

Report Lab 05: GUI Programming

1. Swing components

```

package hust.soict.swing;

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

public class AWTAccumulator extends Frame {
    private TextField tfInput;
    private TextField tfOutput;
    private int sum = 0;

    public AWTAccumulator() {
        setLayout(new GridLayout(rows:2, cols:2, hgap:5, vgap:5)); // Sử dụng GridLayout: 2 hàng, 2 cột, khoảng cách 5px
        add(new Label(text:"Enter a number: "));
        tfInput = new TextField(columns:10); // Ô nhập
        add(tfInput);
        tfInput.addActionListener(new TFInputListener()); // Gắn sự kiện cho ô nhập

        add(new Label(text:"Accumulated sum: "));
        tfOutput = new TextField(columns:10); // Ô hiển thị tổng
        tfOutput.setEditable(b:false); // Chỉ đọc
        add(tfOutput);

        setTitle(title:"AWT Accumulator"); // Đặt tiêu đề
        setSize(width:350, height:120); // Đặt kích thước cửa sổ
        setVisible(b:true); // Hiển thị giao diện

        // Đóng cửa sổ khi nhấn nút Close
        addWindowListener(new java.awt.event.WindowAdapter() {
            public void windowClosing(java.awt.event.WindowEvent windowEvent) {
                System.exit(status:0);
            }
        });
    }

    // Lớp xử lý sự kiện khi nhấn Enter trong ô nhập
    private class TFInputListener implements ActionListener {
        @Override
        public void actionPerformed(ActionEvent evt) {
            try {
                int number = Integer.parseInt(tfInput.getText());
                sum += number; // Cộng dồn vào tổng
                tfInput.setText(""); // Xóa ô nhập
                tfOutput.setText(String.valueOf(sum)); // Hiển thị tổng
            } catch (NumberFormatException e) {
                System.err.println(x:"Invalid input. Please enter a valid number.");
            }
        }
    }

    // Phương thức chạy chương trình
    Run | Debug
    public static void main(String[] args) {
        new AWTAccumulator(); // Tạo giao diện
    }
}

```

```

package hust.soict.swing;

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

public class SwingAccumulator extends JFrame {
    private JTextField inputField;
    private JTextField sumField;
    private int sum = 0;

    public SwingAccumulator() {
        // Thiet lap JFrame
        setTitle(title:"Swing Accumulator");
        setSize(width:400, height:150);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setLayout(new GridLayout(rows:2, cols:2, hgap:5, vgap:5));

        add(new JLabel(text:"Enter a number: "));
        inputField = new JTextField();
        add(inputField);

        add(new JLabel(text:"Accumulated sum: "));
        sumField = new JTextField(text:"0");
        sumField.setEditable(b:false);
        add(sumField);

        inputField.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                try {
                    int number = Integer.parseInt(inputField.getText());
                    sum += number;
                    sumField.setText(String.valueOf(sum));
                    inputField.setText("");
                } catch (NumberFormatException ex) {
                    JOptionPane.showMessageDialog(SwingAccumulator.this, message:"Please enter a valid number!",
                        title:"Invalid Input", JOptionPane.ERROR_MESSAGE);
                }
            }
        });
        setVisible(b:true);
    }

    Run | Debug
    public static void main(String[] args) {
        SwingUtilities.invokeLater(() -> new SwingAccumulator());
    }
}

```

2. Organizing Swing components with Layout Managers

```

1 package hust.soict.swing;
2
3 import javax.swing.*;
4 import java.awt.*;
5 import java.awt.event.ActionEvent;
6 import java.awt.event.ActionListener;
7
8 public class NumberGrid extends JFrame {
9     private JTextField tfDisplay;
10    private JButton[] btnNumbers;
11    private JButton btnDelete, btnReset;
12
13    public NumberGrid() {
14        setTitle(title: "Number Grid");
15        setSize(width: 300, height: 400);
16        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
17        setLayout(new BorderLayout());
18
19        tfDisplay = new JTextField();
20        tfDisplay.setEditable(false);
21        tfDisplay.setHorizontalAlignment(JTextField.RIGHT);
22        add(tfDisplay, BorderLayout.NORTH);
23
24        JPanel panelButtons = new JPanel();
25        panelButtons.setLayout(new GridLayout(rows: 4, cols: 3));
26
27        btnNumbers = new JButton[10];
28        for (int i = 0; i < 10; i++) {
29            btnNumbers[i] = new JButton(String.valueOf(i));
30            panelButtons.add(btnNumbers[i]);
31            btnNumbers[i].addActionListener(new ButtonListener());
32        }
33
34        btnDelete = new JButton(text: "DEL");
35        btnReset = new JButton(text: "C");
36        panelButtons.add(btnDelete);
37        panelButtons.add(btnReset);
38
39        btnDelete.addActionListener(new ButtonListener());
40        btnReset.addActionListener(new ButtonListener());
41        add(panelButtons, BorderLayout.CENTER);
42        setVisible(true);
43    }
44
45    // Lớp lắng nghe sự kiện cho các nút
46    private class ButtonListener implements ActionListener {
47        @Override
48        public void actionPerformed(ActionEvent e) {
49            JButton source = (JButton) e.getSource();
50
51            if (source == btnDelete) {
52                // Xóa ký tự cuối cùng
53                String currentText = tfDisplay.getText();
54                if (!currentText.isEmpty()) {
55                    tfDisplay.setText(currentText.substring(beginIndex: 0, currentText.length() - 1));
56                }
57            } else if (source == btnReset) {
58                // Xóa toàn bộ
59                tfDisplay.setText("");
60            } else {
61                // Nút số: thêm ký tự vào cuối
62                tfDisplay.setText(tfDisplay.getText() + source.getText());
63            }
64        }
65    }
66
67    Run | Debug
68    public static void main(String[] args) {
69        new NumberGrid();
70    }
71

```

3. JavaFX API

```
1 package hust.soict.javafx;
2
3 import javafx.application.Application;
4 import javafx.fxml.FXMLLoader;
5 import javafx.scene.Parent;
6 import javafx.scene.Scene;
7 import javafx.stage.Stage;
8
9 public class Painter extends Application {
10     @Override
11     public void start(Stage stage) throws Exception {
12         Parent root = FXMLLoader.load(getClass().getResource("Painter.fxml"));
13         Scene scene = new Scene(root);
14         stage.setTitle("Painter");
15         stage.setScene(scene);
16         stage.show();
17     }
18
19     Run | Debug
20     public static void main(String[] args) {
21         launch(args);
22     }
23 }
```

```

1  package hust.soict.javaafx;
2
3  import javafx.fxml.FXML;
4  import javafx.scene.layout.Pane;
5  import javafx.scene.input.MouseEvent;
6  import javafx.scene.paint.Color;
7  import javafx.scene.shape.Circle;
8  import javafx.event.ActionEvent;
9
10 public class PainterController {
11     @FXML
12     private Pane drawingAreaPane;
13
14     @FXML
15     void clearButtonPressed(ActionEvent event) {
16         drawingAreaPane.getChildren().clear();
17     }
18
19     @FXML
20     void drawingAreaMouseDragged(MouseEvent event) {
21         Circle newCircle = new Circle(event.getX(), event.getY(), 4, Color.BLACK);
22         drawingAreaPane.getChildren().add(newCircle);
23     }
24 }
25

```

```

1  <?xml version="1.0" encoding="UTF-8"?>
2
3  <?import javafx.geometry.Insets?>
4  <?import javafx.scene.control.Button?>
5  <?import javafx.scene.layout.BorderPane?>
6  <?import javafx.scene.layout.Pane?>
7  <?import javafx.scene.layout.VBox?>
8  <?import javafx.scene.text.Font?>
9
10
11 <BorderPane fx:controller="hust.soict.javaafx.PainterController" maxHeight="-Infinity" maxWidth="-Infinity" minHeight="-Infinity" minWidth="-Infinity" style="-fx-background-color: #f0f0f0">
12     <center>
13         <BorderPane minHeight="-Infinity" minWidth="-Infinity" prefHeight="480.0" prefWidth="640.0" BorderPane.alignment="CENTER">
14             <padding>
15                 <Insets bottom="8.0" left="8.0" right="8.0" top="8.0" />
16             </padding>
17             <center>
18                 <Pane fx:id="drawingAreaPane" onMouseDragged="#drawingAreaMouseDragged" prefHeight="200.0" prefWidth="200.0" style="-fx-background-color: #ffffff;" />
19             </center>
20         </BorderPane>
21     </center>
22     <left>
23         <VBox fx:id="toolMenu" maxHeight="1.7976931348623157E308" spacing="8.0" BorderPane.alignment="CENTER">
24             <BorderPane.margin>
25                 <Insets right="8.0" />
26             </BorderPane.margin>
27             <children>
28                 <Button maxWidth="1.7976931348623157E308" mnemonicParsing="false" onAction="#clearButtonPressed" text="Clear">
29                     <font>
30                         <font size="13.0" />
31                     </font>
32                 </Button>
33             </children>
34         </VBox>
35     </left>
36 </BorderPane>

```

```

Settings | StoreScreen.java U X
AIMSProject > src > hust > soict > dsai > aims > screen > StoreScreen.java > StoreScreen.java
1 package hust.soict.dsai.aims.screen;
2
3 import hust.soict.dsai.aims.cart.Cart;
4 import hust.soict.dsai.aims.media.Book;
5 import hust.soict.dsai.aims.media.CompactDisc;
6 import hust.soict.dsai.aims.media.DigitalVideoDisc;
7 import hust.soict.dsai.aims.store.Store;
8
9 import javax.swing.*;
10 import java.awt.*;
11
12 @SuppressWarnings("unused")
13 public class StoreScreen extends JFrame {
14     private Store store;
15     private Cart cart;
16
17     public StoreScreen(Store store, Cart cart) {
18         this.store = store;
19         this.cart = cart;
20
21         setTitle(title:"Store");
22         setSize(width:1024, height:768);
23         setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
24         setLayout(new BorderLayout());
25
26         add(createNorth(), BorderLayout.NORTH);
27         add(createCenter(), BorderLayout.CENTER);
28
29         setVisible(b:true);
30     }
31
32     private JPanel createNorth() {
33         JPanel north = new JPanel();
34         north.setLayout(new BoxLayout(north, BoxLayout.Y_AXIS));
35         north.add(createMenuBar());
36         north.add(createHeader());
37
38         return north;
39     }
40
41     private JMenuBar createMenuBar() {
42         JMenuBar menuBar = new JMenuBar();
43         JMenu optionsMenu = new JMenu(s:"Options");
44         JMenu smUpdateStore = new JMenu(s:"Update Store");
45
46         JMenuItem addBookItem = new JMenuItem(text:"Add Book");
47         addBookItem.addActionListener(e -> {
48             new AddBookToStoreScreen(store, this);
49         });
50         smUpdateStore.add(addBookItem);

```



```

smUpdateStore.add(addBookItem);

JMenuItem addCDItem = new JMenuItem(text:"Add CD");
addCDItem.addActionListener(e -> {
    new AddCDToStoreScreen(store, this);
});
smUpdateStore.add(addCDItem);

JMenuItem addDVDItem = new JMenuItem(text:"Add DVD");
addDVDItem.addActionListener(e -> {
    new AddDVDToStoreScreen(store, this);
});
smUpdateStore.add(addDVDItem);

optionsMenu.add(smUpdateStore);

JMenuItem viewCartItem = new JMenuItem(text:"View Cart");
viewCartItem.addActionListener(e -> {
    new CartScreen(cart);
});
optionsMenu.add(viewCartItem);

menuBar.add(optionsMenu);

return menuBar;
}

private JPanel createHeader() {
    JPanel header = new JPanel();
    header.setLayout(new BoxLayout(header, BoxLayout.X_AXIS));

    JLabel title = new JLabel(text:"Welcome to AIMS Store");
    title.setFont(new Font(title.getFont().getName(), Font.PLAIN, size:24));
    title.setAlignmentX(CENTER_ALIGNMENT);

    header.add(Box.createRigidArea(new Dimension(width:10, height:10)));
    header.add(title);
    header.add(Box.createRigidArea(new Dimension(width:10, height:10)));

    return header;
}

private JPanel createCenter() {
    JPanel center = new JPanel();
    center.setLayout(new GridLayout(rows:3, cols:3, hgap:10, vgap:10)); // Lưới 3x3 cho các mặt hàng
    store.getItemsInStore().forEach(media -> {
        MediaStore cell = new MediaStore(media, cart);
        center.add(cell);
    });
}

```

```
92     private JPanel createCenter() {
93         MediaStore cell = new MediaStore(media, cart);
94         center.add(cell);
95     };
96     return center;
97 }
98
99 public void refreshCenterPanel() {
100     getContentPane().removeAll();
101     add(createNorth(), BorderLayout.NORTH);
102     add(createCenter(), BorderLayout.CENTER);
103     revalidate();
104     repaint();
105 }
106
107 Run | Debug
108 public static void main(String[] args) {
109     Store store = new Store();
110     Cart cart = new Cart();
111     new StoreScreen(store, cart);
112 }
113
114 }
```

```

1   package hust.soict.dsai.aims.screen;
2
3   import javafx.fxml.FXML;
4   import javafx.scene.control.TableView;
5   import javafx.scene.control.TextField;
6   import javafx.scene.control.Alert;
7   import javafx.scene.control.Button;
8   import javafx.scene.control.RadioButton;
9   import javafx.scene.control.Tab;
10  import javafx.scene.control.TableColumn;
11  import javafx.scene.control.Label;
12
13  import hust.soict.dsai.aims.media.Media;
14  import hust.soict.dsai.aims.media.Playable;
15
16  import hust.soict.dsai.aims.cart.Cart;
17  import hust.soict.dsai.aims.exception.PlayerException;
18  import javafx.beans.value.ChangeListener;
19  import javafx.beans.value.ObservableValue;
20  import javafx.collections.FXCollections;
21  import javafx.collections.ObservableList;
22  import javafx.collections.transformation.FilteredList;
23  import javafx.event.ActionEvent;
24  import javafx.scene.control.cell.PropertyValueFactory;
25
26  @SuppressWarnings("unused")
27  public class CartScreenController {
28      private Cart cart;
29
30      @FXML
31      private TextField tfFilter;
32
33      @FXML
34      private RadioButton radioBtnFilterId;
35
36      @FXML
37      private RadioButton radioBtnFilterTitle;
38
39      @FXML
40      private TableView<Media> tblMedia;
41
42      @FXML
43      private TableColumn<Media, String> colMediaTitle;
44
45      @FXML
46      private TableColumn<Media, String> colMediacategory;
47
48      @FXML
49      private TableColumn<Media, Float> colMediaCost;

```

```

50
51 @FXML
52 private Button btnPlay;
53
54 @FXML
55 private Button btnRemove;
56
57 @FXML
58 private Label lblTotalCost;
59
60 public CartScreenController(Cart cart) {
61     this.cart = cart;
62 }
63
64 @FXML
65 private void initialize() {
66     // Gán giá trị cho từng cột
67     colMediaTitle.setCellValueFactory(new PropertyValueFactory<>(property:"title"));
68     colMediaCategory.setCellValueFactory(new PropertyValueFactory<>(property:"category"));
69     colMediaCost.setCellValueFactory(new PropertyValueFactory<>(property:"cost"));
70
71     // Ẩn các nút ban đầu
72     btnPlay.setVisible(false);
73     btnRemove.setVisible(false);
74
75     // Tạo ObservableList và FilteredList
76     ObservableList<Media> mediaList = FXCollections.observableArrayList(cart.getItemsOrdered());
77     FilteredList<Media> filteredList = new FilteredList<>(mediaList);
78     tblMedia.setItems(filteredList);
79
80     updateTotalCost();
81
82     // Listener cho TableView selection
83     tblMedia.getSelectionModel().selectedItemProperty().addListener((observable, oldValue, newValue) -> {
84         if (newValue != null)
85             updateButtonBar(newValue);
86     });
87     // Listener cho bộ lọc TextField
88     tfFilter.textProperty().addListener((observable, oldValue, newValue) -> {
89         filteredList.setPredicate(media -> {
90             if (newValue == null || newValue.isEmpty())
91                 return true;
92             String lowerCaseFilter = newValue.toLowerCase();
93             if (radioBtnFilterId.isSelected()) {
94                 return String.valueOf(media.getId()).contains(lowerCaseFilter);
95             } else if (radioBtnFilterTitle.isSelected()) {
96                 return media.getTitle().toLowerCase().contains(lowerCaseFilter);

```

```

96         return media.getTitle().toLowerCase().contains(lowerCaseFilter);
97     }
98     return false;
99 });
100 });
101 }
102
103 private void updateButtonBar(Media media) {
104     // Hiển thị nút Remove
105     btnRemove.setVisible(true);
106
107     // Kiểm tra media có phải là Playable hay không
108     if (media instanceof Playable) {
109         btnPlay.setVisible(true);
110     } else {
111         btnPlay.setVisible(false);
112     }
113 }
114
115 @FXML
116 void btnRemovePressed(ActionEvent event) {
117     // Lấy mục được chọn từ TableView
118     Media media = tblMedia.getSelectionModel().getSelectedItem();
119
120     if (media != null) {
121         // Xóa mục khỏi giỏ hàng
122         cart.removeMedia(media);
123         showAlert(Alert.AlertType.INFORMATION, title:"Remove Media", "Removed: " + media.getTitle());
124
125         // Cập nhật lại ObservableList cho TableView
126         ObservableList<Media> updatedList = FXCollections.observableArrayList(cart.getItemsOrdered());
127         tblMedia.setItems(updatedList);
128         updateTotalCost();
129     }
130 }
131
132 @FXML
133 void btnPlaceOrderPressed(ActionEvent event) {
134     System.out.println(x:"Order placed succesfully");
135     cart.getItemsOrdered().clear();
136     tblMedia.getItems().clear();
137     updateTotalCost();
138 }
139
140 private void updateTotalCost() {
141     lblTotalCost.setText(String.format(format:"Total: %.2f $", cart.totalCost()));
142 }
143

```

```
142
143
144 @FXML
145 void btnPlayPressed(ActionEvent event) throws PlayerException {
146     Media media = tblMedia.getSelectionModel().getSelectedItem();
147     if (media != null) {
148         // Kiểm tra nếu mục được chọn là Playable
149         if (media instanceof Playable) {
150             String message = "Now playing: " + media.getTitle();
151             showAlert(Alert.AlertType.INFORMATION, title:"Play Media", message);
152             ((Playable) media).play();
153         }
154     }
155 }
156
157 private void showAlert(Alert.AlertType alertType, String title, String message) {
158     Alert alert = new Alert(alertType);
159     alert.setTitle(title);
160     alert.setHeaderText(headerText:null);
161     alert.setContentText(message);
162     alert.showAndWait();
163 }
164
165 }
```

```
Settings  CartScreen.java U X

AIMSProject > src > hust > soict > dsai > aims > screen > CartScreen.java > ...
1  package hust.soict.dsai.aims.screen;
2
3  import hust.soict.dsai.aims.cart.*;
4  import hust.soict.dsai.aims.media.*;
5
6  import javax.swing.JFrame;
7  import javafx.application.Platform;
8  import javafx.embed.swing.JFXPanel;
9  import javafx.fxml.FXMLLoader;
10 import javafx.scene.Parent;
11 import javafx.scene.Scene;
12
13 import javax.swing.*;
14 import java.io.IOException;
15
16 @SuppressWarnings("unused")
17 public class CartScreen extends JFrame {
18     private Cart cart;
19
20     public CartScreen(Cart cart) {
21         super(); // Gọi constructor của JFrame;
22         this.cart = cart;
23
24         // Tạo JFXPanel để nhúng JavaFX vào Swing
25         JFXPanel fxPanel = new JFXPanel();
26         this.add(fxPanel);
27
28         this.setTitle(title:"Cart");
29         this.setVisible(b:true);
30
31         // Chạy JavaFX trên một luồng riêng biệt
32         Platform.runLater(new Runnable() {
33             @Override
34             public void run() {
35                 try {
36                     FXMLLoader loader = new FXMLLoader(getClass().getResource(name:"cart.fxml"));
37                     CartScreenController controller = new CartScreenController(cart);
38                     loader.setController(controller);
39
40                     Parent root = loader.load(); // nạp root node từ FXML
41                     fxPanel.setScene(new Scene(root));
42                 } catch (IOException e) {
43                     e.printStackTrace();
44                 }
45             }
46         });
47     }
48 }
49
```

```

Settings  x  AddBookToStoreScreen.java U x
AIMSProject > src > hust > soict > dsai > aims > screen > AddBookToStoreScreen.java > AddBookToStoreScreen
1  package hust.soict.dsai.aims.screen;
2
3  import hust.soict.dsai.aims.media.Book;
4  import hust.soict.dsai.aims.store.Store;
5
6  import javax.swing.*;
7  import java.awt.*;
8
9  @SuppressWarnings("unused")
10 public class AddBookToStoreScreen extends AddItemToStoreScreen {
11
12     public AddBookToStoreScreen(Store store, StoreScreen storeScreen) {
13         super(store, storeScreen);
14         setTitle(title:"Add Book to Store");
15     }
16
17     @Override
18     protected void addItemToStore() {
19         try {
20             String title = tfTitle.getText();
21             String category = tfCategory.getText();
22             float cost = Float.parseFloat(tfCost.getText());
23
24             Book book = new Book(title, category, cost);
25             store.addMedia(book); // Có thể ném ngoại lệ
26
27             JOptionPane.showMessageDialog(this, message:"Book added to store successfully!");
28             storeScreen.refreshCenterPanel();
29             dispose();
30         } catch (NumberFormatException e) {
31             JOptionPane.showMessageDialog(this, message:"Invalid cost value!", title:"Error", JOptionPane.ERROR_MESSAGE);
32         } catch (Exception e) {
33             JOptionPane.showMessageDialog(this, message:"Failed to add book to store!", title:"Error", JOptionPane.ERROR_MESSAGE);
34         }
35     }
36 }
37

```

```

Settings  x  AddCDToStoreScreen.java U x
AIMSProject > src > hust > soict > dsai > aims > screen > AddCDToStoreScreen.java > AddCDToStoreScreen
1  package hust.soict.dsai.aims.screen;
2
3  import hust.soict.dsai.aims.media.CompactDisc;
4  import hust.soict.dsai.aims.store.Store;
5
6  import javax.swing.*;
7  import java.awt.*;
8
9  @SuppressWarnings("unused")
10 public class AddCDToStoreScreen extends AddItemToStoreScreen {
11
12     public AddCDToStoreScreen(Store store, StoreScreen storeScreen) {
13         super(store, storeScreen);
14         setTitle(title:"Add CD to Store");
15     }
16
17     @Override
18     protected void addItemToStore() {
19         try {
20             String title = tfTitle.getText();
21             String category = tfCategory.getText();
22             float cost = Float.parseFloat(tfCost.getText());
23             String artist = JOptionPane.showInputDialog(this, message:"Enter artist:");
24
25             try {
26                 CompactDisc cd = new CompactDisc(title, category, cost, artist, length:0, director:"Unknown");
27                 store.addMedia(cd);
28                 JOptionPane.showMessageDialog(this, message:"CD added to store successfully!");
29             } catch (Exception e) {
30                 JOptionPane.showMessageDialog(this, "Error: " + e.getMessage(), title:"Error", JOptionPane.ERROR_MESSAGE);
31             }
32
33             storeScreen.refreshCenterPanel();
34             dispose();
35         } catch (NumberFormatException e) {
36             JOptionPane.showMessageDialog(this, message:"Invalid cost value!", title:"Error", JOptionPane.ERROR_MESSAGE);
37         }
38     }
39 }
40

```



```
Settings x AddDVDToStoreScreen.java U x
AIMSProject > src > hust > soict > dsai > aims > screen > AddDVDToStoreScreen.java > AddDVDToStoreScreen
1 package hust.soict.dsai.aims.screen;
2
3 import hust.soict.dsai.aims.media.DigitalVideoDisc;
4 import hust.soict.dsai.aims.store.Store;
5
6 import javax.swing.*;
7 import java.awt.*;
8
9 @SuppressWarnings("unused")
10 public class AddDVDToStoreScreen extends AddItemToStoreScreen {
11
12     public AddDVDToStoreScreen(Store store, StoreScreen storeScreen) {
13         super(store, storeScreen);
14         setTitle(title:"Add DVD to Store");
15     }
16
17     @Override
18     protected void addItemToStore() {
19         try {
20             String title = tfTitle.getText();
21             String category = tfCategory.getText();
22             float cost = Float.parseFloat(tfCost.getText());
23             String director = JOptionPane.showInputDialog(this, message:"Enter director:");
24             int length = Integer.parseInt(JOptionPane.showInputDialog(this, message:"Enter length in minutes:"));
25
26             DigitalVideoDisc dvd = new DigitalVideoDisc(title, category, director, length, cost);
27             store.addMedia(dvd); // Có thể ném ngoại lệ
28
29             JOptionPane.showMessageDialog(this, message:"DVD added to store successfully!");
30             storeScreen.refreshCenterPanel();
31             dispose();
32         } catch (NumberFormatException e) {
33             JOptionPane.showMessageDialog(this, message:"Invalid input!", title:"Error", JOptionPane.ERROR_MESSAGE);
34         } catch (Exception e) {
35             JOptionPane.showMessageDialog(this, message:"Failed to add DVD to store!", title:"Error", JOptionPane.ERROR_MESSAGE);
36         }
37     }
38
39 }
40
```

```
Settings  AddItemToStoreScreen.java U X
AIMSProject > src > hust > soict > dsai > aims > screen > AddItemToStoreScreen.java > AddItemToStoreScreen > AddItemToStoreScreen(Store
1  package hust.soict.dsai.aims.screen;
2
3  import hust.soict.dsai.aims.store.Store;
4
5  import javax.swing.*;
6  import java.awt.*;
7
8  public abstract class AddItemToStoreScreen extends JFrame {
9      protected Store store;
10     protected StoreScreen storeScreen;
11     protected JTextField tfTitle, tfCategory, tfCost;
12
13     @SuppressWarnings("unused")
14     public AddItemToStoreScreen(Store store, StoreScreen storeScreen) {
15         this.store = store;
16         this.storeScreen = storeScreen;
17
18         setTitle(title:"Add Item to Store");
19         setSize(width:400, height:300);
20         setDefaultCloseOperation(DISPOSE_ON_CLOSE);
21         setLayout(new GridLayout(rows:4, cols:2, hgap:10, vgap:10));
22
23         add(new JLabel(text:"Title:"));
24         tfTitle = new JTextField();
25         add(tfTitle);
26
27         add(new JLabel(text:"Category:"));
28         tfCategory = new JTextField();
29         add(tfCategory);
30
31         add(new JLabel(text:"Cost:"));
32         tfCost = new JTextField();
33         add(tfCost);
34
35         JButton btnAdd = new JButton(text:"Add");
36         btnAdd.addActionListener(e -> addItemToStore());
37         add(btnAdd);
38
39         setVisible(b:true);
40     }
41
42     protected abstract void addItemToStore();
43 }
44
```

```

Settings  X  MediaStore.java U X
AIMSProject > src > hust > soict > dsai > aims > screen > MediaStore.java > MediaStore
1  package hust.soict.dsai.aims.screen;
2
3  import hust.soict.dsai.aims.media.Media;
4  import hust.soict.dsai.aims.cart.Cart;
5  import hust.soict.dsai.aims.exception.*;
6
7  import javax.swing.*;
8  import java.awt.*;
9
10 public class MediaStore extends JPanel {
11     @SuppressWarnings("unused")
12     private final Media media;
13     @SuppressWarnings("unused")
14     private final Cart cart; // Thêm giỏ hàng
15
16     @SuppressWarnings("unused")
17     public MediaStore(Media media, Cart cart) { // Nhận thêm Cart
18         this.media = media;
19         this.cart = cart;
20         this.setLayout(new BorderLayout(this, BorderLayout.V_AXIS));
21
22         // Tạo nhãn hiển thị tiêu đề
23         JLabel title = new JLabel(media.getTitle());
24         title.setFont(new Font(title.getFont().getName(), Font.PLAIN, size:16));
25         title.setAlignmentX(CENTER_ALIGNMENT);
26
27         // Tạo nhãn hiển thị giá
28         JLabel cost = new JLabel(String.format(Format: "%.2f $", media.getCost()));
29         cost.setAlignmentX(CENTER_ALIGNMENT);
30
31         // Tạo nút chức năng
32         JPanel buttonPanel = new JPanel();
33         buttonPanel.setLayout(new FlowLayout(FlowLayout.CENTER));
34
35         // Nút Add to Cart
36         JButton btnAddToCart = new JButton(text: "Add to cart");
37         btnAddToCart.addActionListener(e -> {
38             try {
39                 cart.addMedia(media); // Thêm vào giỏ hàng
40                 JOptionPane.showMessageDialog(this, media.getTitle() + " added to cart.");
41                 new CartScreen(cart); // Mở màn hình giỏ hàng
42             } catch (LimitExceededException ex) {
43                 JOptionPane.showMessageDialog(this, "Cannot add more items: " + ex.getMessage(), title: "Error",
44                     JOptionPane.ERROR_MESSAGE);
45             }
46         });
47
48         buttonPanel.add(btnAddToCart);
49
50         // Nút Play (nếu có)
51         if (media instanceof hust.soict.dsai.aims.media.Playable) {
52             JButton btnPlay = new JButton(text: "Play");
53             btnPlay.addActionListener(e -> JOptionPane.showMessageDialog(this, "Playing " + media.getTitle()));
54             buttonPanel.add(btnPlay);
55         }
56
57         // Thêm các thành phần vào MediaStore
58         this.add(Box.createVerticalGlue());
59         this.add(title);
60         this.add(cost);
61         this.add(buttonPanel);
62         this.add(Box.createVerticalGlue());
63     }
64 }

```

```

Settings X Store.java M X
AIMSProject > src > hust > soict > dsai > aims > store > Store.java > Store > removeMedia(Media)
1 package hust.soict.dsai.aims.store;
2
3 import hust.soict.dsai.aims.media.Media;
4
5 import java.util.ArrayList;
6
7 public class Store {
8     private ArrayList<Media> itemsInStore = new ArrayList<>();
9
10    public void addMedia(Media media) {
11        itemsInStore.add(media);
12    }
13
14    public void removeMedia(Media media) {
15        itemsInStore.remove(media);
16    }
17
18    public ArrayList<Media> getItemsInStore() {
19        return itemsInStore;
20    }
21 }
22

```

```

Settings X Aims.java M X
AIMSProject > src > hust > soict > dsai > aims > Aims.java > Aims > main(String[])
1 package hust.soict.dsai.aims;
2
3 import hust.soict.dsai.aims.media.*;
4 import hust.soict.dsai.aims.exception.PlayerException;
5
6 import javax.swing.*;
7 import java.util.ArrayList;
8
9 public class Aims {
10    Run | Debug
11    public static void main(String[] args) {
12        // Tạo danh sách Media
13        ArrayList<Media> medialist = new ArrayList<>();
14        try {
15            // Thêm DigitalVideoDisc với độ dài hợp lệ
16            DigitalVideoDisc dvd1 = new DigitalVideoDisc(title:"Inception", category:"Sci-Fi", director:"Christopher Nolan", length:148, cost:19.99f);
17            medialist.add(dvd1);
18
19            // Thêm DigitalVideoDisc với độ dài không hợp lệ
20            DigitalVideoDisc dvd2 = new DigitalVideoDisc(title:"Faulty DVD", category:"Drama", director:"Unknown Director", length:0, cost:14.99f);
21            medialist.add(dvd2);
22
23            // Thêm CompactDisc (giả sử độ dài hợp lệ)
24            CompactDisc cd = new CompactDisc(title:"Top Hits", category:"Music", cost:12.99f, artist:"Various Artists", length:50, director:"Music Director");
25            medialist.add(cd);
26
27        } catch (Exception e) {
28            System.err.println("Error creating media: " + e.getMessage());
29        }
30
31        // Gọi play() cho từng Media và xử lý ngoại lệ
32        for (Media media : medialist) {
33            if (media instanceof Playable) {
34                try {
35                    ((Playable) media).play(); // Gọi phương thức play()
36                } catch (PlayerException e) {
37                    System.err.println("Exception occurred while playing media: " + e.getMessage());
38                    e.printStackTrace();
39
40                    // Hiển thị Dialog thông báo lỗi
41                    JOptionPane.showMessageDialog(parentComponent:null, e.getMessage(), title:"Illegal Media Length",
42                                                JOptionPane.ERROR_MESSAGE);
43                }
44            } else {
45                System.out.println("Media " + media.getTitle() + " is not playable.");
46            }
47        }
48    }
49 }

```

```

Settings × TestMediaEquals.java U ×
AIMSProject > src > hust > soict > dsai > aims > TestMediaEquals.java > ...
1 package hust.soict.dsai.aims;
2
3 import hust.soict.dsai.aims.media.Book;
4 import hust.soict.dsai.aims.media.Media;
5
6 public class TestMediaEquals {
7     @SuppressWarnings("unlikely-arg-type")
8     Run | Debug
9     public static void main(String[] args) throws Exception {
10         Media book1 = new Book(title:"Harry Potter", category:"Fantasy", cost:19.99f);
11         Media book2 = new Book(title:"Harry Potter", category:"Adventure", cost:25.99f);
12         Media book3 = new Book(title:"Lord of the Rings", category:"Fantasy", cost:29.99f);
13
14         // Kiểm tra equals()
15         System.out.println("book1 equals book2? " + book1.equals(book2)); // true
16         System.out.println("book1 equals book3? " + book1.equals(book3)); // false
17
18         // Kiểm tra null và khác kiểu
19         System.out.println("book1 equals null? " + book1.equals(obj:null)); // false
20         System.out.println("book1 equals String? " + book1.equals(obj:"Harry Potter")); // false
21     }
22 }

```

pkg



