

# Báo cáo lab5

Họ và tên: Mai Văn Đăng MSSV: 20225699

Lớp: 744520 Kì 2024-1

## I. New code

### 1) Swing Component

#### 1.1. AWATAccumulator

Nhập số và tính tổng các số đã nhập.

```
1 package hust.soict.hedspi.swing;
2
3 import java.awt.*;
4
5
6 public class AWTAccumulator extends Frame {
7     private TextField tfInput;
8     private TextField tfOutput;
9     private int sum = 0; // Accumulated sum, init to 0
10
11     // Constructor to setup the GUI components and event handlers;
12     public AWTAccumulator() {
13         setLayout(new GridLayout(2, 2));
14
15         add(new Label("Enter an Integer: "));
16         tfInput = new TextField(10);
17         add(tfInput);
18         tfInput.addActionListener(new TFInputListener());
19         add(new Label("The Accumulated Sum is: "));
20         tfOutput = new TextField(10);
21         tfOutput.setEditable(false);
22         add(tfOutput);
23         setTitle("AWT Accumulator");
24         setSize(350, 120);
25         setVisible(true);
26     }
27     public static void main(String[] args) {
28         new AWTAccumulator();
29     }
30     private class TFInputListener implements ActionListener {
31         @Override
32         public void actionPerformed(ActionEvent evt) {
33             int numberIn = Integer.parseInt(tfInput.getText());
34             sum += numberIn;
35             tfInput.setText("");
36             tfOutput.setText(sum+"");
37         }
38     }
39 }
```

#### 1.2. SwingAccumulator

Chức năng tương tự AWATAccumulator.

```

package hust.soict.hedspi.swing;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import javax.swing.*;
public class SwingAccumulator extends JFrame{
    private JTextField tfInput;
    private JTextField tfOutput;
    private int sum=0;
    public SwingAccumulator() {
        Container cp = getContentPane();
        cp.setLayout(new GridLayout(2,2));
        cp.add(new JLabel("Enter an Integer: "));
        tfInput = new JTextField(10);
        cp.add(tfInput);
        tfInput.addActionListener(new TFInputListener());
        cp.add(new JLabel("The Accumulated Sum is: "));
        tfOutput = new JTextField(10);
        tfOutput.setEditable(false);
        cp.add(tfOutput);
        setTitle("Swing Accumulator");
        setSize(350, 120);
        setVisible(true);
    }
    public static void main(String[] args) {
        new SwingAccumulator();
    }
    private class TFInputListener implements ActionListener {
        @Override
        public void actionPerformed(ActionEvent evt) {
            int numberIn = Integer.parseInt(tfInput.getText());
            sum += numberIn;
            tfInput.setText("");
            tfOutput.setText(sum+"");
        }
    }
}

```

### 1.3. So sánh các thành phần của Swing và AWT

Lập trình với AWT và Swing khá giống nhau (bao gồm các thành phần container, xử lý sự kiện). Tuy nhiên, có một số khác biệt cần lưu ý:

- Container cấp cao nhất trong Swing và AWT.
- Tên lớp của các thành phần trong AWT và tên lớp tương ứng trong Swing.

## 2) Tổ chức Swing components with Layout Managers

### 2.1. Tạo lớp NumberGrid

```

7 public class NumberGrid extends JFrame {
8     private JButton[] btnNumbers = new JButton[10];
9     private JButton btnDelete, btnReset;
10    private JTextField tfDisplay;
11
12    public NumberGrid() {
13
14        tfDisplay = new JTextField();
15        tfDisplay.setComponentOrientation(ComponentOrientation.RIGHT_TO_LEFT);
16
17        JPanel panelButtons = new JPanel(new GridLayout(4, 3));
18        addButtons(panelButtons);
19
20        Container cp = getContentPane();
21        cp.setLayout(new BorderLayout());
22        cp.add(tfDisplay, BorderLayout.NORTH);
23        cp.add(panelButtons, BorderLayout.CENTER);
24
25        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
26        setTitle("Number Grid");
27        setSize(200, 200);
28        setVisible(true);
29    }

```

## 2.2. Thêm buttons

```

void addButtons(JPanel panelButtons) {
    ButtonListener 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);
}

```

## 2.3. Hoàn thiện lớp bên trong ButtonListener

```

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 delString = tfDisplay.getText();
            if (delString.length() > 0) {
                delString = delString.substring(0, delString.length()-1);
            }
            tfDisplay.setText(delString);
        }
        else if (button.equals("C")){
            tfDisplay.setText("");
        }
    }
}

```

### 3) Tạo GUI cho AIMS với Swing

Tạo ứng dụng AIMS, triển khai ba màn hình:

- Màn hình “View Store” sử dụng Swing.
- Màn hình “View Cart” sử dụng JavaFX.
- Màn hình “Update Store” sử dụng JavaFX.
- Với View Store Screen mã nguồn sẽ được đặt trong package: hust.soict.hedspi.aims.screen
- Đối với màn hình View Store, chúng ta sẽ sử dụng

BorderLayout:

- Component NORTH sẽ chứa menu bar và header.
- Component CENTER sẽ chứa một panel sử dụng GridLayout, mỗi ô trong đó là một mặt hàng của cửa hàng.

#### 3.1. Tạo lớp StoreScreen

Đây là View Store Screen

```

public class StoreScreen extends JFrame{
    private static Store store = new Store();

```

#### 3.2. Component NORTH

```

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

```

- Tạo createMenuBar()

```

JMenuBar createMenuBar() {
    JMenu menu = new JMenu("Options");

    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);

    smAddBook.addActionListener(new btnMenuListener());
    smAddCD.addActionListener(new btnMenuListener());
    smAddDVD.addActionListener(new btnMenuListener());

    menu.add(smUpdateStore);

    JMenuItem viewStoreMenu = new JMenuItem("View store");
    JMenuItem viewCartMenu = new JMenuItem("View cart");
    menu.add(viewStoreMenu);
    menu.add(viewCartMenu);

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

    return menuBar;
}

```

- Tạo createHeader()

```

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

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

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

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

    return header;
}

```

### 3.3. Component CENTER

```

JPanel createCenter() {

    JPanel center = new JPanel();
    center.setLayout(new GridLayout(3, 3, 2, 2));

    ArrayList<Media> mediaInStore = store.getItemsInStore();
    for (int i = 0; i < mediaInStore.size(); i++) {
        MediaStore cell = new MediaStore(mediaInStore.get(i), cart);
        center.add(cell);
    }

    return center;
}

```

- Thêm method `getItemsInStore` vào Store

```

public ArrayList<Media> getItemsInStore() {
    return itemsInStore;
}

```

### 3.4. Lớp MediaStore

```

public class MediaStore extends JPanel {
    public MediaStore(Media media, Cart cart) {

        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));

        // Thêm tương tác cho nút Add to cart
        JButton addToCartButton = new JButton("Add to cart");
        if (media instanceof Playable) {
            JButton playButton = new JButton("Play");
        }

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

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

```

### 3.5. Kết hợp tất cả

```

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

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

    setTitle("Store");
    setSize(1024, 768);
    setVisible(true);
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
}

```

### 3.6. Thêm tương tác người dùng vào các nút trên Media

```

// Thêm tương tác cho nút Add to cart
JButton addToCartButton = new JButton("Add to cart");
addToCartButton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        try {
            String message = cart.addMedia(media);
            JOptionPane.showMessageDialog(null, message);
        } catch (LimitExceededException ex) {
            JOptionPane.showMessageDialog(null, ex.getMessage(), "Error", JOptionPane.ERROR_MESSAGE);
        }
    }
});
container.add(addToCartButton);

// Thêm tương tác cho nút Play
if (media instanceof Playable) {
    JButton playButton = new JButton("Play");
    playButton.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {

            JDialog dialog = new JDialog();
            dialog.setTitle(media.getTitle());
            dialog.setSize(400, 300);

            String mediaInfo = "";
            try {
                mediaInfo = "<html>" + media.playGUI().replace("\n", "<br/>") + "</html>";
                JLabel mediaLabel = new JLabel(mediaInfo);
                mediaLabel.setVerticalAlignment(JLabel.CENTER);
                mediaLabel.setHorizontalAlignment(JLabel.CENTER);
                JScrollPane scrollPane = new JScrollPane(mediaLabel);
                scrollPane.setVerticalScrollBarPolicy(JScrollPane.VERTICAL_SCROLLBAR_AS_NEEDED);

                dialog.add(scrollPane);
                dialog.setVisible(true);
            } catch (PlayerException e1) {
                JOptionPane.showMessageDialog(null, e1.getMessage(), "Error", JOptionPane.ERROR_MESSAGE);
            }
        }
    });
    container.add(playButton);
}

```

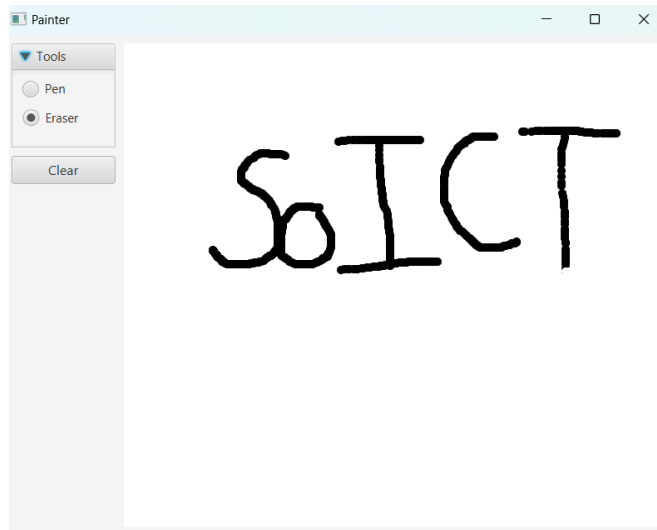
## 4) API JavaFX

Tất cả được đặt trong `hust.soict.hedspi.javafx`

Gồm 3 file:

- Painter.java dùng để kết nối.
- PainterController.java dùng để điều khiển các chức năng.
- Painter.fxml dùng để thiết kế giao diện hiển thị

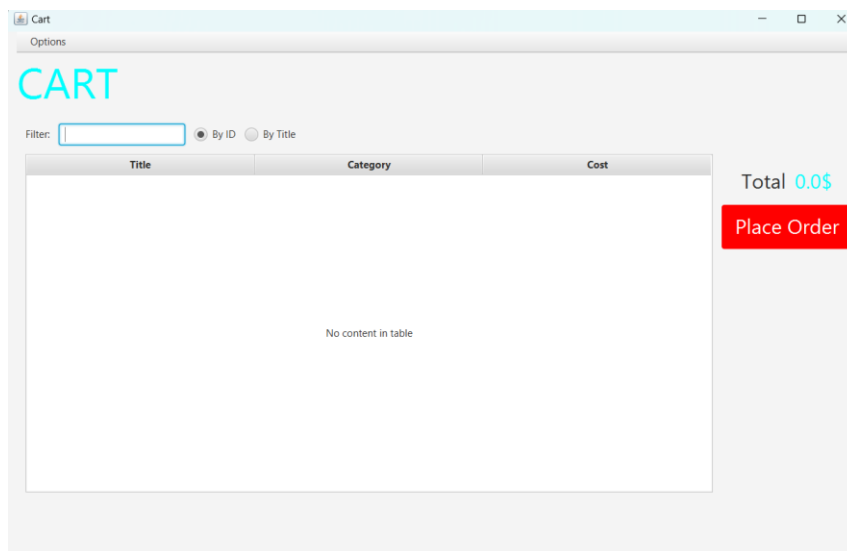
Kết quả hiển thị:



#### 5) View Cart Screen với Scene Builder

Thực hành trong package `hust.soict.hedspi.aims.screen`

- Kết quả hiển thị của file `cart.fxml` trong `hust.soict.hedspi.aims.screen.view`



#### 6) Tích hợp JavaFX vào Swing – Lớp JFXPanel

- Tạo lớp `CartScreen`



```

public class CartScreen extends JFrame{
    private static Cart cart;
    public CartScreen(Cart cart) {
        super();

        this.cart = cart;

        JFXPanel fxPanel = new JFXPanel();
        this.add(fxPanel);

        this.setTitle("Cart");
        this.setSize(1024, 768);
        this.setVisible(true);
        Platform.runLater(new Runnable() {
            @Override
            public void run() {
                try {
                    FXMLLoader loader = new FXMLLoader(getClass()
                        .getResource("/hust/soict/hedspi/aims/screen/view/cart.fxml"));

                    CartScreenController controller = new CartScreenController(cart);
                    loader.setController(controller);
                    Parent root = loader.load();
                    fxPanel.setScene(new Scene(root));
                } catch (IOException e) {
                    e.printStackTrace();
                }
            }
        });
    }

    public static void main(String[] args) {
        new CartScreen(cart);
    }
}

```

## 7) Hiện thị các mục trong Store – JavaFX's data-driven UI

Tạo CartScreenController trong hust.soict.hedspi.aims.screen.controller

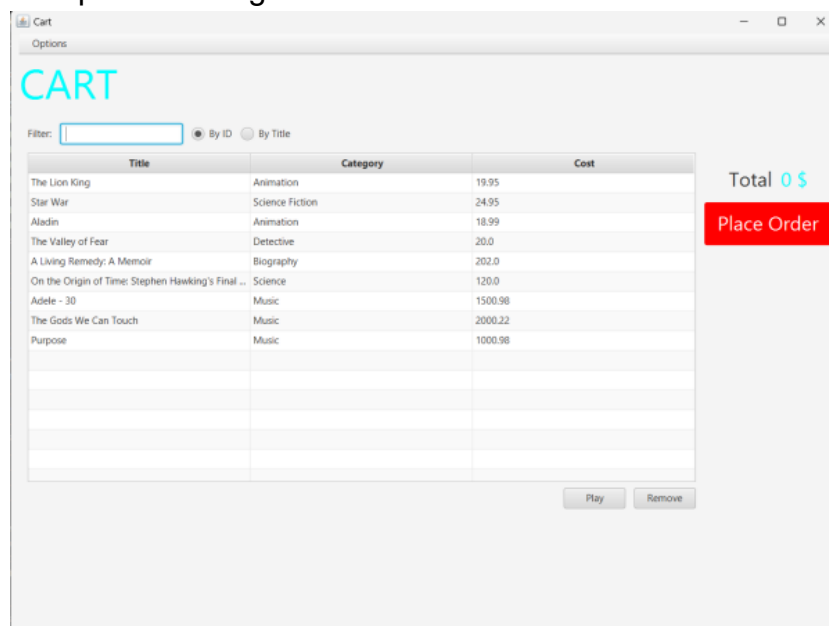
Sau khi tạo xong cần Thay đổi thuộc tính itemsOrdered từ List<Media> thành ObservableList<Media> và thêm method getItemsOrdered() trong lớp Cart

```

private ObservableList<Media> itemsOrdered = FXCollections.observableArrayList();
public ObservableList<Media> getItemsOrdered() {
    return itemsOrdered;
}

```

Kết quả cuối cùng:



## 8) Cập nhật các nút trong TableView – ChangeListener

Cập nhật nút btnPlay và btnRemove trong CartScreenController

```
@FXML
private Button btnPlay;

@FXML
private Button btnRemove;

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);
            }
        }

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

9) Xóa 1 Media – nút btnRemove

- Triển khai sự kiện cho nút Remove

```
@FXML
void btnRemovePressed(ActionEvent event) {
    Media media = tblMedia.getSelectionModel().getSelectedItem();
    cart.removeMedia(media);
    costLabel.setText(cart.totalCost() + " $");
}
```

- Đồng thời thêm Action trong Cart.fxml

10) Lọc các mục trong giỏ hàng

TextField nơi người dùng nhập chuỗi lọc: tfFilter

Hai nút radio để xác định tiêu chí lọc:

- Nút “By ID”: radioBtnFilterId
- Nút “By Title”: radioBtnFilterTitle

```
@FXML
private TextField tfFilter;

@FXML
private RadioButton radioBtnFilterId;

@FXML
private RadioButton radioBtnFilterTitle;
```

Thêm ChangeListener cho tfFilter trong initialize()

```

tfFilter.textProperty().addListener(
    new ChangeListener<String>() {

        @Override
        public void changed(ObservableValue<? extends String> observable, String oldValue, String newValue) {
            showFilteredMedia(newValue);
        }

        private void showFilteredMedia(String keyword) {
            FilteredList<Media> filteredList = new FilteredList<>(cart.getItemsOrdered());

            if (!keyword.isEmpty() && radioBtnFilterId.isSelected()) {
                filteredList.setPredicate(media -> {
                    String idString = String.valueOf(media.getId());
                    return idString.equals(keyword);
                });
            } else if (!keyword.isEmpty() && radioBtnFilterTitle.isSelected()) {
                filteredList.setPredicate(media -> {
                    String title = media.getTitle().toLowerCase();
                    return title.contains(keyword.toLowerCase());
                });
            } else {
                filteredList.setPredicate(null);
            }
            tblMedia.setItems(filteredList);
        }
    });

```

## 11) Hoàn thành ứng dụng GUI của Aims

### 11.1. Hoàn thành Cart Screen

- Nút Place order private Button placeOrder;

```

@FXML
void placeOrderPressed(ActionEvent event) {
    Alert alert = new Alert(Alert.AlertType.INFORMATION, cart.placeOrder());
    alert.setTitle("Order created");
    alert.setHeaderText(null);
    alert.showAndWait();
}

```

- Thêm method placeOrder trong lớp Cart

```

public String placeOrder() {
    if(itemsOrdered.size()==0) {
        return "Your cart is empty!";
    } else {
        qtyOrdered = 0;
        itemsOrdered.clear();
        return "Order created!\n" + "Now your cart will be empty!";
    }
}

```

- Nút play
  - Cập nhật Cost vào thao tác xóa Media trong cart
  - Kết nối StoreScreen với CartScreen
  - Thêm new Cart vào lớp StoreScreen
- ```

public class StoreScreen extends JFrame{
    private static Store store = new Store();
    private static Cart cart = new Cart();

```
- Xóa Cart, initsetup ở lớp CartScreen.java
  - Mỗi lần ấn nút Add to cart thì add Media vào cart vừa tạo (MediaStore.java)

```
// Thêm tương tác cho nút Add to cart
JButton addToCartButton = new JButton("Add to cart");
addToCartButton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        try {
            String message = cart.addMedia(media);
            JOptionPane.showMessageDialog(null, message);
        } catch (LimitExceededException ex) {
            JOptionPane.showMessageDialog(null, ex.getMessage(), "Error", JOptionPane.ERROR_MESSAGE);
        }
    }
});
container.add(addToCartButton);
```

- Sửa lại dạng trả về của method addMedia() trong Cart.java

```
public String addMedia(Media media) throws LimitExceededException {
    if (itemsOrdered.size() >= MAX_NUMBERS_ORDERED) {
        throw new LimitExceededException("ERROR: The number of media has reached its limit");
    } else if (itemsOrdered.contains(media)){
        return media.getTitle() + " is already in the cart!";
    } else {
        itemsOrdered.add(media);
        return (media.getTitle() + "has been added!");
    }
}
```

- Thêm action View Cart thì hiện ra CartScreen với những cart đã thêm

```
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(cart);
    }
});
```

## 11.2. Update Store Screen

Tạo 3 lớp

- AddDigitalVideoDiscToStoreScreen
- AddBookToStoreScreen
- AddCompactDiscToStoreScreen
- AddTrackScreen
- Trong hust.soict.hedspi.aims.screen.view  
→ Các giao diện lần lượt như sau:

addBook.fxml

addCD.fxml

addDVD.fxml

addTracks.fxml

## 12) Kiểm tra source code trước đó để xử lý Exception

- Xem xét tất cả các method, class trong lớp AimsProject để catch/handle tất cả những exception cần thiết.

- Throw Exception cho method addMedia() trong Cart.java và testCart.java

```
public String addMedia(Media media) throws LimitExceededException {
    if (itemsOrdered.size() >= MAX_NUMBERS_ORDERED) {
        throw new LimitExceededException("ERROR: The number of media has reached its limit");
    } else if (itemsOrdered.contains(media)){
        return media.getTitle() + " is already in the cart!";
    } else {
        itemsOrdered.add(media);
        return (media.getTitle() + "has been added!");
    }
}
```

```
package hust.soict.hedspi.test.cart;
import javax.naming.LimitExceededException;

public class CartTest {
    public static void main(String[] args) throws LimitExceededException {

        Cart cart = new Cart();

        DigitalVideoDisc dvd1 = new DigitalVideoDisc("The Lion King",
            "Animation", "Roger Allers", 87, 19.95f);
        cart.addMedia(dvd1);

        DigitalVideoDisc dvd2 = new DigitalVideoDisc("Star War",
            "Science Fiction", "George Lucas", 87, 24.95f);
        cart.addMedia(dvd2);

        DigitalVideoDisc dvd3 = new DigitalVideoDisc("Aladin",
            "Animation", 18.99f);
        cart.addMedia(dvd3);

        DigitalVideoDisc dvd4 = new DigitalVideoDisc("Aladin",
            "Animation", 18.99f);
        cart.addMedia(dvd4);

        cart.print();

        cart.searchByID(3);
        cart.searchByTitle("Lion");
    }
}
```

- Đồng thời catch Exception trong MediaStore.java và Aims.java

```
// Thêm tương tác cho nút Add to cart
JButton addToCartButton = new JButton("Add to cart");
addToCartButton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        try {
            String message = cart.addMedia(media);
            JOptionPane.showMessageDialog(null, message);
        } catch (LimitExceededException ex) {
            JOptionPane.showMessageDialog(null, ex.getMessage(), "Error", JOptionPane.ERROR_MESSAGE);
        }
    }
});
container.add(addToCartButton);
```

```

Media media = store.search(titleForRemove);
if (media != null) {
    clearConsole();
    store.removeMedia(media);
    foundToRemove = true;
} else {
    System.out.println("***MEDIA NOT FOUND***");
}

case 1:
try {
    cart.addMedia(media);
} catch (LimitExceededException e) {
    e.printStackTrace();
}
break;

```

### 13) Tạo một lớp kế thừa từ Exception

#### 13.1. Tạo 1 lớp mới có tên là PlayerException

- Tạo 1 package hust.soict.hedspi.aims.exception, sau đó tạo lớp PlayerException

```

package hust.soict.hedspi.aims.exception;

public class PlayerException extends Exception {
    public PlayerException(String message) {
        super(message);
    }
}

```

#### 13.2. Raise ngoại lệ PlayerException vào method play()

- Thêm vào Media

```

public String playGUI() throws PlayerException {
    return "Playing media";
}

```

- Thêm vào DigitalVideoDisc
- Thêm vào Track

#### 13.3. Cập nhật play() trong interface Playable

- Không cập nhật vì ở đây dùng exception cho playGUI()

#### 13.4. Cập nhật play() trong lớp CompactDisc

```

public String playGUI() throws PlayerException {
    if (this.getLength() > 0) {
        String output = "Playing CD: " + this.getTitle() + "\n" +
            "CD length: " + formatDuration(this.getLength()) + "\n" + "\n";
        for (Track track : tracks) {
            try {
                output += track.playGUI() + "\n";
            } catch (PlayerException e) {
                output += track.getTitle() + "\n" + e.getMessage();
            }
        }
        return output;
    } else {
        throw new PlayerException("ERROR: CD length is non-positive!");
    }
}

```

- Cập nhật các lớp khác

##### • MediaStore

```

// thêm tương tác cho nút Play
if (media instanceof Playable) {
    JButton playButton = new JButton("Play");
    playButton.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {

            JDialog dialog = new JDialog();
            dialog.setTitle(media.getTitle());
            dialog.setSize(400, 300);

            String mediaInfo = "";
            try {
                mediaInfo = "<html>" + media.playGUI().replace("\n", "<br/>") + "</html>";
                JLabel mediaLabel = new JLabel(mediaInfo);
                mediaLabel.setVerticalAlignment(JLabel.CENTER);
                mediaLabel.setHorizontalAlignment(JLabel.CENTER);
                JScrollPane scrollPane = new JScrollPane(mediaLabel);
                scrollPane.setVerticalScrollBarPolicy(JScrollPane.VERTICAL_SCROLLBAR_AS_NEEDED);

                dialog.add(scrollPane);
                dialog.setVisible(true);
            } catch (PlayerException e1) {
                JOptionPane.showMessageDialog(null, e1.getMessage(), "Error", JOptionPane.ERROR_MESSAGE);
            }

        }
    });
    container.add(playButton);
}

```

##### • CartScreenController

```

@FXML
void btnPlayPressed(ActionEvent event) {
    Media media = tblMedia.getSelectionModel().getSelectedItem();
    Alert alert;
    try {
        alert = new Alert(Alert.AlertType.NONE, media.playGUI());
        alert.setTitle("Playing");
        alert.setHeaderText(null);
        alert.getDialogPane().getButtonTypes().add(ButtonType.OK);
        alert.showAndWait();
    } catch (PlayerException e) {
        alert = new Alert(Alert.AlertType.ERROR, e.getMessage());
        alert.setTitle("ERROR");
        alert.setHeaderText(null);
        alert.showAndWait();
    }
}

```

#### 14) Cập nhật lớp Aims

- Vì Exception được throw tại method playGUI() nên không ảnh hưởng lớp Aims.

#### 15) Thay đổi method equal() của lớp Media

- Tránh gặp NullPointerException và ClassCastException

```

@Override
public boolean equals(Object obj) {
    if (obj == this) {
        return true;
    }
    if (obj == null || !(obj instanceof Media)) {
        return false;
    }
    Media otherMedia = (Media) obj;
    return this.getTitle() != null && this.getTitle().equals(otherMedia.getTitle());
}

```

## II. UML Diagram

### Class Diagram

