Báo Cáo Thực Hành

Lab05: GUI Programming

Họ và Tên: Nguyễn Văn Thái

MSSV: 20215135

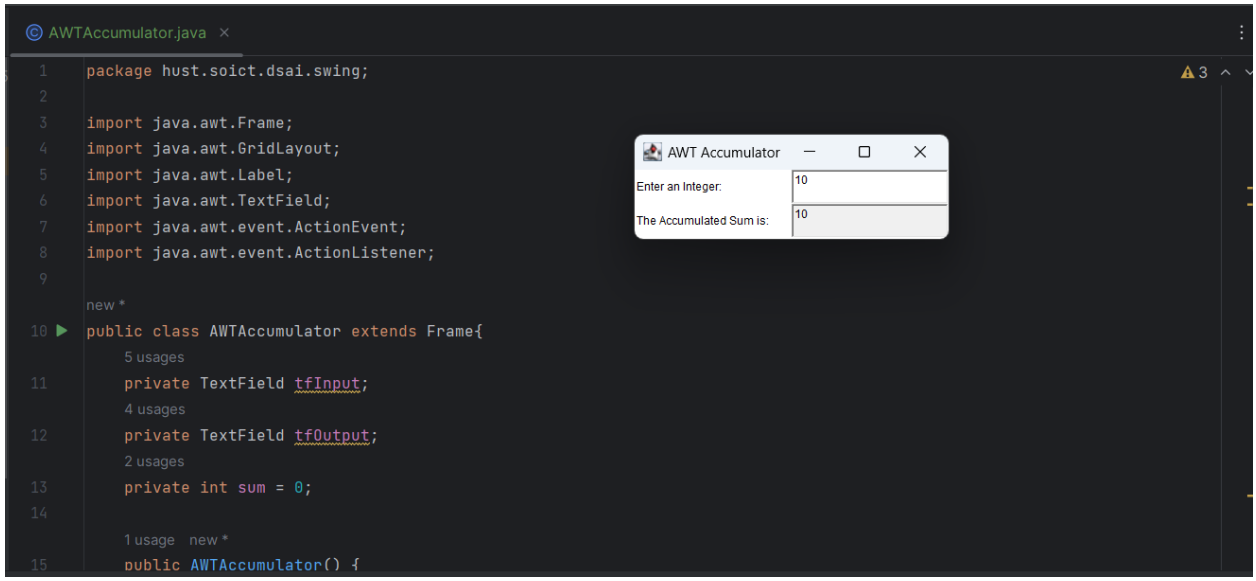
In this lab, you will practice with:

- Create simple GUI applications with Swing
- Create simple GUI applications with JavaFX
- Convert the Aims Project from the console/command-line (CLI) application to the GUI one
- Use Exception in your program

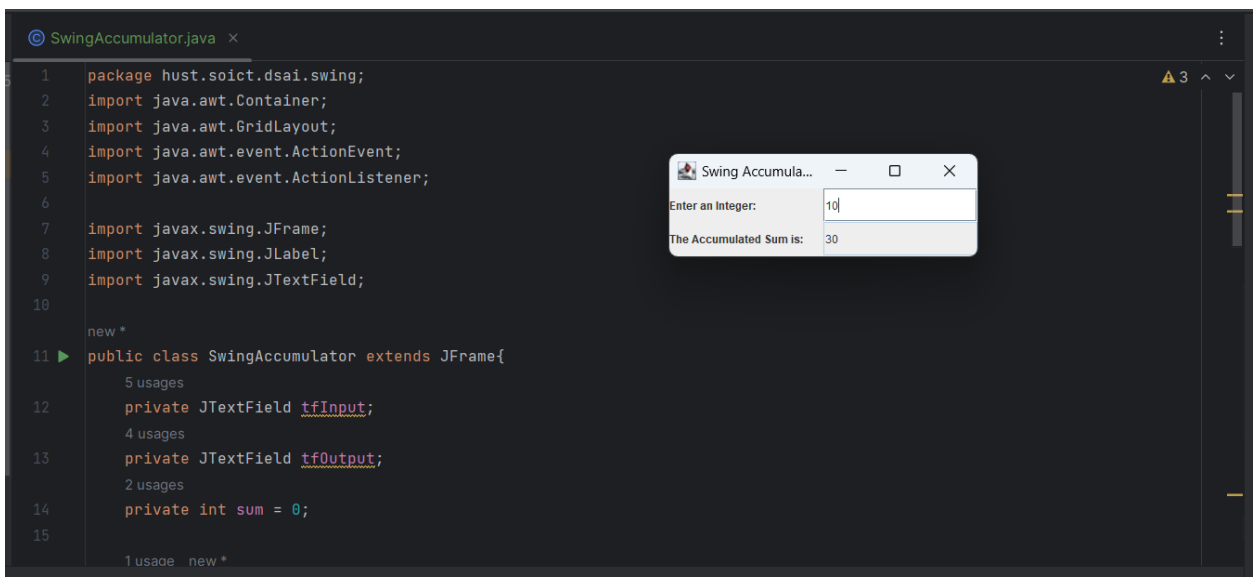
Contents

0. Assignment Submission
1. Swing components

1.1. AWTAccumulator



1.2. SwingAccumulator



1.3. Compare Swing and AWT elements

2. Organizing Swing components with Layout Managers

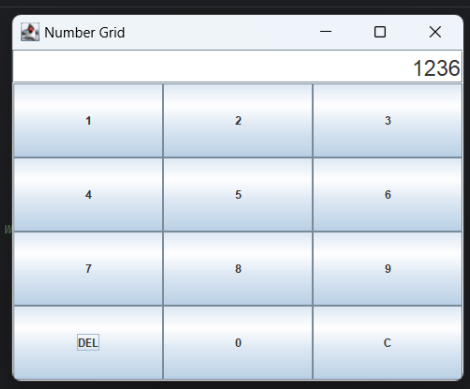
2.1. Swing top-level and secondary-level containers

2.2. Using JPanel as secondary-level container to organize components

```java
    public class ButtonListener implements ActionListener {
        new *
        public void actionPerformed(ActionEvent e) {
            String button = e.getActionCommand();
            if(button.charAt(0) >= '0' && button.charAt(0) <= '9') {
                // Append clicked number to the display field text
                tfDisplay.setText(tfDisplay.getText() + button);
            }
            else if (button.equals("DEL")) {
                // Remove the last character from the display field text
                String s = tfDisplay.getText();
                s = s.substring(0, (s.length()-1));
                tfDisplay.setText(s);
            }
            else {
                // Clear the display field text for the 'C' (reset) button
                tfDisplay.setText("");
            }
        }
    }
```

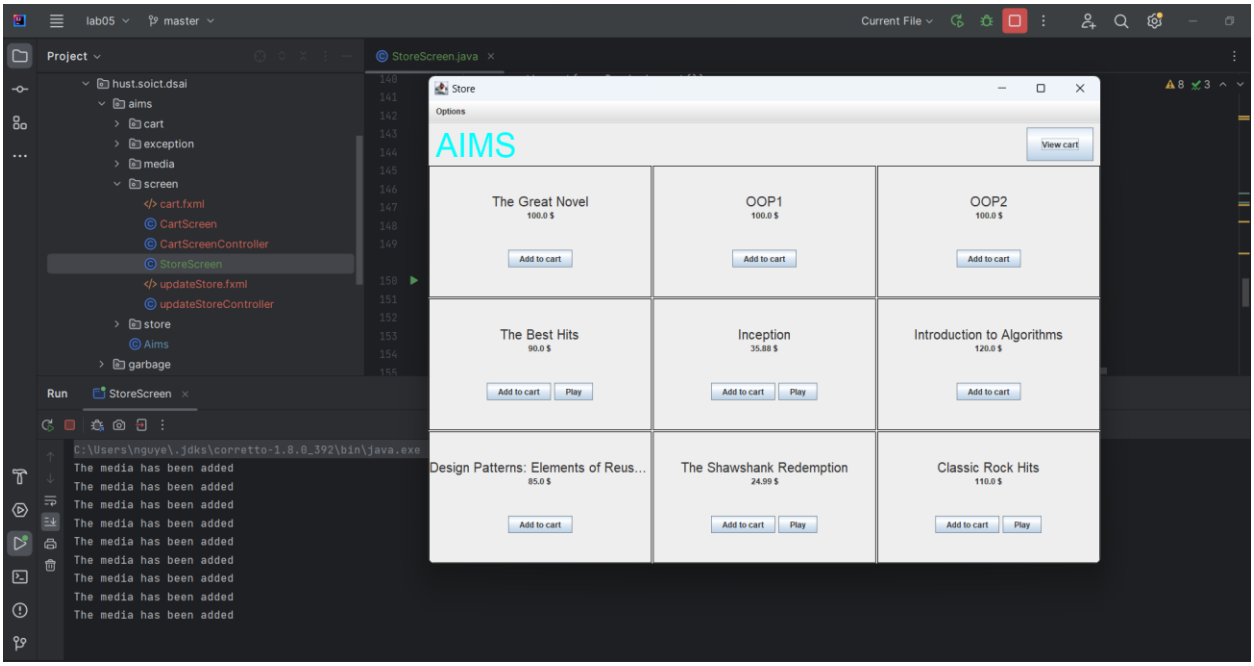
NumberGrid.java

```java
package hust.soict.dsai.swing;

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

/**
 * A simple Swing application that creates a number grid w
 */
new *
public class NumberGrid extends JFrame {
    6 usages
    private JButton[] btnNumbers = new JButton[10];
    3 usages
    private JButton btnDelete, btnReset;
    10 usages
    private JTextField tfDisplay;
```

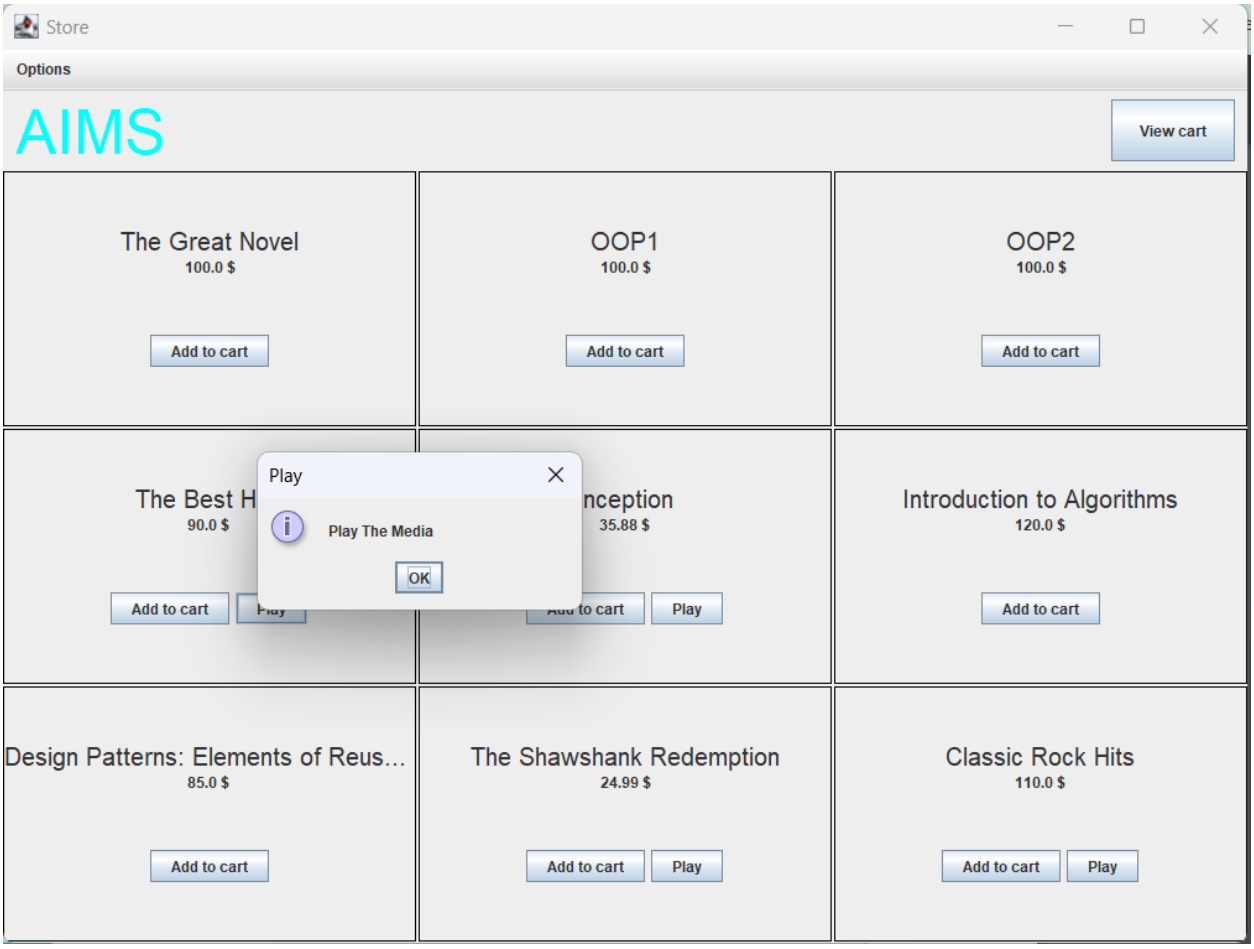


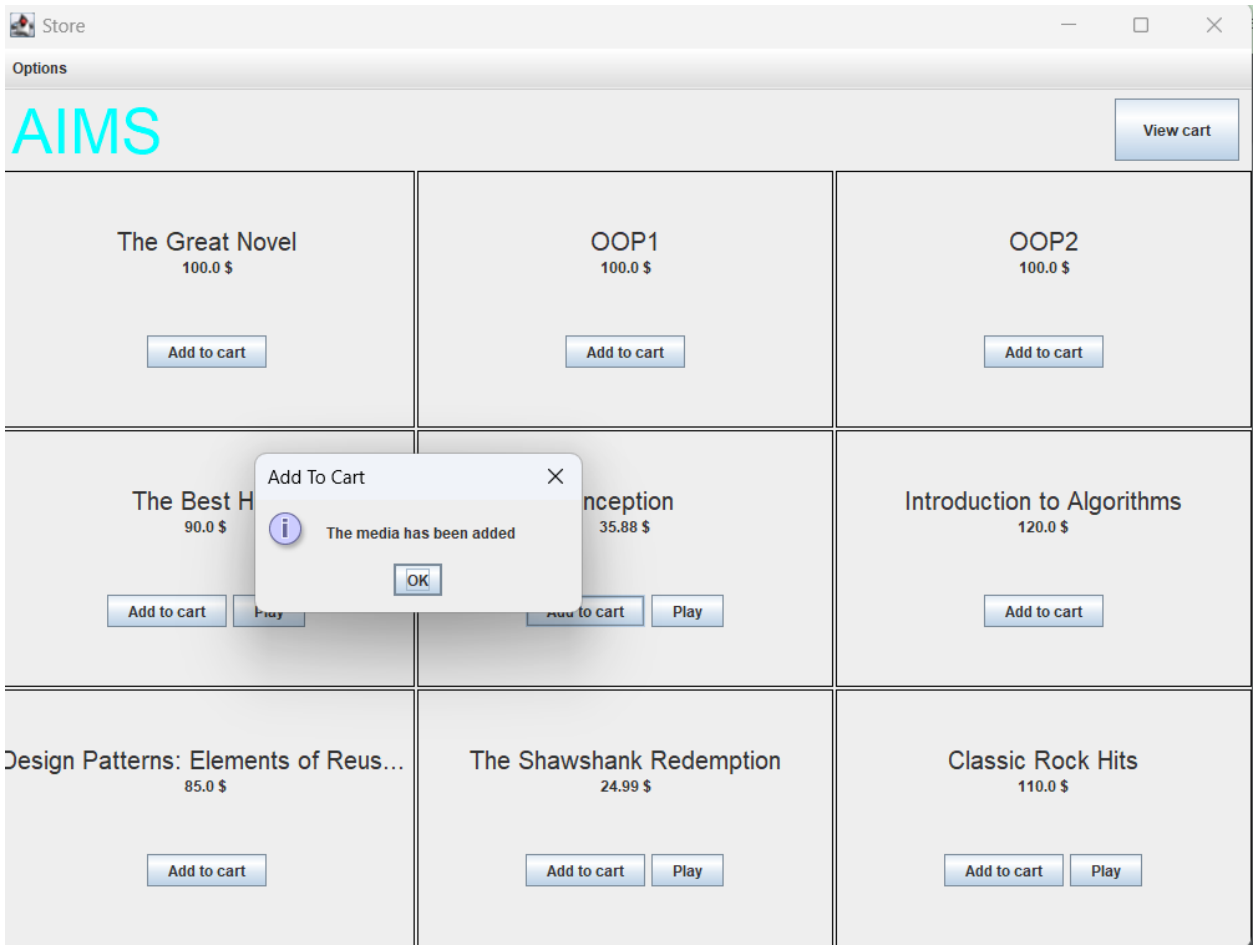
3. Create a graphical user interface for AIMS with Swing

3.1. View Store Screen

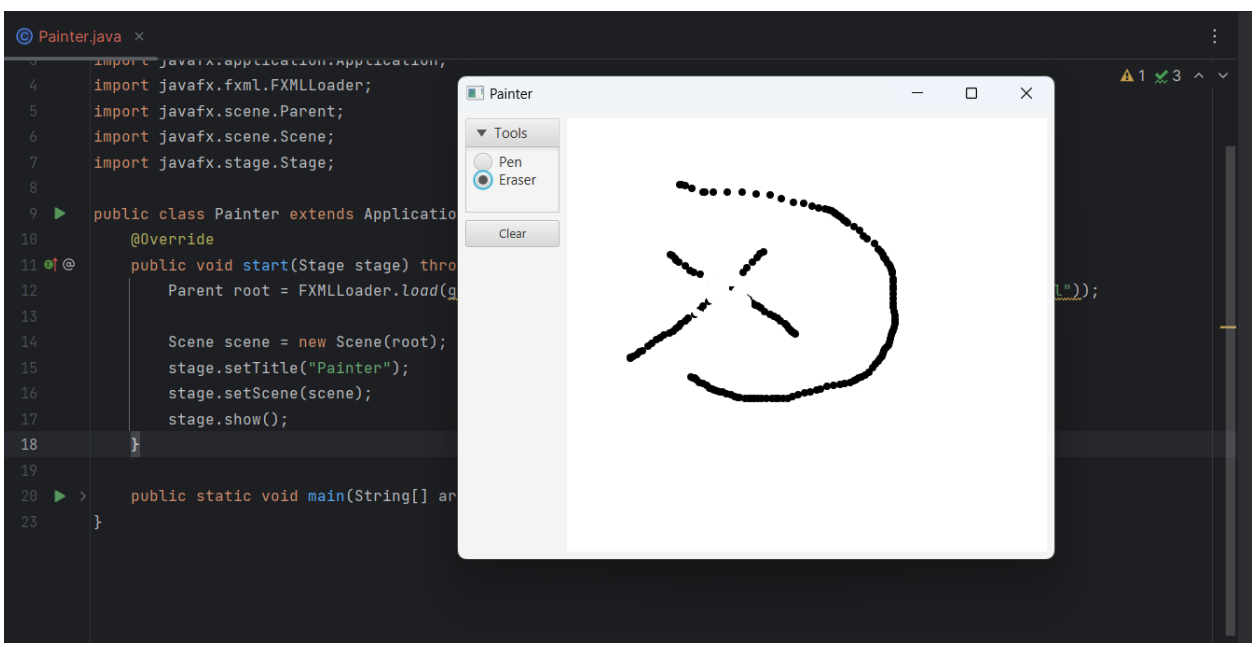


3.2. Adding more user interaction

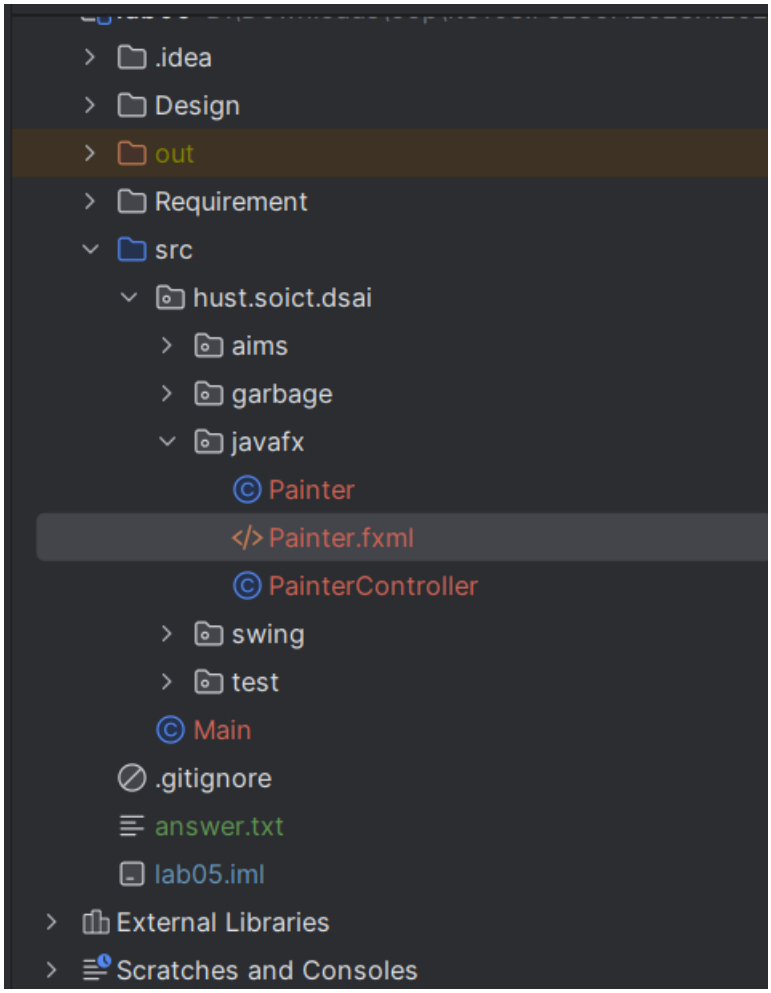



Store
Options
AIMS
View cart
The Great Novel
100.0 $
Add to cart
OOP1
100.0 $
Add to cart
OOP2
100.0 $
Add to cart
The Best H
90.0 $
Add To Cart
The media has been added
OK
nception
35.88 $
Play
Introduction to Algorithms
120.0 $
Add to cart
Design Patterns: Elements of Reus...
85.0 $
Add to cart
The Shawshank Redemption
24.99 $
Add to cart
Play
Classic Rock Hits
110.0 $
Add to cart
Play

4. JavaFX API


Painter.java
import javafx.fxml.FXMLLoader;
import javafx.scene.Parent;
import javafx.scene.Scene;
import javafx.stage.Stage;
public class Painter extends Applicatio
@Override
public void start(Stage stage) thro
Parent root = FXMLLoader.load(g
Scene scene = new Scene(root);
stage.setTitle("Painter");
stage.setScene(scene);
stage.show();
}
public static void main(String[] ar
}
Painter
Tools
Pen
Eraser
Clear

4.1. Create the FXML file



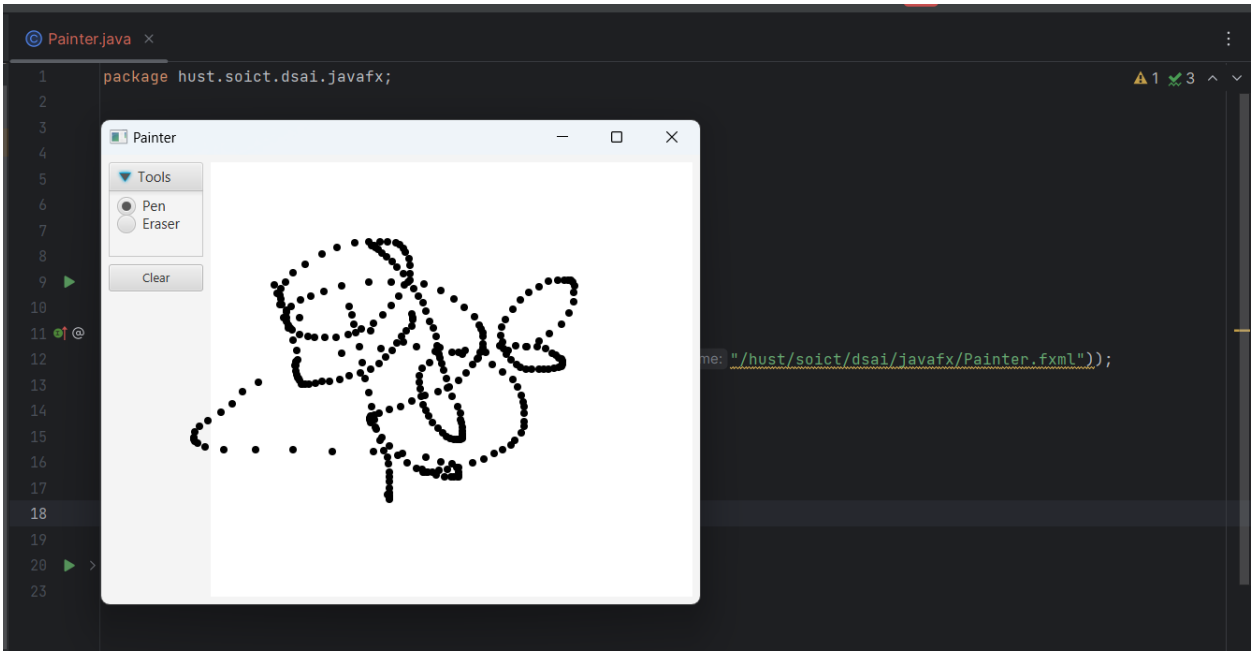
4.2. Create the controller class

```java
package hust.soict.dsai.javafx;

import ...

public class PainterController {
    @FXML
    private RadioButton ButtonPressed1;

    @FXML
    private RadioButton ButtonPressed2;

    @FXML
    private Pane drawingAreaPane;

    @FXML
    void Selected(ActionEvent event) {
        ToggleGroup question= new ToggleGroup();
        ButtonPressed1.setToggleGroup(question);
        ButtonPressed2.setToggleGroup(question);
        if(ButtonPressed1.isSelected()) {
            ButtonPressed2.setSelected(false);
        }
```

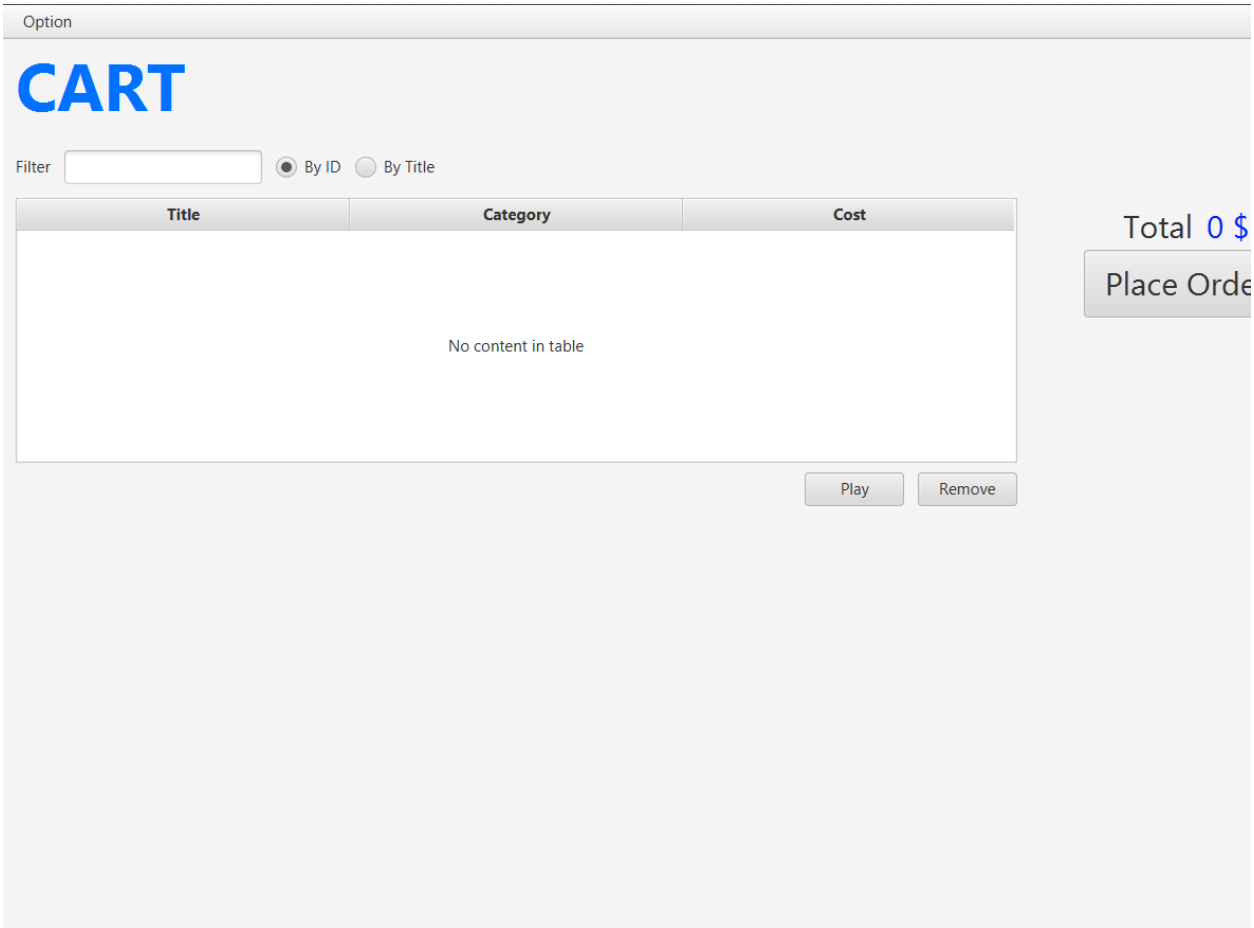
4.3. Create the application

```java
package hust.soict.dsai.javafx;

import javafx.application.Application;
import javafx.fxml.FXMLLoader;
import javafx.scene.Parent;
import javafx.scene.Scene;
import javafx.stage.Stage;

public class Painter extends Application {
    @Override
    public void start(Stage stage) throws Exception {
        Parent root = FXMLLoader.load(getClass().getResource( name: "/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); }
}
```

4.4. Practice exercise



5. Setting up the View Cart Screen with ScreenBuilde



5.1. Setting up the BorderPane

5.2. Setting up the TOP area

6. Integrating JavaFX into Swing application

```java
package hust.soict.dsai.aims.screen;

import ...

no usages
public class CartScreen extends JFrame {
    1 usage
    private Cart cart;

    no usages
    public CartScreen(Cart cart) {
        super();
        this.cart = cart;
        JFXPanel fxPanel = new JFXPanel();
        this.add(fxPanel);

        this.setTitle("Cart");
        this.setVisible(true);

        this.setSize(new Dimension( width: 1024,  height: 768));
        this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        Platform.runLater(new Runnable() {

            @Override
            public void run() {
                // TODO Auto-generated method stub
                try {
```

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

```java
package hust.soict.dsai.aims.screen;

import ...

3 usages
public class CartScreenController {
    9 usages
    private Cart cart;

    1 usage
    public CartScreenController(Cart cart) {
        // TODO Auto-generated constructor stub
        super();
        this.cart = cart;
    }

    @FXML
    private Button btnPlay;

    @FXML
    private Button btnRemove;

    @FXML
    private TableView<Media> tblMedia;

    @FXML
    private TableColumn<Media, String> colMediaTitle;
```

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

```java
    @FXML
    private void initialize() {
        // Show Cart in the Table
        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());
        totalCost.setText(Float.toString(cart.totalCost()) + "$");

        // Set Default the button Play and Remove
        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);
                }
                totalCost.setText(Float.toString(cart.totalCost()) + "$");
            }
        });
        // Filter
        tfFilter.textProperty().addListener(new ChangeListener<String>() {
            @Override
```

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

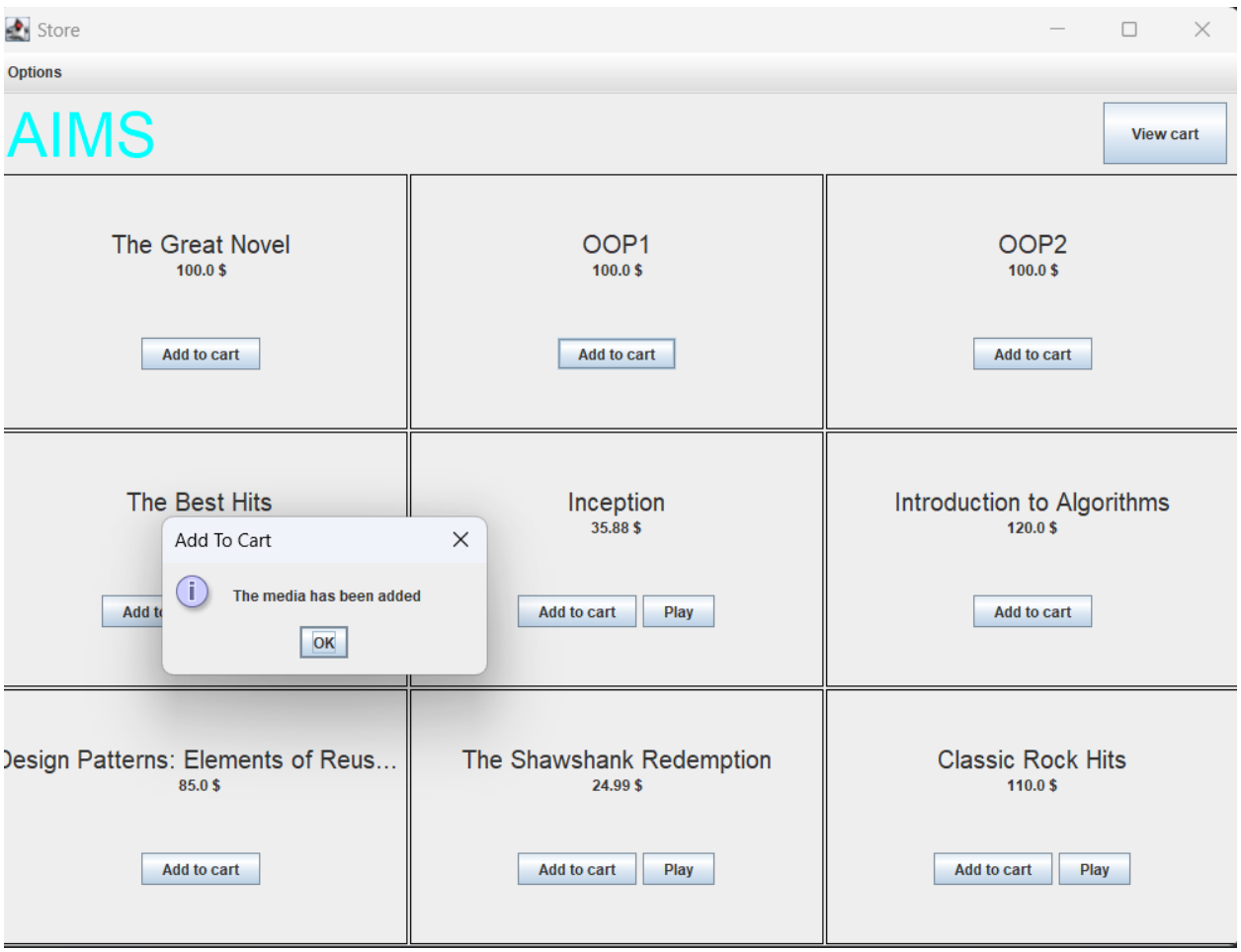
9. Deleting a media

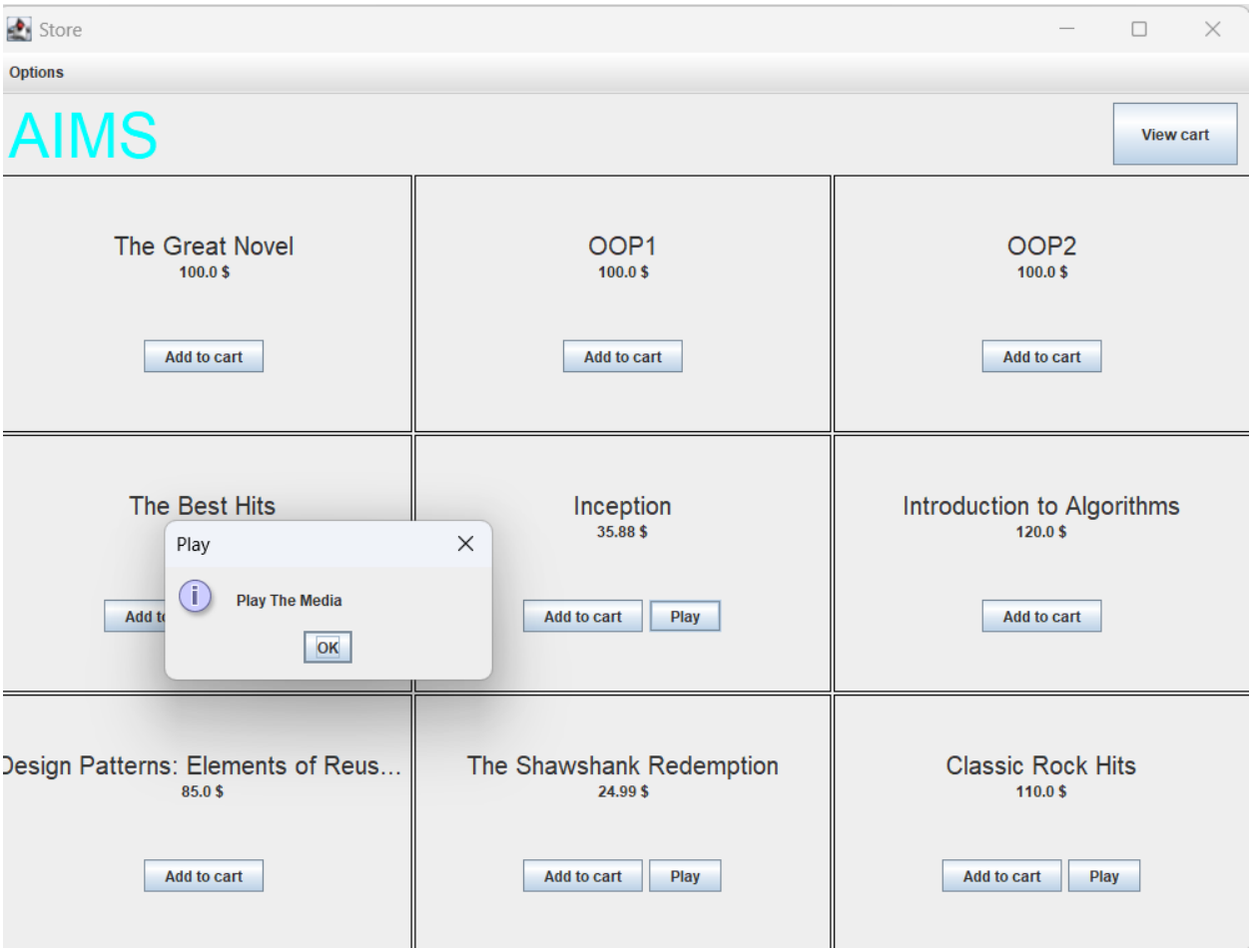
```java
    @FXML
    public void btnRemovePressed(ActionEvent actionEvent) {
        Media media = tblMedia.getSelectionModel().getSelectedItem();
        cart.removeMedia(media);
    }
```

10. Filter items in cart – FilteredList

```java
        // Filter
        tfFilter.textProperty().addListener(new ChangeListener<String>() {
            @Override
            public void changed(ObservableValue<? extends String> arg0, String oldValue, String newValue) {
                showFilteredMedia(newValue);
            }
```

11. Complete the Aims GUI application





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

13. Create a class which inherits from Exception

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

new *
public class PlayerException extends Exception {

    new *
    public static void main(String[] args) {
        // TODO Auto-generated method stub
    }

    no usages  new *
    public PlayerException() {
        super();
        // TODO Auto-generated constructor stub
    }

    no usages  new *
    public PlayerException(String message, Throwable cause, boolean enableSuppression, boolean writableStackTrace) {
        super(message, cause, enableSuppression, writableStackTrace);
        // TODO Auto-generated constructor stub
    }

    no usages  new *
```

```java
    no usages  new *
    public PlayerException(String message, Throwable cause) {
        super(message, cause);
        // TODO Auto-generated constructor stub
    }

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

    no usages  new *
    public PlayerException(Throwable cause) {
        super(cause);
        // TODO Auto-generated constructor stub
    }

}
```

13.1. Create new class named PlayerException

13.2. Raise the PlayerException

13.3. Update play() in the Playable interface

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

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

6 usages  3 implementations
public interface Playable {
    4 usages  3 implementations
    public void play() throws PlayerException;
}
```

13.4. Update play() in CompactDisc

```java
    public void play() throws PlayerException {
        // TODO Auto-generated method stub
        if (this.getLength() > 0) {
            JDialog dialog = new JDialog();
            dialog.setSize( width: 300,  height: 200);

            // create Label
            JLabel text = new JLabel( text: "DVD - Title : " + this.getTitle() + " Length : " + this.getLength());
            dialog.add(text);
            dialog.setTitle("Play DVD");
            dialog.setVisible(true);
        } else
            throw new PlayerException("ERROR : DVD length is non-positive");
    }
}
```

14. Update the Aims class

```java
    public static void handleException(Exception e) {
        String errorMessage = "An exception occurred: " + e.getMessage();
        System.out.println(errorMessage);
        SwingUtilities.invokeLater(() -> {
            JFrame frame = new JFrame( title: "Test Frame");
//            frame.setSize(300, 200);
//            frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
            frame.setVisible(true);
            JOptionPane.showMessageDialog(frame, errorMessage,  title: "Error", JOptionPane.ERROR_MESSAGE);
        });

    }
}
```

```java
    public static void playMedia(Media m) throws PlayerException {
        try {
        if (m instanceof DigitalVideoDisc) {
            System.out.println("\n==============================");
            ((DigitalVideoDisc) m).play();
            System.out.println("==============================\n");
        } else if (m instanceof CompactDisc) {
            System.out.println("\n==============================");
            ((CompactDisc) m).play();
            System.out.println("==============================\n");
        } else {
            System.out.println("=====================================================");
            System.out.println("===>>>This type of media does not have play mode<<<===");
            System.out.println("=====================================================");
        }

        } catch (PlayerException e) {
            handleException(e);
        }
    }
```

15. Modify the equals() method of Media class

```java
    new *
    @Override
    public boolean equals(Object obj) {
        // Check if the object is compared to itself
        if (this == obj) {
            return true;
        }

        // Check if the object is null
        if (obj == null) {
            return false;
        }

        // Check if the object is an instance of Media class
        if (!(obj instanceof Media)) {
            return false;
        }

        // Cast the object to Media type
        Media otherMedia = (Media) obj;

        // Check if the titles are equal
        if (this.getTitle() == null && otherMedia.getTitle() == null) {
            return true;
        } else if (this.getTitle() == null || otherMedia.getTitle() == null) {
            return false;
        } else {
```

16. Reading Document

17. Update Aims class diagram