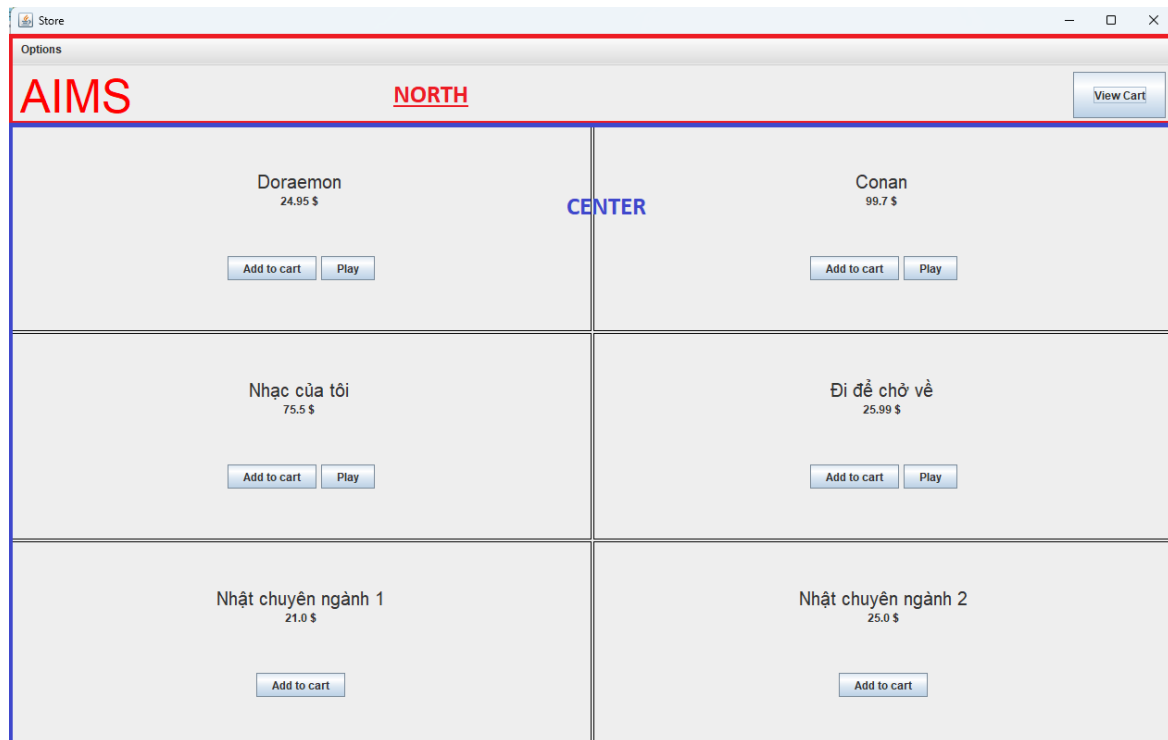
Trong ứng dụng của chúng ta, ta sẽ sử dụng 2 phần

- NORTH: header, menu
- Center: Khu vực hiển thị chính, chứa GridLayout



Kết quả

Thực hành lập trình hướng đối tượng



3.1.1 NORTH Layout

```

3 package hust.soict.dsai.aims.screen;
4
5 public class StoreScreen extends JFrame {
6     private Store store;
7     // Render UI
8     public StoreScreen(Store store) {
9         this.store = store;
10        Container cp = getContentPane();
11        cp.setLayout(new BorderLayout());
12        // North Layout
13        cp.add(createNorth(), BorderLayout.NORTH);
14
15        // Center Layout
16        cp.add(createCenter(), BorderLayout.CENTER);
17
18        setVisible(true);
19        setTitle("Store");
20        setSize(1024, 768);
21    }
22
23    // North Layout
24    JPanel createNorth() {
25        JPanel north = new JPanel();
26        north.setLayout(new BoxLayout(north, BoxLayout.Y_AXIS));
27        // Trong North có menu
28        north.add(createMenuBar());
29        // Vg Header
30        north.add(createHeader());
31        return north;
32    }
33
34    // Tao menu
35    JMenuBar createMenuBar() {
36        JMenu menu = new JMenu("Options");
37        Aims.MenuItemListener menuItemListener = new Aims.MenuItemListener();
38        JMenu smUpdateStore = new JMenu("Update Store");
39        // Add các menu con
40        JMenuItem addBookMenu = new JMenuItem("Add Book");
41        addBookMenu.setActionCommand("add_book");
42        addBookMenu.addActionListener(menuItemListener);
43        smUpdateStore.add(addBookMenu);
44
45        JMenuItem addCDMenu = new JMenuItem("Add CD");
46        addCDMenu.setActionCommand("add_cd");
47        addCDMenu.addActionListener(menuItemListener);
48        smUpdateStore.add(addCDMenu);
49
50        JMenuItem addDvdMenu = new JMenuItem("Add DVD");
51        addDvdMenu.setActionCommand("add_dvd");
52        addDvdMenu.addActionListener(menuItemListener);
53        smUpdateStore.add(addDvdMenu);
54
55        menu.add(smUpdateStore);
56
57        JMenuItem viewStoreMenu = new JMenuItem("View Store");
58        viewStoreMenu.setActionCommand("view_store");
59        viewStoreMenu.addActionListener(menuItemListener);
60        menu.add(viewStoreMenu);
61
62        JMenuItem viewCartMenu = new JMenuItem("View Cart");
63        viewCartMenu.setActionCommand("view_cart");
64        viewCartMenu.addActionListener(menuItemListener);
65        menu.add(viewCartMenu);
66
67        JMenuBar menuBar = new JMenuBar();
68
69        menuBar.setLayout(new FlowLayout(FlowLayout.LEFT));
70        menuBar.add(menu);
71        return menuBar;
72    }
73
74    // Tao Header
75    JPanel createHeader() {
76        JPanel header = new JPanel();
77        header.setLayout(new BoxLayout(header, BoxLayout.X_AXIS));
78        // Logo
79        JLabel title = new JLabel("AIMS");
80        title.setFont(new Font(title.getFont().getName(), Font.PLAIN, 50));
81        title.setForeground(Color.RED);
82
83        // Button ViewCart
84        JButton cart = new JButton("View Cart");
85        cart.setPreferredSize(new Dimension(100, 50));
86        cart.setMaximumSize(new Dimension(100, 50));
87
88        header.add(Box.createRigidArea(new Dimension(10, 10)));
89
90        header.add(title);
91        header.add(Box.createHorizontalGlue());
92        header.add(cart);
93        header.add(Box.createRigidArea(new Dimension(10, 10)));
94        return header;
95    }
96
97    JPanel createCenter() {
98
99    }
100
101
102 }

```


3.1.2 Center Layout

```

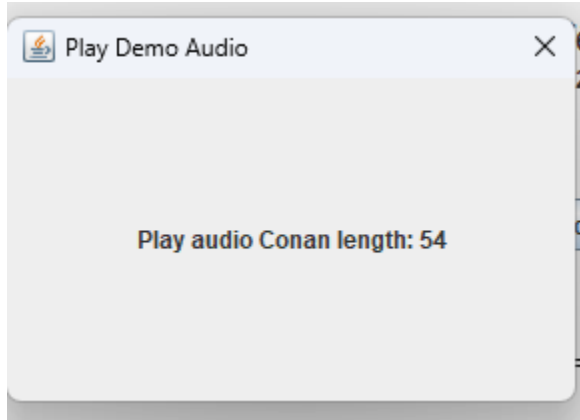
3 package hust.soict.dsai.aims.screen;
4 import hust.soict.dsai.aims.Aims;
5 import hust.soict.dsai.aims.media.Media;
6 import hust.soict.dsai.aims.store.Store;
7 import javafx.collections.ListChangeListener;
8 import javafx.collections.ObservableList;
9
10 import javax.swing.*;
11 import java.awt.*;
12
13 public class StoreScreen extends JFrame {
14     private Store store;
15     // Render UI
16     public StoreScreen(Store store) {
17         this.store = store;
18         Container cp = getContentPane();
19         cp.setLayout(new BorderLayout());
20         // North Layout
21         cp.add(createNorth(), BorderLayout.NORTH);
22
23         // Center Layout
24         cp.add(createCenter(), BorderLayout.CENTER);
25
26         setVisible(true);
27         setTitle("Store");
28         setSize(1024, 768);
29     }
30
31     // North Layout
32     JPanel createNorth() {
33         ...
34     }
35
36     // Top menu
37     JMenuBar createMenuBar() {
38         ...
39     }
40
41     // Top Header
42     JPanel createHeader() {
43         ...
44     }
45
46     // Render Center
47     JPanel createCenter() {
48         JPanel center = new JPanel();
49         // Layout Grid
50         center.setLayout(new GridLayout(3, 3, 2, 2));
51         ObservableList<Media> mediaInStore = store.getItemsInStore();
52         // Lắng nghe sự thay đổi danh sách sản phẩm để update UI
53         mediaInStore.addListener((ListChangeListener<Media>) change -> {
54             while (change.next()) {
55                 if (change.wasAdded()) {
56                     System.out.println(change.getAddedSubList().get(0)
57                         + " was added from the list!");
58                     MediaStore cell = new MediaStore(change.getAddedSubList().get(0), this);
59                     center.add(cell);
60                     center.revalidate();
61
62                 } else if (change.wasRemoved()) {
63                     System.out.println(change.getRemoved().get(0)
64                         + " was removed from the list!");
65                 }
66             }
67         });
68         // Render UI
69         for(int i = 0; i < mediaInStore.size(); i++) {
70             MediaStore cell = new MediaStore(mediaInStore.get(i), this);
71             center.add(cell);
72         }
73         return center;
74     }
75 }
76
77 }

```

Kết quả:

Khi Click vào Button Play:

Thực hành lập trình hướng đối tượng



Khi Add sản phẩm vào Cart thì update danh sách sản phẩm trong Cart

Danh sách sản phẩm trong cart trước khi Click button “Add to cart”

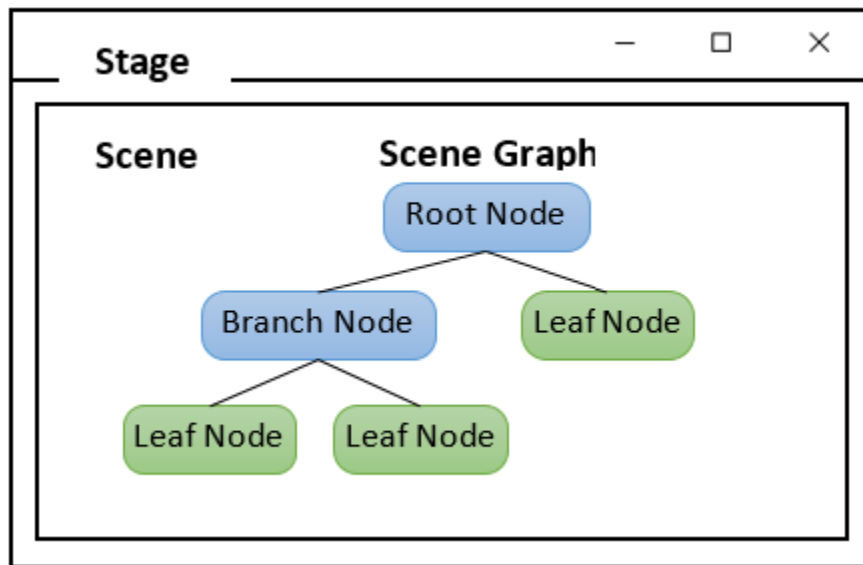
Filter	<input type="text"/>	<input checked="" type="radio"/> By ID <input type="radio"/> By Title	
Title	Category	Cost	Total: 50.940002 \$ Place Order
Doraemon	Hoạt hình	24.95	
Đi để chở về		25.99	

Danh sách sản phẩm sau khi click button “Add to cart”

Filter	<input type="text"/>	<input checked="" type="radio"/> By ID <input type="radio"/> By Title	
Title	Category	Cost	Total: 126.44 \$ Place Order
Doraemon	Hoạt hình	24.95	
Đi để chở về		25.99	
Nhạc của tôi		75.5	

Thực hành lập trình hướng đối tượng

4. JavaFX API

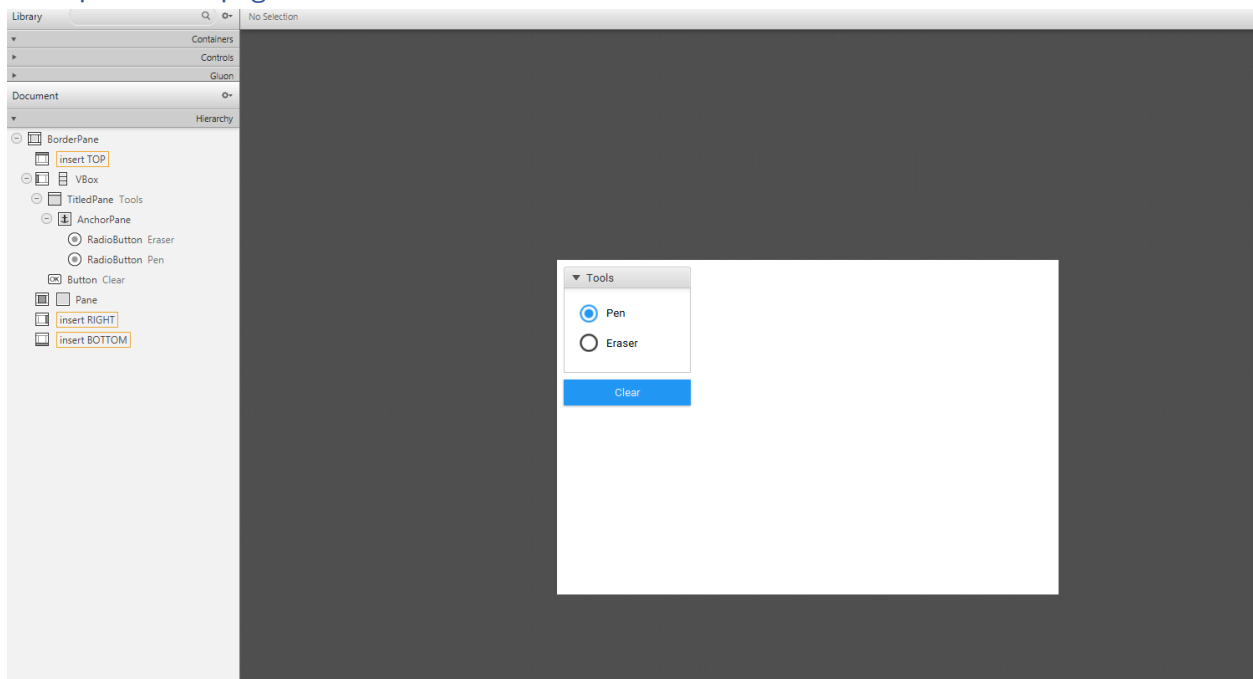


Đề bài: Tạo ứng dụng Painter

Các bước cần làm

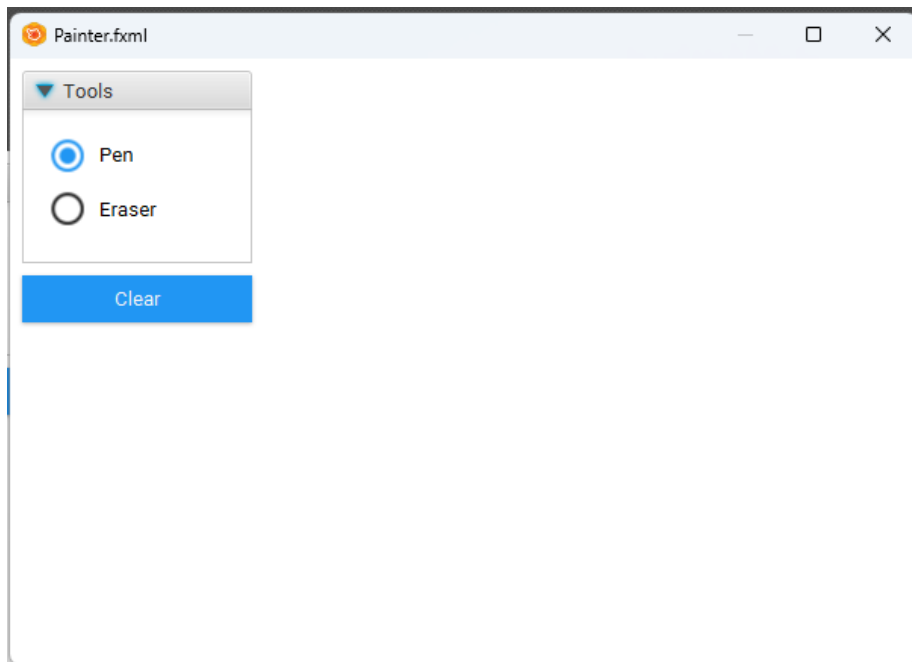
1. Vẽ UI bằng cách sử dụng Scene Builder. Kết thúc ta sẽ lưu file FXML với tên: Painter.fxml
2. Tạo Class PainterController
3. Tạo Class Painter

4.1 Tạo UI sử dụng Scene Builder



Thực hành lập trình hướng đối tượng

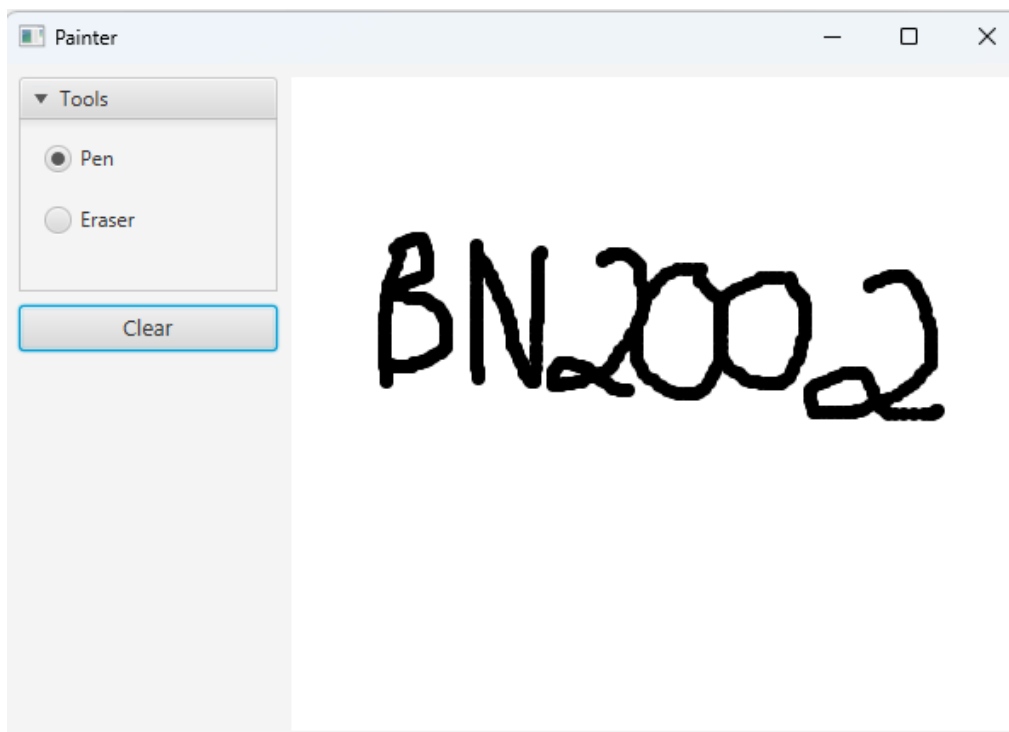
Kết quả:



4.2 Tính năng vẽ

Tính năng vẽ và xóa giống nhau. Bản chất của việc xóa là ta đổi màu mực sang màu **trắng**

Kết quả:



```
3 package hust.soict.globalict.javaafx;
4
5 import javafx.event.ActionEvent;
6 import javafx.fxml.FXML;
7 import javafx.scene.control.RadioButton;
8 import javafx.scene.input.MouseEvent;
9 import javafx.scene.layout.Pane;
10 import javafx.scene.paint.Color;
11 import javafx.scene.shape.Circle;
12
13 public class PainterController {
14
15     // Khu vực vẽ
16     @FXML
17     private Pane drawingAreaPane;
18
19     // Chọn chế độ xóa
20     @FXML
21     private RadioButton radioEraserMode;
22
23     // Chọn chế độ vẽ
24     @FXML
25     private RadioButton radioPenDrawMode;
26
27     // Bút vẽ mặc định có màu đen
28     private Color currentColor = Color.BLACK;
29     @FXML
30     // Khi nhấn clear, xóa tất cả
31     void clearButtonPressed(ActionEvent event) {
32         drawingAreaPane.getChildren().clear();
33     }
34
35     // Sự kiện khi click chuột và di trong vùng vẽ
36     @FXML
37     void drawingAreaMouseDragged(MouseEvent event) {
38         Circle newCircle = new Circle(event.getX(), event.getY(), 4, currentColor);
39         drawingAreaPane.getChildren().add(newCircle);
40     }
41     // Đổi màu mực sang trắng nếu là xóa
42     @FXML
43     void eraserMode(ActionEvent event) {
44         currentColor = Color.WHITE;
45         System.out.println(event);
46     }
47
48     // Nếu vẽ thì đổi màu mực là đen
49     @FXML
50     void penMode(ActionEvent event) {
51         currentColor = Color.BLACK;
52         System.out.println(event);
53     }
54
55 }
56
57
```

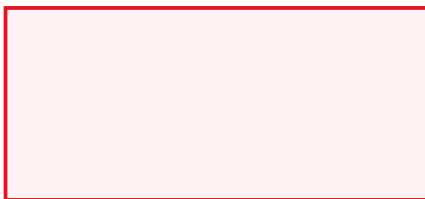
4.3 Tính năng xóa

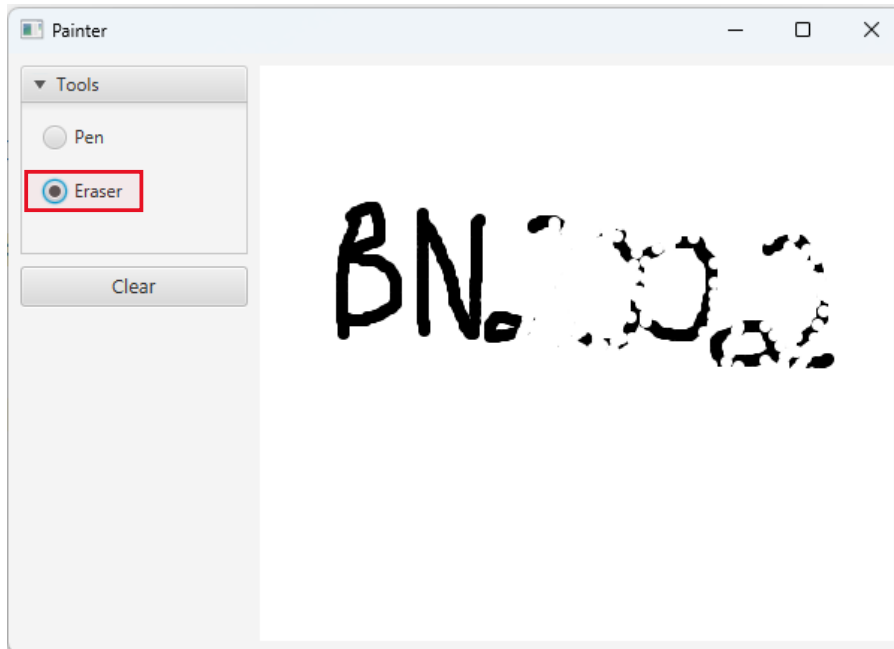
```

3 package hust.soict.globalict.javaafx;
4
5 import javafx.event.ActionEvent;
6 import javafx.fxml.FXML;
7 import javafx.scene.control.RadioButton;
8 import javafx.scene.input.MouseEvent;
9 import javafx.scene.layout.Pane;
10 import javafx.scene.paint.Color;
11 import javafx.scene.shape.Circle;
12
13 public class PainterController {
14
15     // Khu vực vẽ
16     @FXML
17     private Pane drawingAreaPane;
18
19     // Chọn chế độ xóa
20     @FXML
21     private RadioButton radioEraserMode;
22
23     // Chọn chế độ vẽ
24     @FXML
25     private RadioButton radioPenDrawMode;
26
27     // Bút vẽ mặc định có màu đen
28     private Color currentColor = Color.BLACK;
29     @FXML
30     // Khi nhấn clear, xóa tất cả
31     void clearButtonPressed(ActionEvent event) {
32         drawingAreaPane.getChildren().clear();
33     }
34
35     // Sự kiện khi click chuột và di trong vùng vẽ
36     @FXML
37     void drawingAreaMouseDragged(MouseEvent event) {
38         Circle newCircle = new Circle(event.getX(), event.getY(), 4, currentColor);
39         drawingAreaPane.getChildren().add(newCircle);
40     }
41     // Đổi màu mực sang trắng nếu là xóa
42     @FXML
43     void eraserMode(ActionEvent event) {
44         currentColor = Color.WHITE;
45         System.out.println(event);
46     }
47
48     // Nếu vẽ thì đổi màu mực là đen
49     @FXML
50     void penMode(ActionEvent event) {
51         currentColor = Color.BLACK;
52         System.out.println(event);
53     }
54 }
55 }
56
57

```

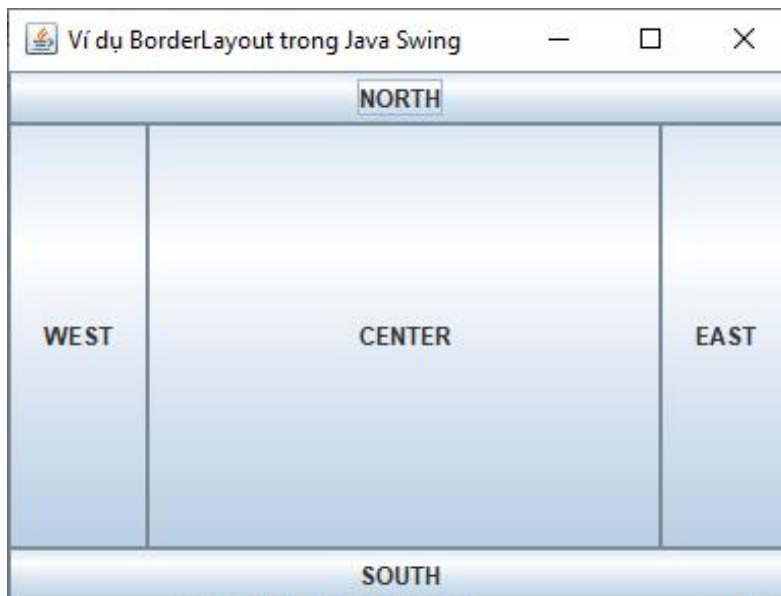
Kết quả:

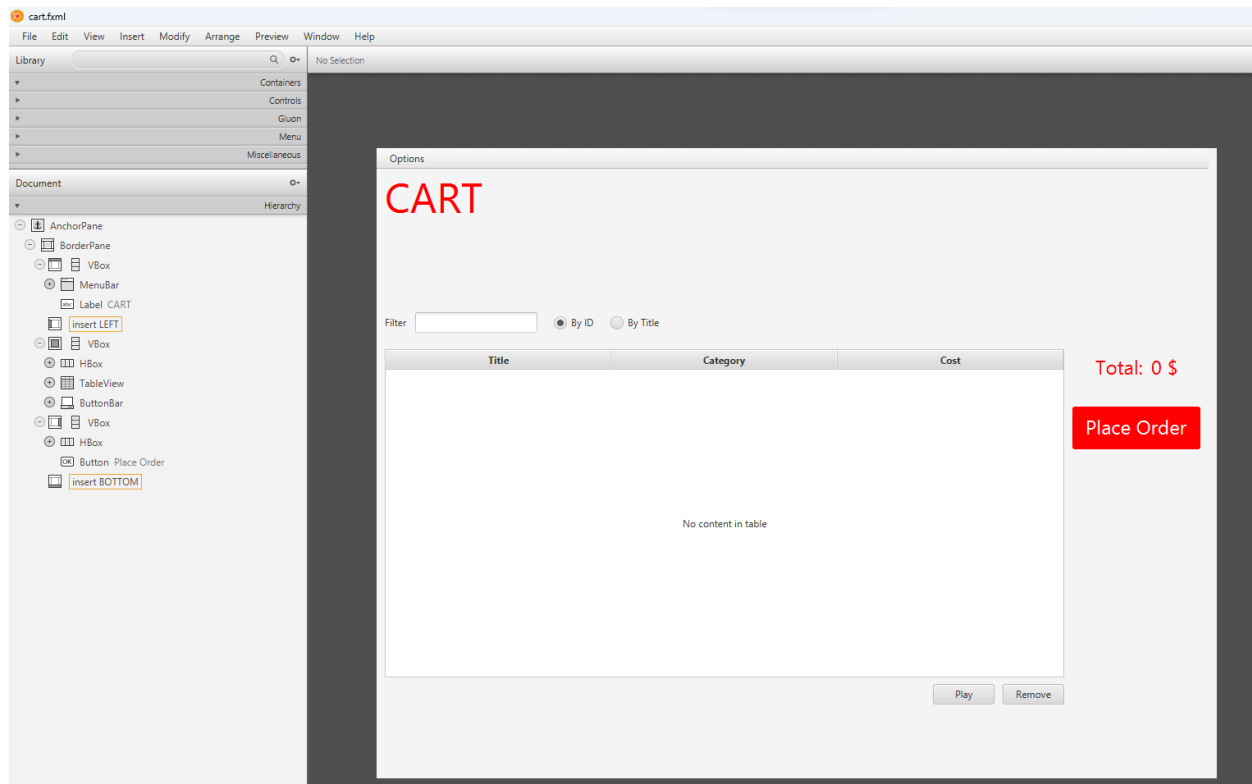




5. Setting up the View Cart Screen with ScreenBuilder

Layout sử dụng ở đây là BorderLayout. 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

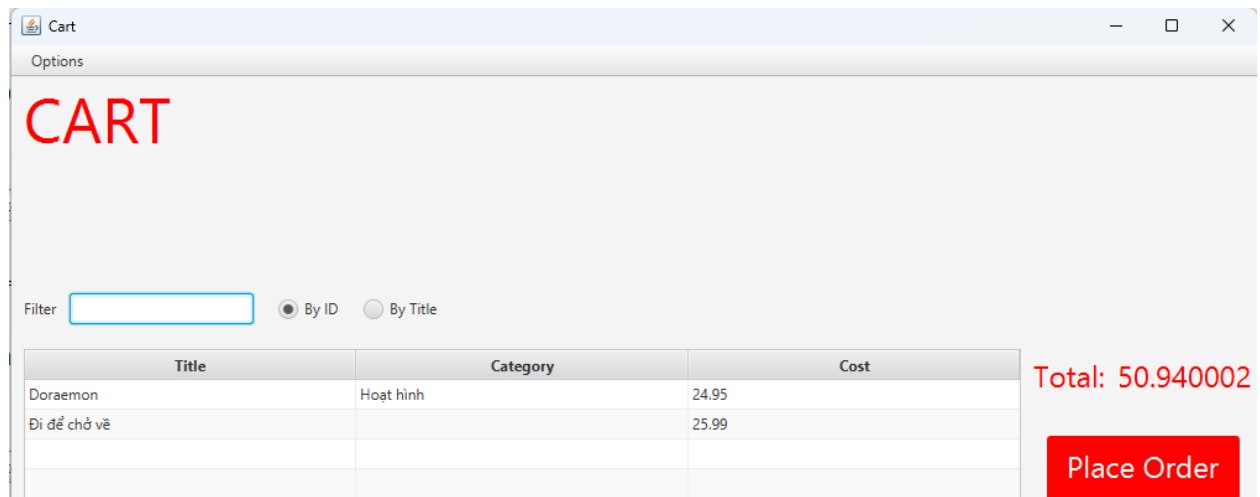
Để làm được việc này, chúng ta cần sử dụng phương thức load() của class FXMLLoader

Mã Nguồn:

Thực hành lập trình hướng đối tượng


```
3 package hust.soict.dsai.aims.screen;
4
5 import hust.soict.dsai.aims.cart.Cart;
6 import javafx.application.Platform;
7 import javafx.embed.swing.JFXPanel;
8 import javafx.fxml.FXMLLoader;
9 import javafx.scene.Parent;
10 import javafx.scene.Scene;
11
12 import javax.swing.*;
13 import java.io.IOException;
14
15 public class CartScreen extends JFrame {
16     private Cart cart;
17     // Render UI
18     public CartScreen(Cart cart) {
19         super();
20         // Set các tham số của cửa sổ: title, size
21         this.cart = cart;
22         JFXPanel fxPanel = new JFXPanel();
23         this.add(fxPanel);
24
25         this.setTitle("Cart");
26         this.setSize(1024, 768);
27         this.setVisible(true);
28         // gọi load của FXMLLoader để sử dụng JavaFX trong ứng dụng Swing
29         Platform.runLater(new Runnable() {
30             @Override
31             public void run() {
32                 try {
33                     FXMLLoader loader = new FXMLLoader(getClass().getResource("./cart.fxml"));
34                     CartScreenController controller = new CartScreenController(cart);
35                     loader.setController(controller);
36                     Parent root = loader.load();
37                     fxPanel.setScene(new Scene(root));
38                 } catch (IOException e) {
39                     e.printStackTrace();
40                 }
41             }
42         });
43     }
44
45 }
46
```

Kết quả



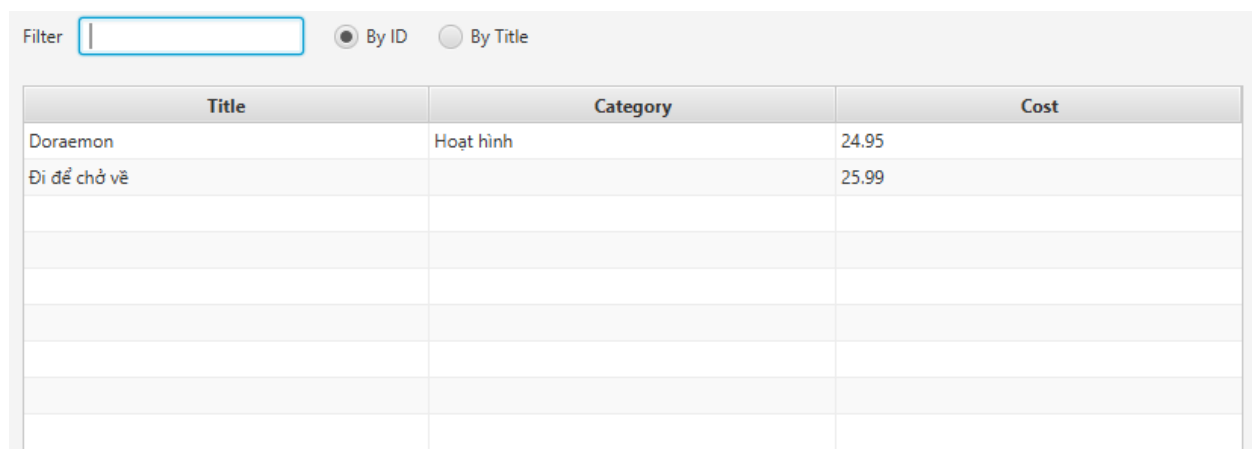
7. Xem danh sách sản phẩm trong giỏ hàng

Trước tiên ta cần set fx:id cho một số cột trong TableView

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

7.1 Tạo class CartScreenController

Kết quả:



Mã nguồn:

```

3 package hust.soict.dsai.aims.screen;
4 import hust.soict.dsai.aims.cart.Cart;
5 import hust.soict.dsai.aims.exception.PlayerException;
6 import hust.soict.dsai.aims.media.Disc;
7 import hust.soict.dsai.aims.media.Media;
8 import hust.soict.dsai.aims.media.Playable;
9 import javafx.application.Platform;
10 import javafx.beans.value.ChangeListener;
11 import javafx.beans.value.ObservableValue;
12 import javafx.collections.ListChangeListener;
13 import javafx.collections.transformation.FilteredList;
14 import javafx.event.ActionEvent;
15 import javafx.fxml.FXML;
16 import javafx.scene.control.*;
17 import javafx.scene.control.cell.PropertyValueFactory;
18
19
20 public class CartScreenController {
21     private Cart cart;
22
23     @FXML
24     private TableColumn<Media, Float> colMediaCost;
25
26     @FXML
27     private TableColumn<Media, String> colMediaTitle;
28
29     @FXML
30     private TableColumn<Media, String> colMediaCategory;
31
32     @FXML
33     private TableView<Media> tblMedia;
34
35     @FXML
36     private Button btnPlay;
37
38     @FXML
39     private Button btnRemove;
40
41     @FXML
42     private Button btnPlaceOrder;
43     @FXML
44     private TextField tfFilter;
45
46     @FXML
47     private ToggleGroup filterCategory;
48
49     @FXML
50     private RadioButton radioBtnFilterId;
51
52     @FXML
53     private RadioButton radioBtnFilterTitle;
54
55     @FXML
56     private Label totalCostLabel;
57
58     private FilteredList<Media> filteredList;
59
60     public CartScreenController(Cart cart) {
61         super();
62         this.cart = cart;
63         this.filteredList = new FilteredList<Media>(cart.getItemsOrdered());
64     }
65
66     // Khởi tạo các event
67     @FXML
68     private void initialize() {
69         // Fill dữ liệu tương ứng với các cột trong bảng
70         colMediaCategory.setCellValueFactory(
71             new PropertyValueFactory<Media, String>("category")
72         );
73
74         colMediaTitle.setCellValueFactory(
75             new PropertyValueFactory<Media, String>("title")
76         );
77
78         colMediaCost.setCellValueFactory(
79             new PropertyValueFactory<Media, Float>("cost")
80         );
81
82         //Đổ dữ liệu ra bảng để hiển thị
83         tblMedia.setItems(filteredList);
84     }
85
86
87
88 }
89

```

Ngoài ra để nhận biết được sự thay đổi của các sản phẩm trong giỏ hàng, ta cần chuyển từ việc sử dụng ArrayList sang ObservableList trong class Cart

Thực hành lập trình hướng đối tượng

```

3 package hust.soict.dsai.aims.cart;
4 import hust.soict.dsai.aims.media.Media;
5 import javafx.collections.FXCollections;
6 import javafx.collections.ObservableList;
7
8 import javax.naming.LimitExceededException;
9
10 public class Cart {
11     public static final int MAX_NUMBERS_ORDERD = 20;
12     // Sử dụng kiểu dữ liệu ObservableList để ta có thể lắng nghe được việc thay đổi dữ liệu
13     private ObservableList<Media> itemsOrdered = FXCollections.observableArrayList();
14     ...
15
16 }
17

```

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”

Title	Category	Cost
Doraemon	Hoạt hình	24.95
Đi để chở về		25.99

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

Title	Category	Cost
Doraemon	Hoạt hình	24.95
Đi để chở về		25.99
Conan	Hoạt hình	99.7

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

```

3 package hust.soict.dsai.aims.screen;
4
5 public class CartScreenController {
6     ...
7
8     @FXML
9     private TableView<Media> tblMedia;
10
11     @FXML
12     private Button btnPlay;
13
14     @FXML
15     private Button btnRemove;
16
17     ...
18
19     public CartScreenController(Cart cart) {
20         super();
21         this.cart = cart;
22         ...
23     }
24
25     // Khởi tạo các event
26     @FXML
27     private void initialize() {
28         ...
29
30         // Mặc định ban đầu là ẩn nút Play và Remove đi
31         btnPlay.setVisible(false);
32         btnRemove.setVisible(false);
33
34         // Ẩn hiện nút play, remove khi ở 1 dòng được select
35         tblMedia.getSelectionModel().selectedItemProperty().addListener(
36             new ChangeListener<Media>() {
37                 @Override
38                 public void changed(ObservableValue<? extends Media> observable, Media oldValue, Media newValue) {
39                     if(newValue != null) {
40                         // Nếu có item được select thì hiện chúng lên
41                         updateButtonBar(newValue);
42                     } else {
43                         //Không có item nào được select thì ẩn cả 2 nút đi
44                         btnPlay.setVisible(false);
45                         btnRemove.setVisible(false);
46                     }
47                 }
48             }
49         );
50
51     }
52
53     ...
54     /**
55     * Hàm hỗ trợ điều khiển việc ẩn hiện của button Play và Remove
56     * @param media
57     */
58
59     void updateButtonBar(Media media) {
60         btnRemove.setVisible(true);
61         if(media instanceof Playable) {
62             btnPlay.setVisible(true);
63         } else {
64             btnPlay.setVisible(false);
65         }
66     }
67
68     ...
69
70
71 }
72

```

Kết quả

Khi không có item nào được select

Thực hành lập trình hướng đối tượng

[illegible]

Khi có item được Select

Title	Category	Cost
Doraemon	Hoạt hình	24.95
Đi để chở về		25.99
Conan	Hoạt hình	99.7

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

Thực hành lập trình hướng đối tượng

```

3 package hust.soict.dsai.aims.screen;
4 public class CartScreenController {
5     ...
6
7     @FXML
8     private Button btnRemove;
9     ...
10
11     public CartScreenController(Cart cart) {
12         super();
13         this.cart = cart;
14     }
15     ...
16     /**
17      * Handle event khi bấm vào nút xóa trong giỏ hàng
18      * @param event
19      */
20     @FXML
21     void btnRemovePressed(ActionEvent event) {
22         // Lấy Item được chọn
23         Media media = tblMedia.getSelectionModel().getSelectedItem();
24         // Xóa
25         cart.removeMedia(media);
26     }
27
28     ...
29 }
30

```

Kết quả

Trước khi xóa

Title	Category	Cost
Doraemon	Hoạt hình	24.95
Đi để chở về		25.99
Conan	Hoạt hình	99.7

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

Thực hành lập trình hướng đối tượng

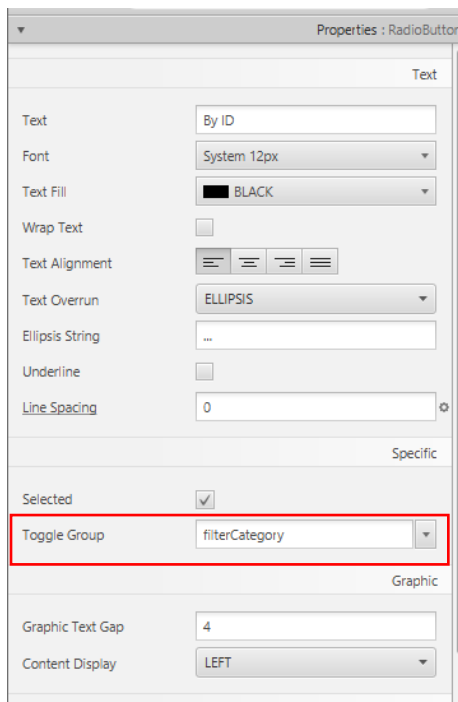
Title	Category	Cost
Doraemon	Hoạt hình	24.95
Conan	Hoạt hình	99.7

9. Tính năng lọc sản phẩm trong giỏ hàng

Chúng ta cần các component sau để thực hiện

- 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

Thực hành lập trình hướng đối tượng


```

3 package hust.soict.dsai.aims.screen;
4
5 public class CartScreenController {
6     ...
7
8     // Chứa danh sách các sản phẩm lọc theo điều kiện
9     private FilteredList<Media> filteredList;
10    ...
11
12
13 }
14

```

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

```

3 package hust.soict.dsai.aims.screen;
4
5 public class CartScreenController {
6     ...
7
8     /**
9      * Handle việc lọc dữ liệu
10     * @param filter
11     */
12     void showFilterMedia(String filter) {
13         filteredList.setPredicate(item -> {
14             if(filter.isEmpty()) {
15                 return true;
16             }
17             try {
18                 // Nếu lọc theo ID
19                 if(radioBtnFilterId.isSelected()) {
20                     return String.valueOf(item.getId()).equals(filter);
21                 } else if(radioBtnFilterTitle.isSelected()) {
22                     // Nếu lọc theo title
23                     return item.getTitle().contains(filter);
24                 }
25             } catch (Exception e) {
26                 e.printStackTrace();
27             }
28
29             return true;
30         });
31     }
32
33 }
34
35
36 }
37

```

Tiếp đó, ta sẽ handle event khi người dùng gõ vào ô input keyword

```

3 package hust.soict.dsai.aims.screen;
4
5 public class CartScreenController {
6     ...
7     @FXML
8     private TextField tfFilter;
9     ...
10
11     // Chứa danh sách các sản phẩm lọc theo điều kiện
12     private FilteredList<Media> filteredList;
13
14     public CartScreenController(Cart cart) {
15         super();
16         this.cart = cart;
17         this.filteredList = new FilteredList<Media>(cart.getItemsOrdered());
18     }
19
20     // Khởi tạo các event
21     @FXML
22     private void initialize() {
23         ...
24         // Bất sự kiện ô input keyword thay đổi, nếu có thay đổi thì tiến hành lọc dữ liệu
25         tfFilter.textProperty().addListener(
26             new ChangeListener<String>() {
27                 @Override
28                 public void changed(ObservableValue<? extends String> observable, String oldValue, String newValue) {
29                     showFilterMedia(newValue);
30                 }
31             }
32         );
33         ...
34     }
35 }
36 }
37

```

Kết quả

Danh sách sản phẩm ban đầu

Title	Category	Cost
Doraemon	Hoạt hình	24.95
Conan	Hoạt hình	99.7
Đi để chờ về		25.99
Nhật chuyên ngành 2	Tiếng Nhật	25.0
Nhật chuyên ngành 1	Tiếng Nhật	21.0

Lọc theo ID:

Filter
☒ By ID
 ☐ By Title

Title	Category	Cost
Đi để chờ về		25.99

Lọc theo title:

Thực hành lập trình hướng đối tượng

Filter ☐ By ID ☒ By Title

Title	Category	Cost
Nhật chuyên ngành 2	Tiếng Nhật	25.0
Nhật chuyên ngành 1	Tiếng Nhật	21.0

10. Hoàn thiện chương trình

Danh sách các tính năng của chương trình

10.1 Màn hình Cart Screen

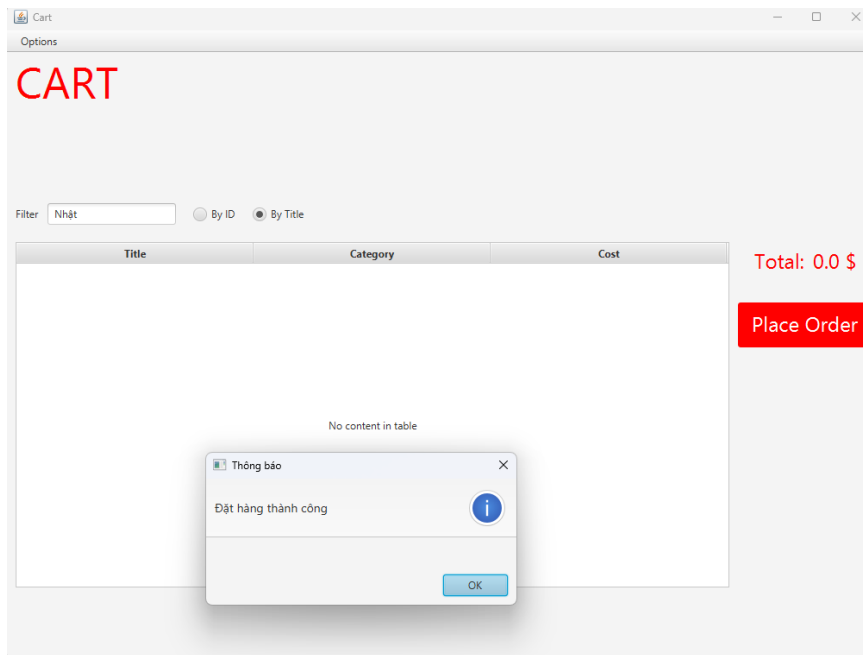
10.1.1 Place order button

```

3 package hust.soict.dsai.aims.screen;
4 public class CartScreenController {
5     @FXML
6     private Button btnPlaceOrder;
7
8     ...
9
10    /**
11     * Handle khi nhấn button Place Order
12     * @param event
13     */
14    @FXML
15    void btnPlaceOrderPressed(ActionEvent event) {
16        cart.placeOrder();
17        Alert alert = new Alert(Alert.AlertType.INFORMATION);
18        alert.setTitle("Thông báo");
19        alert.setHeaderText("Đặt hàng thành công");
20        alert.show();
21    }
22
23
24 }
25

```

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 Play Button

```

3 package hust.soict.dsai.aims.screen;
4
5 public class CartScreenController {
6     @FXML
7     private Button btnPlay;
8
9     /**
10      * Handle khi nhấn vào button Play
11      * @param event
12      */
13     @FXML
14     void btnPlayPressed(ActionEvent event) {
15         // Lấy media hiện tại
16         Playable media = (Playable)tblMedia.getSelectionModel().getSelectedItem();
17         // Lấy title của media hiện tại
18         String mediaTitle = tblMedia.getSelectionModel().getSelectedItem().getTitle();
19         try {
20             // Play
21             media.play();
22             // Hiển thị thông báo
23             Alert alert = new Alert(Alert.AlertType.INFORMATION);
24             alert.setTitle("Trình phát nhạc");
25             Disc disc = (Disc)media;
26             alert.setHeaderText("Bạn đang nghe bản nhạc " + mediaTitle + " của ca sĩ: " + disc.getDirector());
27             alert.show();
28         } catch (PlayerException e) {
29             // Bắt lỗi nếu có
30             Alert alert = new Alert(Alert.AlertType.ERROR);
31             alert.setTitle("Thông báo");
32             alert.setHeaderText(e.getMessage());
33             alert.show();
34         }
35     }
36 }
37
38 }
39

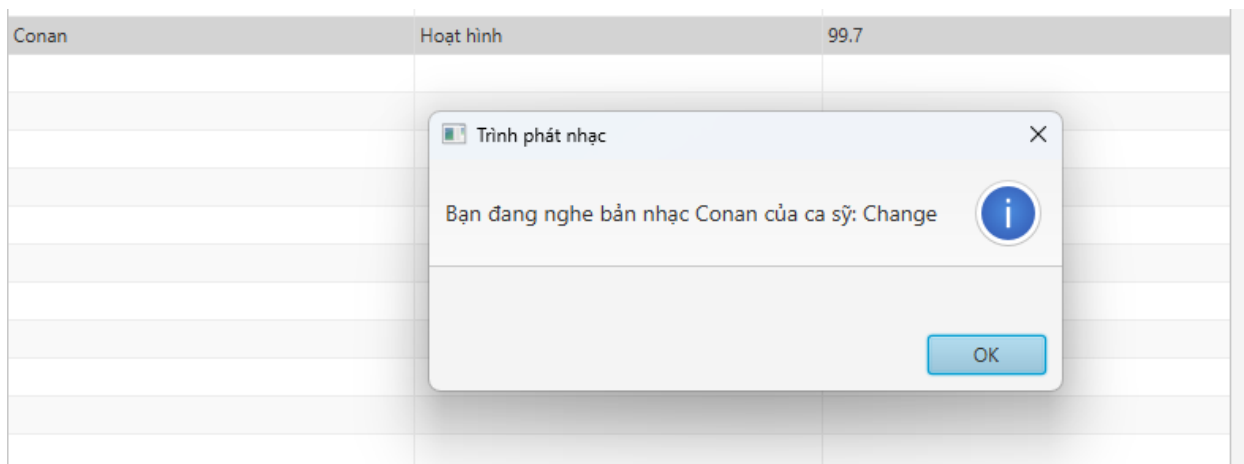
```

Phương thức Play:

Thực hành lập trình hướng đối tượng

```
3 package hust.soict.dsai.aims.media;
4 public class DigitalVideoDisc extends Disc implements Playable {
5
6     /**
7      * Play audio, nếu độ dài media không hợp lệ thì handle lỗi
8      * @throws PlayerException
9      */
10    @Override
11    public void play() throws PlayerException {
12        if(this.getLength() > 0) {
13            System.out.println("Playing DVD: " + this.getTitle());
14            System.out.println("DVD length: " + this.getLength());
15        } else {
16            System.err.println("ERROR: DVD length is non-positive!");
17            throw new PlayerException("ERROR: DVD length is non-positive!");
18        }
19    }
20 }
21 ...
22 }
23 }
```

Kết quả



10.1.3: Tổng tiền sản phẩm

```

3 package hust.soict.dsai.aims.screen;
4
5 public class CartScreenController {
6     private Cart cart;
7     ...
8
9     // Khởi tạo các event
10    @FXML
11    private void initialize() {
12        // Lắng nghe sự thay đổi của danh sách sản phẩm trong giỏ hàng và tính lại tiền
13        cart.getItemsOrdered().addListener((ChangeListener<Media>) change -> {
14            while (change.next()) {
15                //System.out.println(cart.totalCost());
16                Platform.runLater(new Runnable() {
17                    @Override
18                    public void run() {
19                        // Update tiền của đơn hàng
20                        totalCostLabel.setText(String.valueOf(cart.totalCost()) + " $");
21                        System.out.println("Update total cost");
22                    }
23                });
24            }
25        });
26    }
27 }
28 }
29

```

Kết quả

Trước khi thêm 1 sản phẩm:

Filter
☒ By ID
☐ By Title

Title	Category	Cost
Đi để chở về		25.99
Conan	Hoạt hình	99.7

Total: 125.689995 \$

Place Order

Sau khi thêm 1 sản phẩm

Thực hành lập trình hướng đối tượng

Filter
☒ By ID
☐ By Title

Title	Category	Cost
Đi để chở về		25.99
Conan	Hoạt hình	99.7
Nhật chuyển ngành 2	Tiếng Nhật	25.0

Total: 150.69 \$

Place Order

10.2 Màn hình Store Screen

Tính năng: Add to cart

Kết quả:

Store

Options

AIMS

Doraemon
24.95 \$
Add to cart Play

Conan
99.7 \$
Add to cart Play

Nhạc của tôi
75.5 \$
Add to cart Play

Đi để chở về
25.99 \$
Add to cart Play

Nhật chuyển ngành 1
25.0 \$
Add to cart

Nhật chuyển ngành 2
25.0 \$
Add to cart

Cart

Options

Filter
☒ By ID
☐ By Title

Title	Category	Cost
Đi để chở về		25.99
Conan	Hoạt hình	99.7
Nhật chuyển ngành 2	Tiếng Nhật	25.0

Total: 150.69

Place Order

Khi bấm Add To Cart:

Store

Options

AIMS

Doraemon
24.95 \$
Add to cart Play

Conan
99.7 \$
Add to cart Play

Nhạc của tôi
75.5 \$
Add to cart Play

Đi để chở về
25.99 \$
Add to cart Play

Nhật chuyển ngành 1
25.0 \$
Add to cart

Nhật chuyển ngành 2
25.0 \$
Add to cart

Cart

Options

Filter
☒ By ID
☐ By Title

Title	Category	Cost
Đi để chở về		25.99
Conan	Hoạt hình	99.7
Nhật chuyển ngành 2	Tiếng Nhật	25.0
Nhạc của tôi		75.5

Total: 226.19 \$

Place Order

Mã nguồn:

Thực hành lập trình hướng đối tượng

```

3 package hust.soict.dsai.aims.screen;
4
5
6 public class MediaStore extends JPanel {
7     private Media media;
8     private Frame parentFrame;
9     // Render UI
10    public MediaStore(Media media, Frame parentFrame) {
11        ...
12        // Action
13        JButton addToCartBtn = new JButton("Add to cart");
14        container.add(addToCartBtn);
15        addToCartBtn.putClientProperty("media", media);
16        addToCartBtn.putClientProperty("type", "add_to_cart");
17        addToCartBtn.addActionListener(new AddToCartButtonListener());
18        ...
19    }
20
21
22    // Class handle event Add to Cart
23    private class AddToCartButtonListener implements ActionListener {
24        @Override
25        public void actionPerformed(ActionEvent e) {
26            JButton button = (JButton) e.getSource();
27            String type = (String) button.getClientProperty("type");
28            Media selectedMedia = (Media) button.getClientProperty("media");
29            try {
30                // Thêm sản phẩm vào cart
31                InitData.myCart.addMedia(selectedMedia);
32            } catch (LimitExceededException ex) {
33                // Báo lỗi cart đầy nếu có
34                JDialog dialog = new JDialog(parentFrame, "Thông báo");
35                JLabel label = new JLabel(ex.getLocalizedMessage(), SwingConstants.CENTER);
36                dialog.add(label);
37                dialog.setLocationRelativeTo(parentFrame);
38                dialog.setSize(300, 200);
39                dialog.setVisible(true);
40            }
41        }
42    }
43 }
44 }
45 }
46

```

10.3 Màn hình Update Store Screen

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

Cả 3 class trên sẽ kế thừa từ class AddItemToStoreScreen


```

3 package hust.soict.dsai.aims.screen;
4 public class AddItemToStoreScreen extends JFrame {
5     protected JPanel centerLayout;
6     public AddItemToStoreScreen() {
7     }
8     // Render phần North
9     JPanel createNorth() {
10         JPanel north = new JPanel();
11         north.setLayout(new BorderLayout(north, BorderLayout.V_AXIS));
12         north.add(createMenuBar());
13         north.add(createHeader());
14         return north;
15     }
16
17     // Tạo Menu
18     JMenuBar createMenuBar() {
19         JMenu menu = new JMenu("Options");
20
21         JMenu smUpdateStore = new JMenu("Update Store");
22         smUpdateStore.add(new JMenuItem("Add Book"));
23         smUpdateStore.add(new JMenuItem("Add CD"));
24         smUpdateStore.add(new JMenuItem("Add DVD"));
25
26         menu.add(smUpdateStore);
27         Aims.MenuItemListener menuItemListener = new Aims.MenuItemListener();
28         JMenuItem viewStoreMenu = new JMenuItem("View Store");
29         viewStoreMenu.setActionCommand("view_store");
30         viewStoreMenu.addActionListener(menuItemListener);
31         menu.add(viewStoreMenu);
32         JMenuItem viewCartMenu = new JMenuItem("View Cart");
33         viewCartMenu.setActionCommand("view_cart");
34         viewCartMenu.addActionListener(menuItemListener);
35         menu.add(viewCartMenu);
36         JMenuBar menuBar = new JMenuBar();
37         menuBar.setLayout(new FlowLayout(FlowLayout.LEFT));
38         menuBar.add(menu);
39         return menuBar;
40     }
41
42     // Tạo header(logo)
43     JPanel createHeader() {
44         JPanel header = new JPanel();
45         header.setLayout(new BorderLayout(header, BorderLayout.X_AXIS));
46         JLabel title = new JLabel("AIMS");
47         title.setFont(new Font(title.getFont().getName(), Font.PLAIN, 50));
48         title.setForeground(Color.RED);
49         header.add(Box.createRigidArea(new Dimension(10, 10)));
50         header.add(title);
51         header.add(Box.createRigidArea(new Dimension(10, 10)));
52         header.add(Box.createRigidArea(new Dimension(10, 10)));
53         return header;
54     }
55
56     /**
57      * Tạo các input
58      * @param label
59      * @param column
60      * @return
61      */
62     JPanel inputGroup(String label, int column) {
63         JPanel inputGroup = new JPanel();
64         inputGroup.setLayout(new FlowLayout());
65         JLabel inputLabel = new JLabel(label, JLabel.RIGHT);
66         JTextField input = new JTextField(column);
67         inputGroup.add(inputLabel);
68         inputGroup.add(input);
69         return inputGroup;
70     }
71
72     /**
73      * Tạo button và handle event
74      * @param actionListener
75      * @return
76      */
77     JPanel submitButton(ActionListener actionListener) {
78         JPanel buttonGroup = new JPanel();
79         buttonGroup.setLayout(new FlowLayout());
80         JButton submitButton = new JButton("Add Media");
81         submitButton.addActionListener(actionListener);
82         buttonGroup.setPreferredSize(new Dimension(100, 50));
83         buttonGroup.setMaximumSize(new Dimension(100, 50));
84         buttonGroup.add(submitButton);
85         return buttonGroup;
86     }
87
88     /**
89      * Lấy value từ input trong một JPanel component
90      * @param element
91      * @return
92      */
93     String getValueTextField(JPanel element) {
94         String result = "";
95         Component[] components = element.getComponents();
96         for (Component comp : components) {
97             if (comp instanceof JTextField) {
98                 JTextField textField = (JTextField) comp;
99                 result = textField.getText();
100                 break;
101             }
102         }
103         return result;
104     }
105 }
106

```

10.3.1 Menu lựa chọn

```

3 package hust.soict.dsai.aims.screen;
4
5 public class StoreScreen extends JFrame {
6     ...
7
8     // Tạo menu
9     JMenuBar createMenuBar() {
10         JMenu menu = new JMenu("Options");
11         Aims.MenuItemListener menuItemListener = new Aims.MenuItemListener();
12         JMenu smUpdateStore = new JMenu("Update Store");
13
14         // Add các Menu Con
15         JMenuItem addBookMenu = new JMenuItem("Add Book");
16         addBookMenu.setActionCommand("add_book");
17         addBookMenu.addActionListener(menuItemListener);
18         smUpdateStore.add(addBookMenu);
19
20         // Thêm CD
21         JMenuItem addCDMenu = new JMenuItem("Add CD");
22         addCDMenu.setActionCommand("add_cd");
23         addCDMenu.addActionListener(menuItemListener);
24         smUpdateStore.add(addCDMenu);
25
26         // Thêm DVD
27         JMenuItem addDvdMenu = new JMenuItem("Add DVD");
28         addDvdMenu.setActionCommand("add_dvd");
29         addDvdMenu.addActionListener(menuItemListener);
30         smUpdateStore.add(addDvdMenu);
31
32         menu.add(smUpdateStore);
33
34         return menu;
35     }
36 }
37
38 // Xử lý việc gọi các màn hình
39 public static class MenuItemListener implements ActionListener {
40     public void actionPerformed(ActionEvent e) {
41         String command = e.getActionCommand();
42         if(command.equals("view_store")) {
43             currentView = new StoreScreen(myStore);
44         } else if(command.equals("view_cart")) {
45             currentView = new CartScreen(myCart);
46         } else if(command.equals("add_dvd")) {
47             new AddDigitalVideoDiscToStoreScreen();
48         } else if(command.equals("add_cd")) {
49             new AddCompactDiscToStoreScreen();
50         } else if(command.equals("add_book")) {
51             new AddBookToStoreScreen();
52         }
53         System.out.println(e.getActionCommand() + " JMenuItem clicked.");
54     }
55 }
56
57 }
58

```

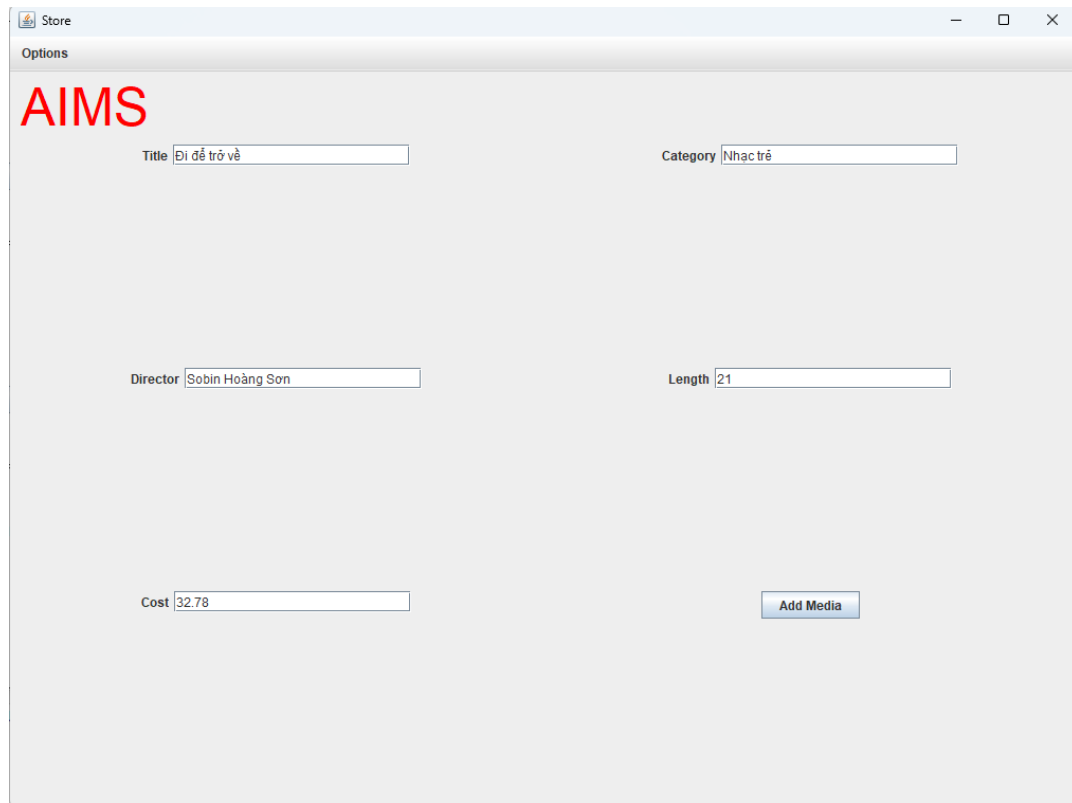
10.3.2 Màn hình thêm DVD

```

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

```

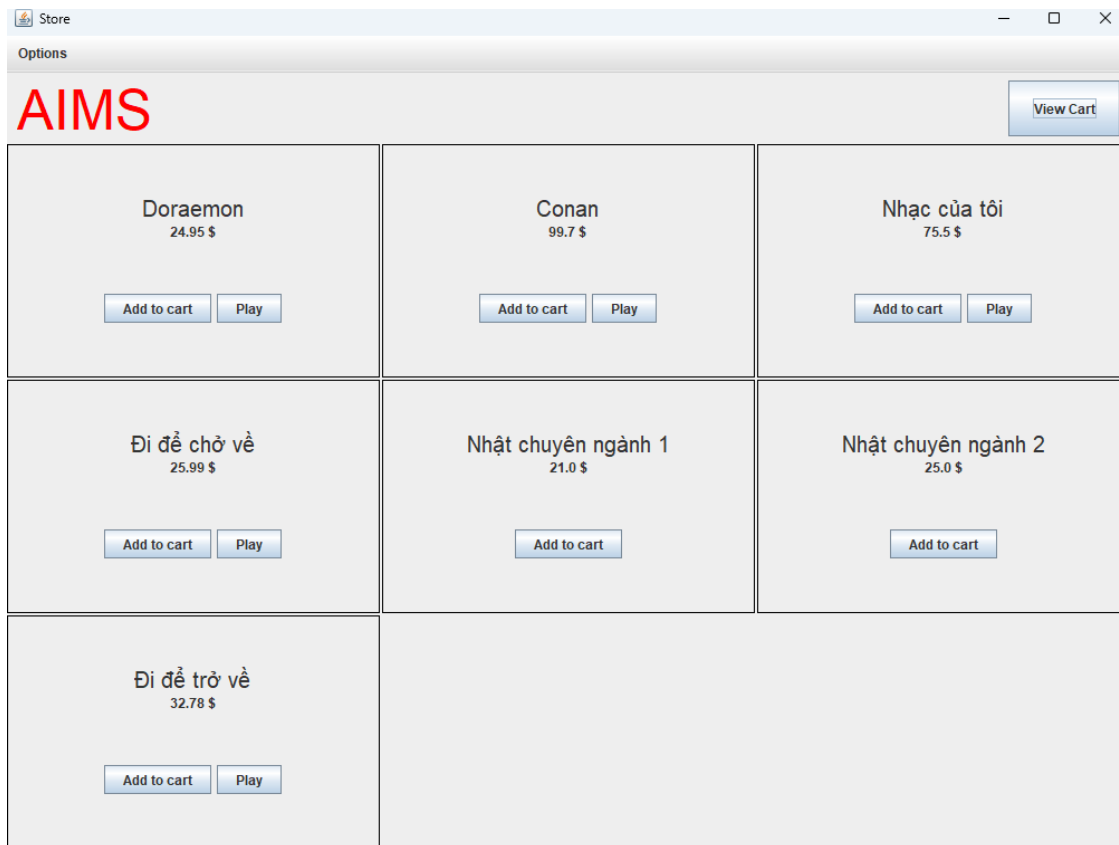
Kết quả



The screenshot shows a window titled "Store" with a sub-header "Options". The "AIMS" logo is displayed in red. The form contains the following fields and controls:

- Title:**
- Category:**
- Director:**
- Length:**
- Cost:**
- Add Media:**

Sau khi nhập thông tin và click Add Media, sản phẩm sẽ có trong cửa hàng



10.3.3 Màn hình thêm CD

Mã nguồn

```

3 package hust.soict.dsai.aims.screen;
4
5 public class AddCompactDiscToStoreScreen extends AddItemToStoreScreen {
6     private Store myStore = InitData.myStore;
7     // Render UI
8     public AddCompactDiscToStoreScreen() {
9         Container cp = getContentPane();
10        cp.setLayout(new BorderLayout());
11        cp.add(createNorth(), BorderLayout.NORTH);
12        cp.add(renderAddCDScreen(), BorderLayout.CENTER);
13        setVisible(true);
14        setTitle("Store");
15        setSize(1024, 768);
16    }
17
18    // Render Form nhập thông tin
19    JPanel renderAddCDScreen() {
20        JPanel layout = new JPanel();
21        layout.setLayout(new GridLayout(3, 3, 2, 2));
22        // Input Title
23        JPanel inputTitleGroup = this.inputGroup("Title", 20);
24        layout.add(inputTitleGroup);
25        // Input category
26        JPanel inputArtistGroup = this.inputGroup("Artist", 20);
27        layout.add(inputArtistGroup);
28        // Input director
29        JPanel inputDirectorGroup = this.inputGroup("Director", 20);
30        layout.add(inputDirectorGroup);
31
32        // Input Cost
33        JPanel inputCostGroup = this.inputGroup("Cost", 20);
34        layout.add(inputCostGroup);
35
36        // Submit button
37        layout.add(submitButton(new ActionListener() {
38            @Override
39            public void actionPerformed(ActionEvent e) {
40                // Tạo CD mới
41                CompactDisc cd = new CompactDisc(getValueTextField(inputArtistGroup), getValueTextField(inputTitleGroup),
42                getValueTextField(inputDirectorGroup), Float.valueOf(getValueTextField(inputCostGroup)));
43                // Thêm CD vào cửa hàng
44                myStore.addMedia(cd);
45            }
46        }));
47        return layout;
48    }
49 }

```

Kết quả

Thực hành lập trình hướng đối tượng

The screenshot shows a window titled 'Store' with a sub-header 'Options'. The 'AIMS' logo is in red. The form contains four input fields: 'Title' with the value 'CD Nhạc Tết 1', 'Artist' with 'Nhiều nghệ sĩ', 'Director' with 'Quốc Khánh', and 'Cost' with '32.99'. An 'Add Media' button is at the bottom.

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

The screenshot shows the 'Store' window with the 'Options' header. The 'AIMS' logo is in red. A 'View Cart' button is in the top right. The product catalog is displayed in a 3x3 grid. The products are:

Product Name	Price	Buttons
Doraemon	24.95 \$	Add to cart, Play
Conan	99.7 \$	Add to cart, Play
Nhạc của tôi	75.5 \$	Add to cart, Play
Đi để chờ về	25.99 \$	Add to cart, Play
Nhật chuyên ngành 1	21.0 \$	Add to cart
Nhật chuyên ngành 2	25.0 \$	Add to cart
Đi để trở về	32.78 \$	Add to cart, Play
CD Nhạc Tết 1	32.99 \$	Add to cart, Play

Thực hành lập trình hướng đối tượng

10.3.4 Màn hình thêm Book

Mã nguồn:

```

3 package hust.soict.dsai.aims.screen;
4
5 public class AddBookToStoreScreen extends AddItemToStoreScreen {
6     private Store myStore = InitData.myStore;
7     // Render UI
8     public AddBookToStoreScreen() {
9         Container cp = getContentPane();
10        cp.setLayout(new BorderLayout());
11        cp.add(createNorth(), BorderLayout.NORTH);
12        cp.add(renderAddBookScreen(), BorderLayout.CENTER);
13
14        setVisible(true);
15        setTitle("Store");
16        setSize(1024, 768);
17
18    }
19
20    /**
21     * Render giao diện form điền thông tin
22     * @return
23     */
24    JPanel renderAddBookScreen() {
25        JPanel layout = new JPanel();
26        layout.setLayout(new GridLayout(3, 3, 2, 2));
27        // Input Title
28        JPanel inputTitleGroup = this.inputGroup("Title", 20);
29        layout.add(inputTitleGroup);
30        // Input category
31        JPanel inputCategoryGroup = this.inputGroup("Category", 20);
32        layout.add(inputCategoryGroup);
33
34        // Input author
35        JPanel inputAuthorGroup = this.inputGroup("Author", 20);
36        layout.add(inputAuthorGroup);
37
38        // Input Cost
39        JPanel inputCostGroup = this.inputGroup("Cost", 20);
40        layout.add(inputCostGroup);
41
42        // Submit button
43        layout.add(submitButton(new ActionListener() {
44            @Override
45            public void actionPerformed(ActionEvent e) {
46                // Tạo sách mới
47                Book book = new Book(getValueTextField(inputTitleGroup), getValueTextField(inputCategoryGroup),
48                Float.valueOf(getValueTextField(inputCostGroup)));
49                // Thêm tác giả
50                book.addAuthor(getValueTextField(inputAuthorGroup));
51                // Thêm vào cửa hàng
52                myStore.addMedia(book);
53            }
54        }));
55        return layout;
56    }
57 }

```

Kết quả

Thực hành lập trình hướng đối tượng

The screenshot shows a window titled 'Store' with a sub-header 'Options'. The 'AIMS' logo is in the top left. The form contains four input fields: 'Title' with the value 'Sách khóc học', 'Category' with the value 'Sách', 'Author' with the value 'Nguyễn Khoa Học', and 'Cost' with the value '77.99'. An 'Add Media' button is located at the bottom center.

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

The screenshot shows the 'Store' window with the 'Options' header. The 'AIMS' logo is on the left, and a 'View Cart' button is on the right. The main area displays a 3x3 grid of product cards. Each card shows the product name, price, and buttons for 'Add to cart' and 'Play'.

Doraemon 24.95 \$ Add to cart Play	Conan 99.7 \$ Add to cart Play	Nhạc của tôi 75.5 \$ Add to cart Play
Đi để chờ về 25.99 \$ Add to cart Play	Nhật chuyên ngành 1 21.0 \$ Add to cart	Nhật chuyên ngành 2 25.0 \$ Add to cart
Đi để trở về 32.78 \$ Add to cart Play	CD Nhạc Tết 1 32.99 \$ Add to cart Play	Sách khóc học 77.99 \$ Add to cart

Thực hành lập trình hướng đối tượng

11. Thêm Exception, hiển thị thông báo lỗi bằng giao diện

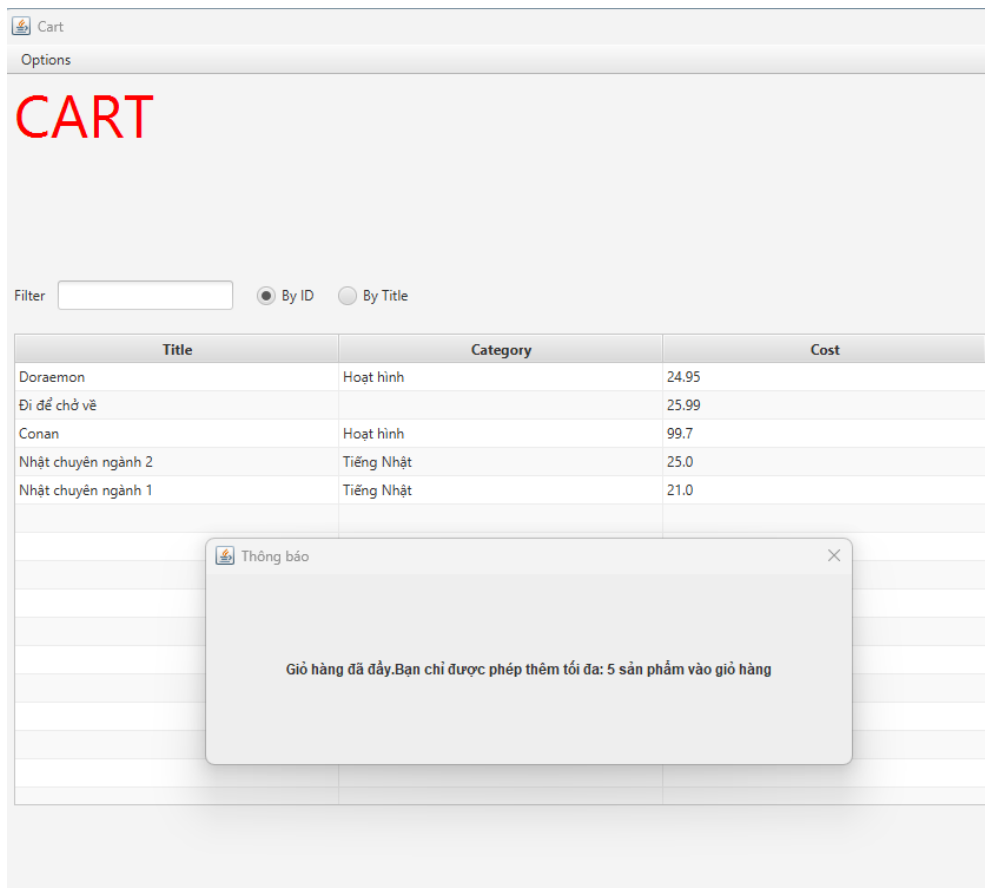
11.1 Hiện thị lỗi quá số lượng sản phẩm trong giỏ hàng

```

1 package hust.soict.dsai.aims.cart;
2
3 import java.util.*;
4 public class Cart {
5     public static final int MAX_NUMBERS_ORDERED = 20;
6     private ObservableList<Media> itemsOrdered = FXCollections.observableArrayList();
7
8     /**
9      * Thêm nhiều media vào giỏ hàng
10      * @param medias
11      * @return
12      */
13     public boolean addMedia(Media ... medias) throws LimitExceededException {
14         // Nếu vượt quá số lượng sản phẩm tối đa thì throw lỗi
15         if(itemsOrdered.size() + medias.length > MAX_NUMBERS_ORDERED) {
16             throw new LimitExceededException("Giỏ hàng đã đầy.Bạn chỉ được phép thêm tối đa: " + (MAX_NUMBERS_ORDERED - itemsOrdered.size()) + " sản phẩm vào giỏ hàng");
17         }
18         for(Media media: medias) {
19             if(!itemsOrdered.contains(media)) {
20                 itemsOrdered.add(media);
21             } else {
22                 System.out.println("Sản phẩm " + media.getTitle() + " đã tồn tại trong giỏ hàng");
23             }
24         }
25
26         return true;
27     }
28 }
29
30

```

Kết quả: Giả sử một giỏ hàng chỉ được chứa tối đa 5 sản phẩm



Thực hành lập trình hướng đối tượng

11.2 Hiện thị lỗi nếu độ dài media không hợp lệ khi Play Mã nguồn

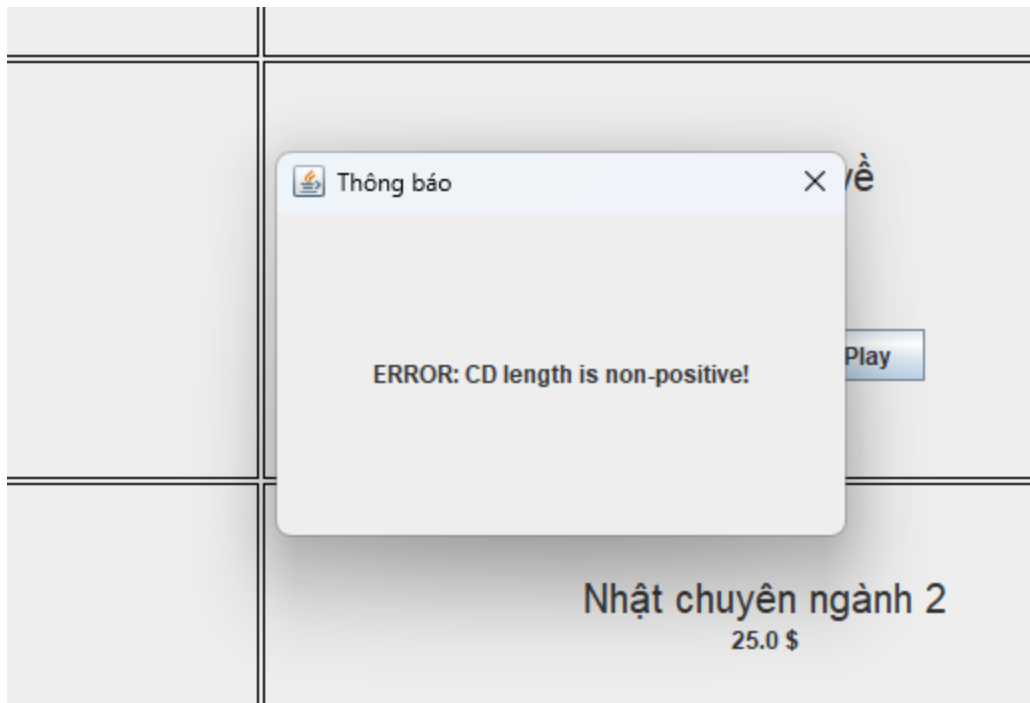
Class PlayerException

```
3 package hust.soict.dsai.aims.exception;
4
5 public class PlayerException extends Exception{
6     public PlayerException(String message) {
7         super(message);
8     }
9 }
10
```

Class DigitalVideoDisc

```
3 package hust.soict.dsai.aims.media;
4
5 public class DigitalVideoDisc extends Disc implements Playable {
6
7     /**
8      * Play audio, nếu độ dài media không hợp lệ thì handle lỗi
9      * @throws PlayerException
10     */
11     @Override
12     public void play() throws PlayerException {
13         if(this.getLength() > 0) {
14             System.out.println("Playing DVD: " + this.getTitle());
15             System.out.println("DVD length: " + this.getLength());
16         } else {
17             System.err.println("ERROR: DVD length is non-positive!");
18             throw new PlayerException("ERROR: DVD length is non-positive!");
19         }
20     }
21 }
22 }
23
```

Kết quả



12. Cập nhật phương thức equals của class Media

Mã nguồn:

```
    * So sánh 2 media.Bằng nhau nếu title giống nhau
    * @param o
    * @return
    */
    @Override
    public boolean equals(Object o) {
        if (this == o) return true;
        if (o == null || getClass() != o.getClass()) return false;
        // Kiểm tra instance xem có phải của class Media không
        if ((o instanceof Media) == false) {
            return false;
        }
        Media media = (Media) o;
        return Objects.equals(title, media.title);
    }
}
```