

ĐẠI HỌC BÁCH KHOA HÀ NỘI
TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

BÁO CÁO THỰC HÀNH
IT3103-744528-2024.1
BÀI THỰC HÀNH 5

Họ và tên sv: Nguyễn Văn Đăng
Phúc Hưng

Lớp: **K67-CNTT Việt Pháp 01**

GVHD: Lê Thị Hoa

TA: **Tạ Trung Hiếu**

Hà Nội 12/2024

BÁO CÁO THỰC HÀNH LAB 5

LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG

Contents

1.	Swing components	4
1.1	AWTAccumulator	4
1.2	SwingAccumulator	5
2	Organizing Swing components with Layout Managers	6
2.1	Code	6
2.2	Demo	8
3	Create a graphical user interface for AIMS with Swing	9
3.1	Create class StoreScreen	9
3.2	Create class MediaStore	13
3.3	Demo	14
4	JavaFX API.....	16
4.1	Create class Painter	16
4.2	Create Painter.fxml.....	16
4.3	Create class PainterController	17
5	View Cart Screen	19
5.1	Create cart.fxml	19
5.2	Create class CartScreen	20
5.3	Create class CartScreenController	21
5.4	Demo	22
6	Updating buttons based on selected item in TableView – ChangeListener	22
6.1	Edit class CartScreenController.....	22
6.2	Demo	23
7	Deleting a media.....	24
7.1	Code	24
7.2	Demo	25
8	Complete the Aims GUI application	26
9	Use case Diagram	30

10	Class Diagram	31
----	---------------------	----

Figure 1.1: Source code of AWTAccumulator.....	4
Figure 1.2: Demo of AWTAccumulator.....	5
Figure 1.3: Source code of SwingAccumulator.....	5
Figure 1.4: Demo of SwingAccumulator.....	6
Figure 2.1: Source code of NumberGrid 1.....	6
Figure 2.2: Source code of NumberGrid 2.....	7
Figure 2.3: Demo buttons 0-9.....	8
Figure 2.4: Demo DEL button.....	8
Figure 2.5: Demo C button.....	8
Figure 3.1: Class StoreScreen 1.....	9
Figure 3.2: Class StoreScreen 2.....	10
Figure 3.3: Class StoreScreen 3.....	10
Figure 3.4: Class StoreScreen 4.....	11
Figure 3.5: Class StoreScreen 5.....	11
Figure 3.6: Class StoreScreen 6.....	12
Figure 3.7: Class MediaStore 1.....	13
Figure 3.8: Class MediaStore 2.....	13
Figure 3.9: Class MediaStore 3.....	14
Figure 3.10: StoreScreen.....	14
Figure 3.11 Demo Add to cart button.....	15
Figure 3.12 Demo Play button.....	15
Figure 3.13 Demo View cart button.....	15
Figure 4.1: Class Painter.....	16
Figure 4.2: Painter.fxml 1.....	16
Figure 4.3: Painter.fxml 2.....	17
Figure 4.4: PainterController.....	17
Figure 4.5: Use Pen.....	18
Figure 4.6: Use Eraser.....	18
Figure 4.7: Clear button.....	18
Figure 5.1: Cart.fxml 1.....	19
Figure 5.2: Cart.fxml 2.....	19
Figure 5.3: Cart.fxml 3.....	20
Figure 5.4: CartScreen class.....	20
Figure 5.5: CartScreenController 1.....	21
Figure 5.6: CartScreenController 2.....	21
Figure 5.7: Demo CartScreen.....	22
Figure 6.1: CartScreenController 1.....	22
Figure 6.2: CartScreenController 2.....	23
Figure 6.3: Demo media playable.....	23
Figure 6.4: Demo media unplayable.....	24
Figure 7.1: btnRemovePressed Method.....	24
Figure 7.2: button Remove.....	25
Figure 7.3: button Remove.....	25
Figure 8.1: Store before add book.....	26

Figure 8.2: Add book	26
Figure 8.3: Store after add book.....	27
Figure 8.4: Add CD.....	27
Figure 8.5: Store after add CD	28
Figure 8.6 Add DVD	28
Figure 8.7: Store after add DVD.....	29
Figure 8.8: Cart	29
Figure 8.9: Exception	30

1. Swing component

1.1. Lớp AWTAccumulator

```
package hust.soict.dsai.swing;
import java.awt.*;

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

    public AWTAccumulator() {
        setLayout(new GridLayout(2, 2));
        add(new Label("Enter an integer: "));

        tfInput = new TextField(10);
        add(tfInput);
        tfInput.addActionListener(new TFInputListener());
        add(new Label("The accumulated sum is: "));

        tfOutput = new TextField(10);
        tfOutput.setEditable(false);
        add(tfOutput);

        setTitle("AWT Accumulator");
        setSize(350, 120);
        setVisible(true);
    }

    private class TFInputListener implements ActionListener {
        @Override
        public void actionPerformed(ActionEvent evt) {
            int numberIn = Integer.parseInt(tfInput.getText());
            sum += numberIn;
            tfInput.setText("");
            tfOutput.setText(sum + "");
        }
    }

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



Figure 1: Mã nguồn và giao diện lớp AWTAccumulator

1.2. Lớp SwingAccumulator

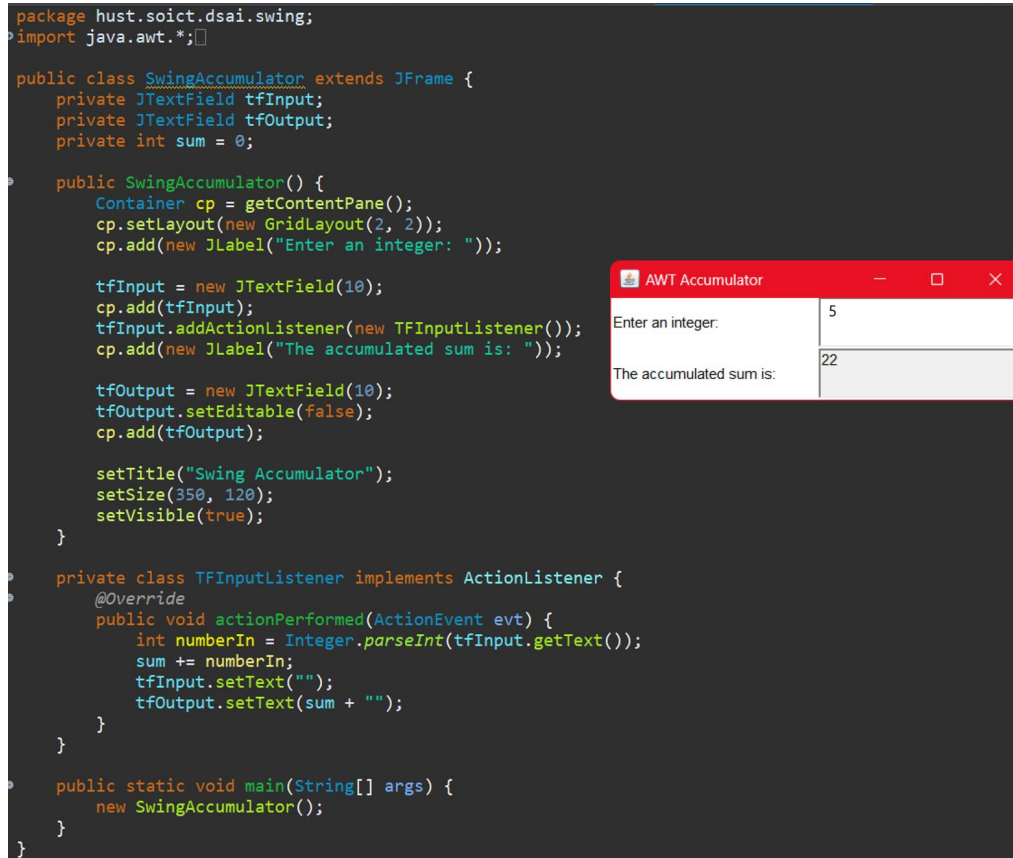


Figure 2: Giao diện và mã nguồn lớp SwingAccumulator

1.3: Compare Swing and AWT elements

- The top-level containers in Swing and AWT:
 - Swing: JFrame, JDialog, JApplet, Jwindow
 - AWT: Frame, Dialog, Applet
- The class name of components in AWT and corresponding class's name in Swing

AWT	Swing
Button	JButton
Label	JLabel
TextField	JTextField
TextArea	JTextArea
Checkbox	JCheckBox
Choice	JComboBox
List	JList
ScrollBar	JScrollBar
Panel	JPanel
Canvas	
MenuBar	JMenuBar
Menu	JMenu
MenuItem	JMenuItem

- The main differences between Swing and AWT:
 - AWT uses native components provided by the operating system (heavyweight), whereas Swing is entirely written in Java and rendered on a lightweight layer.
 - Swing offers many extended components that AWT does not, such as JTable, JTree, JTabbedPane, and more.

2. Organizing Swing components with Layout Managers

```
package hust.soict.dsai.swing;
import java.awt.*;

public class NumberGrid extends JFrame {
    private JButton[] btnNumbers = new JButton[10];
    private JButton btnDelete, btnReset;
    private JTextField tfDisplay;

    public NumberGrid() {
        tfDisplay = new JTextField();
        tfDisplay.setComponentOrientation(ComponentOrientation.RIGHT_TO_LEFT);

        JPanel panelButtons = new JPanel(new GridLayout(4, 3));
        addButtons(panelButtons);

        Container cp = getContentPane();
        cp.setLayout(new BorderLayout());
        cp.add(tfDisplay, BorderLayout.NORTH);
        cp.add(panelButtons, BorderLayout.CENTER);

        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setTitle("Number Grid");
        setSize(200, 200);
        setVisible(true);
    }

    void addButtons(JPanel panelButtons) {
        ActionListener btnListener = new ButtonListener();
        for (int i = 1; i <= 9; i++) {
            btnNumbers[i] = new JButton("" + i);
            panelButtons.add(btnNumbers[i]);
            btnNumbers[i].addActionListener(btnListener);
        }

        btnDelete = new JButton("DEL");
        panelButtons.add(btnDelete);
        btnDelete.addActionListener(btnListener);

        btnNumbers[0] = new JButton("0");
        panelButtons.add(btnNumbers[0]);
        btnNumbers[0].addActionListener(btnListener);

        btnReset = new JButton("C");
        panelButtons.add(btnReset);
        btnReset.addActionListener(btnListener);
    }

    private class ButtonListener implements ActionListener {
        @Override
        public void actionPerformed(ActionEvent e) {
            String button = e.getActionCommand();
            if (button.charAt(0) >= '0' && button.charAt(0) <= '9') {
                tfDisplay.setText(tfDisplay.getText() + button);
            } else if (button.equals("DEL")) {
                String oldText = tfDisplay.getText();
                if (oldText != null && oldText.length() > 0) {
                    String newText = oldText.substring(0, oldText.length() - 1);
                    tfDisplay.setText(newText);
                }
            } else {
                tfDisplay.setText("");
            }
        }
    }

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




Figure 3: Mã nguồn và giao diện lớp NumberGrid

3. Create a graphical user interface for AIMS with Swing

3.1. Lớp StoreScreen

```

package hust.soict.dsai.aims.screen;
import java.awt.*;

public class StoreScreen extends JFrame {
    private Store store;
    private Cart cart;
    private JScrollPane center;

    JPanel createNorth() {
        JPanel north = new JPanel();
        north.setLayout(new BorderLayout(P_AXIS));
        north.add(createMenuBar());
        north.add(createHeader());
        return north;
    }

    JMenuBar createMenuBar() {
        JMenu snUpdateStore = new JMenu("Update store");
        JMenuItem snAddBook = new JMenuItem("Add Book");
        JMenuItem snAddCD = new JMenuItem("Add CD");
        JMenuItem snAddDVD = new JMenuItem("Add DVD");
        snUpdateStore.add(snAddBook);
        snUpdateStore.add(snAddCD);
        snUpdateStore.add(snAddDVD);

        JMenu menu = new JMenu("Options");
        menu.add(snUpdateStore);
        menu.add(new JMenuItem("View store"));
        JMenuItem snViewCart = new JMenuItem("View cart");
        menu.add(snViewCart);

        snViewCart.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                new CartScreen(store, cart);
            }
        });

        snAddBook.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                new AddBookToStoreScreen(store, cart);
            }
        });

        snAddCD.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                new AddCDToStoreScreen(store, cart);
            }
        });

        snAddDVD.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                new AddDVDToStoreScreen(store, cart);
            }
        });

        JMenuBar menubar = new JMenuBar();
        menubar.setLayout(new FlowLayout(FlowLayout.LEFT));
        menubar.add(menu);

        return menubar;
    }

    JPanel createHeader() {
        JPanel header = new JPanel();
        header.setLayout(new BorderLayout(X_AXIS));

        JLabel title = new JLabel("AIMS");
        title.setFont(new Font(title.getFont().getName(), Font.PLAIN, 50));
        title.setForeground(Color.MAGENTA);

        JButton cartBtn = new JButton("View cart");
        cartBtn.setPreferredSize(new Dimension(100, 50));
        cartBtn.setMaximumSize(new Dimension(100, 50));

        cartBtn.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                new CartScreen(store, cart);
            }
        });

        header.add(Box.createRigidArea(new Dimension(10, 10)));
        header.add(title);
        header.add(Box.createRigidArea(new Dimension(10, 10)));
        header.add(cartBtn);
        header.add(Box.createRigidArea(new Dimension(10, 10)));

        return header;
    }

    JScrollPane createCenter(Cart cart) {
        JPanel centerPanel = new JPanel();
        centerPanel.setLayout(new GridLayout(8, 3, 2, 2));

        ArrayList<Media> mediaInStore = store.getItemsInStore();

        for (int i = 0; i < mediaInStore.size(); i++) {
            MediaStore cell = new MediaStore(mediaInStore.get(i), cart);
            centerPanel.add(cell);
        }

        JScrollPane scrollPane = new JScrollPane(centerPanel);
        scrollPane.setPreferredSize(new Dimension(400, 300));

        return scrollPane;
    }

    public StoreScreen(Store store, Cart cart) {
        this.store = store;
        this.cart = cart;
        Container cp = getContentPane();
        cp.setLayout(new BorderLayout());

        cp.add(createNorth(), BorderLayout.NORTH);
        center = createCenter(cart);
        cp.add(center, BorderLayout.CENTER);

        setVisible(true);
        setTitle("Store");
        setSize(1024, 768);
    }
}

```

Figure 4: Mã nguồn lớp StoreScreen

3.2. Lớp MediaStore

```

package hust.soict.dsai.aims.screen;
import java.awt.*;

public class MediaStore extends JPanel {
    private Media media;
    public MediaStore(Media media, Cart cart) {
        this.media = media;
        this.setLayout(new BoxLayout(this, BoxLayout.Y_AXIS));

        JLabel title = new JLabel(media.getTitle());
        title.setFont(new Font(title.getFont().getName(), Font.PLAIN, 20));
        title.setAlignmentX(CENTER_ALIGNMENT);

        JLabel cost = new JLabel("" + media.getCost() + " $");
        cost.setAlignmentX(CENTER_ALIGNMENT);

        JPanel container = new JPanel();
        container.setLayout(new FlowLayout(FlowLayout.CENTER));

        JButton addToCartButton = new JButton("Add to cart");
        container.add(addToCartButton);

        addToCartButton.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                try {
                    cart.addMedia(media);
                } catch (LimitExceededException e1) {
                    e1.printStackTrace();
                }
                JOptionPane.showMessageDialog(null, media.getTitle() + " has been added to the cart.",
                    "Cart", JOptionPane.INFORMATION_MESSAGE);
            }
        });

        if (media instanceof Playable) {
            JButton playButton = new JButton("Play");
            container.add(playButton);

            playButton.addActionListener(new ActionListener() {
                @Override
                public void actionPerformed(ActionEvent e) {
                    playMedia((Playable) media);
                }
            });
        }

        this.add(Box.createVerticalGLue());
        this.add(title);
        this.add(cost);
        this.add(Box.createVerticalGLue());
        this.add(container);

        this.setBorder(BorderFactory.createLineBorder(Color.BLACK));
    }

    private void playMedia(Playable media) {
        JDialog playDialog = new JDialog((Frame) null, "Playing media", true);
        playDialog.setLayout(new BorderLayout());
        playDialog.setSize(300, 200);
        playDialog.setLocationRelativeTo(null);

        JLabel playLabel = new JLabel("Playing: " + media.getTitle(), JLabel.CENTER);
        playLabel.setFont(new Font(playLabel.getFont().getName(), Font.PLAIN, 16));

        JTextArea outputArea = new JTextArea(10, 20);
        outputArea.setEditable(false);
        try {
            outputArea.setText(media.play());
        } catch (PlayerException e) {
            e.printStackTrace();
        }
        outputArea.setWrapStyleWord(true);
        outputArea.setLineWrap(true);
        JScrollPane scrollPane = new JScrollPane(outputArea);

        playDialog.add(playLabel, BorderLayout.NORTH);
        playDialog.add(scrollPane, BorderLayout.CENTER);

        JButton closeButton = new JButton("Close");
        closeButton.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                playDialog.dispose();
            }
        });

        playDialog.add(closeButton, BorderLayout.SOUTH);
        playDialog.setVisible(true);
    }
}

```

Figure 5: Mã nguồn lớp MediaStore

3.3. Chức năng của nút Play và nút Add to cart

3.3.1. Nút Play

```
private void playMedia(Playable media) {
    JDialog playDialog = new JDialog((Frame) null, "Playing media", true);
    playDialog.setLayout(new BorderLayout());
    playDialog.setSize(300, 200);
    playDialog.setLocationRelativeTo(null);

    JLabel playLabel = new JLabel("Playing: " + media.getTitle(), JLabel.CENTER);
    playLabel.setFont(new Font(playLabel.getFont().getName(), Font.PLAIN, 16));

    JTextArea outputArea = new JTextArea(10, 20);
    outputArea.setEditable(false);
    try {
        outputArea.setText(media.play());
    } catch (PlayerException e) {
        e.printStackTrace();
    }
    outputArea.setWrapStyleWord(true);
    outputArea.setLineWrap(true);
    JScrollPane scrollPane = new JScrollPane(outputArea);

    playDialog.add(playLabel, BorderLayout.NORTH);
    playDialog.add(scrollPane, BorderLayout.CENTER);

    JButton closeButton = new JButton("Close");
    closeButton.addActionListener(new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent e) {
            playDialog.dispose();
        }
    });

    playDialog.add(closeButton, BorderLayout.SOUTH);
    playDialog.setVisible(true);
}
}
```

Figure 6: Mã nguồn nút Play của MediaStore

3.3.2. Nút Add to cart

```
JButton addToCartButton = new JButton("Add to cart");
container.add(addToCartButton);

addToCartButton.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        try {
            cart.addMedia(media);
        } catch (LimitExceededException e1) {
            e1.printStackTrace();
        }
        JOptionPane.showMessageDialog(null, media.getTitle() + " has been added to the cart.",
            "Cart", JOptionPane.INFORMATION_MESSAGE);
    }
});
```

Figure 7: Mã nguồn nút Add to cart của MediaStore

3.4. Giao diện

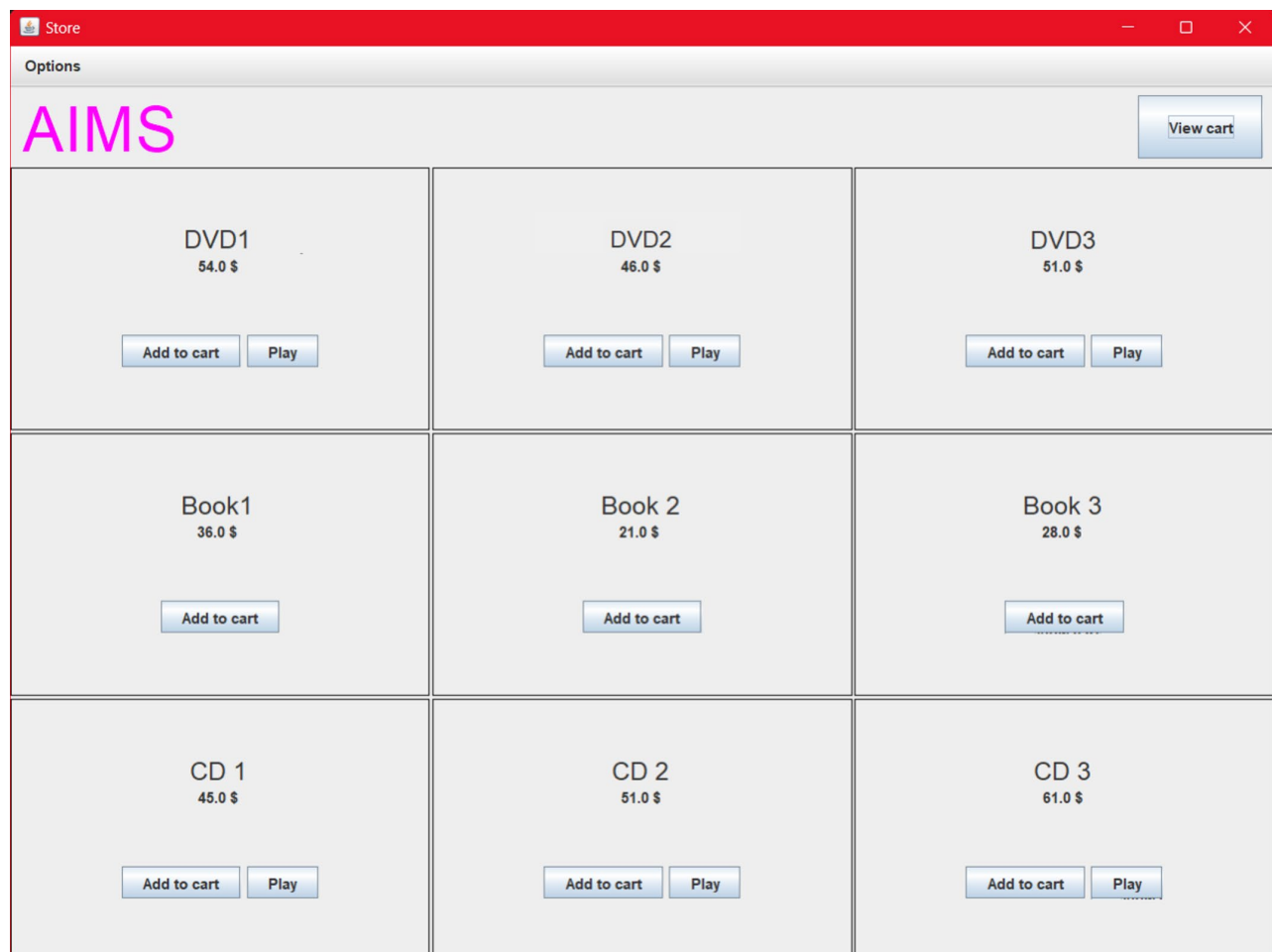


Figure 8: Giao diện Store



Figure 9: Giao diện Play DVD media trong

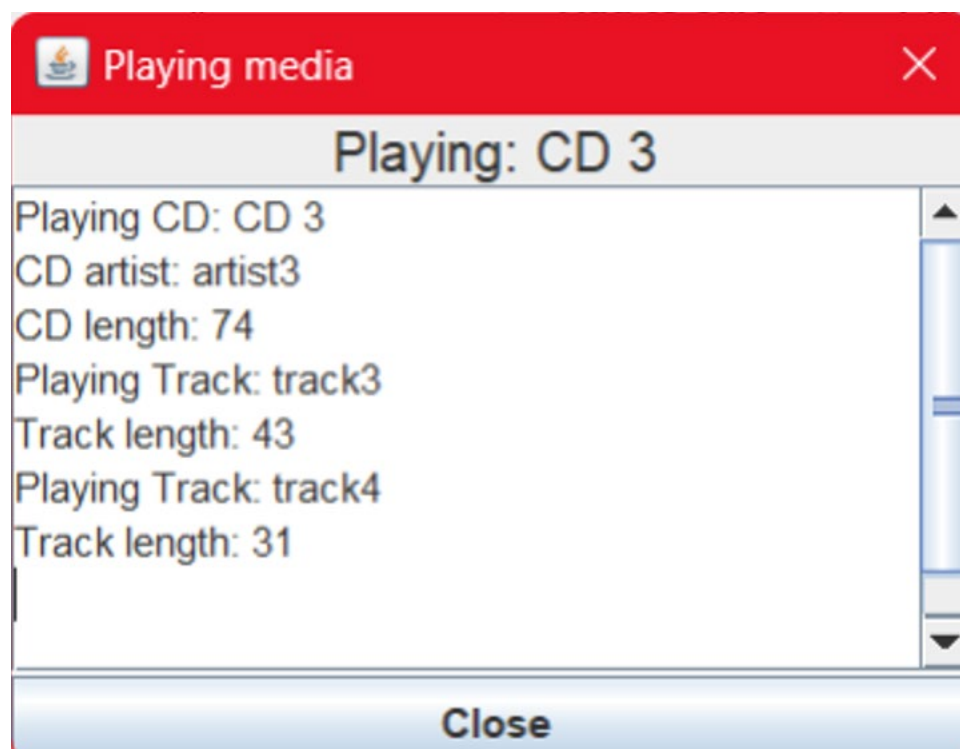


Figure 10: Giao diện Play CD media trong Store

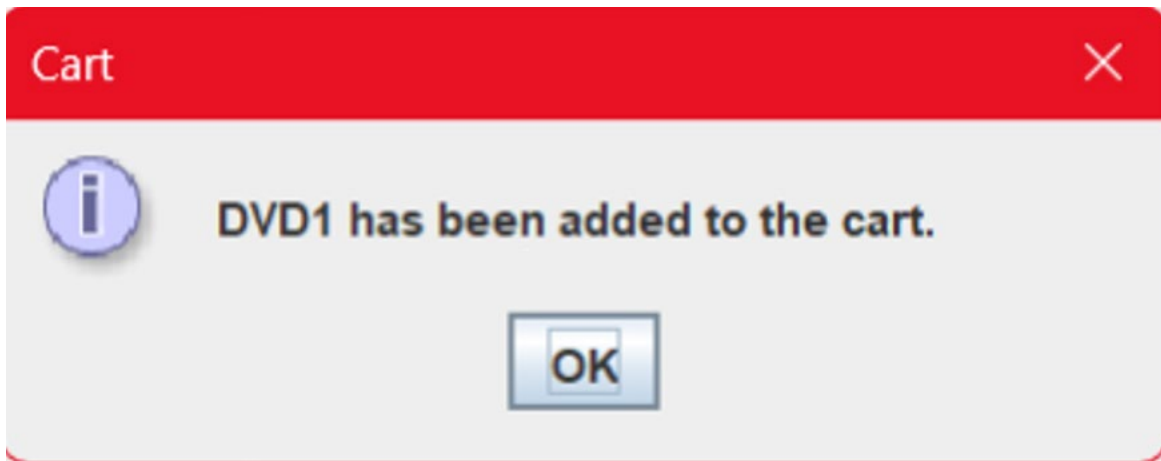


Figure 11: Thông báo thêm media vào cart

4. JavaFX API

```
<?xml version="1.0" encoding="UTF-8"?>

<?import javafx.geometry.Insets?>
<?import javafx.scene.control.Button?>
<?import javafx.scene.control.RadioButton?>
<?import javafx.scene.control.ToggleGroup?>
<?import javafx.scene.layout.AnchorPane?>
<?import javafx.scene.layout.BorderPane?>
<?import javafx.scene.layout.Pane?>
<?import javafx.scene.layout.VBox?>
<?import javafx.scene.text.Font?>

<BorderPane maxHeight="-Infinity" maxWidth="-Infinity" minHeight="-Infinity" minWidth="-Infinity" prefHeight="480.0" prefWidth="640.0"
xmlns="http://javafx.com/javafx/23.0.1" xmlns:fx="http://javafx.com/fxml/1" fx:controller="hust.soict.dsai.javafx.PaintController">
  <padding>
    <Insets bottom="8.0" left="8.0" right="8.0" top="8.0" />
  </padding>
  <center>
    <Pane fx:id="drawingAreaPane" onMouseDragged="#drawingAreaMouseDragged" prefHeight="200.0" prefWidth="200.0" style="-fx-background-color: white;" BorderPane.alignment="CENTER" />
  </center>
  <left>
    <VBox maxHeight="1.7976931348623157E308" spacing="8.0" BorderPane.alignment="CENTER">
      <children>
        <TitledPane animated="false" text="Tools">
          <content>
            <AnchorPane maxHeight="1.7976931348623157E308" maxWidth="1.7976931348623157E308" minHeight="0.0" minWidth="0.0">
              <children>
                <RadioButton layoutX="14.0" layoutY="5.0" mnemonicParsing="false" onAction="#togglePen" selected="true" text="Pen">
                  <toggleGroup>
                    <ToggleGroup fx:id="tools_group" />
                  </toggleGroup>
                </RadioButton>
                <RadioButton layoutX="14.0" layoutY="31.0" mnemonicParsing="false" onAction="#toggleEraser" text="Eraser" toggleGroup="$tools_group" />
              </children>
            </AnchorPane>
          </content>
        </TitledPane>
        <Button maxHeight="1.7976931348623157E308" mnemonicParsing="false" onAction="#clearButtonPressed" text="Clear">
          <font>
            <Font size="13.0" />
          </font>
        </Button>
      </children>
    </VBox>
  </left>
</BorderPane>
```

Figure 12: Mã nguồn Painter.fxml

4.2. Painter.java

```

package hust.soict.dsai.javafx;
import javafx.application.Application;

public class Painter extends Application {
    @Override
    public void start(Stage stage) throws Exception {
        Parent root = FXMLLoader.load(getClass().getResource("/hust/soict/dsai/javafx/Painter.fxml"));

        Scene scene = new Scene(root);
        stage.setTitle("Painter");
        stage.setScene(scene);
        stage.show();
    }

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

```

Figure 13: Mã nguồn Painter.java

4.3. PaintController.java

```

package hust.soict.dsai.javafx;
import javafx.event.ActionEvent;

public class PaintController {
    @FXML
    private Pane drawingAreaPane;
    private boolean drawingMode = true;

    @FXML
    void clearButtonPressed(ActionEvent event) {
        drawingAreaPane.getChildren().clear();
    }

    @FXML
    void drawingAreaMouseDragged(MouseEvent event) {
        Circle newCircle = new Circle(event.getX(), event.getY(), 4);
        if (drawingMode) {
            newCircle.setFill(Color.BLACK);
        } else {
            newCircle.setFill(Color.WHITE);
        }
        drawingAreaPane.getChildren().add(newCircle);
    }

    @FXML
    void togglePen() {
        drawingMode = true;
    }

    @FXML
    void toggleEraser() {
        drawingMode = false;
    }
}

```

Figure 14: Mã nguồn PaintController.java

4.4. Giao diện

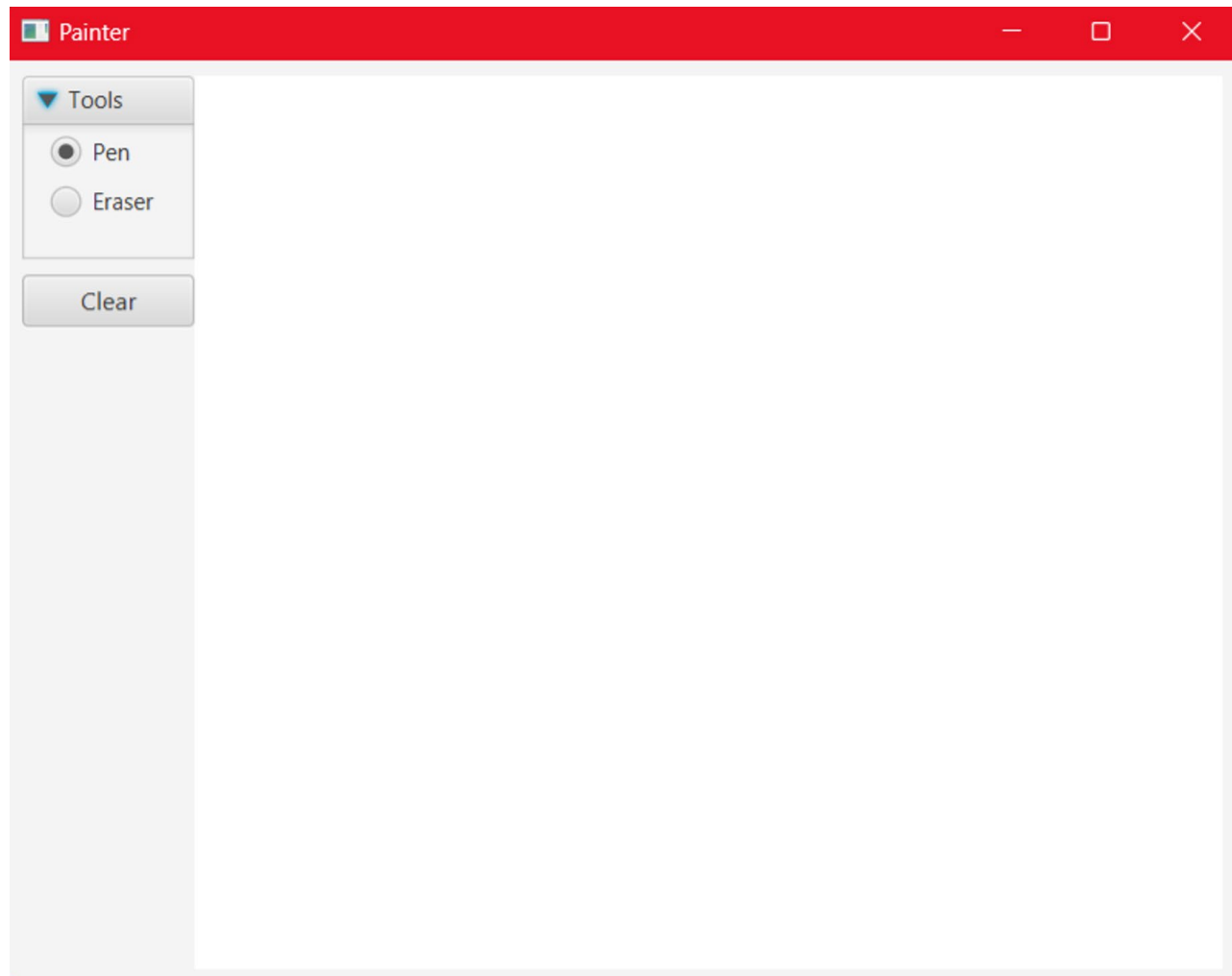


Figure 15: Giao diện Painter

5. Setting up the View Cart Screen with ScreenBuilder

5.1. Cart.fxml


```

<?xml version="1.0" encoding="UTF-8"?>
<?import javafx.geometry.Insets?>
<?import javafx.scene.control.Button?>
<?import javafx.scene.control.ButtonBar?>
<?import javafx.scene.control.Label?>
<?import javafx.scene.control.Menu?>
<?import javafx.scene.control.MenuBar?>
<?import javafx.scene.control.MenuItem?>
<?import javafx.scene.control.RadioButton?>
<?import javafx.scene.control.TableColumn?>
<?import javafx.scene.control.TableView?>
<?import javafx.scene.control.TextField?>
<?import javafx.scene.control.ToggleGroup?>
<?import javafx.scene.layout.BorderPane?>
<?import javafx.scene.layout.HBox?>
<?import javafx.scene.layout.VBox?>
<?import javafx.scene.text.Font?>

<BorderPane prefHeight="768.0" prefWidth="1024.0" xmlns="http://javafx.com/javafx/23.0.1" xmlns:fx="http://javafx.com/fxml/1">
    <top>
        <VBox prefWidth="100.0" BorderPane.alignment="CENTER">
            <children>
                <MenuBar>
                    <menu>
                        <MenuItem mnemonicParsing="false" text="Options">
                            <items>
                                <MenuItem mnemonicParsing="false" text="Update Store">
                                    <items>
                                        <MenuItem mnemonicParsing="false" onAction="#btnAddBookPressed" text="Add Book" />
                                        <MenuItem mnemonicParsing="false" onAction="#btnAddCDPressed" text="Add CD" />
                                        <MenuItem mnemonicParsing="false" onAction="#btnAddDVDPressed" text="Add DVD" />
                                    </items>
                                </MenuItem>
                                <MenuItem mnemonicParsing="false" text="View Cart" />
                                <MenuItem fx:id="menuItemViewStore" mnemonicParsing="false" onAction="#menuItemViewStorePressed" text="View Store" />
                            </items>
                        </MenuItem>
                    </menu>
                </MenuBar>
                <Label text="CART" textFill="AQUA">
                    <font>
                        <font size="50.0" />
                    </font>
                </Label>
            </children>
        </VBox>
    </top>
    <center>
        <VBox prefHeight="200.0" prefWidth="100.0" BorderPane.alignment="CENTER">
            <BorderPane.margin>
                <insets />
            </BorderPane.margin>
            <children>
                <HBox alignment="CENTER_LEFT" prefWidth="200.0" spacing="10.0">
                    <VBox.margin>
                        <insets />
                    </VBox.margin>
                    <padding>
                        <insets bottom="10.0" top="10.0" />
                    </padding>
                    <children>
                        <Label text="Filter" />
                        <TextField promptText="" />
                        <RadioButton mnemonicParsing="false" selected="true" text="By ID">
                            <toggleGroup>
                                <ToggleGroup fx:id="filterCategory" />
                            </toggleGroup>
                        </RadioButton>
                        <RadioButton mnemonicParsing="false" text="By Title" toggleGroup="$filterCategory" />
                    </children>
                </HBox>
                <TableView fx:id="tblMedia">
                    <columns>
                        <TableColumn fx:id="colMediaTitle" prefWidth="75.0" text="Title" />
                        <TableColumn fx:id="colMediaCategory" prefWidth="75.0" text="Category" />
                        <TableColumn fx:id="colMediaCost" prefWidth="75.0" text="Cost" />
                    </columns>
                    <columnResizePolicy>
                        <columnResizePolicy constant="CONSTRAINED_RESIZE_POLICY" />
                    </columnResizePolicy>
                </TableView>
                <ButtonBar prefHeight="40.0" prefWidth="200.0">
                    <buttons>
                        <Button fx:id="btnPlay" mnemonicParsing="false" onAction="#btnPlayPressed" text="Play" />
                        <Button fx:id="btnRemove" mnemonicParsing="false" onAction="#btnRemovePressed" text="Remove" />
                    </buttons>
                </ButtonBar>
            </children>
        </VBox>
    </center>
    <right>
        <VBox alignment="TOP_CENTER" prefHeight="200.0" spacing="10.0" BorderPane.alignment="CENTER">
            <padding>
                <insets top="50.0" />
            </padding>
            <children>
                <HBox alignment="CENTER" spacing="10.0">
                    <children>
                        <Label text="Total">
                            <font>
                                <font size="24.0" />
                            </font>
                        </Label>
                        <Label fx:id="labelTotalCost" text="0$" textFill="AQUA">
                            <font>
                                <font size="24.0" />
                            </font>
                        </Label>
                    </children>
                </HBox>
                <Button fx:id="btnPlaceOrder" mnemonicParsing="false" onAction="#btnPlaceOrderPressed" style="-fx-background-color: red;" text="Place Order" textFill="WHITE">
                    <font>
                        <font size="24.0" />
                    </font>
                </Button>
            </children>
        </VBox>
    </right>
</BorderPane>

```

Figure 16: Mã nguồn Cart.fxml

5.2. Giao diện Cart Screen

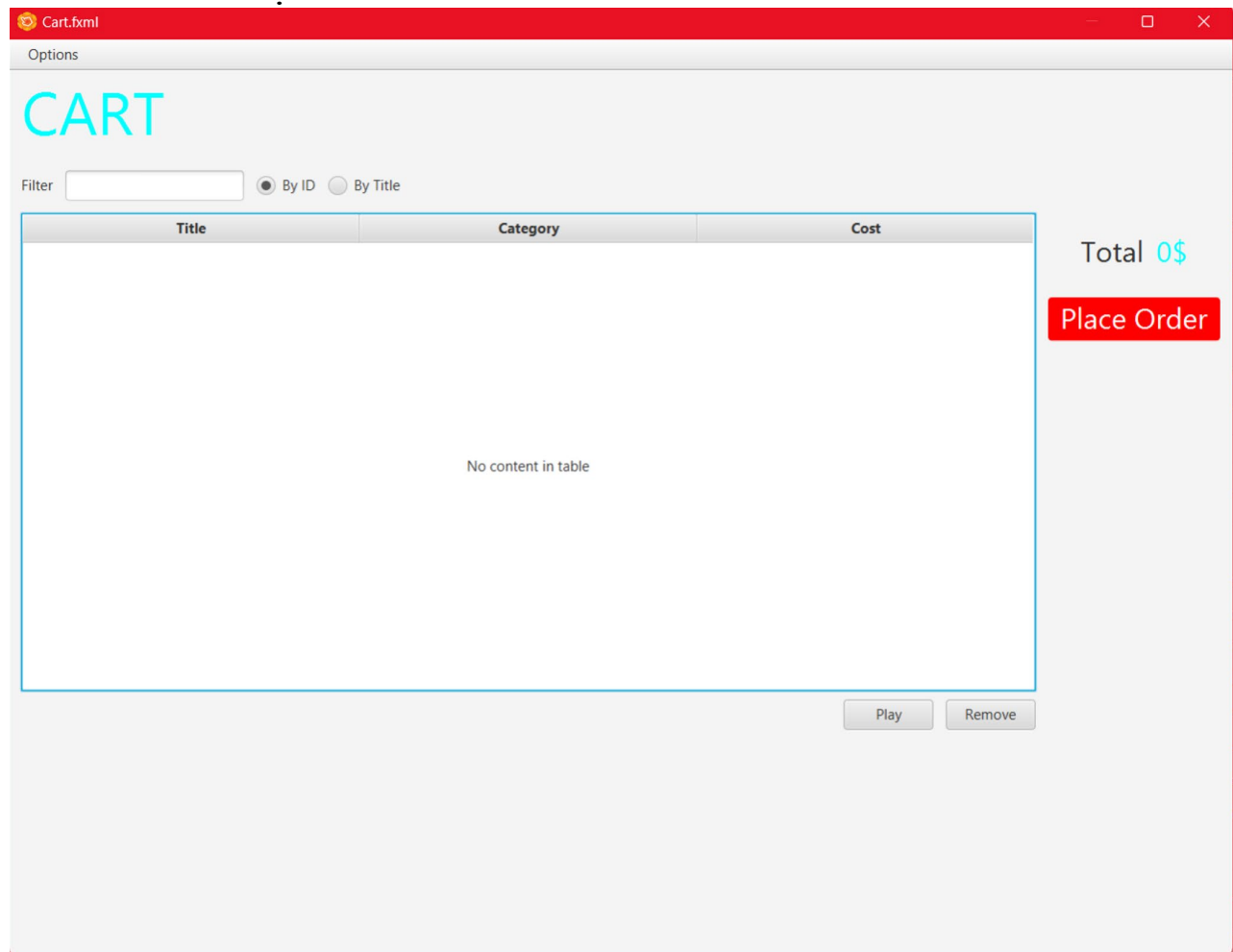


Figure 17: Giao diện Cart Screen

6. Integrating JavaFX into Swing application – The JFXPanel class

```

package hust.soict.dsai.aims.screen;

import java.io.IOException;

public class CartScreen extends JFrame {
    private Cart cart;
    private Store store;

    public CartScreen(Store store, Cart cart) {
        super();
        this.store = store;
        this.cart = cart;

        JFXPanel fxPanel = new JFXPanel();
        this.add(fxPanel);
        this.setTitle("Cart");
        this.setVisible(true);
        Platform.runLater(new Runnable() {
            @Override
            public void run() {
                try {
                    FXMLLoader loader = new FXMLLoader(getClass().getResource("cart.fxml"));
                    CartScreenController controller = new CartScreenController(store, cart);
                    loader.setController(controller);
                    Parent root = loader.load();
                    fxPanel.setScene(new Scene(root));
                } catch (IOException e) {
                    e.printStackTrace();
                }
            }
        });
    }
}

```

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

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

7.1. CartScreenController.java

```

package hust.soict.dsai.aims.screen;
import hust.soict.dsai.aims.cart.Cart;

public class CartScreenController {
    private Cart cart;
    private Store store;
    @FXML
    private Button btnPlay;
    @FXML
    private Button btnRemove;
    @FXML
    private TableView<Media> tblMedia;
    @FXML
    private TableColumn<Media, String> colMediaTitle;
    @FXML
    private TableColumn<Media, String> colMediaCategory;
    @FXML
    private TableColumn<Media, Float> colMediaCost;
    @FXML
    private Label labelTotalCost;

    public CartScreenController(Store store, Cart cart) {
        super();
        this.store = store;
        this.cart = cart;
    }

    @FXML
    private void initialize() {
        colMediaTitle.setCellValueFactory(new PropertyValueFactory<Media, String>("title"));
        colMediaCategory.setCellValueFactory(new PropertyValueFactory<Media, String>("category"));
        colMediaCost.setCellValueFactory(new PropertyValueFactory<Media, Float>("cost"));
        tblMedia.setItems(this.cart.getItemsOrdered());

        btnPlay.setVisible(false);
        btnRemove.setVisible(false);

        tblMedia.getSelectionModel().selectedItemProperty().addListener(
            new ChangeListener<Media>() {
                @Override
                public void changed(ObservableValue<? extends Media> observable, Media oldValue, Media newValue) {
                    if (newValue != null) {
                        updateButtonBar(newValue);
                    }
                }
            }
        );

        cart.getItemsOrdered().addListener((ListChangeListener<? super Media>) change -> updateTotalCost());
        updateTotalCost();
    }

    @FXML
    void btnRemovePressed(ActionEvent event) {
        Media media = tblMedia.getSelectionModel().getSelectedItem();
        cart.removeMedia(media);
    }

    @FXML
    void btnPlayPressed(ActionEvent event) {
        Media media = tblMedia.getSelectionModel().getSelectedItem();
        try {
            ((Playable) media).play();
        } catch (PlayerException e) {
            e.printStackTrace();
        }

        Alert alert = new Alert(Alert.AlertType.INFORMATION);
        alert.setTitle("Thử sản phẩm này (Play this media)");
        alert.setHeaderText(null);
        try {
            alert.setContentText(((Playable) media).play());
        } catch (PlayerException e) {
            e.printStackTrace();
        }
        alert.showAndWait();
    }

    @FXML
    void btnPlaceOrderPressed(ActionEvent event) {
        Alert alert = new Alert(Alert.AlertType.INFORMATION);
        alert.setTitle("Thông báo ()");
        alert.setHeaderText(null);
        alert.setContentText("Đơn hàng của bạn đã được tạo thành công.");
        alert.showAndWait();

        cart.getItemsOrdered().clear();
    }

    @FXML
    void menuItemViewStorePressed(ActionEvent event) {
        new StoreScreen(store, cart);
    }

    @FXML
    void btnAddBookPressed(ActionEvent event) {
        new AddBookToStoreScreen(store, cart);
    }

    @FXML
    void btnAddCDPressed(ActionEvent event) {
        new AddCDToStoreScreen(store, cart);
    }

    @FXML
    void btnAddDVDPressed(ActionEvent event) {
        new AddDVDToStoreScreen(store, cart);
    }

    void updateButtonBar(Media media) {
        btnRemove.setVisible(true);
        if (media instanceof Playable) {
            btnPlay.setVisible(true);
        } else {
            btnPlay.setVisible(false);
        }
    }

    void updateTotalCost() {
        double totalCost = cart.totalCost();
        labelTotalCost.setText(String.format("%.2f$", totalCost));
    }
}

```

Figure 19: Mã nguồn CartScreenController.java

7.2. Sửa lại Cart.java

```
package hust.soict.dsai.aims.cart;
import java.util.*;

public class Cart {
    public static final int MAX_NUMBERS_ORDERED = 20;
    private ObservableList<Media> itemsOrdered = FXCollections.observableArrayList();

    public ObservableList<Media> getItemsOrdered() {
        return itemsOrdered;
    }
}
```

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

7.3. Giao diện CartScreen

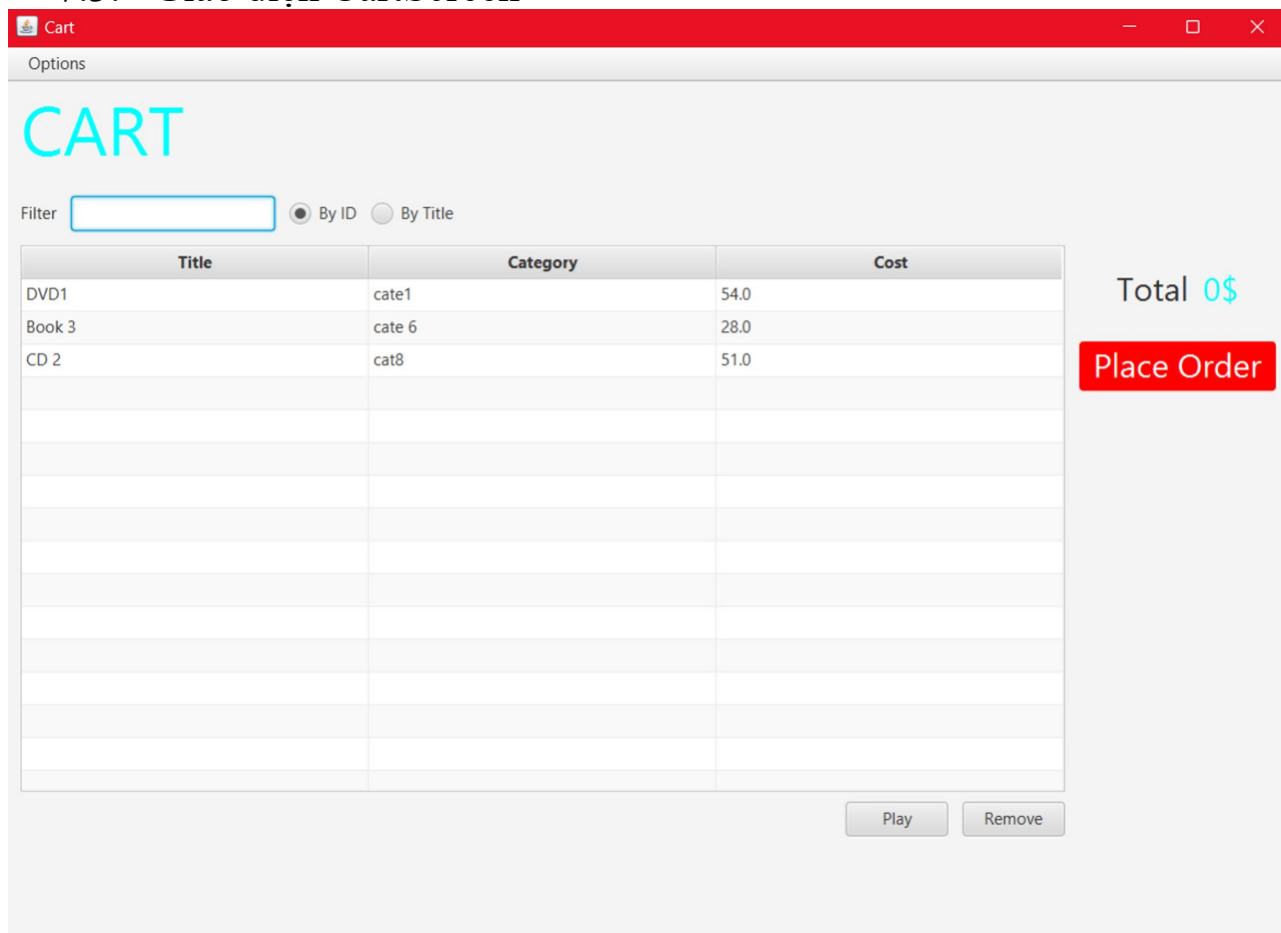


Figure 21: Giao diện Cart (hiển thị sản phẩm trong cart)

8. Updating buttons based on selected item in TableView – ChangeListener

8.1. Sửa lại CartScreenController.java

```

@FXML
private void initialize() {
    colMediaTitle.setCellValueFactory(new PropertyValueFactory<Media, String>("title"));
    colMediaCategory.setCellValueFactory(new PropertyValueFactory<Media, String>("category"));
    colMediaCost.setCellValueFactory(new PropertyValueFactory<Media, Float>("cost"));
    tblMedia.setItems(this.cart.getItemsOrdered());

    btnPlay.setVisible(false);
    btnRemove.setVisible(false);

    tblMedia.getSelectionModel().selectedItemProperty().addListener(
        new ChangeListener<Media>() {
            @Override
            public void changed(ObservableValue<? extends Media> observable, Media oldValue, Media newValue) {
                if (newValue != null) {
                    updateButtonBar(newValue);
                }
            }
        }
    );

    cart.getItemsOrdered().addListener((ListChangeListener<? super Media>) change -> updateTotalCost());
    updateTotalCost();
}

```

Figure 22: Chỉnh sửa CartScreenController.java

8.2. Giao diện CartScreen

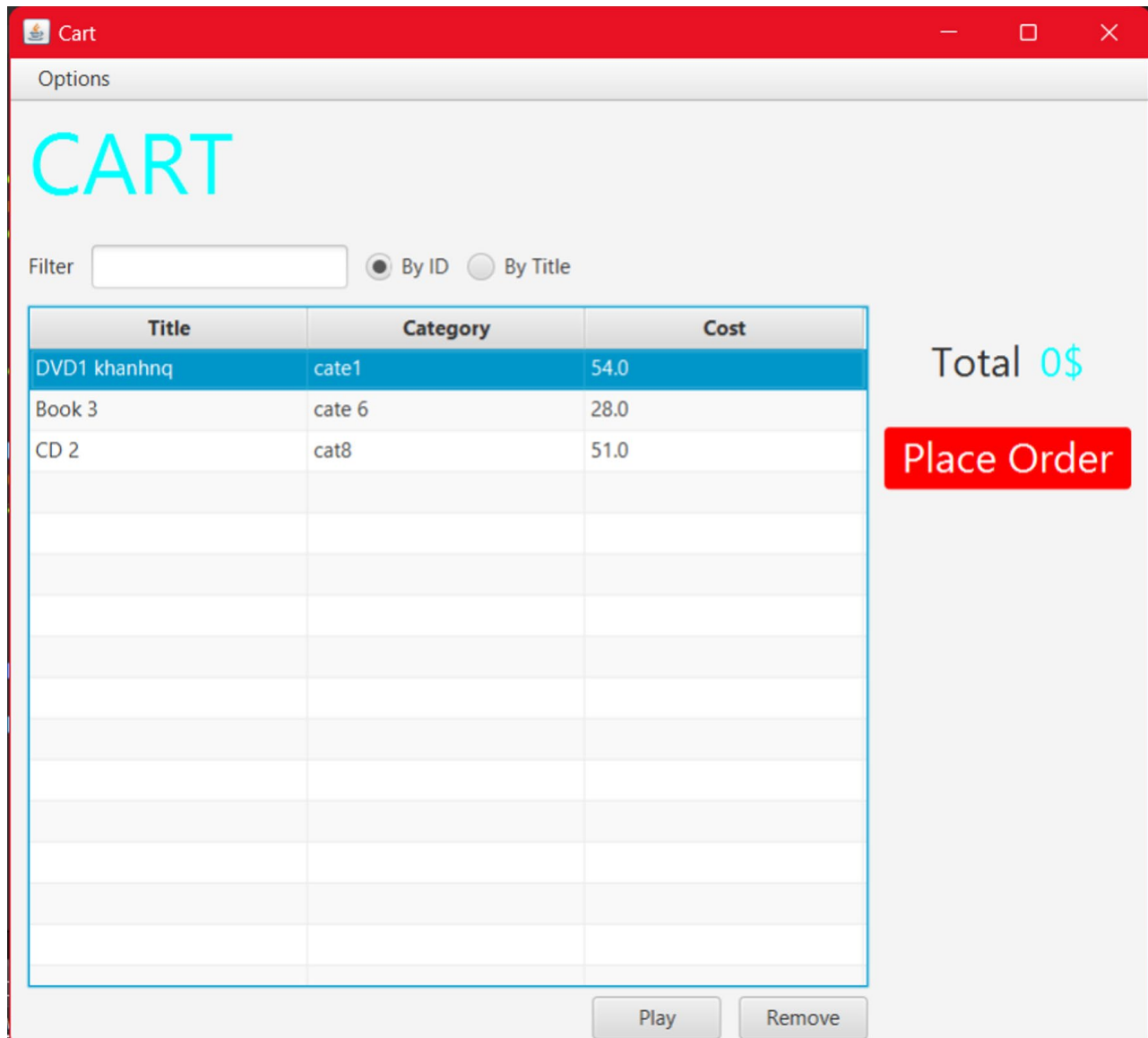


Figure 23: Giao diện Cart (cập nhật nút Play media)

9. Deleting a media

9.1. Thêm phương thức khi ấn nút Remove

```
<buttons>
  <Button fx:id="btnPlay" mnemonicParsing="false" onAction="#btnPlayPressed" text="Play" />
  <Button fx:id="btnRemove" mnemonicParsing="false" onAction="#btnRemovePressed" text="Remove" />
</buttons>
```

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

```
@FXML
void btnRemovePressed(ActionEvent event) {
    Media media = tblMedia.getSelectionModel().getSelectedItem();
    cart.removeMedia(media);
}
```

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

9.2. Giao diện

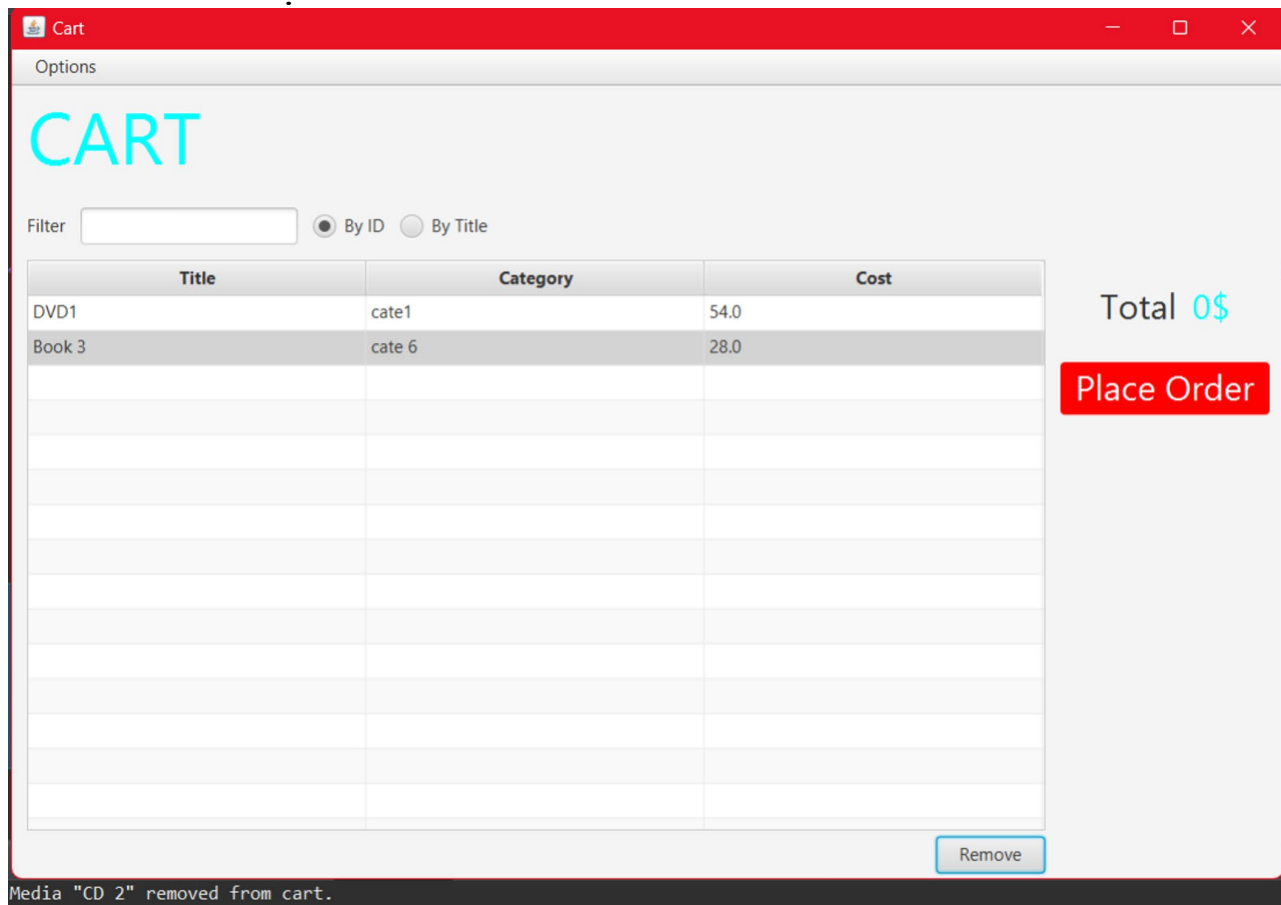


Figure 26: Giao diện Cart sau khi xóa CD2 khỏi cart

10. Filter items in cart – FilteredList (optional)

11. Complete the Aims GUI application

11.1. Cart Screen

11.1.1. “Place order” button

```
@FXML
void btnPlaceOrderPressed(ActionEvent event) {
    Alert alert = new Alert(Alert.AlertType.INFORMATION);
    alert.setTitle("Thông báo ()");
    alert.setHeaderText(null);
    alert.setContentText("Đơn hàng của bạn đã được tạo thành công.");
    alert.showAndWait();

    cart.getItemsOrdered().clear();
}
```

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



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

11.1.2. “Play” button

```
@FXML
void btnPlayPressed(ActionEvent event) {
    Media media = tblMedia.getSelectionModel().getSelectedItem();
    try {
        ((Playable) media).play();
    } catch (PlayerException e) {
        e.printStackTrace();
    }

    Alert alert = new Alert(Alert.AlertType.INFORMATION);
    alert.setTitle("Thử sản phẩm này (Play this media)");
    alert.setHeaderText(null);
    try {
        alert.setContentText(((Playable) media).play());
    } catch (PlayerException e) {
        e.printStackTrace();
    }
    alert.showAndWait();
}
```

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

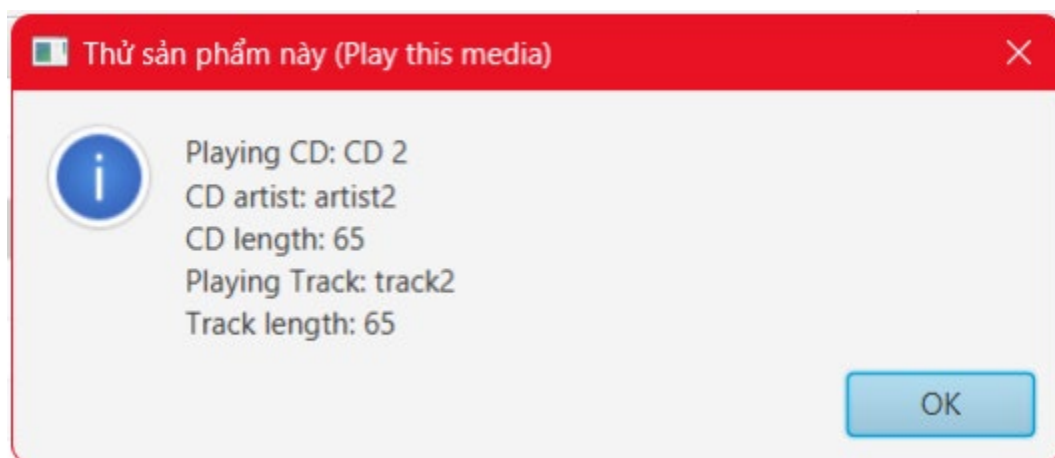


Figure 30: Hộp thoại Play media trong Cart

11.1.3. “Total cost” label

```
@FXML
private void initialize() {
    colMediaTitle.setCellValueFactory(new PropertyValueFactory<Media, String>("title"));
    colMediaCategory.setCellValueFactory(new PropertyValueFactory<Media, String>("category"));
    colMediaCost.setCellValueFactory(new PropertyValueFactory<Media, Float>("cost"));
    tblMedia.setItems(this.cart.getItemsOrdered());

    btnPlay.setVisible(false);
    btnRemove.setVisible(false);

    tblMedia.getSelectionModel().selectedItemProperty().addListener(
        new ChangeListener<Media>() {
            @Override
            public void changed(ObservableValue<? extends Media> observable, Media oldValue, Media newValue) {
                if (newValue != null) {
                    updateButtonBar(newValue);
                }
            }
        }
    );

    cart.getItemsOrdered().addListener((ListChangeListener<? super Media>) change -> updateTotalCost());
    updateTotalCost();
}

void updateTotalCost() {
    double totalCost = cart.totalCost();
    labelTotalCost.setText(String.format("%.2f$", totalCost));
}
```

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

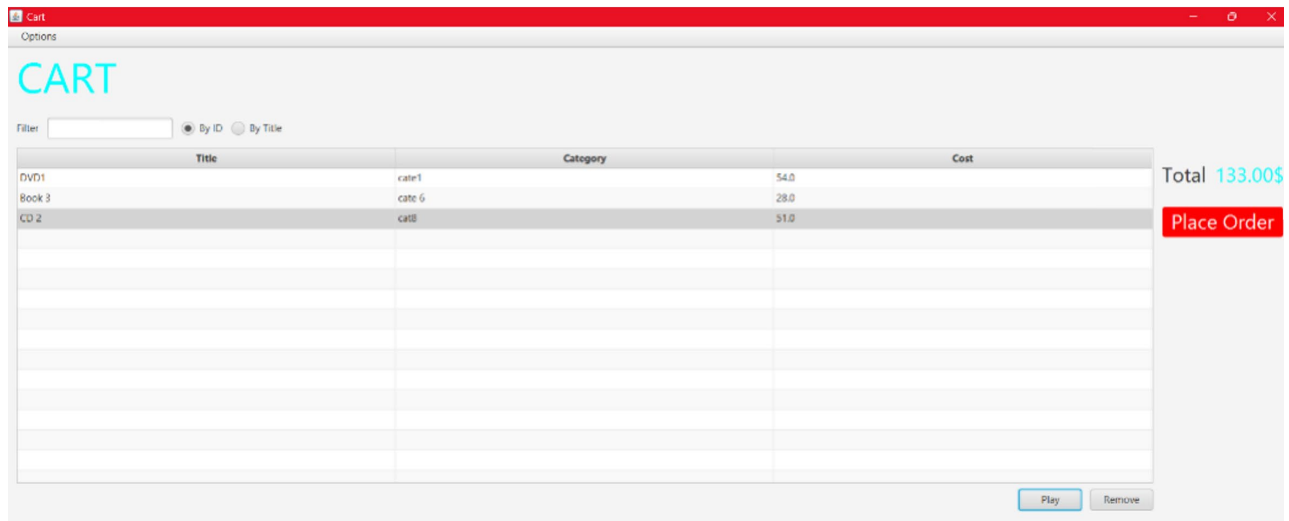


Figure 32: Giao diện Cart (hiển thị tổng giá tiền của đơn hàng)

11.1.4. Menu bar

```

@FXML
void menuItemViewStorePressed(ActionEvent event) {
    new StoreScreen(store, cart);
}

@FXML
void btnAddBookPressed(ActionEvent event) {
    new AddBookToStoreScreen(store, cart);
}

@FXML
void btnAddCDPressed(ActionEvent event) {
    new AddCDToStoreScreen(store, cart);
}

@FXML
void btnAddDVDPressed(ActionEvent event) {
    new AddDVDToStoreScreen(store, cart);
}

```

Figure 33: Mã nguồn MenuBar

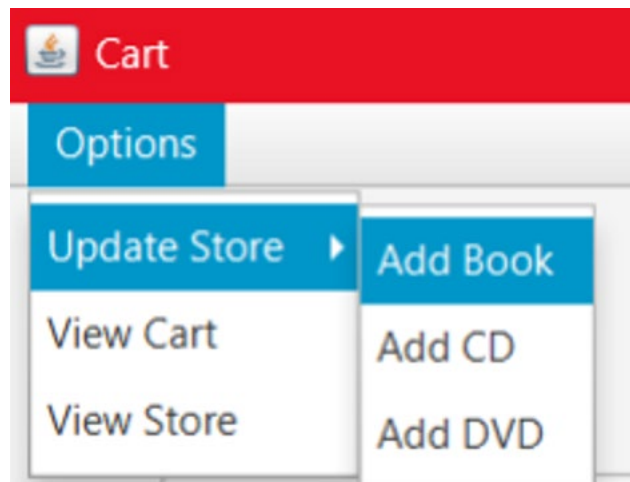


Figure 34: Giao diện MenuBar

11.2. Store Screen

```

addToCartButton.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        try {
            cart.addMedia(media);
        } catch (LimitExceededException e1) {
            e1.printStackTrace();
        }
        JOptionPane.showMessageDialog(null, media.getTitle() + " has been added to the cart.",
            "Cart", JOptionPane.INFORMATION_MESSAGE);
    }
});

```

Figure 35: Mã nguồn thêm sản phẩm vào cart



Figure 36: Thông báo sản phẩm được thêm vào cart

11.3. Update Store Screen

```

package hust.soict.dsai.aims.screen;
import java.awt.*;

public class AddItemToStoreScreen extends JFrame {
    protected Store store;
    protected Cart cart;

    JPanel createNorth() {
        JPanel north = new JPanel();
        north.setLayout(new BorderLayout(north, BorderLayout.Y_AXIS));
        north.add(createMenuBar());
        north.add(createHeader());
        return north;
    }

    JMenuBar createMenuBar() {
        JMenu smUpdateStore = new JMenu("Update store");
        JMenuItem smAddBook = new JMenuItem("Add Book");
        JMenuItem smAddCD = new JMenuItem("Add CD");
        JMenuItem smAddDVD = new JMenuItem("Add DVD");
        smUpdateStore.add(smAddBook);
        smUpdateStore.add(smAddCD);
        smUpdateStore.add(smAddDVD);

        JMenu menu = new JMenu("Options");
        menu.add(smUpdateStore);
        JMenuItem smViewStore = new JMenuItem("View store");
        menu.add(smViewStore);
        JMenuItem smViewCart = new JMenuItem("View cart");
        menu.add(smViewCart);

        smAddBook.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                new AddBookToStoreScreen(store, cart);
            }
        });

        smAddCD.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                new AddCDToStoreScreen(store, cart);
            }
        });

        smAddDVD.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                new AddDVDToStoreScreen(store, cart);
            }
        });

        smViewStore.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                new StoreScreen(store, cart);
            }
        });

        smViewCart.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                new CartScreen(store, cart);
            }
        });

        JMenuBar menuBar = new JMenuBar();
        menuBar.setLayout(new FlowLayout(FlowLayout.LEFT));
        menuBar.add(menu);

        return menuBar;
    }

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

        JLabel title = new JLabel("AIMS");
        title.setFont(new Font(title.getFont().getName(), Font.PLAIN, 50));
        title.setForeground(Color.MAGENTA);

        header.add(Box.createRigidArea(new Dimension(10, 10)));
        header.add(title);
        header.add(Box.createHorizontalGLue());
        header.add(Box.createRigidArea(new Dimension(10, 10)));

        return header;
    }

    JPanel createCenter(Cart cart) {
        JPanel center = new JPanel();
        return center;
    }

    public AddItemToStoreScreen(Store store, Cart cart) {
        this.store = store;
        this.cart = cart;
        Container cp = getContentPane();
        cp.setLayout(new BorderLayout());

        cp.add(createNorth(), BorderLayout.NORTH);
        cp.add(createCenter(cart), BorderLayout.CENTER);

        setVisible(true);
        setTitle("Add item to Store");
        setSize(500, 360);
    }
}

```

Figure 37: Mã nguồn lớp AddItemToStoreScreen

```

package hust.soict.dsai.aims.screen;
import java.awt.GridLayout;

public class AddBookToStoreScreen extends AddItemToStoreScreen {
    public AddBookToStoreScreen(Store store, Cart cart) {
        super(store, cart);
        setTitle("Add Book to Store");
    }

    JPanel createCenter(Cart cart) {
        JPanel center = new JPanel();
        center.setLayout(new GridLayout(7, 2, 5, 5));

        JLabel lblId = new JLabel("ID:");
        JTextField txtId = new JTextField();
        JLabel lblTitle = new JLabel("Title:");
        JTextField txtTitle = new JTextField();
        JLabel lblCategory = new JLabel("Category:");
        JTextField txtCategory = new JTextField();
        JLabel lblCost = new JLabel("Cost:");
        JTextField txtCost = new JTextField();
        JLabel lblAuthors = new JLabel("Author(s) (each author separated by a comma ','):");
        JTextField txtAuthors = new JTextField();

        center.add(lblId);
        center.add(txtId);
        center.add(lblTitle);
        center.add(txtTitle);
        center.add(lblCategory);
        center.add(txtCategory);
        center.add(lblCost);
        center.add(txtCost);
        center.add(lblAuthors);
        center.add(txtAuthors);

        JButton btnAdd = new JButton("Add Book");
        center.add(new JPanel());
        center.add(btnAdd);

        btnAdd.addActionListener(e -> {
            int id;
            String title = txtTitle.getText();
            String category = txtCategory.getText();
            float cost;
            String authors = txtAuthors.getText();

            id = Integer.parseInt(txtId.getText());
            cost = Float.parseFloat(txtCost.getText());

            ArrayList<String> authorList = new ArrayList<>();
            for (String author : authors.split(",")) {
                authorList.add(author.trim());
            }

            store.addMedia(new Book(id, title, category, cost, authorList));
            JOptionPane.showMessageDialog(this, "Book added successfully!", "Success", JOptionPane.INFORMATION_MESSAGE);

            txtId.setText("");
            txtTitle.setText("");
            txtCategory.setText("");
            txtCost.setText("");
            txtAuthors.setText("");
        });

        return center;
    }
}

```

Figure 38: Mã nguồn lớp AddBookToStoreScreen

```

package hust.soict.dsai.aims.screen;
import java.awt.GridLayout;

public class AddDVDToStoreScreen extends AddItemToStoreScreen {
    public AddDVDToStoreScreen(Store store, Cart cart) {
        super(store, cart);
        setTitle("Add DVD to Store");
    }

    JPanel createCenter(Cart cart) {
        JPanel center = new JPanel();
        center.setLayout(new GridLayout(7, 2, 5, 5));

        JLabel lblId = new JLabel("ID:");
        JTextField txtId = new JTextField();
        JLabel lblTitle = new JLabel("Title:");
        JTextField txtTitle = new JTextField();
        JLabel lblCategory = new JLabel("Category:");
        JTextField txtCategory = new JTextField();
        JLabel lblCost = new JLabel("Cost:");
        JTextField txtCost = new JTextField();
        JLabel lblLength = new JLabel("Length:");
        JTextField txtLength = new JTextField();
        JLabel lblDirector = new JLabel("Director:");
        JTextField txtDirector = new JTextField();

        center.add(lblId);
        center.add(txtId);
        center.add(lblTitle);
        center.add(txtTitle);
        center.add(lblCategory);
        center.add(txtCategory);
        center.add(lblCost);
        center.add(txtCost);
        center.add(lblLength);
        center.add(txtLength);
        center.add(lblDirector);
        center.add(txtDirector);

        JButton btnAdd = new JButton("Add DVD");
        center.add(new JPanel());
        center.add(btnAdd);

        btnAdd.addActionListener(e -> {
            int id;
            String title = txtTitle.getText();
            String category = txtCategory.getText();
            float cost;
            int length;
            String director = txtDirector.getText();

            id = Integer.parseInt(txtId.getText());
            cost = Float.parseFloat(txtCost.getText());
            length = Integer.parseInt(txtLength.getText());

            store.addMedia(new DigitalVideoDisc(id, title, category, cost, length, director));
            JOptionPane.showMessageDialog(this, "DVD added successfully!", "Success", JOptionPane.INFORMATION_MESSAGE);

            txtId.setText("");
            txtTitle.setText("");
            txtCategory.setText("");
            txtCost.setText("");
            txtLength.setText("");
            txtDirector.setText("");
        });

        return center;
    }
}

```

Figure 39: Mã nguồn lớp AddDVDToStoreScreen

```

package hust.soict.dsai.aims.screen;
import java.awt.GridLayout;

public class AddCDToStoreScreen extends AddItemToStoreScreen {
    public AddCDToStoreScreen(Store store, Cart cart) {
        super(store, cart);
        setTitle("Add CD to Store");
    }

    JPanel createCenter(Cart cart) {
        JPanel center = new JPanel();
        center.setLayout(new GridLayout(8, 2, 5, 5));

        JLabel lblId = new JLabel("ID:");
        JTextField txtId = new JTextField();
        JLabel lblTitle = new JLabel("Title:");
        JTextField txtTitle = new JTextField();
        JLabel lblCategory = new JLabel("Category:");
        JTextField txtCategory = new JTextField();
        JLabel lblCost = new JLabel("Cost:");
        JTextField txtCost = new JTextField();
        JLabel lblArtist = new JLabel("Artist:");
        JTextField txtArtist = new JTextField();
        JLabel lblDirector = new JLabel("Director:");
        JTextField txtDirector = new JTextField();
        JLabel lblTracks = new JLabel("Tracks (each track separated by ';' | title and length of each track separated by '/')");
        JTextField txtTracks = new JTextField();

        center.add(lblId);
        center.add(txtId);
        center.add(lblTitle);
        center.add(txtTitle);
        center.add(lblCategory);
        center.add(txtCategory);
        center.add(lblCost);
        center.add(txtCost);
        center.add(lblArtist);
        center.add(txtArtist);
        center.add(lblDirector);
        center.add(txtDirector);
        center.add(lblTracks);
        center.add(txtTracks);

        JButton btnAdd = new JButton("Add CD");
        center.add(new JPanel());
        center.add(btnAdd);

        btnAdd.addActionListener(e -> {
            int id;
            String title = txtTitle.getText();
            String category = txtCategory.getText();
            String artist = txtArtist.getText();
            float cost;
            String director = txtDirector.getText();
            String tracks = txtTracks.getText();

            id = Integer.parseInt(txtId.getText());
            cost = Float.parseFloat(txtCost.getText());

            ArrayList<Track> trackList = new ArrayList<Track>();
            for (String track : tracks.split(";")) {
                String trackTitle = track.split("/")[0].trim();
                int trackLength = Integer.parseInt(track.split("/")[1].trim());
                Track newTrack = new Track(trackTitle, trackLength);
                trackList.add(newTrack);
            }

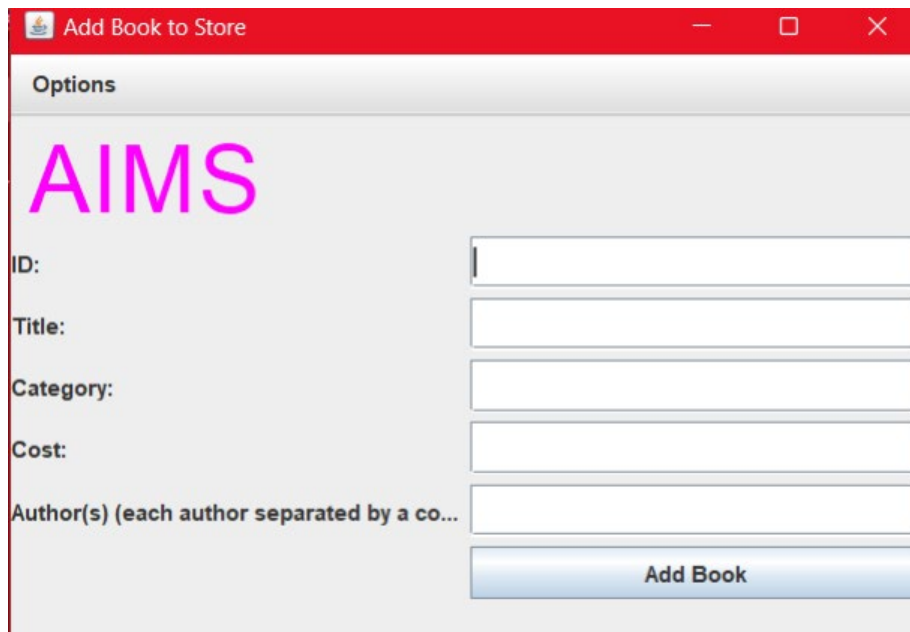
            store.addMedia(new CompactDisc(id, title, category, cost, artist, trackList, director));
            JOptionPane.showMessageDialog(this, "CD added successfully!", "Success", JOptionPane.INFORMATION_MESSAGE);

            txtId.setText("");
            txtTitle.setText("");
            txtCategory.setText("");
            txtCost.setText("");
            txtArtist.setText("");
            txtDirector.setText("");
            txtTracks.setText("");
        });

        return center;
    }
}

```

Figure 40: Mã nguồn lớp AddCDToStoreScreen



Options

AIMS

ID:

Title:

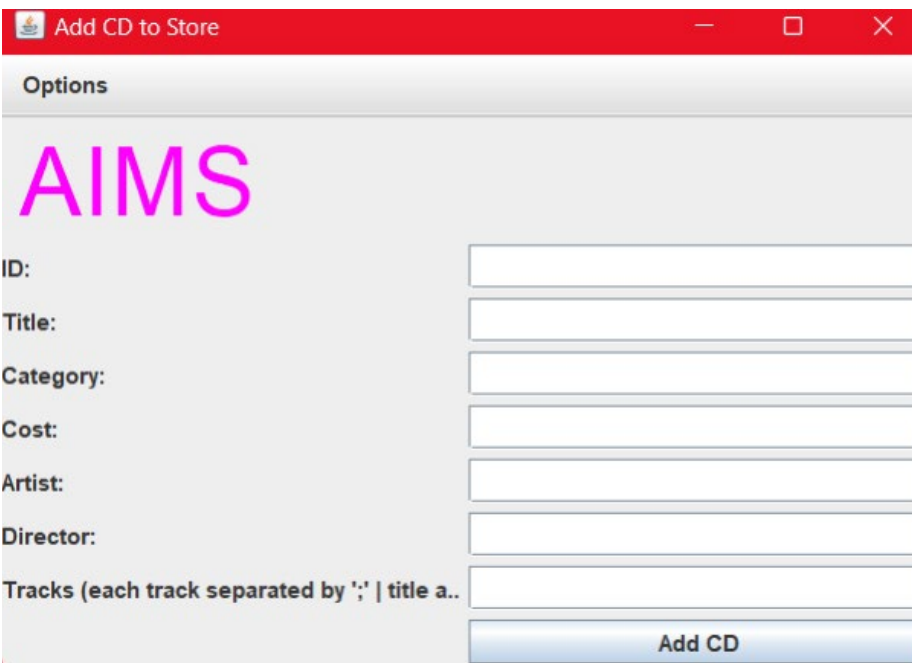
Category:

Cost:

Author(s) (each author separated by a co...

Add Book

Figure 41: Giao diện AddBookToStore



Options

AIMS

ID:

Title:

Category:

Cost:

Artist:

Director:

Tracks (each track separated by \';' | title a...

Add CD

Figure 42: Giao diện AddCDToStore

Figure 43: Giao diện AddDVDToStore

12. Check all the previous source codes to catch/handle/delegate runtime exceptions

```
public void addMedia(Media media) throws LimitExceededException {
    if (itemsOrdered.size() >= MAX_NUMBERS_ORDERED) {
        throw new LimitExceededException("ERROR: Number of medias has reached its limit.");
    } else {
        boolean isIdentical = false;
        for (Media media : itemsOrdered) {
            if (media.equals(media)) {
                isIdentical = true;
                break;
            }
        }

        if (isIdentical) {
            System.out.println("Media \"" + media.getTitle() + "\" is already added to cart.");
            return;
        }

        itemsOrdered.add(media);
        System.out.println("Media \"" + media.getTitle() + "\" added to cart.");
    }
}
```

Figure 44: Mã nguồn ngoại lệ vượt quá số lượng sản phẩm tối đa trong cart

```
public void addMedia(Media media) throws IllegalArgumentException {
    if (media.getCost() < 0) {
        throw new IllegalArgumentException("ERROR: Cost cannot be negative.");
    } else {
        itemsInStore.add(media);
        System.out.println("Media \"" + media.getTitle() + "\" added to store.");
    }
}
```

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

13. Create a class which inherits from Exception

```
package hust.soict.dsai.aims.exception;

public class PlayerException extends MediaException {
    public PlayerException() {
        // TODO Auto-generated constructor stub
    }

    public PlayerException(String message) {
        super(message);
        // TODO Auto-generated constructor stub
    }

    public PlayerException(Throwable cause) {
        super(cause);
        // TODO Auto-generated constructor stub
    }

    public PlayerException(String message, Throwable cause) {
        super(message, cause);
        // TODO Auto-generated constructor stub
    }

    public PlayerException(String message, Throwable cause, boolean enableSuppression, boolean writableStackTrace) {
        super(message, cause, enableSuppression, writableStackTrace);
        // TODO Auto-generated constructor stub
    }
}
```

Figure 46: Chỉnh sửa interface Playable

```
package hust.soict.dsai.aims.media;

import hust.soict.dsai.aims.exception.PlayerException;

public interface Playable {
    public String play() throws PlayerException;
}
```

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

```
public String play() throws PlayerException {
    if (this.getLength() > 0) {
        System.out.println("Playing DVD: " + this.getTitle());
        System.out.println("DVD length: " + this.getLength());

        return "Playing DVD: " + this.getTitle() + "\n" +
            "DVD length: " + this.getLength() + "\n";
    } else {
        throw new PlayerException("ERROR: DVD length is non-positive.");
    }
}
```

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

```

public String play() throws PlayerException {
    if (this.getLength() > 0) {
        System.out.println("Playing CD: " + this.getTitle()); // thông tin về CD
        System.out.println("CD artist: " + this.getArtist());
        System.out.println("CD length: " + this.getLength());
        for (Track track : tracks) {
            try {
                track.play(); // play từng track trong CD
            } catch (PlayerException e) {
                throw e;
            }
        }

        StringBuilder output = new StringBuilder();
        output.append("Playing CD: ").append(this.getTitle()).append("\n");
        output.append("CD artist: ").append(this.getArtist()).append("\n");
        output.append("CD length: ").append(this.getLength()).append("\n");
        for (Track track : tracks) {
            output.append(track.play());
        }
        return output.toString();
    } else {
        throw new PlayerException("ERROR: CD length is non-positive.");
    }
}

```

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

14. Update the Aims class

14.1. Answer the question

Question: What happens if the Aims class is not updated to handle exceptions when the play() method is called?

Answer: The exception will propagate through the call stack until it encounters a try-catch block or reaches the main() method. If the main() method does not catch the exception, the program will terminate immediately, and the console will display the stack trace. This makes the program less robust and harder to use.

14.2. Update the Aims.java code

```

try {
    cart.addMedia(dvd1);
} catch (LimitExceededException e) {
    e.printStackTrace();
}
try {
    cart.addMedia(book3);
} catch (LimitExceededException e) {
    e.printStackTrace();
}
try {
    cart.addMedia(cd2);
} catch (LimitExceededException e) {
    e.printStackTrace();
}

```

```

    if (media instanceof DigitalVideoDisc) {
        DigitalVideoDisc mediadvd = (DigitalVideoDisc) media;
        try {
            mediadvd.play();
        } catch (PlayerException e) {
            e.printStackTrace();
        }
    } else if (media instanceof CompactDisc) {
        CompactDisc mediacd = (CompactDisc) media;
        try {
            mediacd.play();
        } catch (PlayerException e) {
            e.printStackTrace();
        }
    } else {
        System.out.println("Media \"" + media.getTitle() + "\" is unplayable.");
    }

    found = true;
    break;
}

```

Figure 50: Chỉnh sửa lớp Aims

15. Modify the equals() method of Media class

```

@Override
public boolean equals(Object o) {
    try {
        if (this == o) return true;

        if (o == null || getClass() != o.getClass()) return false;

        Media media = (Media) o;
        return this.title != null && this.title.equals(media.title);

    } catch (NullPointerException e) {
        System.err.println("NullPointerException: Title của đối tượng hoặc đối tượng là null.");
        return false;
    } catch (ClassCastException e) {
        System.err.println("ClassCastException: Không thể ép kiểu đối tượng so sánh.");
        return false;
    }
}

```

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

16. Update Aims class diagram

The inheritance of exception classes is clearly demonstrated in the source code of the package `hust.soict.dsai.aims.exception`, as shown in the diagram below.

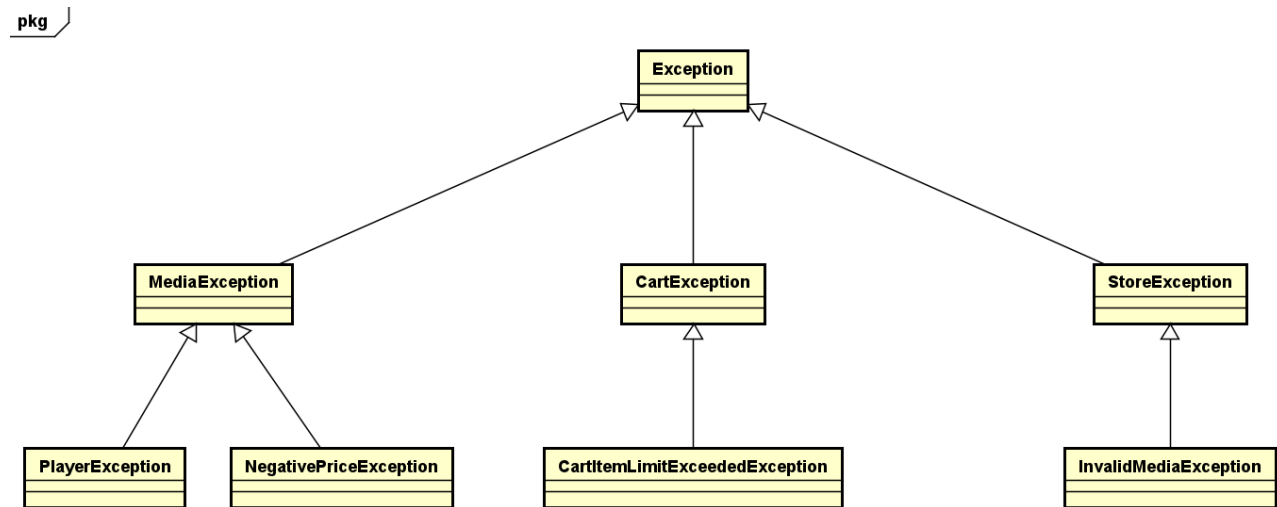


Figure 52: Cây phân cấp ngoại lệ