**ĐẠI HỌC ĐÀ NẴNG**

**ĐẠI HỌC BÁCH KHOA**

**KHOA CÔNG NGHỆ THÔNG TIN**



**PBL2: DỰ ÁN CƠ SỞ LẬP TRÌNH**

**Đề tài: Quản lí điểm học sinh THPT**

**SINH VIÊN THỰC HIỆN:**

**Tên sinh viên 1: Nguyễn Quốc Tiến LỚP: 22T\_KHDL**

**Tên sinh viên 2: Phạm Quốc Vũ LỚP: 22T\_KHDL**

**GIẢNG VIÊN HƯỚNG DẪN: TS. Trương Ngọc Châu**

**Đà Nẵng 2023-2024**

LỜI MỞ ĐẦU

Việc áp dụng các phần mềm tin học vào các lĩnh vực giúp nâng cao tính hiệu quả và chính xác của công việc , ngoài ra còn tiết kiệm thời gian và giảm bớt mệt nhọc cho con người khiến hệ thống công việc hoạt động nhịp nhàng hơn. Đặc biệt hơn chúng còn đóng một vai trò quan trọng trong giáo dục vì nó mang lại nhiều lợi ích cho giáo viên, học sinh và quản lí trường học. Nhận thấy được nhu cầu ấy, chúng em đã chọn đề tài cho “Dự án Cơ sở lập trình” là “Quản lí điểm học sinh THPT”. Cùng với tốc độ phát triển mạnh mẽ của Công nghệ thông tin các lập trình viên đã phát minh ra nhiều phần mềm hữu ích nhằm phục vụ cho công việc của con người và phần mềm quản lí điểm học sinh cũng là một trong những vấn đề quan trọng, nó giúp cho công tác nghiệp vụ của các trường học giảm thiểu tối đa những vất vả trong công việc giúp cho việc lưu trữ hồ sơ dễ dàng hơn, giảm thiểu diện tích các kho và thậm chí là không cần. Chương trình của chúng em được lập trình bằng ngôn ngữ C++. Chương trình này có nhiều chức năng như quản lí hồ sơ, điểm, xếp loại của học sinh giúp giảm tải bớt việc tìm kiếm và tính toán mất thời gian mà độ hiệu quả và tính chính xác vẫn ở mức cao.

MỤC LỤC

[**LỜI MỞ ĐẦU 2**](#_Toc155376761)

[**MỤC LỤC 3**](#_Toc155376762)

[**DANH MỤC HÌNH VẼ 4**](#_Toc155376763)

[**1.GIỚI THIỆU ĐỀ TÀI 5**](#_Toc155376764)

[**2.CƠ SỞ LÝ THUYẾT 5**](#_Toc155376765)

[**3.PHÂN TÍCH CHỨC NĂNG HỆ THỐNG 6**](#_Toc155376766)

[**3.1.Quản lý danh sách lớp học và hồ sơ học sinh:… 6**](#_Toc155376767)

[**3.2.Quản Lí: 6**](#_Toc155376768)

[**4.THIẾT KẾ CẤU TRÚC DỮ LIỆU 8**](#_Toc155376769)

[**4.1.Phát biểu bài toán 8**](#_Toc155376770)

[**4.2.Phân tích và ứng dụng cấu trúc dữ liệu trong hệ thống 8**](#_Toc155376771)

[**5.PHÂN TÍCH HƯỚNG ĐỐI TƯỢNG VÀ TRIỂN KHAI HỆ THỐNG 13**](#_Toc155376772)

[**5.1.Cấu trúc hệ thống hướng đối tượng 13**](#_Toc155376773)

[**5.2.Kết quả 17**](#_Toc155376774)

[**5.2.1.Giao diện chính của chương trình 17**](#_Toc155376775)

[**5.2.2.Kết quả thực thi của chương trình 17**](#_Toc155376776)

[**5.2.3.Nhận xét 21**](#_Toc155376777)

[**KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 22**](#_Toc155376778)

[**a.Kết luận 22**](#_Toc155376779)

[**b.Hướng phát triển 22**](#_Toc155376780)

[**TÀI LIỆU THAM KHẢO 24**](#_Toc155376781)

DANH MỤC HÌNH VẼ

[**LINKED LIST 8**](#_Toc155377116)

[**THUẬT TOÁN LINEAR SEARCH 10**](#_Toc155377117)

[**SƠ ĐỒ KHỐI CỦA THUẬT TOÁN LINEAR SEARCH 11**](#_Toc155377118)

[**FILE THONGTINHOCSINH.TXT 12**](#_Toc155377119)

[**FILE DIEMMONHOC.TXT 12**](#_Toc155377120)

[**FILE SO\_LUONG.TXT 13**](#_Toc155377121)

[**FILE SO\_LUONG.TXT 14**](#_Toc155377705)

[**CẤU TRÚC HỆ THỐNG HƯỚNG ĐỐI TƯỢNG 15**](#_Toc155377706)

[**GIỚI THIỆU ĐỀ TÀI 17**](#_Toc155377707)

**MENU CÁC OPTION** [**17**](#_Toc155377708)

[**OPTION 1 18**](#_Toc155377709)

[**OPTION 2 SAU KHI LẤY DỮ LIỆU TỪ FILE 18**](#_Toc155377710)

[**TÌM KIẾM HỌC SINH THEO ID 19**](#_Toc155377713)

[**HIỂN THỊ NHỮNG ID ĐÃ NHẬP 20**](#_Toc155377716)

# GIỚI THIỆU ĐỀ TÀI

**Giới thiệu đề tài**

Đề tài về quản lí điểm học sinh là một dự án học tập thú vị và thực tế, nhằm tăng cường khả năng quản lí, ghi nhận và phân tích điểm số học sinh. Đề tài này tập trung vào việc xây dựng một hệ thống quản lí điểm học sinh sử dụng phần mềm để ghi nhận và phân tích thông tin điểm số của học sinh. Ngoài ra nó còn tập trung vào việc phát triển một ứng dụng hoặc phần mềm trực tuyến, giúp hỗ trợ giáo viên và quản lí trường học trong việc quản lí điểm số và theo dõi tiến trình học tập của học sinh.

**Mục tiêu đề tài**

Xây dựng một hệ thống quán lí điểm học sinh dễ sử dụng và hiệu quả, tối ưu hóa quá trình ghi nhận điểm số, từ đó giảm thiểu công việc thủ công và lỗi nhầm lẫn, phân tích kết quả học tập và đưa ra báo cáo chi tiết, từ đó tạo ra thông tin hữu ích cho giáo viên, học sinh và phụ huynh, tăng tính minh bạch và tương tác giữa giáo viên, học sinh và phụ huynh.

# CƠ SỞ LÝ THUYẾT

Class hay lớp là một mô tả trừu tượng (abstract) của nhóm các đối tượng (object)

có cùng bản chất, ngược lại mỗi một đối tượng là một thể hiện cụ thể (instance) cho những mô tả trừu tượng đó. Một class trong C++ sẽ có các đặc điểm sau:

▪ **Đối tượng**: một đối tượng gồm thuộc tính và phương thức trong đó bao gồm:

➢ **Thuộc tính**: là những thông tin, đặc điểm, tính chất của đối tượng.

➢ **Phương thức**: là những biểu hiện, thao tác của đối tượng.

▪ **Lớp**: Một lớp là một kiểu dữ liệu bao gồm các thuộc tính và các phương thức

được định nghĩa từ trước. Một lớp là một đơn vị (trừu tượng) bao gồm sự kết

hợp giữa các phương thức và các thuộc tính. Hiểu nôm na hơn là các đối

tượng có các đặc tính tương tự nhau được gom lại thành một lớp đối tượng.

▪ **Ví dụ**: lớp Xe bao gồm nhiều đối tượng xe như xe con, xe tải, xe máy…..,

với mỗi loại xe có mỗi chức năng khác nhau chính là các phương thức khác

nhau

❖ **Các tính chất cơ bản của OOP**:

**Tính đóng gói (Encapsulation)**: Các dữ liệu và phương thức có liên quan với

nhau được đóng gói thành các lớp để tiện cho việc quản lý và sử dụng. Tức là mỗi lớp được xây dựng để thực hiện một nhóm chức năng đặc trưng của riêng

lớp đó. Ngoài ra, đóng gói còn để che giấu một số thông tin và chi tiết cài đặt nội bộ để bên ngoài không thể nhìn thấy.

**Tính kế thừa (Inheritance)**: Nó cho phép xây dựng một lớp mới dựa trên cácđịnh nghĩa của lớp đã có. Có nghĩa là lớp cha có thể chia sẽ dữ liệu và phương thức cho các lớp con. Các lớp con khỏi phải định nghĩa lại, ngoài ra có thể mở rộng các thành phần kế thừa và bổ sung thêm các thành phần mới. Tái sử dụng mã nguồn 1 cách tối ưu,

tận dụng được mã nguồn.

**Tính đa hình (Polymorphism)**: Tính đa hình là một hành động có thể được thực hiện bằng nhiều cách khác nhau. Đây lại là một tính chất có thể nói là chứa đựng hầu hết sức mạnh của lập trình hướng đối tượng. Hiểu một cách đơn giản hơn đa hình là khái niệm mà hai hoặc nhiều lớp có những phương thức giống nhau nhưng có thể thực thi theo những cách thức khác nhau.

**Tính trừu tượng (Abstraction)**: Trừu tượng có nghĩ là tổng quát hóa một cái gì

đó lên, không cần chú ý chi tiết bên trong. Nó không màng đến chi tiết bên trong là gì và người ta vẫn hiểu nó mỗi khi nghe về nó.

Kết luận: Từ lý thuyết trên ta xây dựng các đối tượng để quản lí thông tin và điểm học sinh.

**▪ HocSinh**: gồm thông tin học sinh, học sinh thực hiện chức năng cập nhật thông tin cá nhân gồm họ tên ngày tháng năm sinh và ID.

**▪ MonHoc**: gồm điểm, điểm trung bình, học lực,… của học sinh.

**▪ QuanLi**: class chính để quản lí thông tin và điểm của học sinh. Gồm nhiều tính năng như thêm, sửa, xóa, tìm kiếm.

# PHÂN TÍCH CHỨC NĂNG HỆ THỐNG

## Quản lý danh sách lớp học và hồ sơ học sinh: Hệ thống cho phép quản lý viên và giáo viên tạo và quản lý danh sách lớp học. Họ có thể thêm, sửa đổi hoặc xóa thông tin về học sinh, bao gồm thông tin cá nhân,…

## Quản Lí:

**3.2.1 Các chức năng quản lí học sinh:**

**3.2.1.1 Nhập thông tin học sinh**

Chương trình cung cấp giao diện để nhập thông tin học sinh.

**3.2.1.2 Nhập điểm cho học sinh**

Chương trình cung cấp giao diện để nhập điểm cho học sinh.

**3.2.1.3 Xem thông tin học sinh**

Chức năng cho phép người dùng xem toàn bộ thông tin của học sinh có trong hệ thống bao gồm các thông tin như: ID, Name, Date.

**3.2.1.4 Xem điểm học sinh**

Chức năng cho phép người dùng xem điểm của học sinh có trong hệ thống bao

gồm các thông tin như: Điểm các môn học, điểm trung bình, học lực,….

**3.2.1.5 Hiển thị ID các học sinh đã nhập hoặc thêm vào từ file**

Chức năng cho phép xem các ID đã nhập trực tiếp hoặc gián tiếp từ file.

**3.2.1.6 Thêm học sinh vào đầu danh sách**

Đây là chức năng sử dụng cấu trúc dữ liệu link list để quản lí học sinh, và đưa thông tin học sinh vào đầu danh sách.

**3.2.1.7 Thêm học sinh vào cuối danh sách**

Đây là chức năng sử dụng cấu trúc dữ liệu link list để quản lí học sinh, và đưa thông tin học sinh vào cuối danh sách.

**3.2.1.8 Thêm học sinh vào vị trí bất kì trong danh sách**

Đây là chức năng sử dụng cấu trúc dữ liệu link list để quản lí học sinh, và đưa thông tin học sinh vào vị trí bất kì trong danh sách.

**3.2.1.9 Xóa học sinh ở đầu danh sách**

Chương trình cho phép người dùng xóa thông tin và điểm của học sinh ở đầu danh sách. Sau khi xóa thì thông tin và điểm của học sinh đó sẽ biến mất khỏi danh sách.

**3.2.1.10 Xóa học sinh ở cuối danh sách**

Chương trình cho phép người dùng xóa thông tin và điểm của học sinh ở cuối danh sách. Sau khi xóa thì thông tin và điểm của học sinh đó sẽ biến mất khỏi danh sách.

**3.2.1.11 Xóa học sinh ở vị trí bất kì trong danh sách**

Chương trình cho phép người dùng xóa thông tin học sinh thông qua ID của học sinh.

Sau khi xóa toàn bộ thông tin thuộc về bệnh nhân sẽ biến mất khỏi database của

hệ thống.

**3.2.1.12 Sửa điểm cho học sinh**

Chương trình cho phép người dùng sửa điểm của học sinh thông qua ID của học sinh đó. Sửa dụng phương thức set, get để thực hiện.

**3.2.1.13 Tìm kiếm học sinh**

Chươn trình cho phép người dùng tìm kiếm học sinh thông qua ID của học sinh đó. ở đây bọn em sử dụng thuật toán **Linear Search.**

**3.2.1.14 Sắp xếp điểm**

Sử dụng thuật toán **Select Sort** để sắp xếp theo thứ tự tăng dần về điểm của học sinh đó.

**3.2.2 Hiển thi danh sách:**

Chương trình cho phép người dùng xem thông tin và điểm của những học sinh đã được được thêm vào danh sách như: name,id,date,point,….

# THIẾT KẾ CẤU TRÚC DỮ LIỆU

## Phát biểu bài toán

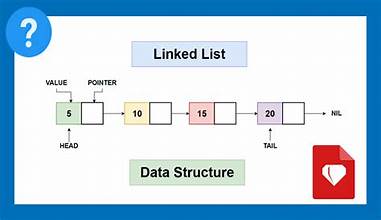
Xây dựng một hệ thống quản lý điểm học sinh để theo dõi và ghi chép thông tin điểm số của học sinh trong một học kỳ. Hệ thống cần có khả năng lưu trữ điểm số từng môn học, quản lý trọng số điểm, tính điểm trung bình tổng kết, và hiển thị thông tin điểm một cách rõ ràng.

Hệ thống cũng cần hỗ trợ các chức năng như nhập điểm, sửa điểm, xóa điểm, và xem bảng điểm cá nhân cho từng học sinh. Ngoài ra, cần có khả năng tạo báo cáo về kết quả học tập của học sinh, thống kê điểm số theo lớp, môn học, và khóa học.

Input: Nhập thủ công hoặc đọc file chứa thông tin và điểm số của từng học sinh.

Output: Các thao tác chức năng quản lí thông tin và điểm số của học sinh.

## Phân tích và ứng dụng cấu trúc dữ liệu trong hệ thống



Link list là một cấu trúc dữ liệu trong lập trình, được sử dụng để lưu trữ và quản lý dữ liệu theo cách linh hoạt hơn so với một mảng thông thường. Mỗi node trong link list chứa dữ liệu và một con trỏ trỏ tới node tiếp theo trong link list.

Cấu trúc dữ liệu link list có các loại chính như link list đơn, link list đôi và link list liên kết ngẫu nhiên.

Cấu trúc node trong link list đơn bao gồm hai thành phần chính: dữ liệu và con trỏ. Dữ liệu là thông tin mà cần lưu trữ trong mỗi node. Con trỏ là một biến đặc biệt, có chức năng trỏ tới node tiếp theo trong link list. Đối với node cuối cùng của link list, con trỏ sẽ trỏ tới null để đánh dấu kết thúc của link list.

Độ phức tạp của các thao tác trên link list đơn phụ thuộc vào loại thao tác và kích thước của link list. Dưới đây là độ phức tạp trung bình của các thao tác cơ bản trên link list đơn:

Thêm một phần tử vào đầu link list (insertion at the beginning): O(1)

Thêm một phần tử vào cuối link list (insertion at the end): O(n)

Thêm một phần tử vào một vị trí cụ thể trong link list (insertion at a specific position): O(n) Xóa một phần tử khỏi link list: O(n)

Truy cập/tìm kiếm một phần tử trong link list: O(n)

Sắp xếp link list (ví dụ: sắp xếp theo thứ tự tăng dần): O(n^2) hoặc O(nlogn) tùy thuộc vào thuật toán sắp xếp được sử dụng.

Điểm chính của link list là khả năng thêm và xóa phần tử một cách linh hoạt, tuy nhiên, việc truy cập phần tử bất kỳ trong link list có thể mất nhiều thời gian hơn so với mảng thông thường. Điều này đòi hỏi phải lặp qua các node từ đầu đến vị trí cần truy cập, dẫn đến độ phức tạp O(n).

Do đó, link list thường được sử dụng trong các trường hợp cần sự linh hoạt trong việc chèn và xóa phần tử, mà không yêu cầu truy cập ngẫu nhiên nhanh chóng.

Tuy nhiên, các thay đổi và tìm kiếm phần tử trong link list có thể tốn nhiều thời gian hơn so với mảng thông thường.

**Thuật toán:**

Thuật toán tìm kiếm tuần tự (**Linear Search**)

▪ Là phương pháp tìm kiếm một phần tử cho trước trong một danh sách

bằng cách duyệt lần lượt từng phần từ của danh sách đó đến khi nào tìm

được giá trị mong muốn hay đã duyệt hết qua hết danh sách.

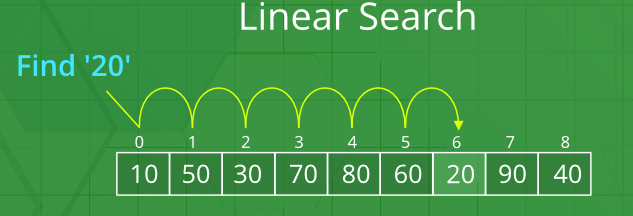
▪ Ý tưởng:

➢ Bắt đầu từ bản ghi đầu tiên của mảng, duyệt từ đầu mảng đến cuối

mảng với x.

➢ Nếu phần tử đang duyệt bằng x thì trả về vị trí.

➢ Nếu không tìm thấy phần từ nào khi đã duyệt hết thì trả về -1.



Độ phức tạp thời gian: O(n) khi phần tử tìm kiếm nằm cuối danh sách

hoặc không có trong danh sách.

Thời gian chạy tốt nhất: O(1) khi phần tử cần tìm nằm ngay đầu danh

Sách.

Độ phức tạp không gian: O(n).

**Thuật toán tìm kiếm tuần tự (Selection Sort)**

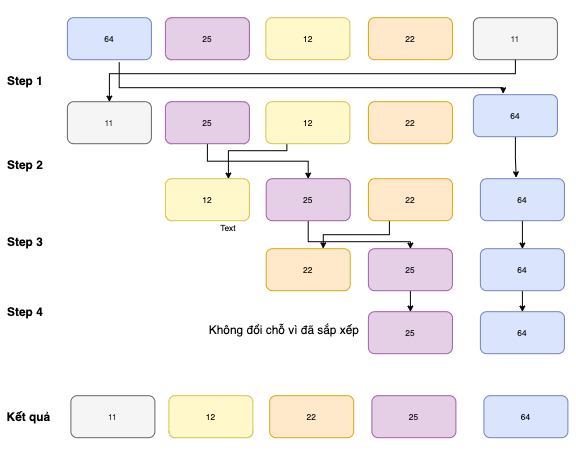
Thuật toán **Selection Sort** sắp xếp một mảng bằng cách liên tục tìm phần tử nhỏ nhất (xét theo thứ tự tăng dần) từ phần chưa được sắp xếp và đặt nó ở đầu. Thuật toán duy trì hai mảng con trong một mảng nhất định.

1.Mảng con đã được sắp xếp.

2.Mảng con còn lại chưa được sắp xếp.

Trong mỗi lần lặp lại sắp xếp lựa chọn, phần tử tối thiểu (xét theo thứ tự tăng dần) từ mảng con chưa được sắp xếp sẽ được chọn và chuyển đến mảng con đã sắp xếp.

Dưới đây là sơ đồ khối theo từng bước thực hiện của thuật toán trên:



**Đánh giá độ phức tạp của thuật toán:**

**Độ phức tạp thuật toán:**

* Trường hợp tốt: O(n^2)
* Trung bình: O(n^2)
* Trường hợp xấu: O(n^2)

**4.3 Quản lý tổ chức dữ liệu đầu vào:**

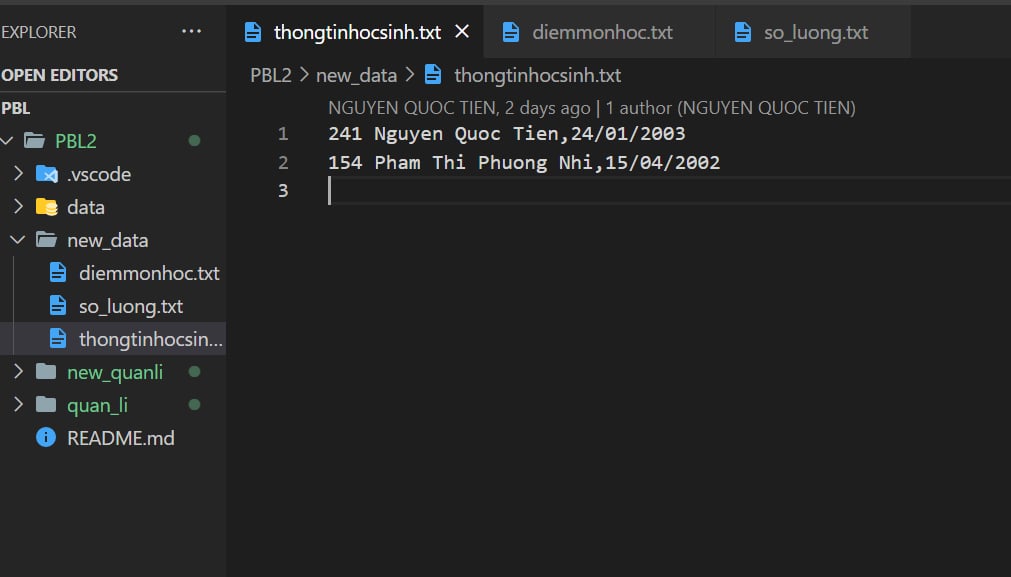
**4.3.1 file thongtinhocsinh.txt:**

Các thông tin bao gồm:

▪ ID: id của học sinh trong hệ thống

▪ Họ tên: Họ tên của học sinh

▪ Date: Ngày tháng năm sinh của học sinh

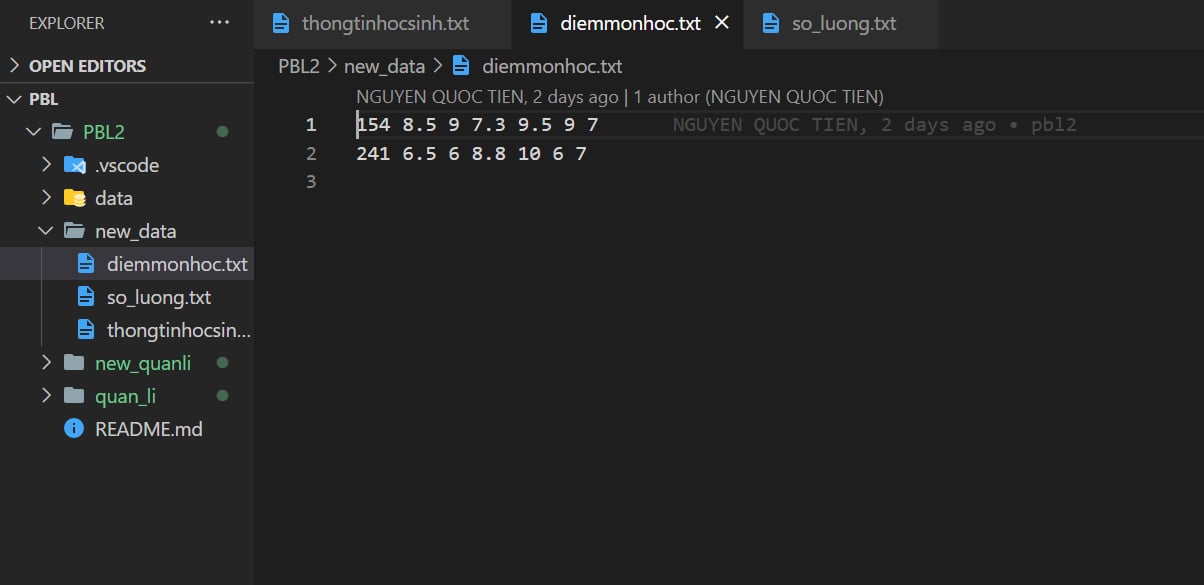


**4.3.2 file diemmonhoc.txt**

Các thông tin bao gồm:

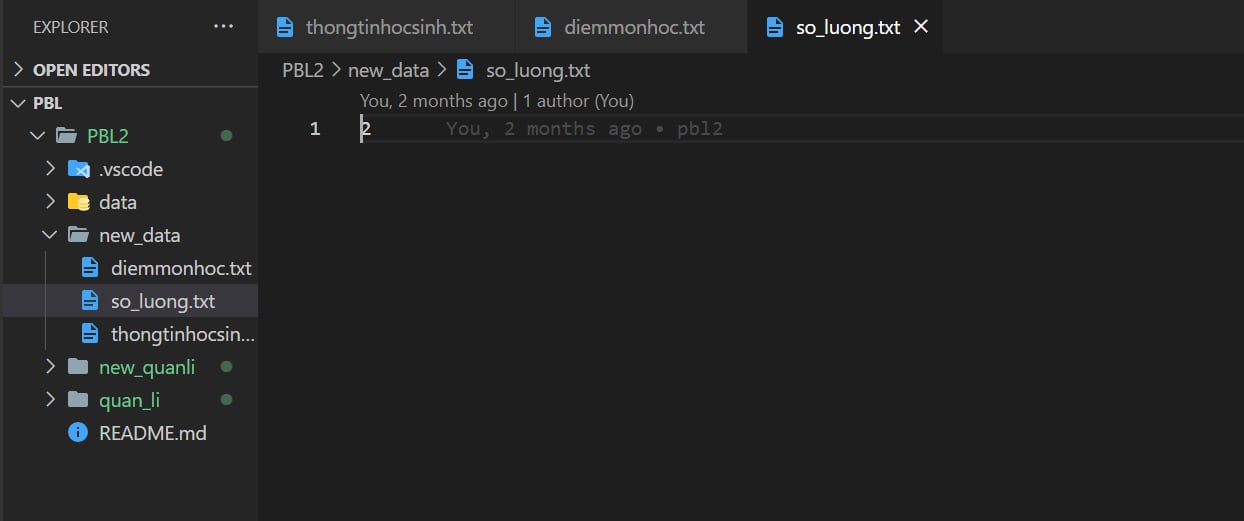
▪ Điểm: Điểm các môn học của học sinh

▪ ID: id của học sinh trong hệ thống



**4.3.3 file so\_luong.txt:**

Số lượng học sinh.



# PHÂN TÍCH HƯỚNG ĐỐI TƯỢNG VÀ TRIỂN KHAI HỆ THỐNG

## Cấu trúc hệ thống hướng đối tượng

|  |
| --- |
| **QuanLi** |
| Node \*Head; |
| QuanLi();  int dem\_so\_luong\_hoc\_sinh();  void them\_hoc\_sinh\_vao\_dau(HocSinh,MonHoc);  void them\_hoc\_sinh\_vao\_cuoi(HocSinh,MonHoc);  void them\_hoc\_sinh\_vao\_vi\_tri\_bat\_ki(HocSinh,MonHoc,int);  void xoa\_hoc\_sinh\_o\_vi\_tri\_dau\_tien();  void xoa\_hoc\_sinh\_o\_vi\_tri\_cuoi\_cung();  void xoa\_hoc\_sinh\_o\_vi\_tri\_bat\_ki(int);  void sua\_diem\_cho\_hoc\_sinh(int);  void tim\_kiem\_hoc\_sinh(int);  void HienThiDanhSach();  void sap\_xep\_diem();  ~QuanLi(); |

|  |
| --- |
| **Node** |
|  |
| HocSinh student; MonHoc subject;  Node\*next;Node();  Node(HocSinh,MonHoc);  ~Node(); |

|  |
| --- |
| **HocSinh** |
| string ten,ngaysinh; int id |
| HocSinh();  HocSinh(string,string,int);  string get\_ten();  string get\_ngaysinh();  int get\_id();  void nhapthongtin(int);  void set\_ten(string);  void set\_ngaysinh(string);  void set\_id(int);  void xuat\_thong\_tin();  Void docFile\_hocsinh(ifstream&);  ~HocSinh(); |

|  |
| --- |
| **MonHoc** |
| double toan,li,hoa,van,anh,tin;  double diemtrungbinh;  string hocluc;  int id; |
| MonHoc();  MonHoc(double,double,double,double,double,double,int);  void nhapdiem(int);double get\_toan();double get\_li();double get\_hoa();  double get\_van();double get\_anh();double get\_tin();  double get\_diem\_trung\_binh();string get\_hoc\_luc();int get\_id();  void set\_toan(double);void set\_li(double);void set\_hoa(double);  void set\_van(double);void set\_anh(double);void set\_tin(double);  void set\_diem\_trung\_binh(double);void set\_hoc\_luc(string);void set\_id(int);  void xuatdiem();void docFile\_MonHoc(ifstream&);  ~MonHoc(); |

**Chương trình có 4 class chính:**

* **Class HocSinh:**

**Các thuộc tính:**

+ ten, ngaysinh, id: họ tên, ngày tháng năm sinh, id của học sinh.

**Các phương thức:**

+HocSinh(): Constructor

+get\_ten(), get\_ngaysinh(), get\_ID(): lấy ra họ tên ngày tháng năm sinh và ID của học sinh.

+set\_ten(), set\_ngaysinh(), set\_ID(): sửa họ tên ngày tháng năm sinh và ID của học sinh.

+nhapthongtin(int), xuat\_thong\_tin(): nhập và xuất thông tin của học sinh.

+docfile\_hocsinh(ifstream&): đọc file thongtinhocsinh.txt.

+~HocSinh(): Destructor.

* **Class MonHoc:**

**Các thuộc tính:**

+id: id của học sinh.

+ toan, li, hoa, van, anh, tin: điểm toán lí hóa văn anh tin của học sinh.

+ diemtrungbinh: điểm trung bình của học sinh.

+ hocluc: học lực của học sinh.

**Các phương thức:**

+ MonHoc(): Constructor.

+ void nhapdiem(int): nhập điểm cho học sinh.

+get\_toan();get\_li();get\_hoa();get\_van();get\_anh();get\_tin();get\_diem\_trung\_binh();get\_hoc\_luc();get\_id(): lấy ra điểm Toán Lí Hóa Văn Anh Tin, điểm trung bình, học lực và ID của học sinh.

+ set\_toan(double); set\_li(double); set\_hoa(double); set\_van(double); void set\_anh(double);set\_tin(double);set\_diem\_trung\_binh(double);set\_hoc\_luc(string); set\_id(int): sửa điểm Toán Lí Hóa Văn Anh Tin, điểm trung bình, học lực và ID của học sinh.

+ xuatdiem(): xuất ra điểm của học sinh.

+docFile\_MonHoc(ifstream&): đọc file diemmonhoc.txt.

+ ~MonHoc(): Destructor.

* **Class Node:**

**Các thuộc tính:**

**Các phương thức:**

+HocSinh student; MonHoc subject: đối tượng student, subject.

+Node\*next: con trỏ next.

+Node(HocSinh,MonHoc): Constructor.

+~Node(): Destructor.

* **Class QuanLi:**

**Các thuộc tính:**

+ Node \*Head: Con trỏ Head.

**Các phương thức:**

+ QuanLi(): Constructor.

+ int dem\_so\_luong\_hoc\_sinh(): Hàm đếm số lượng học sinh.

+ void them\_hoc\_sinh\_vao\_dau(HocSinh,MonHoc);void them\_hoc\_sinh\_vao\_cuoi(HocSinh,MonHoc);void them\_hoc\_sinh\_vao\_vi\_tri\_bat\_ki(HocSinh,MonHoc,int): thêm học sinh vào đầu,cuối , bất kì vị trí nào trong danh sách.

+ void xoa\_hoc\_sinh\_o\_vi\_tri\_dau\_tien();void xoa\_hoc\_sinh\_o\_vi\_tri\_cuoi\_cung();void xoa\_hoc\_sinh\_o\_vi\_tri\_bat\_ki(int): xóa học sinh ở vị trí đầu,cuối cùng, vị trí bất kì trong danh sách.

+ void sua\_diem\_cho\_hoc\_sinh(int): sửa; điểm cho học sinh.

+ void tim\_kiem\_hoc\_sinh(int): tìm kiếm học sinh.

+ void HienThiDanhSach(): hiển thị danh sách.

+ void sap\_xep\_diem(): sắp xếp điểm.

+ ~QuanLi(): Destructor.

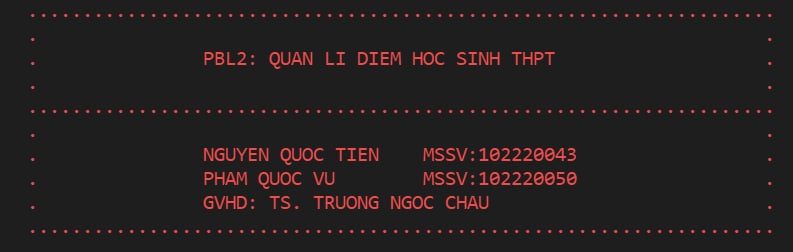
**Triển khai hệ thống:**

Chương trình được cấu thành từ các Folder và File con nhằm phục vụ các mục đích khác nhau. Khi chạy file main.cpp chương trình sẽ đảm nhiệm những chức năng riêng biệt ở cửa sổ **LUA CHON**. Mỗi option trên sẽ gọi tới những chức năng tương ứng để quản lí điểm học sinh. Các file diemmonhoc.txt, so\_luong.txt, thongtinhocsinh.txt được sử dụng để tương tác với các class MonHoc,HocSinh. Bên cạnh đó, nhằm mục đích tạo một cấu trúc dữ liệu tối ưu để lưu trữ thông tin (điểm, học sinh), ta cũng cần gọi đến Template Linked List được lưu trữ tại file node.h,node.cpp. Và đó là cách chương trình chúng em hoạt động.

## Kết quả

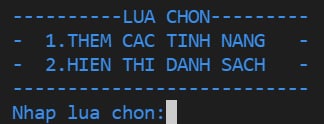
### Giao diện chính của chương trình

**Giới thiệu đề tài**

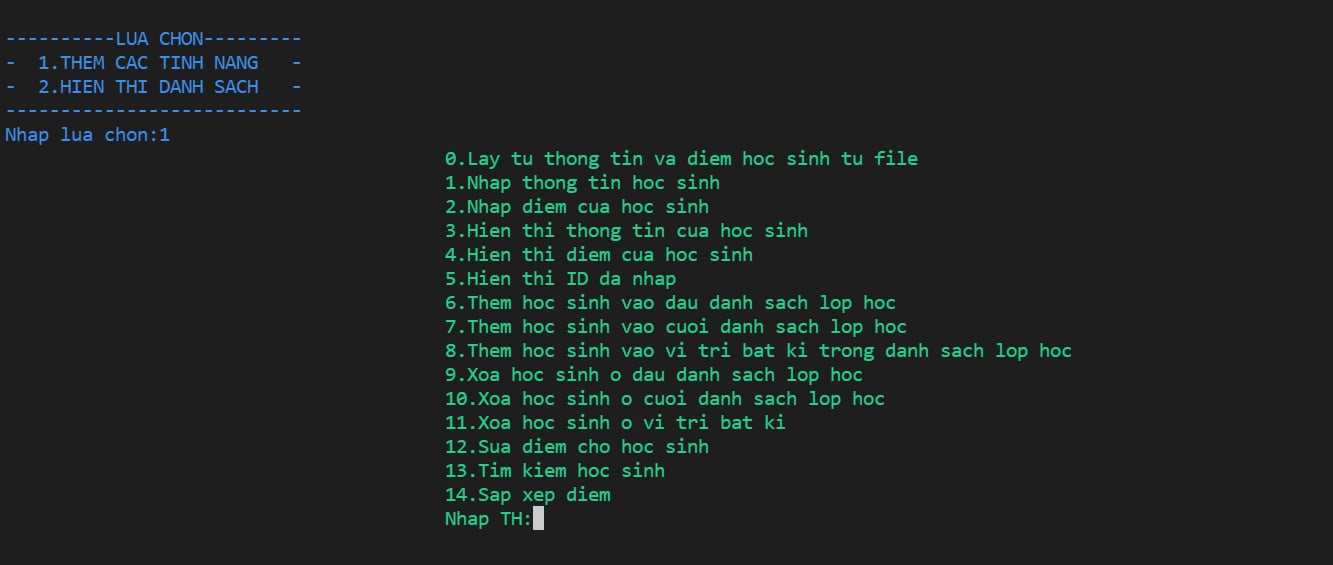


### Kết quả thực thi của chương trình

**Menu các Option :**



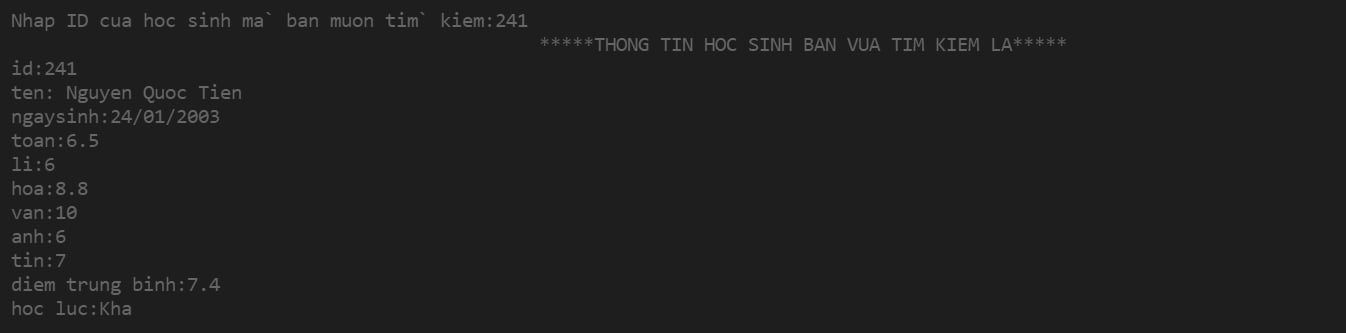
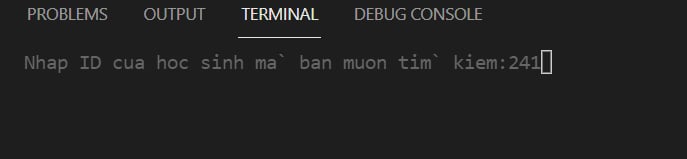
**Option 1:**



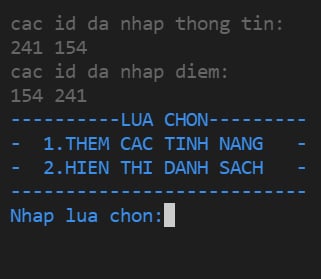
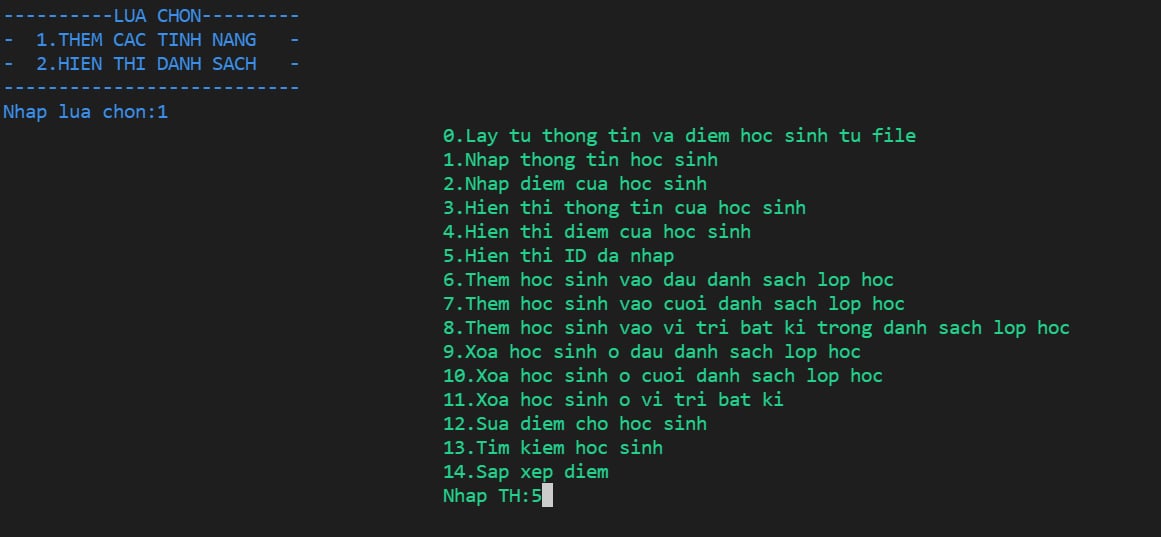
**Option 2 sau khi lấy dữ liệu từ file :**



**Tìm kiếm học sinh theo ID:**



**Hiển thị những ID đã nhập:**



### Nhận xét

Sau khi qua hàng loạt các chỉnh sửa và tổng hợp để hoàn thành đề tài trên, chúng em thấy rằng một phần mềm quản lí điểm học sinh hoàn thiện và tối ưu thì phải đáp ứng được các yêu cầu sau:

**Tính linh hoạt và tùy chỉnh:** Phần mềm quản lí điểm học sinh nên có tính linh hoạt và khả năng tùy chỉnh cao để phù hợp với các yêu cầu và quy trình quản lí của từng trường học. Nó cần cho phép quản lý viên và giáo viên điều chỉnh cách tính điểm, ghi nhận điểm và tạo các báo cáo phù hợp với nhu cầu cụ thể.

**Giao diện người dùng thân thiện:** Phần mềm quản lí điểm học sinh nên có giao diện người dùng thân thiện và dễ sử dụng. Điều này giúp giáo viên và quản lý trường học tiết kiệm thời gian và năng lượng trong việc ghi nhận và phân tích điểm số. Giao diện nên được thiết kế một cách rõ ràng và trực quan để người dùng dễ dàng tìm hiểu và sử dụng phần mềm một cách hiệu quả.

**Tính bảo mật và quyền riêng tư:** Phần mềm quản lí điểm học sinh chứa nhiều thông tin quan trọng về học sinh, do đó tính bảo mật và quyền riêng tư là rất quan trọng. Hệ thống phải đảm bảo rằng dữ liệu đươc bảo vệ một cách an toàn và chỉ có những người được ủy quyền mới có thể truy cập và thực hiện thay đổi trong hệ thống. Các biện pháp bảo mật như mã hóa dữ liệu và chứng thực người dùng nên được áp dụng.

**Khả năng tạo báo cáo và phân tích:** Phần mềm quản lí điểm học sinh nên cung cấp khả năng tạo báo cáo và phân tích kết quả học tập. Việc tạo ra các báo cáo chi tiết và biểu đồ phân tích giúp giáo viên, quản lý trường học và phụ huynh có cái nhìn tổng quan về tiến trình học tập và đánh giá chất lượng giáo dục. Những thông tin này có thể giúp điều chỉnh phương pháp giảng dạy, hỗ trợ học sinh và phụ huynh trong việc theo dõi tiến bộ và đưa ra quyết định phù hợp.

**Tích hợp với hệ thống khác:** Phần mềm quản lí điểm học sinh nên có khả năng tích hợp với các hệ thống khác trong trường học, chẳng hạn như hệ thống quản lí sinh viên hay hệ thống quản lý lớp học. Điều này giúp tạo sự liên kết và thông tin liên tục giữa các phần mềm và hệ thống trong trường học, giúp rút ngắn thời gian và công sức trong quá trình quản lí và gia tự dữ liệu.

# KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

## Kết luận

Phần mềm quản lí điểm học sinh là một công cụ quan trọng trong quá trình quản lí và ghi nhận điểm học tập của học sinh. Nó mang lại nhiều lợi ích và tiện ích cho giáo viên, quản lý trường học và phụ huynh. Tuy nhiên, để đáp ứng được nhu cầu đa dạng của từng trường học, phần mềm này cần có tính linh hoạt, tùy chỉnh và giao diện thân thiện. Đồng thời, tính bảo mật và quyền riêng tư là yếu tố quan trọng phải được chú ý trong quá trình phát triển phần mềm quản lí điểm học sinh.

## Hướng phát triển

**Tăng tính tương tác:** Phát triển một giao diện người dùng tương tác cao, cho phép học sinh và phụ huynh truy cập vào phần mềm quản lí điểm học sinh. Người dùng có thể tra cứu điểm số, theo dõi tiến trình học tập và nhận thông báo về kết quả qua các kênh truyền thông khác nhau. Tích hợp trên nền tảng di động: Xây dựng ứng dụng di động để hỗ trợ việc quản lí điểm học sinh. Điều này giúp giáo viên, quản lý trường học và phụ huynh có thể truy cập và cập nhật thông tin từ mọi nơi và mọi thiết bị di động.

**Phân tích dữ liệu học tập:** Mở rộng khả năng phân tích dữ liệu để cung cấp thông tin sâu hơn về tiến trình học tập và hiệu suất của học sinh. Ví dụ, phân tích đánh giá kỹ năng, khả năng và khuyết điểm của học sinh để giúp xác định các phương pháp giảng dạy và hỗ trợ tốt hơn.

**Kết nối với hệ thống quản lý lớp học và trường học:** Phát triển tính năng kết nối dữ liệu với các hệ thống khác trong trường học như hệ thống quản lí lớp học, trang web trường học hay hệ thống quản lí sinh viên. Điều này giúp thông tin được cập nhật liên tục và truy xuất một cách thuận tiện trong môi trường trường học.

**Tích hợp trí tuệ nhân tạo và học máy:** Áp dụng trí tuệ nhân tạo và học máy để tự động phân loại và đánh giá điểm học sinh, giúp giảm công việc thủ công của giáo viên và tăng tính chính xác trong quá trình ghi nhận và đánh giá điểm học tập.

**Tích hợp tính năng gửi tin nhắn và thông báo tự động:** Phát triển tính năng gửi tin nhắn và thông báo tự động để thông báo kết quả học tập, nhắc nhở về các sự kiện liên quan đến học tập và cung cấp thông tin mới nhất về học sinh cho phụ huynh và học sinh.

Đơn giản hóa quy trình nhập liệu: Tối ưu hóa quá trình nhập liệu và ghi nhận điểm học tập, đảm bảo tính chính xác và giảm thiểu sự mất mát thông tin. Áp dụng các công nghệ như quét mã vạch hoặc hỗ trợ nhập liệu tự động để giảm thiểu công việc thủ công và tăng hiệu suất.

TÀI LIỆU THAM KHẢO

[1] Phan Chí Tùng, *Cấu trúc dữ liệu*, Khoa Công nghệ thông tin trường Đại học Bách khoa Đà Nẵng, năm 2018.

[2] Lê Thị Mỹ Hạnh, Giáo trình môn học Lập trình hướng đối tượng, 2002.

[3] LinkedList, Linked List Data Structure – GeeksforGeeks, https://viblo.asia/p/tim-hieu-ve-hash-table-RnB5p0gY5PG

[4] Template in C++, Templates in C++ with Examples – GeeksforGeeks, https://www.geeksforgeeks.org/templates-cpp/

[5] Vector in C++, Vector in C++ STL – GeeksforGeeks, https://www.geeksforgeeks.org/vector-in-cpp-stl/

[6] Set và Map, Set và Map trong C++ (viblo.asia),https://viblo.asia/p/set-va-map-trong-c-djeZ1jPJlWz