Design Principles

- Software Design and Construction
- GVHD: Nguyễn Thị Thu Trang
- SVTH: Nguyễn Đình Quang - 20146574

Design Principles

- Tại sao cần phải thiết kế ?
- Mối liên hệ giữa Design Principles & Design Patterns ?
- Class Design principles - 5 nguyên lý.
  - The Single Responsibility Principle - Nguyên lý đơn nhiệm.
    - Một class chỉ nên có một trách nhiệm duy nhất.
  - The Open Closed principle - Nguyên lý mở rộng/hạn chế.
    - Các thực thể phần mềm (các lớp, mô-đun, chức năng, v.v.) nên được Open để mở rộng, nhưng Closed để sửa đổi.
  - The Liskov Substitution Principle - Nguyên lý thay thế Liskov.
    - Class con có thể thay thế class cha mà không làm thay đổi tính đúng đắn của chương trình.
  - The Interface Segregation Principle - Nguyên lý phân tách giao tiếp.
    - Phân tách các Interface lớn thành nhiều các interface nhỏ với mục đích cụ thể. Khi đó việc implement và quản lý sẽ dễ hơn.
  - The Dependency inversion principle - Nguyên lý nghịch đảo phụ thuộc.
    - Các module cấp cao không nên phụ thuộc vào các modules cấp thấp. Cả 2 nên phụ thuộc vào những cái trừu tượng (abstractions).
    - Những cái trừu tượng không nên phụ thuộc vào chi tiết, mà ngược lại.

Design Principles

- Các nguyên lý đã áp dụng:
  - The Single Responsibility Principle - Nguyên lý đơn nhiệm.
    - Một class chỉ nên có một trách nhiệm duy nhất.
  - The Open Closed principle - Nguyên lý mở rộng/hạn chế.
    - Các thực thể phần mềm (các lớp, mô-đun, chức năng, v.v.) nên được Open để mở rộng, nhưng Closed để sửa đổi.
  - The Interface Segregation Principle - Nguyên lý phân tách giao tiếp.
    - Phân tách các Interface lớn thành nhiều các interface nhỏ với mục đích cụ thể. Khi đó việc implement và quản lý sẽ dễ hơn.

Design Principles

- The Single Responsibility Principle - Nguyên lý đơn nhiệm.
  - Một class chỉ nên có một trách nhiệm duy nhất.
  - Các class trong package ui chứa các giao diện tương tác với người dùng.
  - Các class trong package logic chứa các ràng buộc, chuyển đổi, điều hướng, nhận thông tin từ ui rồi chuyển sang cho tầng data xử lý. rồi nhận thông tin từ tầng data chuyển ngược lại cho tầng BLL thông qua các Value Object.

Design Principles

- The Single Responsibility Principle - Nguyên lý đơn nhiệm.
  - Một class chỉ nên có một trách nhiệm duy nhất.
  - Các class trong package ui chứa các giao diện tương tác với người dùng
  - Các class trong package logic chứa các ràng buộc, chuyển đổi, điều hướng, nhận thông tin từ ui rồi chuyển sang cho tầng data xử lý. rồi nhận thông tin từ tầng data chuyển ngược lại cho tầng BLL thông qua các Value Object.
  - Các class trong package dao chứa thông tin truy cập dữ liệu thông qua các Value Object.

Design Principles

- The Interface Segregation Principle - Nguyên lý phân tách giao tiếp.
- Ex: Chức năng quản lý sinh viên

```
package edu.hust.soict.dao;

import edu.hust.soict.objects.Student;
import java.util.ArrayList;

public interface InterfaceStudentDAO {

    public ArrayList<Student> getAll();

    public ArrayList<Student> findByIDStudent(String maSV);

    public boolean addNew(Student sv);

    public boolean updateByID(Student sv);

    public ArrayList<Student> checkID(String masv);
}
```

Design Principles

- The Interface Segregation Principle - Nguyên lý phân tách giao tiếp.
- Ex: Chức năng quản lý User Account.

```
package edu.hust.soict.dao;

import edu.hust.soict.objects.AddUser;
import java.sql.ResultSet;
import java.util.ArrayList;

public interface InterfaceAddUserDAO {

    public ArrayList<AddUser> getAll();

    public ArrayList<AddUser> findUserName(String userName);

    public boolean addNew(AddUser user);

    public AddUser updateByID(AddUser user);

    public ArrayList<AddUser> checkID(String tenDangNhap);

    public boolean updateUser2(AddUser user);

    public AddUser extractUserFromResultSet(ResultSet rs);

}
```

Design Principles

- Phương thức Update user.

```
public boolean updateUser2(AddUser user) {

    if (DBConnect.open()) {
        PreparedStatement ps = null;
        try {
            ps = cnn.prepareStatement("UPDATE dang_nhap SET password=? WHERE user_name=?");
            String encryptedPassword = PasswordUtils.encryptPassword(user.getPassword());
            ps.setString(1, encryptedPassword);
            ps.setString(2, user.getUser_name());
            int i = ps.executeUpdate();
            if (i == 1) {
                return true;
            }else{
            return false;
            }
        } catch (SQLException ex) {
            Logger.getLogger(AddUserDAO.class.getName()).log(Level.SEVERE, null, ex);
        } finally {
            DBConnect.close(ps);
        }
    }
    return false;
}
```