**Báo cáo thực hành Lập trình hướng đối tượng Lab 5**

Họ và tên: Nguyễn Minh Khôi

MSSV: 20225642

Mục lục

[I. Java swing and java awt 3](#_Toc154861776)

[1. AWT Accumlator 3](#_Toc154861777)

[a. Code 3](#_Toc154861778)

[b. Result 4](#_Toc154861779)

[2. Swing Accumulator 4](#_Toc154861780)

[a. Code 4](#_Toc154861781)

[b. Result 5](#_Toc154861782)

[3. Number Grid 5](#_Toc154861783)

[a. Code 5](#_Toc154861784)

[b. Result 7](#_Toc154861785)

[II. Java FX 8](#_Toc154861786)

[1. Tạo lớp Painter 8](#_Toc154861787)

[2. Tạo lớp PainterController 8](#_Toc154861788)

[3. Tạo file Painter.fxml 9](#_Toc154861789)

[4. Kết quả 11](#_Toc154861790)

[III. Xử lý ngoại lệ 12](#_Toc154861791)

[1. Tạo lớp ngoại lệ PlayException trong package ../aims/exception 12](#_Toc154861792)

[2. Từ ngoại lệ đấy, sửa các phương thức có liên quan 12](#_Toc154861793)

[a. Trong interface Playable 12](#_Toc154861794)

[b. Trong class DigitalVideoDisc 12](#_Toc154861795)

[c. Trong class CompactDisc 12](#_Toc154861796)

[d. Trong class Track 13](#_Toc154861797)

[e. Trong class chính AIMS 13](#_Toc154861798)

[IV. Tạo tất cả giao diện trong project AIMS 15](#_Toc154861799)

[1. Tạo package screen trong project AIMS 15](#_Toc154861800)

[2. Tạo class MediaStore 15](#_Toc154861801)

[3. Tạo giao diện giỏ hàng 17](#_Toc154861802)

[a. Tạo class CartScreenController để điều khiển file fxml 17](#_Toc154861803)

[b. Tạo file Cart.fxml 21](#_Toc154861804)

[c. Tạo class CartScreen 24](#_Toc154861805)

[4. Tạo class StoreScreen, trong file StoreScreen.java thì sẽ tạo ra một vài chức năng như thêm xoá sửa tìm kiếm, hoàn thành các chức năng của ứng dụng AIMS trên GUI (Sau khi tạo xong các lớp như CartScreen, …) 25](#_Toc154861806)

[a. Source code 25](#_Toc154861807)

[b. Result 30](#_Toc154861808)

[V. Các biểu đồ 31](#_Toc154861809)

[1. Biểu đồ Use-case 31](#_Toc154861810)

[2. Biểu đồ Class 32](#_Toc154861811)

1. Java swing and java awt
2. AWT Accumlator
3. Kết quả

A screenshot of a computer

Description automatically generated A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

1. Swing Accumulator
2. Kết quả

A screenshot of a computer

Description automatically generated A screenshot of a computer

Description automatically generated

1. Number Grid
2. Result

A screenshot of a grid

Description automatically generated

Sau khi ấn DEL

A screenshot of a computer

Description automatically generated

Sau khi ấn C

A screenshot of a grid

Description automatically generated

1. Java FX

Tạo 1 ứng dụng vẽ tranh sử dụng JavaFX

Kết quả:

Ban đầu

A screenshot of a computer

Description automatically generated

Sau khi vẽ

A screenshot of a computer

Description automatically generated

Sau khi sử dụng Eraser

A screenshot of a computer

Description automatically generated

1. Xử lý ngoại lệ
2. Tạo lớp ngoại lệ PlayException trong  
   }

A screenshot of a computer program

Description automatically generated

1. Từ ngoại lệ đấy, sửa các phương thức có liên quan

* Trong class interface Playable
* Trong class DigitalVideoDisc
* Trong class CompactDisc
* Trong class Track
* Trong class AimsTest

1. Tạo tất cả giao diện trong project AIMS
2. Tạo package screen trong project AIMA screenshot of a computer

   Description automatically generated
3. Tạo class MediaStore
4. package AimsProject.src.hust.soict.hedspi.main.screen;  
     
   import AimsProject.src.hust.soict.hedspi.main.Cart.Cart;  
   import AimsProject.src.hust.soict.hedspi.main.aims.media.\*;  
   import AimsProject.src.hust.soict.hedspi.main.exception.PlayerException;  
     
   import javax.swing.\*;  
   import java.awt.\*;  
   import java.awt.event.ActionEvent;  
   import java.awt.event.ActionListener;  
     
   public class MediaStore extends JPanel {  
    private Media media;  
     
    public MediaStore(Media media, Cart myCart) {  
    this.media = media;  
    this.setLayout(new BoxLayout(this, BoxLayout.*Y\_AXIS*));  
    this.setBackground(new Color(255, 255, 255));  
    this.setBorder(BorderFactory.*createCompoundBorder*(  
    BorderFactory.*createLineBorder*(new Color(200, 200, 200), 1, true),  
    BorderFactory.*createEmptyBorder*(15, 15, 15, 15)  
    ));  
     
    JLabel title = new JLabel(media.getTitle());  
    title.setFont(new Font("Segoe UI", Font.*BOLD*, 20));  
    title.setForeground(new Color(44, 62, 80));  
    title.setAlignmentX(*CENTER\_ALIGNMENT*);  
     
    JLabel cost = new JLabel(String.*format*("%.2f $", media.getCost()));  
    cost.setFont(new Font("Segoe UI", Font.*PLAIN*, 16));  
    cost.setForeground(new Color(127, 140, 141));  
    cost.setAlignmentX(*CENTER\_ALIGNMENT*);  
     
    JPanel container = new JPanel();  
    container.setLayout(new FlowLayout(FlowLayout.*CENTER*));  
    container.setBackground(new Color(255, 255, 255));  
     
    JButton btnAdd = createStyledButton("Add to cart", new Color(52, 152, 219));  
    btnAdd.addActionListener(new ActionListener() {  
    @Override  
    public void actionPerformed(ActionEvent e) {  
    myCart.addMedia(media);  
    JOptionPane.*showMessageDialog*(null,  
    "Added " + media.getTitle() + " to cart!",  
    "Success",  
    JOptionPane.*INFORMATION\_MESSAGE*);  
    }  
    });  
    container.add(btnAdd);  
     
    if (media instanceof Playable) {  
    JButton playBtn = createStyledButton("Play", new Color(46, 204, 113));  
    playBtn.addActionListener(new ActionListener() {  
    @Override  
    public void actionPerformed(ActionEvent e) {  
    JDialog playDialog = *createPlayDialog*(media);  
    playDialog.setVisible(true);  
    playDialog.setSize(300, 200);  
    playDialog.pack();  
    }  
    });  
    container.add(playBtn);  
    }  
     
    this.add(Box.*createVerticalGlue*());  
    this.add(title);  
    this.add(cost);  
    this.add(Box.*createVerticalGlue*());  
    this.add(container);  
    }  
     
    private JButton createStyledButton(String text, Color backgroundColor) {  
    JButton button = new JButton(text);  
    button.setFont(new Font("Segoe UI", Font.*BOLD*, 12));  
    button.setForeground(Color.*WHITE*);  
    button.setBackground(backgroundColor);  
    button.setBorder(BorderFactory.*createEmptyBorder*(5, 10, 5, 10));  
    button.setFocusPainted(false);  
    button.setCursor(new Cursor(Cursor.*HAND\_CURSOR*));  
     
    button.addMouseListener(new java.awt.event.MouseAdapter() {  
    public void mouseEntered(java.awt.event.MouseEvent evt) {  
    button.setBackground(backgroundColor.darker());  
    }  
     
    public void mouseExited(java.awt.event.MouseEvent evt) {  
    button.setBackground(backgroundColor);  
    }  
    });  
     
    return button;  
    }  
     
    static JDialog createPlayDialog(Media media) {  
    JDialog playDialog = new JDialog();  
    Container container = playDialog.getContentPane();  
    playDialog.setDefaultCloseOperation(JDialog.*DISPOSE\_ON\_CLOSE*);  
    container.setLayout(new BoxLayout(container, BoxLayout.*Y\_AXIS*));  
    container.setBackground(new Color(255, 255, 255));  
    container.add(Box.*createRigidArea*(new Dimension(10, 10)));  
     
    if (media instanceof DigitalVideoDisc dvd) {  
    container.add(*createStyledLabel*("Playing DVD: " + dvd.getTitle(), new Font("Segoe UI", Font.*BOLD*, 14)));  
    container.add(*createStyledLabel*("DVD length: " + dvd.getLength() + " min", new Font("Segoe UI", Font.*PLAIN*, 12)));  
    } else if (media instanceof CompactDisc cd) {  
    container.add(*createStyledLabel*("Title: " + cd.getTitle(), new Font("Segoe UI", Font.*BOLD*, 14)));  
    container.add(*createStyledLabel*("Artist: " + cd.getArtist(), new Font("Segoe UI", Font.*PLAIN*, 12)));  
    for (Track track : cd.getTracks()) {  
    container.add(*createStyledLabel*("Play: " + track.getTitle() + ". Length: " + track.getLength() + " min", new Font("Segoe UI", Font.*PLAIN*, 12)));  
    }  
    }  
     
    playDialog.setTitle("Play " + media.getTitle());  
    return playDialog;  
    }  
     
    private static JLabel createStyledLabel(String text, Font font) {  
    JLabel label = new JLabel(text);  
    label.setFont(font);  
    label.setForeground(new Color(44, 62, 80));  
    return label;  
    }  
   }
5. Tạo giao diện
6. Tạo class CartScreenController điều khiển file fxml

package AimsProject.src.hust.soict.hedspi.main.screen;  
  
import AimsProject.src.hust.soict.hedspi.main.Cart.Cart;  
import AimsProject.src.hust.soict.hedspi.main.aims.media.Media;  
import AimsProject.src.hust.soict.hedspi.main.aims.media.Playable;  
import javafx.beans.value.ChangeListener;  
import javafx.beans.value.ObservableValue;  
import javafx.collections.FXCollections;  
import javafx.event.ActionEvent;  
import javafx.event.EventHandler;  
import javafx.fxml.FXML;  
import javafx.geometry.Pos;  
import javafx.scene.Scene;  
import javafx.scene.control.\*;  
import javafx.scene.control.cell.PropertyValueFactory;  
import javafx.scene.layout.VBox;  
import javafx.scene.paint.Color;  
import javafx.scene.text.Font;  
import javafx.scene.text.FontWeight;  
import javafx.stage.Modality;  
import javafx.stage.Stage;  
  
import javax.swing.\*;  
import java.util.ArrayList;  
  
public class CartScreenController {  
 private Cart cart;  
 @FXML  
 private Button btnPlaceOrder;  
 @FXML  
 private TextField tfFilter;  
 @FXML  
 private ToggleGroup filterCategory;  
 @FXML  
 private RadioButton radioBtnFilterId;  
 @FXML  
 private RadioButton radioBtnFilterTitle;  
 @FXML  
 private Button btnPlay;  
 @FXML  
 private Button btnRemove;  
 @FXML  
 private TableView<Media> tblMedia;  
 @FXML  
 private TableColumn<Media, Integer> colMediaId;  
 @FXML  
 private TableColumn<Media, String> colMediaTitle;  
 @FXML  
 private TableColumn<Media, String> colMediacategory;  
 @FXML  
 private TableColumn<Media, Float> colMediaCost;  
 @FXML  
 private Label totalPrice;  
  
 public CartScreenController(Cart cart) {  
 super();  
 this.cart = cart;  
 }  
  
 @FXML  
 private void initialize() {  
 colMediaId.setCellValueFactory(new PropertyValueFactory<Media, Integer>("id"));  
 colMediaTitle.setCellValueFactory(new PropertyValueFactory<Media, String>("title"));  
 colMediacategory.setCellValueFactory(new PropertyValueFactory<Media, String>("category"));  
 colMediaCost.setCellValueFactory(new PropertyValueFactory<Media, Float>("cost"));  
 tblMedia.setItems(FXCollections.*observableList*(this.cart.getItemsOrdered()));  
 tblMedia.setPlaceholder(new Label("No item in cart"));  
 btnPlay.setVisible(false);  
 btnRemove.setVisible(false);  
 btnRemove.setOnAction(new EventHandler<javafx.event.ActionEvent>() {  
 @Override  
 public void handle(javafx.event.ActionEvent event) {  
 Media media = tblMedia.getSelectionModel().getSelectedItem();  
 cart.removeMedia(media);  
 totalPrice.setText(Float.*toString*(cart.totalCost()) + "$");  
 tblMedia.setItems(FXCollections.*observableList*(cart.getItemsOrdered()));  
 }  
 });  
 tfFilter.textProperty().addListener(new ChangeListener<String>() {  
 @Override  
 public void changed(ObservableValue<? extends String> observableValue, String s, String t1) {  
 showFilterMedia(t1);  
 }  
 });  
  
 tblMedia.getSelectionModel().selectedItemProperty().addListener(  
 new ChangeListener<Media>() {  
 @Override  
 public void changed(ObservableValue<? extends Media> observableValue, Media media, Media t1) {  
 updateButtonBar(t1);  
 }  
 }  
 );  
 totalPrice.setText(Float.*toString*(cart.totalCost()) + "$");  
  
 btnPlay.setOnAction(new EventHandler<ActionEvent>() {  
 @Override  
 public void handle(ActionEvent event) {  
 JDialog playDialog = MediaStore.*createPlayDialog*(tblMedia.getSelectionModel().getSelectedItem());  
 playDialog.setVisible(true);  
 playDialog.setSize(300, 200);  
 playDialog.pack();  
 }  
 });  
  
 btnPlaceOrder.setOnAction(new EventHandler<ActionEvent>() {  
 @Override  
 public void handle(ActionEvent event) {  
 createPopUp();  
 cart.getItemsOrdered().clear();  
 tblMedia.setItems(FXCollections.*observableList*(cart.getItemsOrdered()));  
 totalPrice.setText(Float.*toString*(cart.totalCost()) + "$");  
 }  
 });  
 }  
  
 @FXML  
 void updateButtonBar(Media media) {  
 btnRemove.setVisible(true);  
 if (media instanceof Playable) {  
 btnPlay.setVisible(true);  
 } else {  
 btnPlay.setVisible(false);  
 }  
 }  
  
 @FXML  
 void showFilterMedia(String t1) {  
 if (filterCategory.getSelectedToggle() == radioBtnFilterTitle) {  
 ArrayList<Media> filterByTitle = new ArrayList<Media>();  
 for (Media item : cart.getItemsOrdered()) {  
 if (item.getTitle().contains(t1)) {  
 filterByTitle.add(item);  
 }  
 }  
 tblMedia.setItems(FXCollections.*observableList*(filterByTitle));  
 } else if (filterCategory.getSelectedToggle() == radioBtnFilterId) {  
 ArrayList<Media> filterByID = new ArrayList<Media>();  
 for (Media item : cart.getItemsOrdered()) {  
 if (item.getId() == Integer.*parseInt*(t1)) {  
 filterByID.add(item);  
 }  
 }  
 tblMedia.setItems(FXCollections.*observableList*(filterByID));  
 }  
 }  
  
 @FXML  
 void createPopUp() {  
 Stage popupwindow = new Stage();  
 popupwindow.initModality(Modality.*APPLICATION\_MODAL*);  
 popupwindow.setTitle("Place order");  
  
 Label label1 = new Label("You have placed your order!");  
 label1.setFont(Font.*font*("Arial", FontWeight.*BOLD*, 14));  
 Label label2 = new Label("Your bill total is " + Float.*toString*(cart.totalCost()) + "$");  
 Button button1 = new Button("OK!");  
 label2.setTextFill(Color.*RED*);  
 button1.setOnAction(e -> popupwindow.close());  
 VBox layout = new VBox(10);  
 layout.getChildren().addAll(label1, label2, button1);  
 layout.setAlignment(Pos.*CENTER*);  
 Scene scene1 = new Scene(layout, 300, 200);  
 popupwindow.setScene(scene1);  
 popupwindow.show();  
 }  
}

1. Tạo file Cart.fxml làm giao diện chính

*<?*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 maxHeight="-Infinity" maxWidth="-Infinity" minHeight="-Infinity" minWidth="-Infinity" prefHeight="768.0" prefWidth="1024.0" xmlns="http://javafx.com/javafx/21" xmlns:fx="http://javafx.com/fxml/1">  
 <top>  
 <VBox prefWidth="100.0" BorderPane.alignment="CENTER">  
 <children>  
 <MenuBar>  
 <menus>  
 <Menu mnemonicParsing="false" text="Options">  
 <items>  
 <Menu mnemonicParsing="false" text="Update store">  
 <items>  
 <MenuItem mnemonicParsing="false" text="Add book" />  
 <MenuItem mnemonicParsing="false" text="Add DVD" />  
 <MenuItem mnemonicParsing="false" text="Add CD" />  
 </items>  
 </Menu>  
 <Menu mnemonicParsing="false" text="View store">  
 <items>  
 <MenuItem mnemonicParsing="false" text="Action 1" />  
 </items>  
 </Menu>  
 <Menu mnemonicParsing="false" text="View cart">  
 <items>  
 <MenuItem mnemonicParsing="false" text="Action 1" />  
 </items>  
 </Menu>  
 </items>  
 </Menu>  
 </menus>  
 </MenuBar>  
 <Label text="CART" textFill="#000fff">  
 <font>  
 <Font size="50.0" />  
 </font>  
 </Label>  
 </children>  
 </VBox>  
 </top>  
 <center>  
 <VBox BorderPane.alignment="CENTER">  
 <padding>  
 <Insets left="10.0" />  
 </padding>  
 <children>  
 <HBox alignment="CENTER\_LEFT" prefWidth="200.0" spacing="10.0">  
 <padding>  
 <Insets bottom="10.0" top="10.0" />  
 </padding>  
 <children>  
 <Label text="Filter" />  
 <TextField fx:id="tfFilter" />  
 <RadioButton fx:id="radioBtnFilterId" mnemonicParsing="false" selected="true" text="By ID">  
 <toggleGroup>  
 <ToggleGroup fx:id="filterCategory" />  
 </toggleGroup>  
 </RadioButton>  
 <RadioButton fx:id="radioBtnFilterTitle" mnemonicParsing="false" text="By Title" toggleGroup="$filterCategory" />  
 </children>  
 </HBox>  
 <TableView fx:id="tblMedia" prefHeight="200.0" prefWidth="200.0">  
 <columns>  
 <TableColumn fx:id="colMediaId" prefWidth="50.0" text="ID" />  
 <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>  
 <TableView fx:constant="CONSTRAINED\_RESIZE\_POLICY" />  
 </columnResizePolicy>  
 </TableView>  
 <ButtonBar prefHeight="40.0" prefWidth="200.0">  
 <buttons>  
 <Button fx:id="btnPlay" mnemonicParsing="false" text="Play" />  
 <Button fx:id="btnRemove" mnemonicParsing="false" text="Remove" />  
 </buttons>  
 </ButtonBar>  
 </children>  
 </VBox>  
 </center>  
 <right>  
 <VBox alignment="TOP\_CENTER" BorderPane.alignment="CENTER">  
 <padding>  
 <Insets left="10.0" right="10.0" top="50.0" />  
 </padding>  
 <children>  
 <HBox alignment="CENTER">  
 <children>  
 <Label text="Total:">  
 <font>  
 <Font size="24.0" />  
 </font>  
 <padding>  
 <Insets right="10.0" />  
 </padding>  
 </Label>  
 <Label fx:id="totalPrice" text="0 $" textFill="#000fff">  
 <font>  
 <Font size="24.0" />  
 </font>  
 </Label>  
 </children>  
 </HBox>  
 <Button fx:id="btnPlaceOrder" mnemonicParsing="false" style="-fx-background-color: red;" text="Place Order" textFill="WHITE">  
 <font>  
 <Font size="24.0" />  
 </font>  
 </Button>  
 </children>  
 </VBox>  
 </right>  
</BorderPane>

1. Tạo class CartScreen xử lý sau khi người dùng ấn vào “Add to Cart” và khi xem “View Cart”

package AimsProject.src.hust.soict.hedspi.main.screen;  
  
import AimsProject.src.hust.soict.hedspi.main.Cart.Cart;  
import javafx.application.Platform;  
import javafx.embed.swing.JFXPanel;  
import javafx.fxml.FXMLLoader;  
import javafx.scene.Parent;  
import javafx.scene.Scene;  
  
import javax.swing.\*;  
import java.io.IOException;  
  
public class CartScreen extends JFrame {  
 private Cart cart;  
  
 public CartScreen(Cart cart) {  
 super();  
 this.cart = cart;  
 JFXPanel fxPanel = new JFXPanel();  
 this.add(fxPanel);  
 this.setTitle("Cart");  
 this.setVisible(true);  
 setSize(1100,768);  
 Platform.*runLater*(new Runnable() {  
 @Override  
 public void run() {  
 try {  
 FXMLLoader loader = new FXMLLoader(getClass().getResource("cart.fxml"));  
 CartScreenController controller = new CartScreenController(cart);  
 loader.setController(controller);  
 Parent root = loader.load();  
 fxPanel.setScene(new Scene(root));  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
 }  
 });  
 }  
}

Kết quả

A screenshot of a computer

Description automatically generated

Play 1 playlist nào đó

A screenshot of a computer

Description automatically generated

Thêm sách vào Store

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

View CartScreen

A screenshot of a computer

Description automatically generated

Sort theo ID

A screenshot of a computer

Description automatically generated

Sort theo Title

A screenshot of a computer

Description automatically generated

Filter theo ID

A screenshot of a computer

Description automatically generated

Filter theo Title

A screenshot of a computer

Description automatically generated

Remove Item trong Cart

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Chốt đơn

A screenshot of a computer

Description automatically generated

1. Các biểu đồ
2. Biểu đồ Use-case

A diagram of a company

Description automatically generated

1. Biểu đồ Class

A yellow folders with text

Description automatically generated