**Báo cáo về kiến trúc phần mềm**

**I. Chương trình**

Chương trình được viết theo mô hình MVC ( Model, Controller, View)

1. **Model (Lớp Process)**

*Các hàm kiểm tra*

***- bool KiemTraSinhVien(string sbd, int &vitri);***

+ Hai tham số truyền vào

+ Dùng vòng lặp FOR để duyệt mảng kết hợp IF kiểm tra xem số báo danh truyền vào đã tồn tại chưa

+ Trường hợp SBD này có tồn tại trong mảng thì gán &vitri bằng với phần tử ở lần duyệt mảng đó, thao tác này quan trọng vì sau này gọi hàm Sua(); thì ta căn cứ vào vị trí của SBD có tồn tại đó để sửa thông tin sinh viên.

+ Nếu SBD có tồn tại trong mảng thì hàm này trả về TRUE ngược lại trả về FALSE

***- bool KiemTraSoBaoDanh(string sobaodanh);***

+ Hàm Kiểm tra chuỗi truyền vào (số báo danh) có tồn tại trong mảng hay không

+ Dùng vòng lặp FOR duyệt các phần tử của mảng, Mỗi lần duyệt dùng IF kiểm tra xem chuỗi truyền vào có trùng với số báo danh trong mảng hay không

+ Số báo danh trong mảng là kết quả khi gọi phương thức ***MangSV[i].getSBD()***

+ Nếu có tồn tại chuỗi này trong mảng thì hàm ***CheckSbd(string sobaodanh)*** trả về ***TRUE,*** ngược lại trả về ***FALSE***

***\* Các hàm sau tương tự***

***- bool KiemTraHoTen(string hoten);***

***- bool KiemTraTongDiem(double tong);***

***- bool KiemTraDiemHoa(double hoa);***

***- bool KiemTraDiemToan(double toan);***

***- bool KiemTraDiemLy(double ly);***

*Các hàm Tìm kiếm*

***- void TimKiemSoBaoDanh(string sbd);***

+ Hàm có tham số truyền vào là một chuỗi

+ Thực hiện gọi hàm ***KiemTraSoBaoDanh(sbd)*** với tham số tryền vào chính là tham số truyền vào của hàm ***TimKieSoBaoDanh()*** kết quả trả về lưu vào biến kiểu bool

+ Nếu kết quả (biến bool vừa lưu vào) thu được là **TRUE** Dùng IF-FOR lồng nhau để kiểm tra xem số báo danh truyền vao có tồn tại trong mảng hay không, thao tác này cũng tương tự như hàm kiểm tra bên trên

+ Với mỗi lần lặp, kiểm tra bằng IF nếu thỏa mãn thì gọi hàm **Xuat()**

+ Nếu kết quả (biến bool vừa lưu vào) thu được là **FALSE** thì thông báo số báo danh không tồn tại

\* ***Các hàm sau tương tự***

***- void TimKiemTongDiem(double tong);***

***- void TimKiemHoTen(string hoten);***

***- void TimKiemDiemToan(double diemToan);***

***- void TimKiemDiemLy(double diemLy);***

***- void TimKiemDiemHoa(double diemHoa);***

*Các hàm sắp xếp*

***- void SapXepDiemTong (SinhVien mangsv[], int n);***

+ Tham số truyền vào là đối tượng kiểu mảng của lớp SinhVien đã định nghĩa và biến kiểu nguyên hiểu như đây chính là kích thước mảng

+ Dùng giải thuật sắp xếp bằng so sánh trực tiếp: So sánh 2 phần tử kề nhau, nếu phần tử đứng trước (j) lớn hơn phần tử đứng sau (j+1) thì:

\* gán phần tử đứng trước vào biến tạm

\* gán phần tử đứng sau bằng phần tử đứng trước

\* gán phần tử đứng sau bằng biến tạm

+ Mảng sau khi được sắp xếp sẽ theo thứ tự tăng dần

+ Dùng vòng lặp FOR để Xuất mảng

\* ***Các hàm sau tương tự***

***- void SapXepSoBaoDanh(SinhVien mangsv[], int n);***

***- void SapXepAnphabe(SinhVien mangsv[], int n);***

*Các hàm khác*

***- void ThemSinhVien(int n);***

+ Gọi hàm NhapSinhVien();

+ Sau mỗi lần nhập tăng chỉ số mảng lên 1

***- void SuaSinhVien(string sbd);***

+ Sửa theo số báo danh người dùng nhập

+ Kiểm tra xem số báo danh có tồn tại hay không

+ Nếu tồn tại gọi hàm NhapSinhVien() để sửa

+ Nếu không tồn tại thì thông báo đến người dùng

***- void XoaSinhVien(string sbd);***

+ Xóa theo số báo danh người dùng nhập

+ Dùng kỹ thuật dồn mảng để xóa

+ Sau mỗi lần xóa giảm chỉ số mảng đi 1

***- void ThongKeTatCa();***

+ Thống kê toàn bộ

+ Thống kê theo điểm

+ Thống kê số lượng sinh viên đạt mức điểm nào đó: Dùng biến đếm kết hợp IF và FOR để Xuất mảng

***- void NhapSinhVienTuFile();***

+ Nhập danh sách sinh viên từ một file cho trước, người dùng nhập tên file sau đó hệ thống tự động nhập danh sách sinh viên.

+ Bên trong file danh sách sinh viên, các trường được nhập cách nhau bằng dấu cách hoặc dấu tab

***- void XuatSinhVien();***

+ Xuất danh sánh sinh viên đang có trong mảng ra file với tên do ngườ dùng nhập (lưu tại cùng thư mục với file chạy exe)

1. **Model (Lớp SinhVien)**

- Các thuộc tính Private: ***HoTen; SBD; DiaChi; GioiTinh; Toan; Ly; Hoa;***

+ Các thuộc tính lưu các giá trị người dùng nhập

+ Các thuộc tính này như là biến toàn cục

- ***Các phương thức Public:***

+ Các phương thức ***SET***: Set giá trị mỗi thuộc tính tương ứng với tham số truyền vào khi gọi hàm

+ Các phương thức ***GET***: Trả về các giá trị thuộc tính khi có gọi hàm (phương thức)

+ ***NhapSinhVien();*** Nhập thông tin sinh viên

+ ***XuatSinhVien();*** Xuất thông tin sinh viên

+ ***TongDiem();*** Tổng: toán + lý + hóa

1. **Controller**

Phương thức main đóng vai trò là trung gian khi gọi các phương thức hiển thị ở lớp Menu và gọi các phương thức xử lý từ Process và SinhVien

1. **View (Menu)**

Có các Phương thức hiển thị Menu chính, menu sắp xếp, menu tìm kiếm . . .để người dùng chọn