

LAB 05 REPORT

GUI Programming with Java Swing

Student: Le Viet Anh

Student ID: 202416774

Course: Object-Oriented Programming (OOP)

Instructor: Ho Viet Duc Luong

Semester: 2024–2025

Contents

1	Introduction	2
2	Objectives	2
3	Development Environment	2
4	System Architecture	2
4.1	Relationship with Lab 04	2
4.2	Package Structure	2
5	Implementation	3
5.1	StoreScreen Class	3
5.2	MediaStore Class	3
5.3	CartScreen Class	4
6	Results	4
6.1	Store Interface	4
6.2	Cart Interface	5
7	Evaluation	5
8	Conclusion	6

1 Introduction

Lab 05 focuses on building a graphical user interface (GUI) for the AIMS system developed in Lab 04. In this lab, Java Swing is used to design interactive windows, while reusing the core object-oriented classes such as **Store**, **Cart**, and **Media**.

2 Objectives

The main objectives of Lab 05 are:

- To become familiar with GUI programming using Java Swing
- To build Store and Cart interfaces for the AIMS system
- To connect GUI components with the business logic implemented in Lab 04
- To practice event handling in Java

3 Development Environment

- Programming Language: Java
- IDE: Eclipse
- GUI Library: Java Swing (AWT & Swing)
- Operating System: Windows

4 System Architecture

4.1 Relationship with Lab 04

Lab 05 reuses all model classes implemented in Lab 04, including:

- `Media`, `Book`, `CompactDisc`, `DigitalVideoDisc`
- `Store`
- `Cart`

No changes are made to the core logic of these classes; they are only integrated into the GUI layer.

4.2 Package Structure

- `hust.soict.cs.dsai.aims.media`: Media-related classes
- `hust.soict.cs.dsai.aims.store`: Store class
- `hust.soict.cs.dsai.aims.cart`: Cart class
- `hust.soict.cs.dsai.aims.screen`: GUI classes (Lab 05)

5 Implementation

5.1 StoreScreen Class

The `StoreScreen` class extends `JFrame` and represents the main window of the application. It displays all available media in the store and allows users to:

- View media information
- Add media to the cart
- Play media if it implements the `Playable` interface

```
1 public class StoreScreen extends JFrame {  
2  
3     private Store store;  
4     private Cart cart;  
5  
6     public StoreScreen(Store store) {  
7         this.store = store;  
8         this.cart = new Cart();  
9         setTitle("AIMS Store");  
10        setSize(1024, 768);  
11        setVisible(true);  
12    }  
13 }
```

5.2 MediaStore Class

The `MediaStore` class extends `JPanel` and represents a single media item displayed in the `StoreScreen`. Each `MediaStore` panel includes:

- Media title
- Media cost
- An Add to Cart button
- A Play button (if applicable)

```
1 public class MediaStore extends JPanel {  
2  
3     public MediaStore(Media media) {  
4         JLabel title = new JLabel(media.getTitle());  
5         JButton addToCart = new JButton("Add to Cart");  
6     }  
7 }
```

5.3 CartScreen Class

The `CartScreen` class extends `JFrame` and displays all media items currently stored in the cart, along with the total cost.

```
1 public class CartScreen extends JFrame {  
2  
3     private Cart cart;  
4  
5     public CartScreen(Cart cart) {  
6         this.cart = cart;  
7         setTitle("Cart");  
8         setSize(600, 400);  
9         setVisible(true);  
10    }  
11 }
```

6 Results

6.1 Store Interface

Figure 1 shows the `StoreScreen` interface displaying available media items.

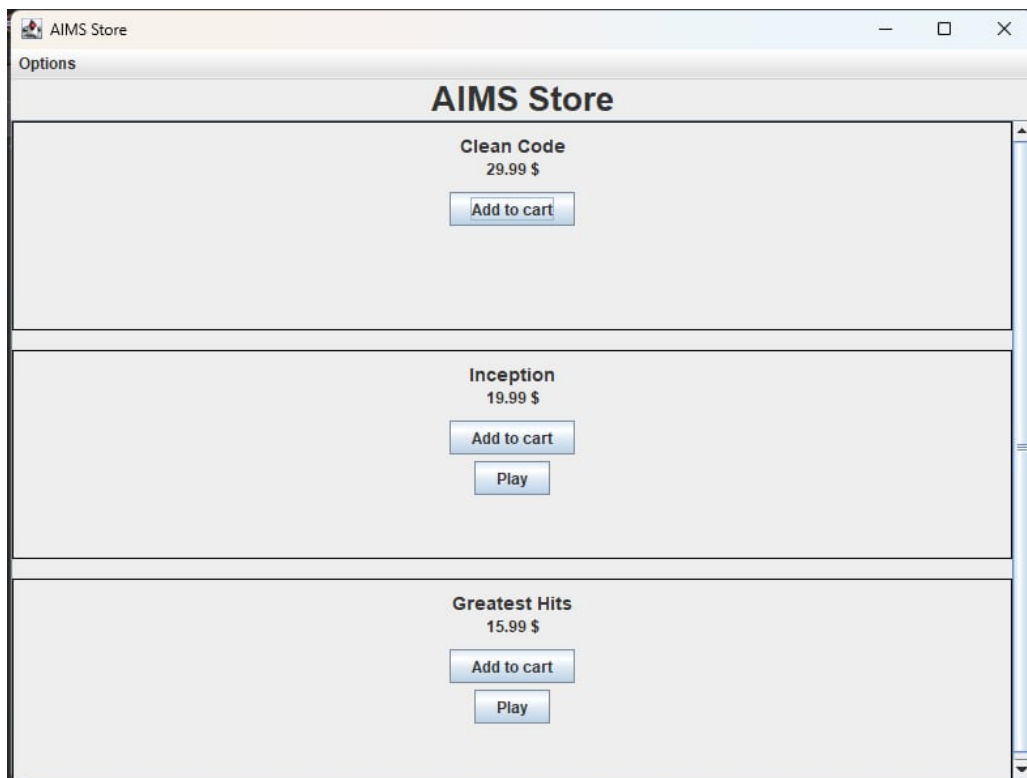


Figure 1: `StoreScreen` Interface

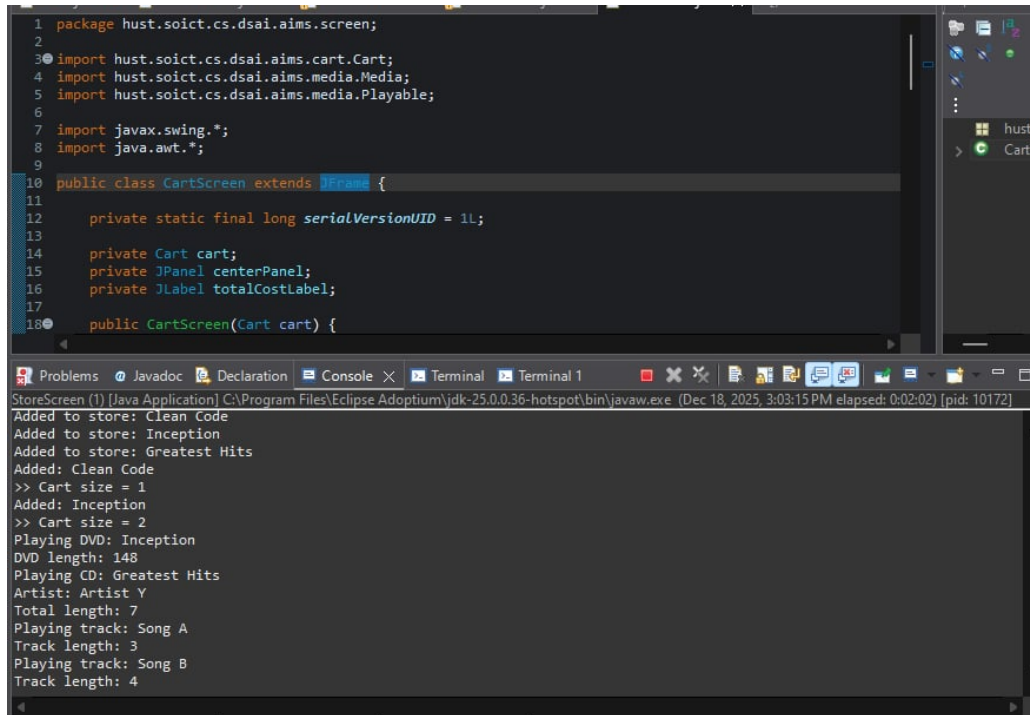
Screenshot instructions:

- Run the `StoreScreen` class

- Ensure multiple media items are displayed
- Capture the full Store window
- Save the image as `Store.jpg`

6.2 Cart Interface

Figure 2 shows the CartScreen after adding media items.



```

1 package hust.soict.cs.dsai.aims.screen;
2
3 import hust.soict.cs.dsai.aims.cart.Cart;
4 import hust.soict.cs.dsai.aims.media.Media;
5 import hust.soict.cs.dsai.aims.media.Playable;
6
7 import javax.swing.*;
8 import java.awt.*;
9
10 public class CartScreen extends JFrame {
11
12     private static final long serialVersionUID = 1L;
13
14     private Cart cart;
15     private JPanel centerPanel;
16     private JLabel totalCostLabel;
17
18     public CartScreen(Cart cart) {

```

```

StoreScreen (1) [Java Application] C:\Program Files\Eclipse Adoptium\jdk-25.0.0-hotspot\bin\javaw.exe (Dec 18, 2025, 3:03:15 PM elapsed: 0:02:02) [pid: 10172]
Added to store: Clean Code
Added to store: Inception
Added to store: Greatest Hits
Added: Clean Code
>> Cart size = 1
Added: Inception
>> Cart size = 2
Playing DVD: Inception
DVD length: 148
Playing CD: Greatest Hits
Artist: Artist Y
Total length: 7
Playing track: Song A
Track length: 3
Playing track: Song B
Track length: 4

```

Figure 2: CartScreen Interface

Screenshot instructions:

- Add at least two media items to the cart
- Open the CartScreen
- Capture the full Cart window
- Save the image as `Cart.jpg`

7 Evaluation

The application fully satisfies the requirements of Lab 05. The GUI is responsive and user-friendly, and all core functionalities such as Add to Cart, Play Media, and View Cart work correctly. Reusing the model layer from Lab 04 improves maintainability and extensibility.

8 Conclusion

Through Lab 05, students gain practical experience in building graphical user interfaces using Java Swing and integrating them with object-oriented application logic. This lab demonstrates how GUI programming complements core OOP principles in real-world applications.