Object-Oriented Programming Laboratory Project

Võ Nhật Tân

1 General Requirements

- Execute as specified in the project guidelines
- Each group must have a minimum of 3 members and a maximum of 5 members.
- The group should designate one representative to submit the project.
- The submission should be a single compressed ZIP file named **[GroupID].zip**, which includes:
  - Source Code: a link to group's source code **[GroupID - SC].txt**
  - The technical report: **[GroupID - Report].pdf**
  - The slides: **[GroupID - Slide].pptx** or **[GroupID - Slide].pdf**
  - The video demo: a link to group's demo video **[GroupID - Video].txt**

2 Objectives

- Apply object-oriented programming (OOP) concepts to develop a functional and maintainable application.
- Understand and demonstrate the four pillars of OOP: encapsulation, inheritance, polymorphism, and abstraction.
- Explore and apply some appropriate software design patterns to solve real-world programming problems effectively.
- Practice programming, software design, teamwork, documentation, and technical presentation skills.

3 Project Requirements

3.1 Problem Statement

Students must develop a management system application using OOP principles. The topic is open but must be related to management. Suggested examples include:

- Library Management System
- Student Information Management
- Book or Inventory Management System
- Hospital or Clinic Management
- Employee or Attendance Management
- ...

3.2 Programming and Design Requirements

- Programming Language: C++, Java, or C# (choose one).
- The system must demonstrate the four core OOP principles:
  - **Encapsulation**
  - **Inheritance**
  - **Polymorphism**
  - **Abstraction**
- At least two design patterns must be applied.

3.3 User Interface Requirements

- The application must include a friendly, clearly formatted, and fully interactive user interface.
- Console-based UI is acceptable but must:
  - Support multi-level menu navigation and structured data input.
  - Clearly display output with proper formatting and user guidance.
  - Allow smooth and error-checked interaction.
- GUI is strongly encouraged. Recommended GUI frameworks for each language include:
  - **C++**: Qt, wxWidgets
  - **Java**: Java Swing, JavaFX
  - **C#**: WinForms, WPF

3.4 Technical Report

The report must include the following sections:

a. Group Introduction

- Group details: name, member names, student IDs.
- Individual contributions to the project.
- Self-evaluation of how well each requirement was met.

b. System Design Overview

- Description of the problem domain and chosen management topic.
- Show the main features that your system will support
- Draw UML class diagrams (**Document Reference**)

c. Application Flow

- Flowcharts of user interactions and main program logic.
- Description of key modules and components.
- Example input/output with screenshots.

d. Challenges and Future Enhancements

- Problems faced during development.
- Ideas for improvement or additional features.

3.5 Video Requirement

A short video (10–20 minutes) must be submitted along with the project to demonstrate the functionality and features of your application.

4 Contact

If you have any questions during the project, please feel free to send an email to vntan.work@gmail.com.

— GOOD LUCK! —