

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA CÔNG NGHỆ THÔNG TIN



BÁO CÁO BÀI TẬP VỀ NHÀ
MÔN HỌC: PHƯƠNG PHÁP LẬP TRÌNH
HƯỚNG ĐỐI TƯỢNG
STUDENT LIST MANAGEMENT

Lớp: 21_4

Giảng viên: Lê Tuấn Thu

| Nhóm sinh viên thực hiện |

Nguyễn Huy Hoàng – 21120458

Hoàng Thị Khôn – 21120485

Thành phố Hồ Chí Minh, tháng 3 năm 2023

❖ CÁC THƯ VIỆN, LỚP VÀ HÀM CẦN THIẾT ĐỂ THỰC THI CHƯƠNG TRÌNH

➤ Thư viện

- Thư viện **fstream**: sử dụng để đọc / ghi file
- Thư viện **cstring**: sử dụng kiểu dữ liệu `char*` và dùng hàm `getline()`
- Thư viện **ctime**: sử dụng để tính ngày hiện tại
- **Pragma warning(disable: 1996)**: khắc phục lỗi cảnh báo compiler khi sử dụng `ctime`

➤ Lớp

- **Lớp Student:**
 - Dùng để lưu trữ thông tin của một sinh viên, bao gồm họ và tên, mã số sinh viên, ngày sinh, điểm số
 - Cung cấp các phương thức để đọc và ghi thông tin của một sinh viên, tính điểm trung bình, xếp loại sinh viên dựa trên điểm trung bình
 - Là thành phần của lớp `StudentList` dùng để quản lý danh sách sinh viên
- **Lớp StudentList:**
 - Dùng để lưu trữ và quản lý thông tin của danh sách sinh viên
 - Cung cấp các phương thức để đọc, ghi danh sách sinh viên, tính điểm trung bình của danh sách, thêm sinh viên vào danh sách, tìm ra những sinh viên có cùng ngày sinh trong ngày hiện tại,...

❖ CHỨC NĂNG CHÍNH CỦA TỪNG HÀM

➤ Các hàm trong lớp Student:

• Các hàm dựng, hàm hủy

`Student()`:

- Khởi tạo vùng nhớ con trỏ kiểu `char` `fullName` để lưu họ tên với kích thước tối đa 100.
- Khởi tạo vùng nhớ con trỏ kiểu `char` `id` để lưu mã số sinh viên với kích thước tối đa 10.
- Khởi tạo vùng nhớ con trỏ kiểu `char` `dateOfBirth` để lưu ngày tháng năm sinh với kích thước tối đa 30.
- Khởi tạo điểm bài tập, điểm giữa kỳ, điểm cuối kỳ bằng 0.

`Student(const Student& s):`

- Khởi tạo vùng nhớ cho fullName, id, dateOfBirth với kích thước tối đa lần lượt 100, 10, 30.
- Sao chép thông tin từ Student s sang con trỏ this.

~Student(): Hủy các vùng nhớ fullName, id, dateOfBirth.

- **Hàm nhập / xuất thông tin sinh viên**

void inputInfo(): Nhập thông tin sinh viên từ bàn phím có họ tên, id, ngày tháng năm sinh, điểm bài tập, điểm giữa kỳ, điểm cuối kỳ.

void printInfo() **const**: Xuất thông tin sinh viên với họ tên, id, ngày tháng năm sinh, điểm bài tập, điểm giữa kỳ, điểm cuối kỳ và điểm trung bình.

- **Hàm tính điểm trung bình của sinh viên, trả về ngày sinh của sinh viên**

double getAvgMark() **const**: Hàm lấy điểm trung bình trả về double.

const char* getDateOfBirth() **const**: Hàm lấy ngày tháng năm sinh trả về char*

- **Các hàm đọc / ghi file sinh viên**

void writeFile(ofstream& fileOut) **const**: Hàm ghi thông tin vào file với đối số fileOut: **ofstream**

- Ghi thông tin sinh viên từ fullName, id, dateOfBirth, marks[0], marks[1], marks[2], getAvgMarks() lần lượt họ tên, id, ngày tháng năm sinh, điểm bài tập, điểm giữa kỳ, điểm cuối kỳ và điểm trung bình.

void readFile(ifstream& fileIn): Hàm đọc thông tin sinh viên từ file ra với đối số fileIn: **ifstream**

- Đọc thông tin sinh viên lần lượt họ tên lưu vào fullName, id lưu vào id, ngày tháng năm sinh lưu vào dateOfBirth, điểm bài tập lưu vào marks[0], điểm giữa kỳ marks[1], điểm cuối kỳ marks[2].

- **Hàm xếp loại sinh viên**

string getRate() **const**: Hàm lấy xếp loại sinh viên trả về string.

- Đầu tiên, khai báo một biến rate kiểu string để lưu xếp loại sinh viên.
- Gọi hàm getAvgMarks() để lấy điểm trung bình vào biến avg.
- Kiểm tra các trường hợp avg gán giá trị xếp loại tương ứng vào biến rate.
- Cuối cùng, trả về rate.

➤ Các hàm trong lớp StudentList:

- **Các hàm dựng, hàm hủy**

`StudentList()`: Khởi tạo danh sách sinh viên với con trỏ list trỏ tới NULL và kích thước của danh sách là 0

`~StudentList()`: Giải phóng vùng nhớ đã cấp phát

- **Hàm nhập / xuất danh sách sinh viên**

`void inputList()`: Nhập thông tin cho danh sách sinh viên từ bàn phím, bao gồm số lượng sinh viên và thông tin của từng sinh viên

`void printList()`: In ra màn hình danh sách sinh viên

- **Hàm đọc/ ghi danh sách sinh viên**

`void writeList(const char* fileName) const:`

- Mở tập tin fileName để ghi dữ liệu thông qua đối tượng ofstream
- Lặp qua danh sách sinh viên và gọi hàm writeFile() đã viết ở lớp Student để ghi thông tin của từng sinh viên vào tập tin

`void readList(const char* fileName)`

- Mở tệp đầu vào với cú pháp ifstream fileIn(fileName)
- Đọc số lượng sinh viên được lưu trong tệp với fileIn>>size
- Dùng vòng lặp để đọc thông tin của từng sinh viên thông qua hàm readFile của lớp Student, sau đó lưu vào mảng list để đọc thông tin của danh sách sinh viên

- **Hàm tính điểm trung bình của danh sách và ghi danh sách sinh viên có điểm trung bình nhỏ hơn điểm trung bình của danh sách vào file**

`double getSumList() const:` Lặp từ đầu đến cuối danh sách để lấy ra tổng điểm trung bình của danh sách

`double getAvgList() const:` Lấy tổng điểm trung bình đã trả về ở hàm getSumList() chia cho số lượng sinh viên → Điểm trung bình của danh sách

`void writeListLowerAvg(const char* fileName) const:`

- Dùng vòng lặp chạy từ đầu đến cuối danh sách để kiểm tra và so sánh điểm trung bình của từng sinh viên với điểm trung bình của danh sách

- Nếu điểm trung bình của sinh viên nhỏ hơn điểm trung bình của danh sách thì ghi thông tin của các sinh viên này vào file bằng cách gọi đến hàm `writeFile` đã viết ở lớp `Student`

- **Hàm thêm sinh viên vào danh sách**

`void addNewStudent():`

- Tạo ra một đối tượng sinh viên mới bằng cách gọi hàm `inputInfo()`
- Khởi tạo một vùng nhớ mới có kích thước `size + 1` để lưu trữ danh sách sinh viên mới
- Sử dụng vòng lặp `for` để sao chép toàn bộ danh sách sinh viên ở mảng cũ vào mảng mới `newList`. Sau đó thêm sinh viên mới vào cuối mảng `newList` và tăng biến `size` lên 1
- Giải phóng bộ nhớ của danh sách cũ bằng cú pháp `delete[]` rồi gán địa chỉ của mảng mới `newList` cho con trỏ `list` để cập nhật sinh viên mới vào danh sách
- Dùng hàm `writeList()` để ghi thông tin sinh viên mới vào danh sách

- **Hàm in ra danh sách nhưng sinh viên có cùng ngày sinh nhật ở ngày hiện tại**

`void studentWithSameBirthday() const;`

- Dùng hàm `time()` để lấy thời gian hiện tại và chuyển đổi sang định dạng 'tm' với các thành phần ngày, tháng, năm lưu trữ trong các biến `'currentDay'`, `'currentMonth'`, `'currentYear'`
- Lặp qua danh sách sinh viên và so sánh ngày sinh của từng sinh viên với ngày hiện tại, nếu có sinh viên có ngày sinh giống với ngày hiện tại thì in ra danh sách sinh viên, đặt cờ "found" thành `true`. Nếu không có sinh viên nào có ngày sinh giống với ngày hiện tại thì in ra thông báo không tìm thấy, thay đổi cờ "found" thành `false`

- **Hàm in ra danh sách nhưng sinh viên có thông tin xếp loại**

`void printListWithGrade() const:` dùng vòng lặp `for` để duyệt qua danh sách sinh viên và in ra thông tin của từng sinh viên, gọi lại hàm `getRate()` đã viết ở lớp `Student` để bổ sung thêm thông tin xếp loại của sinh viên