**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**

**KHOA CÔNG NGHỆ THÔNG TIN**

Ảnh có chứa biểu tượng, Đồ họa, Phông chữ, thiết kế đồ họa

Mô tả được tạo tự động

**BÀI TẬP LỚN**

**Đề tài: Student Management Application**

**Môn học: Lập trình hướng đối tượng**

|  |  |
| --- | --- |
| **Lê Đức Anh** | **B22DCDT015** |
| **Nguyễn Đức Anh** | **B22DCDT017** |
| **Trần Minh Tùng** | **B22DCVT501** |
| **Nguyễn Minh Vũ** | **B22DCVT594** |
| **Tạ Cao Sơn** | **B22DCVT445** |
|  |  |

**Hà Nội - 2024**

**Lời cảm ơn**

Nhóm chúng em xin chân thành cảm ơn cô Đỗ Thị Liên đã dành thời gian và tâm huyết hỗ trợ chúng em trong quá trình học tập và nghiên cứu môn lập trình hướng đối tượng. Sự hướng dẫn tận tình của cô đã giúp chúng em tích lũy được nhiều kiến thức quý giá, là nền tảng vững chắc cho việc hoàn thành bài tiểu luận với đề tài: "Student Management Application."

Chúng em ý thức rằng trong quá trình thực hiện, không thể tránh khỏi những thiếu sót. Vì vậy, chúng em rất mong nhận được những ý kiến đóng góp từ cô để có thể hoàn thiện bài làm của mình.

I, Xác định yêu cầu

. Quản lý sinh viên

• Thêm mới sinh viên vào hệ thống.

• Cập nhật thông tin sinh viên (tên, ngày sinh, email, số điện thoại).

• Xóa sinh viên khỏi hệ thống.

• Tìm kiếm sinh viên theo các tiêu chí:

• Tên, ID, email, hoặc số điện thoại.

2. Quản lý lớp học

• Tạo mới lớp học (tên lớp, thời gian bắt đầu/kết thúc, giáo viên phụ trách).

• Cập nhật thông tin lớp học (đổi tên lớp, thời gian học…).

• Xóa lớp học.

• Gán sinh viên vào lớp học.

• Hiển thị danh sách lớp học với thông tin chi tiết:

3. Quản lý giáo viên

• Thêm mới giáo viên vào hệ thống.

• Cập nhật thông tin giáo viên (tên, email, số điện thoại…).

• Xóa giáo viên khỏi hệ thống.

• Gán giáo viên phụ trách lớp học.

• Hiển thị danh sách giáo viên và thông tin chi tiết.

II, Phân tích thiết kế ra lớp

1. Các đối tượng (Class) chính:

a. Account

• Thuộc tính:

• password: Mật khẩu đăng nhập (string).

• userID: ID người dùng (string).

• Phương thức:

• login(): Đăng nhập (trả về kiểu bool).

• signUp(): Đăng ký tài khoản mới (void).

b. Person

• Đây là một lớp cha (superclass) chung cho các đối tượng người dùng như Admin, Teacher, và Student.

• Thuộc tính:

• name: Tên (string).

• role: Vai trò (string).

• phone: Số điện thoại (string).

• email: Email (string).

• id: ID người dùng (string).

• birthDate: Ngày sinh (date).

• Phương thức:

• viewPersonalInfo(): Xem thông tin cá nhân (void).

• updatePersonalInfo(): Cập nhật thông tin cá nhân (void).

c. Student (Kế thừa từ Person)

• Thuộc tính:

• enrollmentDate: Ngày nhập học (date).

• cpa: Điểm trung bình tích lũy (float).

• Phương thức:

• viewPersonalInfo(): Xem thông tin cá nhân của sinh viên (override từ Person).

d. Teacher (Kế thừa từ Person)

• Phương thức:

• CRUD Class: Thêm, sửa, xóa hoặc xem thông tin lớp học mà giáo viên phụ trách.

e. Class

• Thuộc tính:

• className: Tên lớp học (string).

• beginTime: Thời gian bắt đầu lớp học (date).

• endTime: Thời gian kết thúc lớp học (date).

• Phương thức:

• viewStudentList(): Xem danh sách sinh viên trong lớp.

• CRUD Student: Quản lý sinh viên trong lớp.

f. Admin

• Phương thức:

• viewTeacherInfo(): Xem thông tin giáo viên.

• updateTeacher(): Cập nhật thông tin giáo viên.

• deleteTeacher(): Xóa giáo viên.

• CRUD Teacher: Thêm, sửa, xóa hoặc xem thông tin giáo viên.

2. Các mối quan hệ chính:

1. Account → Person:

• Quan hệ 1-1: Một tài khoản chỉ thuộc về một cá nhân (Person).

2. Person → Student/Teacher/Admin:

• Quan hệ kế thừa: Các lớp Student, Teacher, và Admin kế thừa từ lớp cha Person.

3. Class → Student:

• Quan hệ 1-N: Một lớp có thể có nhiều sinh viên, nhưng mỗi sinh viên chỉ thuộc một lớp.

4. Class → Teacher:

• Quan hệ 1-1: Mỗi lớp học được quản lý bởi một giáo viên.

5. Admin → Teacher:

• Quan hệ 1-N: Một admin có thể quản lý nhiều giáo viên.

III, Biểu đồ lớp (UML)

Ảnh có chứa văn bản, biểu đồ, Song song, hàng

Mô tả được tạo tự động

IV,Xem xét hiện thực các tính chất OOP và project

1. Tính đóng gói (Encapsulation)

Áp dụng:

• Khai báo thuộc tính private/protected trong các lớp như Account, Person, Student, Class,… để bảo vệ dữ liệu.

• Sử dụng các getter và setter để kiểm soát truy cập vào dữ liệu.

2. Tính kế thừa (Inheritance)

Áp dụng:

• Lớp Person là lớp cha của Student, Teacher, và Admin.

• Các lớp con kế thừa và có thể ghi đè (override) các phương thức từ lớp cha khi cần thiết.

3. Tính đa hình (Polymorphism)

Áp dụng:

• Ghi đè (Override):

• viewPersonalInfo() được ghi đè trong lớp Student và Teacher.

• Nạp chồng (Overloading):

• Phương thức crudStudent() trong lớp Class có thể được nạp chồng với các tham số khác nhau để hỗ trợ các thao tác khác nhau.

4. Tính trừu tượng (Abstraction)

Áp dụng:

• Lớp Person có thể được định nghĩa là lớp abstract để các lớp con Student, Teacher, và Admin triển khai chi tiết.

• Sử dụng interface để đảm bảo tất cả các đối tượng hỗ trợ thao tác CRUD (thêm, sửa, xóa).