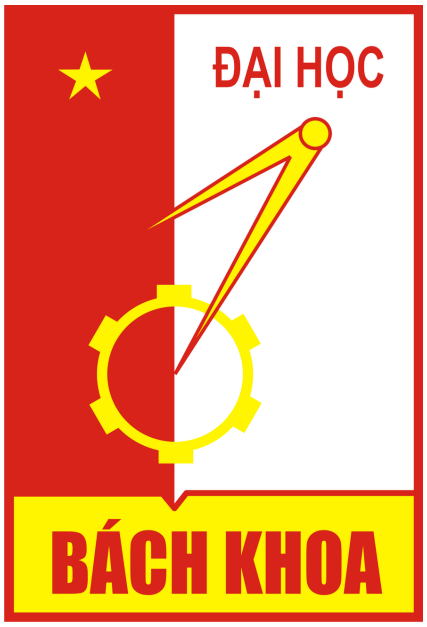
ĐẠI HỌC BÁCH KHOA HÀ NỘI
KHOA TOÁN-TIN

----- 📖 -----



BÁO CÁO BÀI TẬP LỚN
BỘ MÔN: KỸ THUẬT LẬP TRÌNH
ĐỀ TÀI: HỆ THỐNG QUẢN LÝ ĐIỂM DANH

Giảng viên hướng dẫn: **Vũ Thành Nam**

Sinh viên thực hiện:
Họ và tên: Vũ Xuân Thái
MSSV: 20237387
Lớp: 158241

Hà Nội, 6/2025

MỤC LỤC

MỞ ĐẦU

Trong bối cảnh giáo dục hiện đại, việc quản lý thông tin sinh viên và theo dõi chuyên cần là một yếu tố quan trọng góp phần nâng cao chất lượng đào tạo và quản lý lớp học hiệu quả. Các phương pháp điểm danh thủ công truyền thống thường tốn thời gian, dễ xảy ra sai sót và khó khăn trong việc tổng hợp, thống kê. Nhằm giải quyết những vấn đề này, đề tài "Hệ thống quản lý điểm danh lớp học" được thực hiện với mục tiêu phát triển một hệ thống phần mềm hỗ trợ công tác điểm danh một cách tự động, chính xác và tiện lợi.

Ứng dụng được xây dựng bằng ngôn ngữ lập trình C++, tập trung vào việc cung cấp các chức năng cơ bản nhưng thiết yếu cho việc quản lý lớp học, quản lý danh sách sinh viên, thực hiện điểm danh theo ngày, và tổng hợp thống kê chuyên cần. Hệ thống được thiết kế theo kiến trúc module hóa, đảm bảo tính dễ bảo trì, dễ mở rộng và khả năng tái sử dụng cao.

Báo cáo này sẽ trình bày chi tiết về quá trình phân tích, thiết kế, triển khai và các kết quả đạt được của dự án, cùng với những đánh giá và hướng phát triển trong tương lai.

Thông qua đề tài này, em có cơ hội trau dồi kiến thức đã học về lập trình, cấu trúc dữ liệu và giải thuật vào một sản phẩm thực tiễn. Đồng thời, đây cũng là dịp để rèn luyện kỹ năng thiết kế phần mềm, xử lý dữ liệu và tư duy giải quyết vấn đề một cách hệ thống.

Chương I: Giới thiệu

1. Mô tả bài toán

Dự án xây dựng một hệ thống quản lý điểm danh cho sinh viên các lớp học. Hệ thống cho phép người dùng (giảng viên hoặc quản trị viên) thực hiện các chức năng chính bao gồm:

- Tạo và quản lý danh sách các lớp học.
- Thêm sinh viên vào từng lớp học.
- Thực hiện điểm danh cho các buổi học, ghi nhận trạng thái có mặt/vắng mặt của sinh viên.
- Cho phép sửa đổi thông tin điểm danh nếu có sai sót.
- Xem lại lịch sử điểm danh của một lớp.
- Thống kê tình hình điểm danh của lớp (ví dụ: tỷ lệ chuyên cần, số buổi vắng của từng sinh viên).
- Lưu trữ và tải dữ liệu điểm danh từ tệp để đảm bảo tính bền vững của dữ liệu.

2. Phân tích yêu cầu

2.1 Yêu cầu chức năng:

a) Quản lý Lớp học

- Hệ thống cho phép người dùng (ví dụ: giáo vụ, giảng viên) thêm một lớp học mới.
  - Đầu vào: Mã lớp (duy nhất), tên lớp.
  - Đầu ra: Thông báo thêm lớp thành công hoặc lỗi (nếu mã lớp đã tồn tại).
  - Mô tả: Mỗi lớp học có một mã định danh duy nhất và một tên mô tả.
- Hệ thống cho phép người dùng xem danh sách các lớp học hiện có.
  - Đầu vào: Yêu cầu xem danh sách lớp.
  - Đầu ra: Hiển thị danh sách mã lớp và tên lớp.

b) Quản lý Sinh viên

- Hệ thống cho phép người dùng thêm một sinh viên mới vào một lớp học cụ thể.
  - Đầu vào: Mã lớp, mã sinh viên (duy nhất trong lớp đó), họ tên sinh viên, khoa.
  - Đầu ra: Thông báo thêm sinh viên thành công hoặc lỗi (ví dụ: lớp không tồn tại, mã sinh viên đã tồn tại trong lớp).
- Hệ thống cho phép người dùng xem danh sách sinh viên của một lớp học.
  - Đầu vào: Mã lớp.
  - Đầu ra: Hiển thị danh sách mã sinh viên, họ tên, khoa của tất cả sinh viên trong lớp.

c) Quản lý Điểm danh

- Hệ thống cho phép người dùng thực hiện điểm danh cho một lớp học vào một ngày cụ thể.
  - Đầu vào: Mã lớp, ngày điểm danh.
  - Mô tả: Hệ thống sẽ hiển thị danh sách sinh viên của lớp. Với mỗi sinh viên, người dùng nhập trạng thái điểm danh (ví dụ: có mặt, vắng mặt, có phép). Ngày điểm danh phải hợp lệ.
  - Đầu ra: Thông báo điểm danh thành công hoặc lỗi (ví dụ: lớp không tồnaitại, ngày không hợp lệ, không có sinh viên trong lớp).
- Hệ thống cho phép người dùng sửa đổi trạng thái điểm danh của một sinh viên cụ thể trong một buổi điểm danh đã được ghi nhận.
  - Đầu vào: Mã lớp, ngày điểm danh, mã sinh viên, trạng thái điểm danh mới.
  - Đầu ra: Thông báo sửa thành công hoặc lỗi (ví dụ: không tìm thấy buổi điểm danh, không tìm thấy sinh viên).
- Hệ thống cho phép người dùng xem lại chi tiết một buổi điểm danh của một lớp.
  - Đầu vào: Mã lớp, ngày điểm danh.
  - Đầu ra: Hiển thị danh sách sinh viên của lớp cùng với trạng thái điểm danh của họ vào ngày đó.
- Hệ thống cho phép người dùng xem lại chi tiết điểm danh của một sinh viên cụ thể của một lớp
  - Đầu vào: Mã lớp, mã sinh viên.
  - Đầu ra: Hiển thị tình trạng điểm danh của sinh viên từng ngày và đưa ra tổng số buổi có mặt, vắng mắt và tỉ lệ điểm danh.
- Hệ thống cho phép người dùng xem lịch sử tất cả các buổi điểm danh của một lớp học.
  - Đầu vào: Mã lớp.
  - Đầu ra: Hiển thị danh sách các ngày đã điểm danh và có thể là tóm tắt số lượng sinh viên có mặt/vắng mỗi ngày.

d) Thống kê và Báo cáo

- Hệ thống cho phép người dùng xem thống kê chuyên cần cho một lớp học cụ thể.
  - Đầu vào: Mã lớp.
  - Đầu ra: Báo cáo thống kê bao gồm: Tổng số sinh viên trong lớp, tổng số buổi đã thực hiện điểm danh, tỷ lệ điểm danh trung bình của cả lớp, số lần vắng của từng sinh viên và tổng kết chi tiết trạng thái các buổi học của từng sinh viên (ví dụ: số buổi có mặt, số buổi vắng).

e) Quản lý Dữ liệu

- Hệ thống phải có khả năng lưu trữ toàn bộ dữ liệu (thông tin lớp, danh sách sinh viên, lịch sử điểm danh) một cách bền vững để có thể sử dụng lại sau khi chương trình tắt.
  - Mô tả: Dữ liệu được lưu vào các tệp trên ổ cứng.
- Hệ thống phải có khả năng tải (đọc) dữ liệu đã lưu khi khởi động hoặc theo yêu cầu của người dùng.
  - Mô tả: Đảm bảo tính toàn vẹn của dữ liệu khi đọc từ tệp.

2.2. YÊU CẦU PHI CHỨC NĂNG

a) Tính dễ sử dụng

- Giao diện người dùng (dòng lệnh) phải rõ ràng, dễ hiểu và dễ thao tác.
  - Mô tả: Các menu và thông báo phải sử dụng ngôn ngữ tự nhiên (Tiếng Việt), hướng dẫn người dùng thực hiện các thao tác một cách trực quan.
- Hệ thống cần cung cấp phản hồi rõ ràng cho mỗi hành động của người dùng (ví dụ: thông báo thành công, cảnh báo lỗi, hướng dẫn nhập liệu).
- Quá trình nhập liệu cần được hỗ trợ xác thực cơ bản để giảm thiểu lỗi (ví dụ: kiểm tra định dạng ngày, kiểm tra kiểu dữ liệu số).

b) Độ tin cậy

- Hệ thống phải hoạt động ổn định và không bị treo hoặc lỗi đột ngột trong quá trình sử dụng thông thường.
- Dữ liệu điểm danh và thông tin sinh viên phải được lưu trữ một cách chính xác và toàn vẹn.
- Hệ thống cần có cơ chế xử lý lỗi cơ bản (ví dụ: nhập liệu sai định dạng, không tìm thấy tệp dữ liệu) và thông báo cho người dùng một cách thân thiện thay vì dừng đột ngột. (Đã thể hiện qua try-catch ở main).

c) Hiệu năng

- Thêm lớp học mới: Thời gian thêm lớp học vào bộ nhớ có độ phức trung bình O(1), tệ nhất O(L) với L là số lượng lớp học trong hệ thống
- Thêm sinh viên mới vào lớp: Để thêm sinh viên mới vào lớp ta cần phải kiểm tra sự tồn tại của mã sinh viên trong danh sách sinh viên có độ phức tạp trung bình là O(1), tệ nhất là O(S) với S là số sinh viên hiện có trong lớp và sau đó thêm sinh viên mới vào với độ phức tạp trung bình là O(1), tệ nhất là O(S)..
- Thực hiện điểm danh mới: Lặp qua toàn bộ danh sách sinh viên của lớp để lấy mã sinh viên và thiết lập trạng thái ban đầu nên độ phức tạp trung bình là O(S), tệ nhất là $O(S^2)$ khi bị xung đột key với S là số sinh viên trong lớp
- Tìm một buổi điểm danh cụ thể theo ngày: Độ phức tạp trung bình là O(D) khi phải duyệt và so sánh với D là số lượng buổi điểm danh

- Thao tác Thống kê: Duyệt qua các lớp trong danh sách lớp, ghi thông tin cơ bản của từng lớp, duyệt qua danh sách sinh viên để lấy thông tin sinh viên, duyệt qua danh sách điểm danh nên độ phức tạp trung bình là O(L * (S + D * S)), với L là số lượng lớp trong danh sách, S là số lượng sinh viên trong lớp, D là số lượng buổi điểm danh. Đây là thao tác tốn kém nhất nếu dữ liệu lớn, nhưng thường chỉ thực hiện một lần khi kết thúc hoặc theo yêu cầu.

d) Khả năng bảo trì

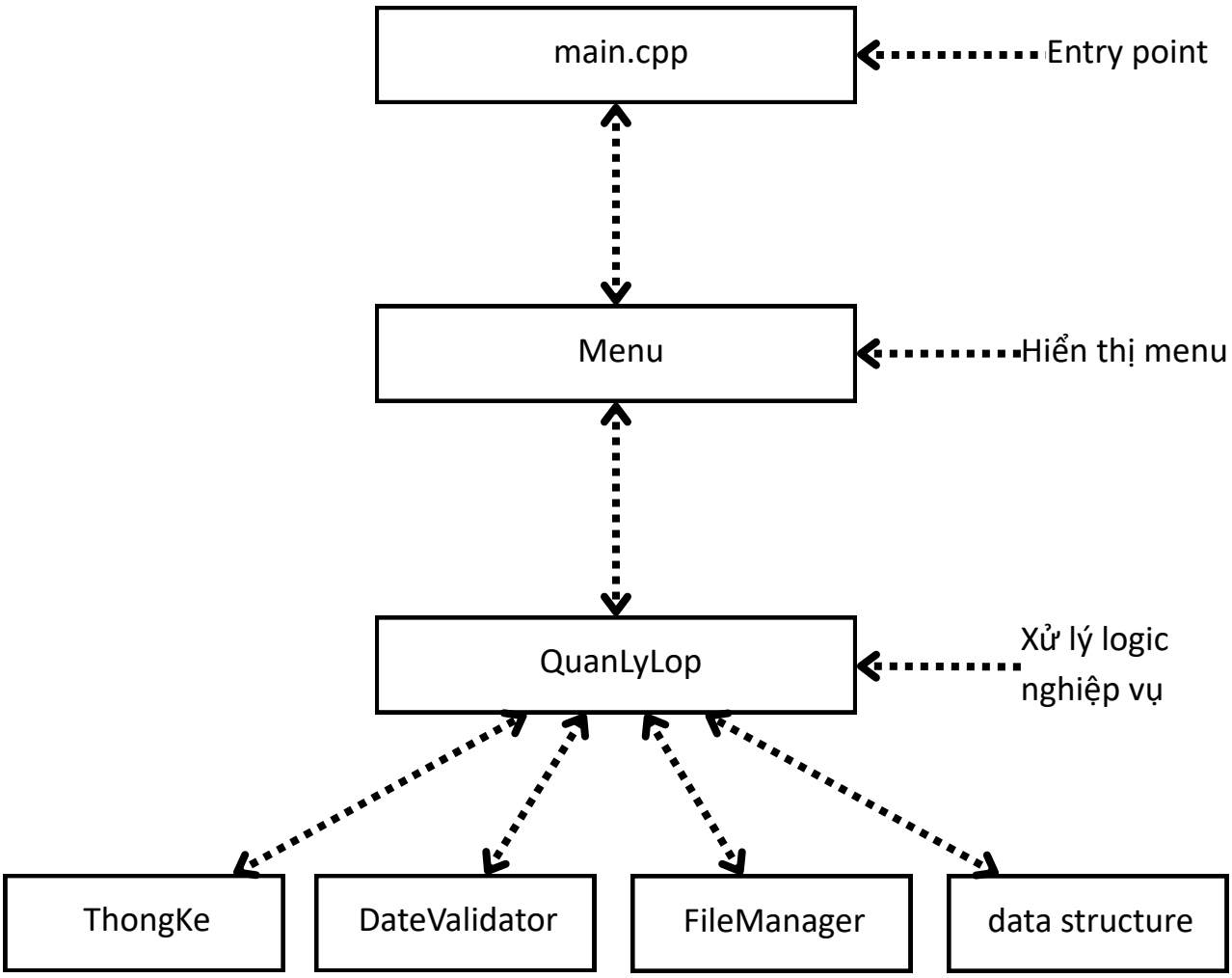
- Mã nguồn phải được tổ chức rõ ràng, theo module (đã thể hiện qua cấu trúc thư mục và các lớp riêng biệt).
- Mã nguồn cần có chú thích (comment) hợp lý ở những đoạn code phức tạp hoặc quan trọng để người khác (hoặc chính người phát triển sau này) dễ hiểu và bảo trì.
- Thiết kế phải cho phép dễ dàng mở rộng thêm các chức năng mới trong tương lai (ví dụ: thêm các loại báo cáo mới, tích hợp các phương thức điểm danh khác).

3. Cấu trúc hệ thống:

Hệ thống được thiết kế theo kiến trúc module hóa, chia thành các thành phần chính với vai trò rõ ràng:

- Giao diện người dùng (UI): Chịu trách nhiệm tương tác với người dùng thông qua một menu dòng lệnh, tiếp nhận yêu cầu và hiển thị kết quả.
- Lớp Dịch vụ (Service Layer): Xử lý logic nghiệp vụ chính của ứng dụng, bao gồm quản lý lớp học, quản lý sinh viên, và các nghiệp vụ điểm danh.
- Lớp Điều khiển (Utils): Quản lý các tương tác với hệ thống tệp, ví dụ như đọc và ghi dữ liệu điểm danh.Cung cấp các tiện ích hỗ trợ, ví dụ như kiểm tra tính hợp lệ của dữ liệu (ngày tháng).
- Mô-đun Dữ liệu (Data Modules): Định nghĩa các cấu trúc dữ liệu cốt lõi được sử dụng trong toàn hệ thống.
- Điểm vào (Main Entry Point): Khởi tạo và điều phối hoạt động của các thành phần.
- Thư mục dữ liệu (DanhSachLop): Nơi lưu trữ các tệp dữ liệu của từng lớp học.

Sơ đồ tổng quan



Luồng hoạt động chính người dùng bắt đầu từ main.cpp, khởi tạo và hiển thị Menu (UI). Người dùng tương tác với menu, các yêu cầu sẽ được chuyển đến các lớp Services (ví dụ: QuanLyLop) để xử lý logic nghiệp vụ. Các lớp Services sử dụng các Modules (cấu trúc dữ liệu) và có thể gọi đến Utils (ví dụ: FileManager để đọc/ghi hoặc DateValidator để kiểm tra ngày).

Chương II: Thiết kế chương trình

1. Chức năng tổng thể

Hệ thống cung cấp một giao diện dòng lệnh (thông qua lớp Menu) cho phép người dùng thực hiện các tác vụ sau:

1. Quản lý Lớp học:
- Thêm mới lớp học với thông tin mã lớp và tên lớp.
- Hiển thị danh sách các lớp hiện có.

2. Quản lý Sinh viên:
- Thêm sinh viên vào một lớp học cụ thể với các thông tin: mã sinh viên, họ và tên, tên ngành của sinh viên.
- Hiển thị danh sách sinh viên của một lớp.

3. Quản lý Điểm danh:
- Thực hiện điểm danh cho một lớp vào một ngày cụ thể.
- Cho phép chỉnh sửa thông tin điểm danh đã ghi nhận.
- Xem chi tiết điểm danh của một lớp vào một ngày cụ thể.
- Xem điểm danh của một sinh viên trong lớp.
- Lưu trữ lịch sử các buổi điểm danh.

4. Tra cứu và Thống kê:
- Xem lịch sử điểm danh của toàn bộ sinh viên trong một lớp.
- Xem thống kê điểm danh của một lớp (tổng số sinh viên, tổng buổi điểm danh, tỷ lệ điểm danh trung bình, số lần vắng của từng sinh viên).

5. Quản lý Dữ liệu:
- Đọc dữ liệu điểm danh của một lớp từ tệp (khi khởi tạo hoặc theo yêu cầu).
- Lưu dữ liệu điểm danh hiện tại của một lớp vào tệp (ngầm định sau một số thao tác hoặc theo yêu cầu).

2. Tổ chức chương trình

Hoạt động chính: Chương trình bắt đầu từ hàm main(), nơi đối tượng Menu được tạo và phương thức menu() của nó được gọi. Lớp Menu hiển thị các lựa chọn cho người dùng và dựa trên lựa chọn đó, gọi các phương thức tương ứng của đối tượng QuanLyLop (mà nó sở hữu).

- Các lớp chính và vai trò:
  - Menu: Lớp giao diện người dùng, điều hướng các chức năng.
  - QuanLyLop: Lớp dịch vụ trung tâm, xử lý tất cả logic nghiệp vụ liên quan đến lớp học, sinh viên, và điểm danh. Nó sử dụng các đối tượng Lop, Sinhvien, DiemDanh.
  - Lop: Đại diện cho một lớp học, chứa danh sách sinh viên và lịch sử điểm danh của lớp đó.
  - Sinhvien: Đại diện cho một sinh viên.
  - DiemDanh: Đại diện cho một bản ghi điểm danh của một lớp vào một ngày cụ thể.
  - ThongKe: Cung cấp các phương thức tĩnh để tính toán và trả về dữ liệu thống kê dựa trên thông tin của một Lop.
  - DateValidator: Cung cấp các tiện ích tĩnh để kiểm tra tính hợp lệ của ngày và lấy ngày hiện tại.
  - FileManager: Quản lý việc đọc/ghi dữ liệu lớp học từ/vào các tệp trong hệ thống, xác định đường dẫn và tạo thư mục nếu cần.
- Sự tương tác giữa các lớp:
  - main.cpp khởi tạo Menu.
  - Menu gọi các phương thức của QuanLyLop.
  - QuanLyLop tạo và quản lý các đối tượng Lop, Sinhvien, DiemDanh.
  - QuanLyLop sử dụng DateValidator để xác thực ngày tháng khi điểm danh.
  - QuanLyLop sử dụng FileManager để lưu và tải dữ liệu.
  - QuanLyLop sử dụng ThongKe để lấy dữ liệu thống kê cho Menu hiển thị.

3. Kiểu dữ liệu

3.1. Cấu trúc dữ liệu:

- struct Sinhvien:
  - int maSV: Mã số sinh viên (khóa chính).
  - string hoTen: Họ và tên sinh viên.
  - string khoa: ngành của sinh viên.
- struct DiemDanh:
  - string ngay: Ngày thực hiện điểm danh (ví dụ: "dd/mm/yyyy").
  - unordered_map<int, int> trangThai: Lưu trạng thái điểm danh của từng sinh viên trong buổi đó với key là maSV (mã sinh viên) và value là dữ liệu kiểu int (đại diện cho trạng thái, ví dụ: 0 - vắng, 1 - có mặt).

- struct Lop:
  - int maLop: Mã lớp học (khóa chính).
  - string tenLop: Tên lớp học.
  - unordered_map<int, Sinhvien> danhSachSV: Danh sách sinh viên thuộc lớp với key là maSV và value là đối tượng SinhVien.
  - vector<DiemDanh> danhSachDiemDanh: Lịch sử các buổi điểm danh của lớp.
- Trong QuanLyLop:
  - unordered_map<int, Lop> danhSachLop: Cấu trúc dữ liệu trung tâm, quản lý tất cả các lớp học với key là maLop và value là đối tượng Lop.

3.2. Lưu trữ dữ liệu

- Dữ liệu của chương trình (thông tin lớp, sinh viên, lịch sử điểm danh) được lưu trữ bền vững vào tệp để có thể sử dụng lại giữa các phiên làm việc.
- Lớp FileManager chịu trách nhiệm quản lý việc này.
- Mỗi lớp học sẽ có một tệp dữ liệu riêng, tên tệp có thể dựa trên maLop (ví dụ: DATA_DIR/101.dat, DATA_DIR/102.dat). DATA_DIR có thể là một thư mục con như "data/" hoặc "DanhSachLop/".
- Định dạng tệp có thể là văn bản để dễ đọc và gỡ lỗi, hoặc tệp nhị phân (binary file) để tăng tốc độ đọc/ghi và giảm kích thước tệp, tùy thuộc vào lựa chọn triển khai trong FileManager.cpp.
- Nếu là text file: có thể sử dụng các định dạng như CSV, JSON (cần thư viện ngoài), hoặc một định dạng tự định nghĩa phân tách bằng ký tự đặc biệt.
- Việc lưu trữ bao gồm thông tin của lớp, danh sách sinh viên của lớp đó, và toàn bộ lịch sử điểm danh (vector<DiemDanh>).
- Hàm saveDiemDanh() trong QuanLyLop sẽ gọi các hàm trong FileManager để ghi dữ liệu hiện tại của các lớp ra tệp.
- Hàm docFile(int maLop) trong QuanLyLop sẽ gọi các hàm trong FileManager để đọc dữ liệu của một lớp cụ thể từ tệp vào bộ nhớ.

4. Thiết kế module

- **Module Khởi chạy (main.cpp)**
  - Vai trò: Là điểm vào (entry point) của ứng dụng.
  - Thiết kế: Chứa hàm main tối giản, có nhiệm vụ duy nhất là khởi tạo đối tượng Menu và gọi phương thức menu() để bắt đầu vòng lặp chính của chương trình. Toàn bộ logic được bọc trong một khối try-catch để bắt các ngoại lệ không mong muốn, giúp chương trình kết thúc một cách an toàn thay vì bị dừng đột ngột (crash).

- **Module Giao diện Người dùng (ui/menu)**
  - Vai trò: Chịu trách nhiệm hoàn toàn về việc tương tác với người dùng qua giao diện dòng lệnh (CLI).
  - Thiết kế: Lớp này chứa một vòng lặp hiển thị các chức năng chính. Nó nhận đầu vào từ người dùng, sau đó gọi các phương thức dịch vụ công khai (public) từ một đối tượng QuanLyLop mà nó sở hữu. Lớp Menu không chứa logic nghiệp vụ, mà chỉ đóng vai trò điều phối viên. Các hàm xử lý (handler) bên trong Menu được chia nhỏ cho từng tác vụ (ví dụ: handleThemLop, handleDiemDanh), giúp mã nguồn rõ ràng và dễ quản lý.
- **Module Dịch vụ (services/)**
- **class QuanLyLop**:
  - Vai trò: Là "bộ não" của ứng dụng, chứa đựng toàn bộ logic nghiệp vụ cốt lõi.
  - Thiết kế:
  - Trạng thái nội bộ (private): Quản lý một unordered_map<int, Lop> tên là danhSachLop, cho phép truy cập, thêm, xóa một lớp học theo mã lớp với hiệu năng cao (độ phức tạp O(1) trung bình).
  - Logic nghiệp vụ chi tiết: Các phương thức trong lớp này được triển khai chi tiết. Ví dụ, hàm diemDanh(ngay, maLop) không chỉ điểm danh mà còn kiểm tra xem ngày đó đã được điểm danh chưa. Nếu đã tồn tại, nó sẽ hỏi người dùng có muốn điểm danh lại hay không, thể hiện sự tương tác chi tiết và logic phức tạp. Nó sử dụng các hàm riêng tư (private) là diemDanhMoi và diemDanhLai để thực hiện các luồng logic khác nhau, tuân thủ nguyên tắc chia để trị.
  - Tương tác Console: Lớp này trực tiếp sử dụng cout và cin để thực hiện các chức năng xuất và nhập với người dùng (ví dụ: hỏi trạng thái từng sinh viên khi điểm danh).
  - Định dạng đầu ra: Sử dụng thư viện <iomanip> để tạo bảng dữ liệu (ví dụ trong xuatDiemDanh, xemThongKe), tạo ra giao diện gọn gàng, dễ hiểu.
- **class ThongKe**:
  - Vai trò: Cung cấp chức năng tính toán thống kê.
  - Thiết kế: QuanLyLop không tự tính toán thống kê mà ủy thác nhiệm vụ này cho lớp ThongKe bằng cách gọi phương thức tĩnh (static) tinhThongKe(lop). Điều này giúp tách biệt logic thống kê khỏi logic quản lý chung, làm cho cả hai lớp trở nên gọn gàng và tập trung hơn và dễ phát triển tính năng thống kê cho sau này.

- **Module Tiện ích (utils/)**
- **Lớp FileManager:**
  - Vai trò: Trừu tượng hóa các hoạt động liên quan đến hệ thống tệp.
  - Thiết kế: Cung cấp các phương thức tĩnh (static) để lấy đường dẫn tệp, kiểm tra sự tồn tại của tệp, và tạo thư mục. Các hàm saveDiemDanh() và docFile() sử dụng lớp này. Logic đọc/ghi tệp được viết trong này, nhưng về mặt thiết kế, nó tuân thủ định dạng và đường dẫn do FileManager quy định.
  - Định dạng lưu trữ: Dữ liệu được lưu dưới dạng tệp văn bản (text file) có cấu trúc rõ ràng cho mỗi lớp:
  - Dòng đầu tiên: MaLop - TenLop.
  - Các dòng tiếp theo: MaSV|HoTen|Khoa cho mỗi sinh viên.
  - Dấu phân cách: ---DIEMDANH---.
  - Các khối điểm danh, mỗi khối bắt đầu bằng dòng Ngay, theo sau là các dòng MaSV-TrangThai, và kết thúc bằng --ENDNGAY--.
- **Lớp DateValidator**:
  - Vai trò: Cung cấp các tiện ích xác thực và xử lý ngày tháng.
  - Thiết kế: Lớp này cung cấp các phương thức tĩnh như isValidDate. Trước khi thực hiện bất kỳ thao tác nào liên quan đến ngày (điểm danh, sửa điểm danh, xem điểm danh), QuanLyLop đều gọi hàm này để đảm bảo dữ liệu đầu vào là hợp lệ, ngăn ngừa lỗi và đảm bảo tính toàn vẹn dữ liệu.
- **Module Thực thể Dữ liệu (Entities - modules/)**
- **Tệp defaults.h**:
  - Vai trò: Là nơi định nghĩa các khuôn mẫu dữ liệu của toàn hệ thống.
  - Thiết kế: Chứa các cấu trúc Lop, Sinhvien, và DiemDanh. Các struct này được thiết kế để làm các đối tượng chứa dữ liệu thuần túy . Chúng sử dụng các container của thư viện C++:
    - Lop chứa unordered_map danhSachSV và vector danhSachDiemDanh.
    - DiemDanh chứa unordered_map trangThai.
    - Việc sử dụng các cấu trúc dữ liệu hiệu năng cao này là nền tảng cho hiệu suất tốt của các thao tác tra cứu và cập nhật trong ứng dụng.

Chương III: Kết quả thực nghiệm

1. Khởi động chương trình và hiển thị Menu chính

```
========================================
MENU CHÍNH
========================================
1. Thêm lớp học
2. Thêm sinh viên vào lớp
3. Điểm danh
4. Sửa điểm danh
5. Xuất bảng điểm danh theo ngày
6. Xuất bảng điểm danh của sinh viên
7. Xem lịch sử điểm danh của lớp
8. Xem thống kê điểm danh của lớp
9. Đọc dữ liệu lớp từ file
0. Thoát chương trình
========================================
Nhập lựa chọn (0-9):
```

2. Thêm lớp học mới

2.1. Thêm lớp học mới thành công

```
--- THÊM LỚP HỌC MỚI ---
Nhập mã lớp (số nguyên): 158241
Nhập tên lớp: Ky thuat lap trinh
Thêm lớp Ky thuat lap trinh (mã: 158241) thành công.
Đã lưu thành công dữ liệu cho 1/1 lớp.

Nhấn Enter để tiếp tục...
```

2.2. Thêm lớp học thất bại do đã tồn tại

```
--- THÊM LỚP HỌC MỚI ---
Nhập mã lớp (số nguyên): 158241
Nhập tên lớp: Ky nang mem
Lớp với mã 158241 đã tồn tại.

Nhấn Enter để tiếp tục...
```

2.3. Thêm lớp thất bại do đầu vào sai

```
--- THÊM LỚP HỌC MỚI ---
Nhập mã lớp (số nguyên): T150
Vui lòng nhập một số nguyên hợp lệ: T150
Vui lòng nhập một số nguyên hợp lệ:
```

3. Thêm sinh viên vào lớp

3.1. Thêm sinh viên vào lớp thành công

```
 --- THÊM SINH VIÊN VÀO LỚP ---
Nhập mã lớp cần thêm sinh viên: 158241
Nhập số lượng sinh viên cần thêm: 1

--- Nhập thông tin cho sinh viên thứ 1 ---
Mã sinh viên (số nguyên): 20237387
Họ và tên: Vu Xuan Thai
Khoa: Toan tin

Hoàn tất thêm sinh viên. Đã thêm thành công 1/1 sinh viên vào lớp 158241.
Đã lưu thành công dữ liệu cho 1/1 lớp.

Nhấn Enter để tiếp tục...
```

3.2. Thêm sinh viên vào lớp thất bại do đã tồn tại

```
 --- THÊM SINH VIÊN VÀO LỚP ---
Nhập mã lớp cần thêm sinh viên: 158241
Nhập số lượng sinh viên cần thêm: 1

--- Nhập thông tin cho sinh viên thứ 1 ---
Mã sinh viên (số nguyên): 20237387
Họ và tên: Vu Xuan Thai
Khoa: Toan tin
Sinh viên với mã 20237387 đã tồn tại trong lớp 158241!

Hoàn tất thêm sinh viên. Đã thêm thành công 0/1 sinh viên vào lớp 158241.

Nhấn Enter để tiếp tục...
```

4. Thực hiện điểm danh

4.1. Điểm danh thành công

```
 --- THỰC HIỆN ĐIỂM DANH ---
Ngày hiện tại theo hệ thống: 06/06/2025
Nhập ngày cần điểm danh (định dạng dd/mm/yyyy, nhấn Enter để dùng ngày hiện tại):
Sử dụng ngày hiện tại: 06/06/2025
Nhập mã lớp cần điểm danh: 158241

ĐIỂM DANH LỚP Ky thuat lap trinh - NGÀY 06/06/2025
================================================
1. Sam Quoc Su (MSV: 20237386) - Trạng thái (1=Có mặt/0=Vắng): 1
2. Vu Xuan Thai (MSV: 20237387) - Trạng thái (1=Có mặt/0=Vắng): 1
3. Pham Thi Quynh (MSV: 20237478) - Trạng thái (1=Có mặt/0=Vắng): 1
4. Le Minh Quan (MSV: 20237382) - Trạng thái (1=Có mặt/0=Vắng): 1
5. Tran Thanh Phu (MSV: 20237473) - Trạng thái (1=Có mặt/0=Vắng): 1
6. Nguyen Doanh Thai (MSV: 20237483) - Trạng thái (1=Có mặt/0=Vắng): 1
7. Vuong Van Phong (MSV: 20237379) - Trạng thái (1=Có mặt/0=Vắng): 1
8. Nguyen Thanh Son (MSV: 20216880) - Trạng thái (1=Có mặt/0=Vắng): 1
9. Trinh Dinh Phong (MSV: 20237472) - Trạng thái (1=Có mặt/0=Vắng): 1
10. Pham Nam Phong (MSV: 20237378) - Trạng thái (1=Có mặt/0=Vắng): 1
Điểm danh thành công!
Đã lưu thành công dữ liệu cho 1/1 lớp.

Nhấn Enter để tiếp tục...
```

4.1. Điểm danh bất bại do không tìm thấy sinh viên

```
 --- THỰC HIỆN ĐIỂM DANH ---
Ngày hiện tại theo hệ thống: 06/06/2025
Nhập ngày cần điểm danh (định dạng dd/mm/yyyy, nhấn Enter để dùng ngày hiện tại):
Sử dụng ngày hiện tại: 06/06/2025
Nhập mã lớp cần điểm danh: 150
Lớp 150 chưa có sinh viên nào!

Nhấn Enter để tiếp tục...
```

5. Tìm kiếm điểm danh theo ngày

5.1. Nếu buổi điểm danh đó tồn tại

```
 --- XUẤT BẢNG ĐIỂM DANH THEO NGÀY ---
Nhập ngày cần xuất bảng điểm danh (dd/mm/yyyy): 07/06/2025
Nhập mã lớp: 158241

BẢNG ĐIỂM DANH
Lớp: Ky thuat lap trinh (Mã: 158241)
Ngày: 07/06/2025
================================================================================
Ma SV     Ho va ten                 Khoa               Trang thai
--------------------------------------------------------------------------------
20237378  Pham Nam Phong            Toan tin           Co mat
20237472  Trinh Dinh Phong          Toan tin           Co mat
20216880  Nguyen Thanh Son          Toan tin           Co mat
20237379  Vuong Van Phong           Toan tin           Co mat
20237483  Nguyen Doanh Thai         Toan tin           Co mat
20237473  Tran Thanh Phu            Toan tin           Vang
20237382  Le Minh Quan              Toan tin           Co mat
20237478  Pham Thi Quynh            Toan tin           Co mat
20237387  Vu Xuan Thai              Toan tin           Co mat
20237386  Sam Quoc Su               Toan tin           Vang
--------------------------------------------------------------------------------
Tổng số sinh viên: 10
Có mặt: 8, Vắng: 2
Tỷ lệ có mặt: 80.0%

Nhấn Enter để tiếp tục...
```

5.2. Nếu buổi điểm danh đó không tồn tại

```
 --- XUẤT BẢNG ĐIỂM DANH THEO NGÀY ---
Nhập ngày cần xuất bảng điểm danh (dd/mm/yyyy): 02/02/2022
Nhập mã lớp: 158241
Không tìm thấy bản ghi điểm danh cho lớp 158241 vào ngày 02/02/2022.

Nhấn Enter để tiếp tục...
```

5.3. Nếu người dùng chưa đọc dữ liệu của lớp

```
 --- XUẤT BẢNG ĐIỂM DANH THEO NGÀY ---
Nhập ngày cần xuất bảng điểm danh (dd/mm/yyyy): 06/06/2025
Nhập mã lớp: 150149
Lớp 150149 không tồn tại trong bộ nhớ!

Nhấn Enter để tiếp tục...
```

6. Tìm kiếm điểm danh của sinh viên trong lớp

6.1. Nếu sinh viên tồn tại trong lớp có các buổi điểm danh

```
 --- XUẤT BẢNG ĐIỂM DANH CỦA SINH VIÊN ---
Nhập mã sinh viên: 20237387
Nhập mã lớp: 158241

BẢNG ĐIỂM DANH CỦA SINH VIÊN Vu Xuan Thai (MSV: 20237387)
============================================================
Ngày: 05/06/2025 - Trạng thái: Có mặt
Ngày: 06/06/2025 - Trạng thái: Có mặt
Ngày: 07/06/2025 - Trạng thái: Có mặt
Ngày: 08/06/2025 - Trạng thái: Có mặt
Ngày: 09/06/2025 - Trạng thái: Vắng
Tổng số buổi vắng: 1
Tổng số buổi có mặt: 4
Tỷ lệ điểm danh: 80.00%

Nhấn Enter để tiếp tục...
```

6.2. Nếu sinh viên không tồn tại trong lớp

```
 --- XUẤT BẢNG ĐIỂM DANH CỦA SINH VIÊN ---
Nhập mã sinh viên: 20237380
Nhập mã lớp: 158241
Sinh viên 20237380 không tồn tại trong lớp 158241!

Nhấn Enter để tiếp tục...
```

6.3. Nếu sinh viên tồn tại trong lớp nhưng không có buổi điểm danh

```
 --- XUẤT BẢNG ĐIỂM DANH CỦA SINH VIÊN ---
Nhập mã sinh viên: 20237387
Nhập mã lớp: 150149

BẢNG ĐIỂM DANH CỦA SINH VIÊN Vu Xuan Thai (MSV: 20237387)
============================================================
Chưa có lịch sử điểm danh nào cho lớp này.

Nhấn Enter để tiếp tục...
```

7. Xem lịch sử điểm danh của lớp

7.1. Nếu lớp đã điểm danh

```
 --- XEM LỊCH SỬ CÁC NGÀY ĐÃ ĐIỂM DANH ---
Nhập mã lớp cần xem lịch sử: 158241

LỊCH SỬ ĐIỂM DANH LỚP Ky thuat lap trinh (Mã: 158241)
============================================================
Ngày: 05/06/2025 - Số SV được điểm danh: 10 (Có mặt: 7, Vắng: 3)
Ngày: 06/06/2025 - Số SV được điểm danh: 10 (Có mặt: 10, Vắng: 0)
Ngày: 07/06/2025 - Số SV được điểm danh: 10 (Có mặt: 8, Vắng: 2)
Ngày: 08/06/2025 - Số SV được điểm danh: 10 (Có mặt: 6, Vắng: 4)
Ngày: 09/06/2025 - Số SV được điểm danh: 10 (Có mặt: 7, Vắng: 3)

Nhấn Enter để tiếp tục...
```

7.2. Nếu lớp chưa điểm danh

```
 --- XEM LỊCH SỬ CÁC NGÀY ĐÃ ĐIỂM DANH ---
Nhập mã lớp cần xem lịch sử: 150149

LỊCH SỬ ĐIỂM DANH LỚP Lich Su Dang (Mã: 150149)
============================================================
Chưa có lịch sử điểm danh nào cho lớp này.

Nhấn Enter để tiếp tục...
```

8. Xem thống kê điểm danh của lớp

8.1. Nếu lớp có sinh viên và các buổi điểm danh

```
 --- XEM THỐNG KÊ ĐIỂM DANH CỦA LỚP ---
Nhập mã lớp cần xem thống kê: 158241

THỐNG KÊ LỚP Ky thuat lap trinh (Mã: 158241)
============================================================
Tổng số sinh viên: 10
Tổng số buổi điểm danh: 5
Tỷ lệ điểm danh trung bình toàn lớp: 76.00%
================================================================================================
Ho va ten               Ma SV       05/06/2025  06/06/2025  07/06/2025  08/06/2025  09/06/2025  Tong
------------------------------------------------------------------------------------------------
Pham Nam Phong          20237378    Co mat      Co mat      Co mat      Vang        Co mat      4
Trinh Dinh Phong        20237472    Co mat      Co mat      Co mat      Vang        Co mat      4
Nguyen Thanh Son        20216880    Vang        Co mat      Co mat      Co mat      Vang        3
Vuong Van Phong         20237379    Co mat      Co mat      Co mat      Co mat      Co mat      5
Nguyen Doanh Thai       20237483    Co mat      Co mat      Co mat      Co mat      Vang        4
Tran Thanh Phu          20237473    Vang        Co mat      Vang        Co mat      Co mat      3
Le Minh Quan            20237382    Co mat      Co mat      Co mat      Vang        Co mat      4
Pham Thi Quynh          20237478    Co mat      Co mat      Co mat      Co mat      Co mat      5
Vu Xuan Thai            20237387    Co mat      Co mat      Co mat      Co mat      Vang        4
Sam Quoc Su             20237386    Vang        Co mat      Vang        Vang        Co mat      2
================================================================================================

Nhấn Enter để tiếp tục...
```

8.1. Nếu lớp không có sinh viên hoặc các buổi điểm danh

```
 --- XEM THỐNG KÊ ĐIỂM DANH CỦA LỚP ---
Nhập mã lớp cần xem thống kê: 150

THỐNG KÊ LỚP 150 (Mã: 150)
============================================================
Lớp chưa có sinh viên.

Nhấn Enter để tiếp tục...
```

```
 --- XEM THỐNG KÊ ĐIỂM DANH CỦA LỚP ---
Nhập mã lớp cần xem thống kê: 150149

THỐNG KÊ LỚP Lich Su Dang (Mã: 150149)
============================================================
Lớp chưa có buổi điểm danh nào.

Nhấn Enter để tiếp tục...
```

9. Đọc dữ liệu của lớp từ file

9.1. Nếu lớp tồn tại

```
 --- ĐỌC DỮ LIỆU LỚP TỪ FILE ---
Nhập mã lớp cần đọc dữ liệu từ file: 158241
Đọc dữ liệu cho lớp 158241 - 'Ky thuat lap trinh' thành công.

Nhấn Enter để tiếp tục...
```

9.2. Nếu lớp không tồn tại

```
 --- ĐỌC DỮ LIỆU LỚP TỪ FILE ---
Nhập mã lớp cần đọc dữ liệu từ file: 152141
File không tồn tại cho lớp 152141.

Nhấn Enter để tiếp tục...
```

10. Xử lý đầu vào không hợp lệ

10.1. Nhập sai kiểu mã lớp

```
 --- THÊM SINH VIÊN VÀO LỚP ---
Nhập mã lớp cần thêm sinh viên: T150
Vui lòng nhập một số nguyên hợp lệ:
```

10.2. Nhập sai số lượng sinh viên cần thêm

```
 --- THÊM SINH VIÊN VÀO LỚP ---
Nhập mã lớp cần thêm sinh viên: T150
Vui lòng nhập một số nguyên hợp lệ: 150149
Nhập số lượng sinh viên cần thêm: 0
Số lượng sinh viên không hợp lệ! Thao tác bị hủy.

Nhấn Enter để tiếp tục...
```

10.3. Nhập sai mã sinh viên

```
 --- THÊM SINH VIÊN VÀO LỚP ---
Nhập mã lớp cần thêm sinh viên: 150149
Nhập số lượng sinh viên cần thêm: 1

--- Nhập thông tin cho sinh viên thứ 1 ---
Mã sinh viên (số nguyên): T2023
Vui lòng nhập một số nguyên hợp lệ:
```

10.3. Nhập sai định dạng ngày

```
 --- XUẤT BẢNG ĐIỂM DANH THEO NGÀY ---
Nhập ngày cần xuất bảng điểm danh (dd/mm/yyyy): 5/5/2025
Định dạng ngày không hợp lệ: '5/5/2025'. Thao tác bị hủy.

Nhấn Enter để tiếp tục...
```

Chương IV: Đánh giá hiệu năng

Hệ thống được thiết kế với hiệu năng tốt, phù hợp với phạm vi và mục tiêu của một ứng dụng quản lý dành cho môi trường giáo dục có quy mô vừa phải. Các lựa chọn về cấu trúc dữ liệu đã cân bằng giữa tốc độ thực thi, sự đơn giản trong triển khai và tính hiệu quả trong quản lý bộ nhớ.

1. Điểm mạnh về hiệu năng:

- Truy xuất dữ liệu cơ bản cực nhanh: Việc sử dụng unordered_map làm cấu trúc dữ liệu chính để quản lý danh sách lớp (danhSachLop) và danh sách sinh viên trong mỗi lớp (danhSachSV) là một lựa chọn thiết kế xuất sắc. Nó đảm bảo rằng các thao tác tra cứu, thêm mới, và truy cập một lớp học hoặc một sinh viên cụ thể (khi đã biết mã định danh) có độ phức tạp trung bình là O(1). Điều này có nghĩa là thời gian để tìm một lớp trong hàng ngàn lớp, hay tìm một sinh viên trong một lớp hàng trăm người gần như là tức thời và không bị ảnh hưởng bởi sự gia tăng của dữ liệu. Đây là yếu tố then chốt đảm bảo trải nghiệm người dùng mượt mà cho các tác vụ thường xuyên.
- Hiệu quả trong các thao tác tương tác thường xuyên: Các chức năng mà người dùng tương tác nhiều nhất như thêm lớp, thêm sinh viên, chọn một lớp để thao tác đều có hiệu năng rất cao. Điều này giúp hệ thống phản hồi nhanh, không gây cảm giác trễ hay chờ đợi.

2. Các điểm cần cân nhắc và đánh đổi về hiệu năng:

- Tìm kiếm tuyến tính trong lịch sử điểm danh: Việc lưu trữ các buổi điểm danh (DiemDanh) trong một vector là một lựa chọn tự nhiên và đơn giản, giữ được thứ tự thời gian. Tuy nhiên, điều này dẫn đến việc tìm kiếm một buổi điểm danh theo ngày cụ thể phải thực hiện tìm kiếm tuyến tính với độ phức tạp O(D) (với D là tổng số buổi đã điểm danh). Trong thực tế, các thao tác như xem lại hoặc sửa một buổi điểm danh cũ sẽ chậm đi tương ứng. Dù vậy, với quy mô của một môn học thông thường (D < 100), độ trễ này là không đáng kể và hoàn toàn chấp nhận được.
- Các tác vụ xử lý toàn cục có độ phức tạp cao: Các chức năng yêu cầu xử lý trên toàn bộ dữ liệu của một lớp, như tính toán thống kê (tinhThongKe) và xem toàn bộ lịch sử (xemLichSu), có độ phức tạp lên đến O(S*D). Điều này là không thể tránh khỏi vì bản chất của các tác vụ này là phải duyệt qua mọi bản ghi điểm danh của mọi sinh viên. Khi số lượng sinh viên (S) và số buổi điểm danh (D) cùng tăng lên, thời gian thực thi sẽ tăng theo cấp số nhân. Tuy nhiên, đây là những chức năng không được thực hiện liên tục, người dùng thường chỉ chạy chúng vào cuối kỳ hoặc khi cần tổng hợp nên người dùng có thể đợi được.

- Thao tác đọc/ghi tệp: Đây là thao tác có độ phức tạp cao nhất trong toàn hệ thống, O(L*S*D) nếu lưu tất cả dữ liệu. Việc đọc/ghi tệp vốn dĩ chậm hơn nhiều so với các thao tác trên bộ nhớ RAM. Tuy nhiên, đây là yêu cầu bắt buộc để đảm bảo tính bền vững của dữ liệu. Thiết kế hiện tại (lưu riêng từng lớp) có thể tối ưu hóa bằng cách chỉ lưu những lớp có thay đổi, nhưng việc lưu toàn bộ khi kết thúc chương trình vẫn là một giải pháp đơn giản và an toàn cho phạm vi đồ án.

3. Kết luận về hiệu năng:

- Đối với mục đích và bối cảnh của một đồ án môn học Kỹ thuật Lập trình, các lựa chọn thiết kế hiện tại là hoàn toàn hợp lý và hiệu quả. Hệ thống ưu tiên tốc độ cho các thao tác tương tác thời gian thực, trong khi chấp nhận hiệu năng thấp hơn cho các tác vụ xử lý hàng loạt (batch processing) ít được sử dụng hơn.
- Cấu trúc hiện tại đủ sức phục vụ tốt cho vài chục lớp học, mỗi lớp có khoảng 100 sinh viên và lịch sử điểm danh trong một năm học mà không gặp bất kỳ vấn đề nào về hiệu năng. Nếu trong tương lai, hệ thống cần mở rộng để quản lý dữ liệu ở quy mô cấp trường với hàng chục ngàn sinh viên và lịch sử nhiều năm, các giải pháp nâng cao hơn như sử dụng cơ sở dữ liệu (ví dụ: SQLite, MySQL, PostgreSQL) hoặc tối ưu hóa cấu trúc tệp và chỉ mục hóa dữ liệu sẽ cần được xem xét. Tuy nhiên, trong phạm vi hiện tại, hệ thống đã đạt được mục tiêu về hiệu năng một cách xuất sắc.

Chương V: Các kỹ thuật đã sử dụng

1. Lập trình Hướng đối tượng

Đây là tư tưởng thiết kế chủ đạo của toàn bộ dự án. Thay vì viết code dưới dạng các hàm thủ tục rời rạc, dự án mô hình hóa các khái niệm của bài toán thành các đối tượng (object) có thuộc tính (data) và hành vi (method) riêng.

- Tính trừu tượng: Các đối tượng trong thế giới thực như "Lớp học", "Sinh viên", "Menu" được trừu tượng hóa thành các class trong C++ (Lop, Sinhvien, Menu). Mỗi lớp chỉ phơi bày ra bên ngoài các giao diện cần thiết (hàm public), trong khi che giấu hoàn toàn các chi tiết cài đặt phức tạp bên trong (hàm và biến private). Ví dụ, người dùng lớp QuanLyLop chỉ cần gọi themLop() mà không cần biết bên trong nó đang sử dụng unordered_map hay một cấu trúc dữ liệu nào khác để lưu trữ.
- Tính đóng gói: Dữ liệu và các phương thức xử lý dữ liệu đó được đóng gói chặt chẽ bên trong các lớp. Ví dụ, dữ liệu của một lớp học (danhSachSV, danhSachDiemDanh) được khai báo là private trong lớp Lop. Việc truy cập và thay đổi dữ liệu này phải thông qua các phương thức public do chính lớp Lop cung cấp. Điều này giúp bảo vệ sự toàn vẹn của dữ liệu, tránh các thay đổi không mong muốn từ bên ngoài.
- Tính module hóa thông qua lớp: Mỗi lớp đảm nhiệm một vai trò và trách nhiệm riêng biệt, tạo nên các module chức năng độc lập. Menu lo về giao diện, QuanLyLop xử lý nghiệp vụ, FileManager quản lý file, v.v.

2. Lập trình Module hóa (Modular Programming)

Kỹ thuật này được áp dụng triệt để để tăng tính tổ chức và khả năng tái sử dụng của mã nguồn.

- Phân tách tệp Header (.h) và tệp Triển khai (.cpp): Mỗi lớp hoặc module chức năng được chia thành hai tệp:
- Tệp .h (header file): Chứa khai báo của lớp, bao gồm các thuộc tính và phương thức. Tệp này đóng vai trò như một "bản hợp đồng" về chức năng mà module cung cấp.
- Tệp .cpp (source file): Chứa phần định nghĩa và cài đặt chi tiết của các phương thức đã khai báo trong tệp header.
- Cách làm này giúp tách biệt giao diện và cài đặt, giảm thời gian biên dịch (khi chỉ thay đổi tệp .cpp), và giúp mã nguồn trở nên sạch sẽ, dễ đọc hơn.
- Tổ chức mã nguồn theo thư mục chức năng: Dự án được cấu trúc thành các thư mục có ý nghĩa rõ ràng . Điều này giúp dễ quản lý và phát triển các thành phần của hệ thống. Bất kỳ ai đọc dự án đều nắm bắt được cấu trúc tổng thể.

3. Xử lý ngoại lệ

- Để tăng cường độ tin cậy cho chương trình, cơ chế xử lý ngoại lệ đã được áp dụng.
- Trong hàm main, toàn bộ logic thực thi chính được đặt trong một khối try. Nếu có bất kỳ lỗi nghiêm trọng nào xảy ra ở các lớp bên dưới và một ngoại lệ (exception) được ném ra, khối catch sẽ bắt lại.
- Điều này ngăn chương trình không bị crash (dừng đột ngột), thay vào đó nó sẽ in ra một thông báo lỗi thân thiện và kết thúc một cách an toàn. Đây là một kỹ thuật quan trọng để xây dựng các ứng dụng mạnh mẽ, có khả năng phục hồi sau lỗi.

4. Xác thực dữ liệu đầu vào

- Đây là một kỹ thuật cơ bản nhưng cực kỳ quan trọng để đảm bảo tính đúng đắn và toàn vẹn của dữ liệu.
- Hệ thống không mù quáng tin tưởng vào dữ liệu người dùng nhập. Thay vào đó, nó thực hiện các bước kiểm tra cần thiết.
- Ví dụ điển hình là lớp DateValidator dùng để kiểm tra xem chuỗi ngày người dùng nhập có đúng định dạng và là một ngày hợp lệ trong lịch hay không. Tương tự, các hàm helper trong lớp Menu (như readInt) đảm bảo rằng người dùng phải nhập đúng kiểu dữ liệu số khi được yêu cầu.

5. Quy tắc đặt tên.

- Để đảm bảo mã nguồn có tính thống nhất, dễ đọc và dễ bảo trì trên toàn bộ dự án, một bộ quy tắc đặt tên rõ ràng đã được tuân thủ nghiêm ngặt. Đây là một kỹ thuật quan trọng giúp giảm thiểu sự mơ hồ và tăng tốc độ làm quen với mã nguồn.
- Lớp và Cấu trúc: Sử dụng quy tắc PascalCase, trong đó ký tự đầu tiên của mỗi từ trong tên đều được viết hoa. *Ví dụ:* Menu, QuanLyLop, FileManager, Sinhkhoa, DiemDanh, ThongKeLop.
- Biến và Hàm/Phương thức: Sử dụng quy tắc camelCase, trong đó ký tự đầu tiên của từ đầu tiên được viết thường, và ký tự đầu tiên của các từ tiếp theo được viết hoa. *Ví dụ:* danhSachLop, maSV, hoTen, trangThai.
- Hằng số và Macro của Header Guard: Sử dụng quy tắc UPPER_SNAKE_CASE, trong đó tất cả các ký tự đều được viết hoa và các từ được ngăn cách bởi dấu gạch dưới (_). *Ví dụ:* DATA_DIR (hằng số).
- Tên tệp: Sử dụng lowercase, và tên tệp thường tương ứng với tên của lớp hoặc module chính mà nó chứa đựng. *Ví dụ:* menu.h, quanlylop.cpp.

6. Cấu trúc dữ liệu và Giải thuật

- Việc lựa chọn đúng cấu trúc dữ liệu và giải thuật là yếu tố quyết định đến hiệu năng của chương trình. Các lựa chọn trong dự án được thực hiện dựa trên sự phân tích kỹ lưỡng về yêu cầu của từng tác vụ.
- Sử dụng Cấu trúc dữ liệu bảng băm (unordered_map) để Tối ưu hóa Truy cập:
  - Kỹ thuật: Thay vì sử dụng mảng hoặc vector và tìm kiếm tuyến tính (độ phức tạp O(n)), dự án đã tận dụng unordered_map cho các tác vụ tra cứu thường xuyên.
  - Áp dụng:
    - danhSachLop (trong QuanLyLop): Giúp tìm kiếm một lớp học theo maLop với thời gian trung bình là O(1).
    - danhSachSV (trong Lop): Giúp tìm kiếm một sinh viên theo maSV với thời gian trung bình là O(1).
    - trangThai (trong DiemDanh): Giúp tra cứu trạng thái điểm danh của một sinh viên cụ thể trong một buổi học với thời gian trung bình là O(1).
  - Tác động: Lựa chọn này là nền tảng cho hiệu năng của toàn bộ hệ thống, đảm bảo rằng chương trình vẫn phản hồi nhanh ngay cả khi số lượng lớp và sinh viên tăng lên đáng kể.
- Sử dụng Mảng động (vector) để Lưu trữ Dữ liệu Tuần tự:
  - Kỹ thuật: Khi thứ tự của các phần tử là quan trọng và cần được bảo toàn, vector là lựa chọn tối ưu.
  - Áp dụng: danh sách điểm danh trong lớp Lop được khai báo là DiemDanh. Điều này đảm bảo rằng lịch sử các buổi điểm danh được lưu trữ theo đúng trình tự thời gian chúng được tạo ra.
  - Sự đánh đổi: Kỹ thuật này đi kèm với một sự đánh đổi có chủ đích. Việc truy cập một buổi điểm danh theo ngày sẽ yêu cầu một giải thuật tìm kiếm tuyến tính (duyệt qua vector) với độ phức tạp O(D) (D là số buổi điểm danh). Tuy nhiên, với quy mô của một môn học, đây là một sự đánh đổi hoàn toàn chấp nhận được để đổi lấy sự đơn giản và khả năng duy trì thứ tự.
- Giải thuật Duyệt và Tổng hợp Dữ liệu:
  - Kỹ thuật: Nhiều chức năng của chương trình được triển khai dưới dạng các giải thuật duyệt và xử lý dữ liệu.
  - Áp dụng:
  - Giải thuật duyệt lồng nhau: Chức năng tính toán thống kê (tinhThongKe) là một ví dụ điển hình. Nó sử dụng một vòng lặp ngoài để duyệt qua tất cả

- các buổi điểm danh (danhSachDiemDanh) và một vòng lặp trong để duyệt qua trạng thái của từng sinh viên trong buổi đó (trangThai). Đây là một giải thuật duyệt dữ liệu có cấu trúc để thu thập thông tin.
- Giải thuật tổng hợp (Aggregation): Hàm tinhThongKe không chỉ duyệt, mà còn tổng hợp dữ liệu. Nó khởi tạo các biến đếm (ví dụ: tongSoLuotCoMat, soLanVang) và tích lũy kết quả qua mỗi lần lặp để tạo ra một bản tóm tắt (báo cáo thống kê) từ dữ liệu thô.

7. Kỹ thuật Bình luận và Chú thích Mã nguồn

- Dự án áp dụng một chiến lược bình luận (comment) có chủ đích, nhằm mục tiêu làm rõ những phần mã nguồn phức tạp hoặc những quyết định thiết kế quan trọng, thay vì bình luận một cách tràn lan. Nguyên tắc chính là "để mã nguồn tự diễn giải ở những nơi có thể, và chỉ bình luận khi cần thiết để giải thích 'tại sao' chứ không phải 'cái gì'".
- Chú thích Chức năng của Lớp và Module:
  - Kỹ thuật: Ở đầu các tệp header (.h), các bình luận ngắn gọn được sử dụng để mô tả trách nhiệm chung của lớp hoặc module đó.
- Giải thích Mục đích của Biến hoặc Cấu trúc Phức tạp:
  - Kỹ thuật: Khi một biến thành viên có ý nghĩa không thể hiện rõ hoàn toàn qua tên của nó, một bình luận ngắn sẽ được thêm vào để làm rõ.
- Nguyên tắc Tránh Bình luận Thừa:
  - Kỹ thuật: Dự án chủ động tránh việc viết những bình luận giải thích những gì mã nguồn đã thể hiện rõ ràng.

Chương VI: Phụ lục

1. Hàm main

```
#include "ui/menu.h"
#include <iostream>
using namespace std;
int main(){
   try {
      Menu m;
      m.menu();
   } catch(const exception& e) {
      cout << "Lỗi nghiêm trọng: " << e.what() << endl;
      return 1;
   }
   return 0;
}
```

2. Chương trình con

Lớp QuanLyLop (services/quanlylop.h)

- bool themLop(int maLop, const string& tenLop);
  - Mô tả: Thêm một lớp học mới vào danh sách quản lý.
  - Tham số:
    - maLop (int): Mã của lớp mới.
    - tenLop (const string&): Tên của lớp mới.
  - Trả về: true nếu thêm thành công, false nếu mã lớp đã tồn tại.
- bool themSinhvien(int maLop, const Sinhvien& sv);
  - Mô tả: Thêm một sinh viên mới vào một lớp học đã có.
  - Tham số:
    - maLop(int): Mã lớp cần thêm sinh viên.
    - sv (const Sinhvien&): Đối tượng sinh viên cần thêm.
  - Trả về: true nếu thêm thành công, false nếu thất bại.
- void xuatDiemDanh(const string& ngay, int maLop);
  - Mô tả: Hiển thị thông tin điểm danh của một lớp vào một ngày cụ thể.
  - Tham số:
    - ngay (const string&): Ngày cần xem điểm danh.
    - maLop (int): Mã lớp cần xem.
  - Trả về: Không có (xuất ra console).

- bool diemDanh(const string& ngay, int maLop);
  - Mô tả: Thực hiện điểm danh cho tất cả sinh viên trong một lớp vào một ngày cụ thể. Hàm này có thể gọi diemDanhMoi hoặc các hàm helper khác.
  - Tham số:
    - ngay (const string&): Ngày thực hiện điểm danh (định dạng "dd/mm/yyyy").
    - maLop (int): Mã lớp cần điểm danh.
  - Trả về: true nếu điểm danh (hoặc bắt đầu quá trình điểm danh) thành công, false nếu có lỗi (ví dụ: lớp không tồn tại, ngày không hợp lệ).
- bool suaDiemDanh(const string& ngay, int maLop, int maSV, int trangThai);
  - Mô tả: Sửa trạng thái điểm danh của một sinh viên cụ thể trong một buổi điểm danh đã có.
  - Tham số:
    - ngay (const string&): Ngày của buổi điểm danh cần sửa.
    - maLop (int): Mã lớp.
    - maSV (int): Mã sinh viên cần sửa trạng thái.
    - trangThai (int): Trạng thái điểm danh mới.
  - Trả về: true nếu sửa thành công, false nếu không tìm thấy buổi điểm danh hoặc sinh viên.
- void xemLichSu(int maLop);
  - Mô tả: Hiển thị toàn bộ lịch sử các buổi điểm danh của một lớp.
  - Tham số:
    - maLop (int): Mã lớp cần xem lịch sử.
  - Trả về: Không có (xuất ra console).
- void xemThongKe(int maLop);
  - Mô tả: Hiển thị báo cáo thống kê điểm danh cho một lớp.
  - Tham số:
    - maLop (int): Mã lớp cần xem thống kê.
  - Trả về: Không có (xuất ra console, sử dụng lớp ThongKe).
- bool saveDiemDanh();
  - Mô tả: Gọi hàm lưu dữ liệu (saveLop) trong lớp FileManager
  - Tham số: Không có.
  - Trả về: true nếu lưu thành công, false nếu có lỗi.

- bool docFile(int maLop);
  - Mô tả: Gọi hàm đọc dữ liệu (readLop) từ lớp FileManager.
  - Tham số:
    - maLop (int): Mã lớp cần đọc dữ liệu.
  - Trả về: true nếu đọc thành công, false nếu tệp không tồn tại hoặc có lỗi.
- bool lopTonTai(int maLop) const;
  - Mô tả: Kiểm tra xem một mã lớp có tồn tại trong danh sách quản lý hay không.
  - Tham số:
    - maLop (int): Mã lớp cần kiểm tra.
  - Trả về: true nếu lớp tồn tại, false nếu không.
- vector<int> getDanhSachMaLop() const;
  - Mô tả: Lấy danh sách các mã lớp hiện có.
  - Tham số: Không.
  - Trả về: vector<int> chứa các mã lớp.

Lớp ThongKe (services/thongke.h)

- static ThongKeLop tinhThongKe(const Lop& lop);
  - Mô tả: Tính toán và trả về các thông tin thống kê cho một đối tượng lớp học.
  - Tham số:
    - lop (const Lop&): Đối tượng lớp cần tính thống kê.
  - Trả về: ThongKeLop (struct) chứa các dữ liệu thống kê đã tính toán.

Lớp FileManager (utils/filemanager.h)

- static bool createDirectory();
  - Mô tả: Tạo thư mục lưu trữ dữ liệu nếu nó chưa tồn tại.
  - Tham số: Không có.
  - Trả về: true nếu tạo thành công hoặc thư mục đã tồn tại, false nếu có lỗi.
- static string getFilePath(int maLop);
  - Mô tả: Trả về đường dẫn đầy đủ đến tệp dữ liệu của một lớp cụ thể.
  - Tham số:
    - maLop (int): Mã lớp.
  - Trả về: string là đường dẫn tệp.
- static bool fileExists(int maLop);
  - Mô tả: Kiểm tra xem tệp dữ liệu của một lớp có tồn tại hay không.
  - Tham số:
    - maLop (int): Mã lớp.
  - Trả về: true nếu tệp tồn tại, false nếu không.

- static bool saveLop(const Lop& lop);
  - Mô tả: Lưu toàn bộ dữ liệu điểm danh (của tất cả các lớp đang quản lý) ra tệp.
  - Tham số:
    - lop (const Lop&): Đối tượng lớp cần lưu.
  - Trả về: True nếu lưu thành công, false nếu lưu thất bại
- static bool readLop(int *maLop*, Lop& *outLop*);
  - Mô tả: Đọc dữ liệu của một lớp cụ thể từ tệp vào bộ nhớ.
  - Tham số:
    - maLop(int): Mã lớp cần đọc dữ liệu.
    - outLop(Lop&): Đối lượng lớp để lưu dữ liệu của mã lớp đọc vào.
  - Trả về: True nếu đọc dữ liệu thành công, false nếu đọc dữ liệu thất bại.

Lớp DateValidator (utils/datevalidator.h)

- static bool isValidDate(const string& date);
  - Mô tả: Kiểm tra xem một chuỗi ngày tháng có hợp lệ về định dạng và giá trị hay không.
  - Tham số:
    - date (const string&): Chuỗi ngày tháng cần kiểm tra.
  - Trả về: true nếu ngày hợp lệ, false nếu không.
- static string getCurrentDate();
  - Mô tả: Trả về ngày hiện tại của hệ thống dưới dạng chuỗi (theo một định dạng nhất định).
  - Tham số: Không có.
  - Trả về: string biểu thị ngày hiện tại.

Lớp Menu (ui/menu.h)

- void menu();
  - Mô tả: Hiển thị menu chính của ứng dụng, nhận lựa chọn của người dùng và điều hướng đến các chức năng tương ứng. Lặp lại cho đến khi người dùng chọn thoát.
  - Tham số: Không có.
  - Trả về: Không có**.**
- Các chức năng chuyển màn hình hiển thị chức năng
  - void handleThemLop();
  - void handleThemSinhvien();
  - void handleDiemDanh();
  - void handleSuaDiemDanh();
  - void handleXuatDiemDanh();

- void handleXuatDiemDanhSV();
- void handleTimLichSu();
- void handleXemThongKe();
- void handleDocFile();
- Mô tả: Các hàm thực hiện các chức năng gọi các hàm chính từ class quản lý lớp để tạo khung hình nhập dữ liệu thao tác cho các chức năng chính.
- Tham số: Không có.
- Trả về: Không có.