BỘ THÔNG TIN VÀ TRUYỀN THÔNG

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG CƠ SỞ TẠI TP.HCM

**BÁO CÁO Học Phần: Cấu Trúc Dữ Liệu & Giải Thuật**

**Đề tài: 13/ Trung tâm ngoại ngữ**

GVHD: Lê Văn Hạnh

SVTH: Nguyễn Quang Chính \_ N19DCAT011 \_ D19CQAT01-N

**Phần 1: Giới thiệu về mục tiêu và nội dung của đề tài, các yêu cầu cần thực hiện**

* Đề tài là một chương trình của một trung tâm ngoại ngữ dùng để quản lý các cấp lớp, các lớp, các học viên và điểm của các học viên, và các chức năng thống kê hiện xuất danh sach từng phần.
* Các yêu cần cần được thực hiện:

- Về cấp lớp: thêm / xóa / hiệu chỉnh thông tin cấp lớp.

- Về lớp: thêm / xóa / hiệu chỉnh thông tin lớp.

- Về học viên: thêm / xóa / hiệu chỉnh thông tin học viên.

- Về điểm: thêm / xóa / hiệu chỉnh thông tin về điểm của học viên.

- In danh sách học viên theo lớp.

- Tìm và In kết quả học tập của học viên dựa trên mã số của học viên.

- Thống kê số lượng học viên theo từng cấp lớp.

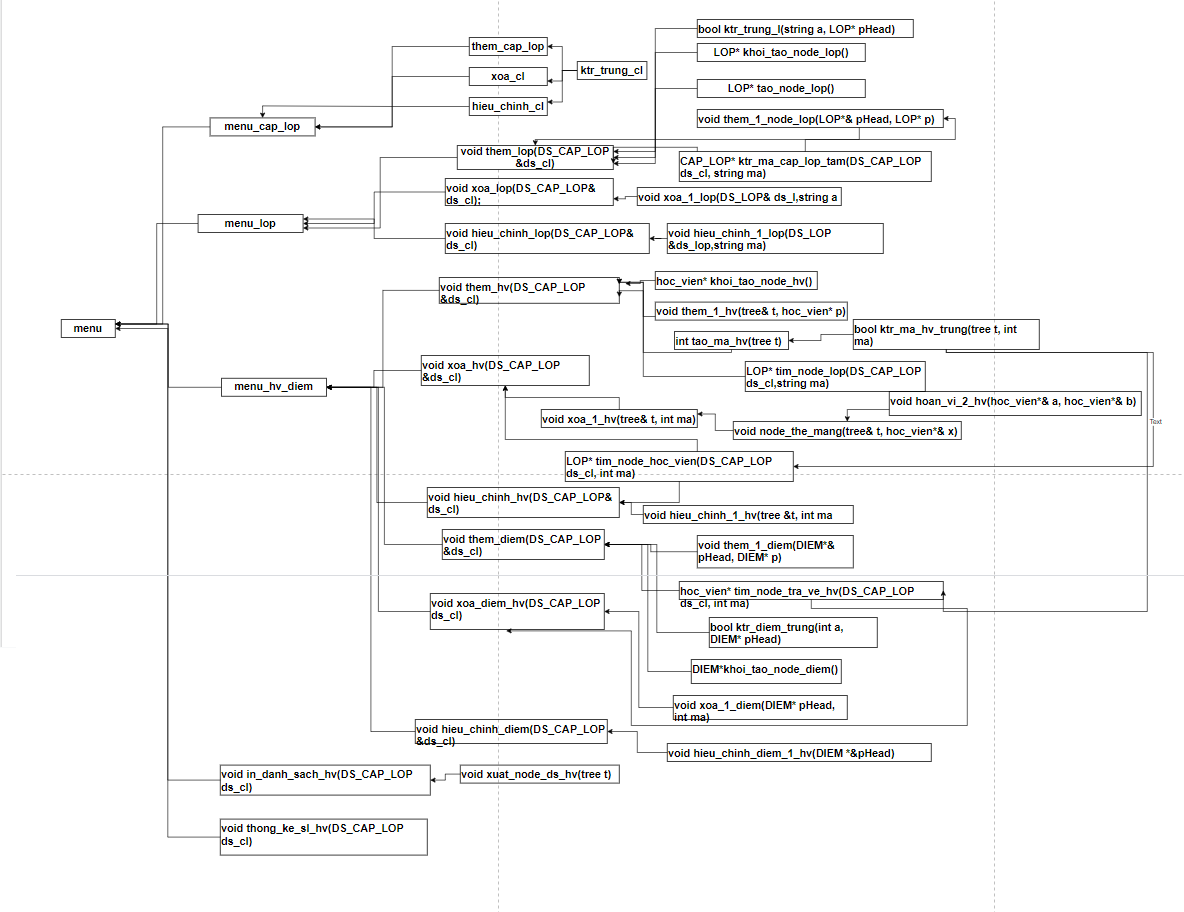
- In ra 10 lớp có số lượng học viên là đông nhất.

**Phần 2:**

**Mô tả các cấu trúc dữ liệu chính đã dùng trong chương trình:**

* Cấp lớp: danh sách tuyến tính là 1 mảng con trỏ, có con trỏ trỏ tới danh sach lớp tương ứng
* Lớp: danh sách liên kết đơn, có con trỏ trỏ tới danh sach hoc vien tương ứng
* HọcViên: cây nhị phân tìm kiếm, có con trỏ trỏ tới danh sach điểm của mỗi học viên
* Điểm: danh sách liên kết đơn.

**Vẽ sơ đồ liên kết (gọi hàm) giữa các hàm chính có trong chương trình**



**Giới thiệu các hàm chính mà đã cài đặt:**

* **void menu\_cap\_lop(DS\_CAP\_LOP &ds\_cl):** hàm này dùng để truy cập menu cho phép chọn những thao tác của cấp lớp với tham chiếu đưa vào là một biến danh sách cấp lớp. biến này sẽ là biến chính trong chương trình bởi vì có thay đổi thông tin nên phải truyền tham chiếu.
* void them\_cap\_lop(DS\_CAP\_LOP& ds\_cl); : là hàm thêm cấp lớp mới với biến tham chiếu ds\_cl để truy cập các thành phần của struct DS\_CAP\_LOP. Với giải thuật thêm phần tử trong danh sách tuyến tính.
* void xoa\_cl(DS\_CAP\_LOP& ds\_cl);: là hàm xóa cấp lớp. Với giải thuật xóa phần tử trong danh sách tuyến tính. sử dụng hàm int ktr\_trung\_cl(string a, DS\_CAP\_LOP ds\_cl); để trả về mã cấp lớp mới có trùng hay không
* void hieu\_chinh\_cl(DS\_CAP\_LOP& ds\_cl);: hàm điều chỉnh những thông tin của cấp lớp. Với giải thuật tìm và chỉnh sửa phần tử trong danh sách tuyến tính.
* void menu\_lop(DS\_CAP\_LOP &ds\_cl): hàm menu cho chọn những thao tác về lớp. những hàm liên quan tới cấp lớp sẽ được thực thi trong nó và thay đổi tham chiếu ds\_cl.
* void them\_lop(DS\_CAP\_LOP &ds\_cl); hàm dùng để thêm lớp vào một cấp lớp tương ứng. cho nhập mã cấp lớp. dùng hàm **ktr\_ma\_cap\_lop\_tam(ds\_cl,a),** hàm này dùng giải thuật tìm kiếm trong danh sách truyến tính, kết quả trả về bool**;** để ktr mã cấp lớp và trả về cấp lớp có mã tương ứng rồi tiến hành. **thêm một lớp** vào danh sách lớp của cấp lớp đó, sử dụng giải thuật thêm một node trong danh sách liên kết đơn. Với hàm **ktr\_trung\_l**  giúp kiểm tra mã lớp đó có trùng với lớp nào không.
* void xoa\_lop(DS\_CAP\_LOP &ds\_cl): cho nhập mã cấp lớp có chứa lớp cần xóa. Cho duyệt hết danh sách cấp lớp. Mỗi cấp lớp đi tìm mã lớp cần xóa. tìm ra node lớp có mã lớp cần xóa. sử dùng hàm xóa 1 lớp **xoa\_1\_lop(DS\_LOP& ds\_l,string a),** dùng giải thuật xóa một node trong danh sách liên kết đơn.
* void hieu\_chinh\_lop(DS\_CAP\_LOP& ds\_cl); : nhập mã lớp cân hiệu chỉnh với tam chiếu triền vào là **DS\_CAP\_LOP& ds\_cl** để tim ra node lớp có mã lớp trùng rồi dùng hàm void hieu\_chinh\_1\_lop(DS\_LOP &ds\_lop,string ma) dùng giải thuật tìm và chỉnh sửa một node trong danh sách liên kết đơn để tiến hành hiệu chỉnh node lớp.
* void menu\_hv\_diem(DS\_CAP\_LOP &ds\_cl);: hàm menu cho chọn những thao tác về học viên và điểm học viên;
* void them\_hv(DS\_CAP\_LOP &ds\_cl);: cần nhập mã lớp cần thêm học viên; tìm ra node LOP chứ lớp có mã lớp cần thêm bằng hàm **LOP\* tim\_node\_lop(DS\_CAP\_LOP ds\_cl,string ma)** học viên tiến hành thêm học viên vào. Với mã học viên được tạo tự động là số ngẫu nhiên từ 100-999. Sau đó thêm các thông tin của học viên dùng hàm **them\_1\_hv** để thêm vào, bằng giải thuật thêm node trong cây tìm kiếm nhị phân.
* void xoa\_hv(DS\_CAP\_LOP &ds\_cl): hàm xóa học viên. Nhập mã học viên cần xóa. dừng hàm LOP\* tim\_node\_hoc\_vien(DS\_CAP\_LOP ds\_cl, int ma) Lớp có mã học viên tương ứng trả về LOP\*. void xoa\_1\_hv(tree& t, int ma): dùng xóa 1 hoc viên sau khi đã tìm ra node học viên đó. Sử dụng giải thuật xóa node trong cây nhị phân tìm kiếm
* void hieu\_chinh\_hv(DS\_CAP\_LOP& ds\_cl): hiệu chỉnh điểm học viên cũng cần tìm mã học viên rồi dùng hàm LOP\* tim\_node\_hoc\_vien(DS\_CAP\_LOP ds\_cl, int ma) để tìm ra vị trí của học viên có mã tương ứng. Dùng hàm void hieu\_chinh\_1\_hv(tree &t, int ma) để tìm ra node có mã học viên tương ứng và xóa node trong cây nhị phân tìm kiếm.
* void them\_diem(DS\_CAP\_LOP &ds\_cl): thêm điểm cho học viên. Yêu cầu nhập mã học viên cần thêm điểm. dùng hàm hoc\_vien\* tim\_node\_tra\_ve\_hv(DS\_CAP\_LOP ds\_cl, int ma) tìm thấy node học viên có mã tương ứng trả về **hoc\_vien** với giải thuật duyệt tìm của danh sách tuyến tính và danh sách liên kết đơn. Khi thêm tên kỹ năng bool ktr\_diem\_trung(int a, DIEM\* pHead) cần kiếm tra kỹ năng đó có trùng trong danh sách điểm không. Dùng hàm void them\_1\_diem(DIEM\*& pHead, DIEM\* p) dùng giải thuật thêm 1 node trong danh sách liên kết đơn.
* void xoa\_diem\_hv(DS\_CAP\_LOP ds\_cl);:yêu cầu mã học viên cần xóa điểm, dùng hàm hoc\_vien\* tim\_node\_tra\_ve\_hv(DS\_CAP\_LOP ds\_cl, int ma) để tìm ra node hc viên cần xóa điểm. truy cập danh sách điểm của hc viên đó. Nhập tên kỹ năng để xóa. void xoa\_1\_diem(DIEM\* pHead, int ma) dùng hàm này để tìm ra kỹ năng và xóa điểm trong ds điểm với giải thuật xóa node trong danh sách liên kết đơn.
* void hieu\_chinh\_diem(DS\_CAP\_LOP &ds\_cl);: yêu cầu mã học viên cần xóa điểm, dùng hàm hoc\_vien\* tim\_node\_tra\_ve\_hv(DS\_CAP\_LOP ds\_cl, int ma) để tìm ra node hc viên cần hiệu chỉnh điểm. và dùng hàm void hieu\_chinh\_diem\_1\_hv(DIEM \*&pHead) với giải thuật tìm và sửa thông tin trong node của danh sách liên kê đơn.
* void xuat\_danh\_sach\_diem(DS\_CAP\_LOP ds\_cl): in danh sách điểm học viên dựa trên mã học viên. Yêu cầu nhập mã học viên cần xuất điểm. dùng hàm hoc\_vien\* tim\_node\_tra\_ve\_hv(DS\_CAP\_LOP ds\_cl, int ma) tìm thấy node học viên có mã tương ứng trả về **hoc\_vien** với giải thuật duyệt tìm của danh sách tuyến tính và danh sách liên kết đơn. Dùng giải thuật in danh sách trong danh sách liên kết đơn.
* void in\_danh\_sach\_hv(DS\_CAP\_LOP ds\_cl): in danh sách học viên theo lớp. Duyệt danh sách tuyến tính cap lớp và lấy được danh sách lớp trong từng cấp lớp rồi in ra danh sách học viên dùng hàm void xuat\_node\_ds\_hv(tree t) dùng giải thuật duyệt cây tìm kiếm nhị phân để in ra danh sách học viên.
* void thong\_ke\_sl\_hv(DS\_CAP\_LOP ds\_cl): thông kê số lượng học viên theo cấp lớp. Duyệt trong danh sách tuyến tính in ra thông cấp lớp. tiếp tục truy cập tới được danh sách lớp. duyệt danh sách lớp tới danh sách học viên truy cập thành phần số lượng học viên trong từng lớp cộng lại của cấp lớp.