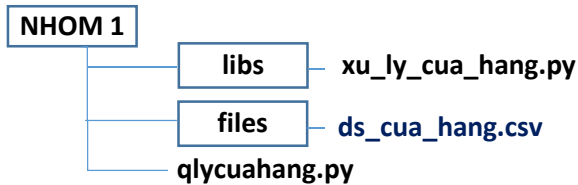


## Nhóm\_1

Để quản lý cửa hàng kinh doanh người ta lưu trữ thông tin của mỗi cửa hàng bao gồm: *Mã cửa hàng; tên cửa hàng; vốn đầu tư; doanh thu; tiền thuế.*

Trong đó: Mã cửa hàng; tên cửa hàng; vốn đầu tư; doanh thu được nhập vào từ bàn phím.

Tạo một project (thư mục) **NHOM01**, có cấu trúc như sau:



**I.** Viết chương trình quản lý cửa hàng version1 đặt tên là **glycuahang.py** thực hiện các chức năng sau đây:

- Tạo menu chọn chức năng cần thực hiện
- Mở file quản lý cửa hàng
- Nhập thông tin các cửa hàng từ bàn phím, không nhập trường thuế.
- Tính thuế biết: Biết thuế = 10% doanh thu với doanh nghiệp có số vốn đầu tư  $\geq$  50 triệu, còn lại thuế = 5% doanh thu.
- Lưu danh sách các cửa hàng với đầy đủ thông tin vào tệp **ds\_cua\_hang.csv** trong thư mục files.
- Sắp xếp các cửa hàng theo thứ tự giảm dần của doanh thu. In ra màn hình danh sách trước và sau khi sắp xếp.

**II.** Tổ chức viết lại chương trình ở phần I với nội dung như sau:

Trong thư mục **libs** viết module **xu\_ly\_cua\_hang.py** chứa các hàm (function) thực hiện các chức năng sau:

- Mở file quản lý cửa hàng (**cuahang.csv**)
- Thêm thông tin các cửa hàng (macuahang, tencuahang, vondautu, doanhthu, tienthue)
- In danh sách cửa hàng với đầy đủ thông tin.
- Lưu file quản lý cửa hàng (lưu vào thư mục files với tên dscuahang.csv)
- Tìm cửa hàng theo mã cửa hàng (ma\_ch).
- Xóa cửa hàng theo mã cửa hàng (ma\_ch).
- Thống kê doanh thu
- Lọc cửa hàng theo vốn đầu tư
- Cửa hàng có doanh thu cao nhất

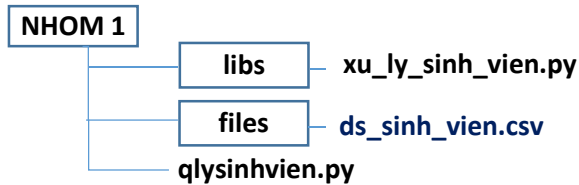
Viết lại file **glycuahang\_v2.py** với các chức năng cần thực hiện được gọi từ các function đã viết trong module **xu\_ly\_cua\_hang.py**

## Nhóm 2.

Đề quản lý quản lý sinh viên trong một lớp học với thông tin của một sinh viên gồm: mã sinh viên (masv: kiểu xâu), tên sinh viên (tensv: kiểu xâu), năm sinh (số nguyên), giới tính (kiểu logic bool: True là nam và False là nữ), điểm trung bình cả năm (tbcn: số thực), học bổng (số thực).

Trong đó: mã sinh viên, tên sinh viên, năm sinh, giới tính} được nhập vào từ bàn phím.

Tạo một project (thư mục) **NHOM02**, có cấu trúc như sau:



**I.** Viết chương trình quản lý sinh viên version1 đặt tên là **qlsinhvien.py** thực hiện các chức năng sau đây:

- Tạo menu chọn chức năng cần thực hiện
- Mở file danh sách sinh viên. (ds\_sinh\_vien.csv)
- Nhập thông tin sinh viên từ bàn phím, không nhập trường học bổng.
- Tính học bổng biết:
  - Nếu điểm trung bình cả năm  $\geq 9.0$  : Học bổng = 8 000 000
  - Nếu  $8.0 \leq$  điểm trung bình cả năm  $< 9.0$  : Học bổng = 5 000 000
  - Nếu  $7.0 \leq$  điểm trung bình cả năm  $< 8.0$  : Học bổng = 3 000 000
  - Nếu điểm trung bình cả năm  $< 7.0$  : Học bổng = 0
- Lưu danh sách các sinh viên với đầy đủ thông tin vào tệp **ds\_sinh\_vien.csv** trong thư mục files.
- Sắp xếp các sinh viên theo thứ tự giảm dần của điểm trung bình cả năm. In ra màn hình danh sách trước và sau khi sắp xếp.

**II.** Tổ chức viết lại chương trình ở phần I với nội dung như sau:

Trong thư mục **libs** viết module **xu\_ly\_sinh\_vien.py** chứa các hàm (function) thực hiện các chức năng sau:

- Mở file danh sách sinh viên (**ds\_sinh\_vien.csv**)
- Thêm thông tin sinh viên (masv, tensv, nam\_sinh, gioi\_tinh, tbcn, hoc\_bong)
- In danh sách các sinh viên với đầy đủ thông tin.
- Lưu file danh sách sinh viên (lưu vào thư mục files với tên **ds\_sinh\_vien.csv**)
- Tìm sinh viên theo mã sinh viên (masv).
- Xóa sinh viên theo mã sinh viên (masv).
- Thống kê sinh viên theo học bổng
- Lọc sinh viên theo điểm trung bình cả năm.
- Sinh viên có học bổng cao nhất

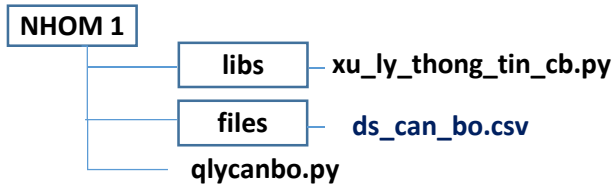
Viết lại thành file **qlsinhvien\_v2.py** với các chức năng cần thực hiện được gọi từ các function đã viết trong module **xu\_ly\_sinh\_vien.py**

### Nhóm 3

Để quản lý cán bộ trong một cơ quan với thông tin của mỗi cán bộ bao gồm: Mã cán bộ (macb: kiểu xâu), Tên cán bộ (tencb: kiểu xâu), Hệ số lương (hsl: số thực), Chức vụ (chucvu: kiểu xâu), Phụ cấp chức vụ (phucap: số thực), Lương (luong: số thực)

Trong đó: Mã cán bộ, Tên cán bộ, Hệ số lương, Chức vụ được nhập vào từ bàn phím.

Tạo một project (thư mục) **NHOM03**, có cấu trúc như sau:



**I.** Viết chương trình quản lý cán bộ version1 đặt tên là **glycanbo.py** thực hiện các chức năng sau đây:

- Tạo menu chọn chức năng cần thực hiện
- Mở file danh sách cán bộ
- Nhập thông tin cán bộ từ bàn phím, không nhập trường lương.
- Tính lương của từng cán bộ trong danh sách cán bộ, với  
Lương = hệ số lương \* 1 500 000 + phụ cấp chức vụ. Biết:
  - + Nếu chức vụ là giám đốc “Giam doc” thì phụ cấp là 10 000 000
  - + Nếu chức vụ là trưởng phòng “Truong phong” thì phụ cấp là 5 000 000
  - + còn lại là 2 000 000.
- Lưu danh sách các cán bộ với đầy đủ thông tin vào tệp **ds\_can\_bo.csv** trong thư mục files.
- Sắp xếp cán bộ theo chiều tăng dần của lương. In ra màn hình danh sách trước và sau khi sắp xếp.

**II.** Tổ chức viết lại chương trình ở phần I với nội dung như sau:

Trong thư mục **libs** viết module **xu\_ly\_thong\_tin\_cb.py** chứa các hàm (function) thực hiện các chức năng sau:

- Mở file danh sách cán bộ (**ds\_can\_bo.csv**)
- Thêm thông tin các cán bộ  
(macb, tencb, hsl, chucvu, phucap, luong)
- In danh sách cán bộ với đầy đủ thông tin.
- Lưu file danh sách cán bộ  
(lưu vào thư mục files với tên **ds\_can\_bo.csv**)
- Tính và hiển thị tổng lương của các cán bộ làm việc ở phòng “hành chính”
- Xóa cán bộ theo mã cán bộ (macb).
- Thêm vào danh sách một cán bộ với thông tin cán bộ nhập vào từ bàn phím sao cho vẫn đảm bảo thứ tự sắp xếp.

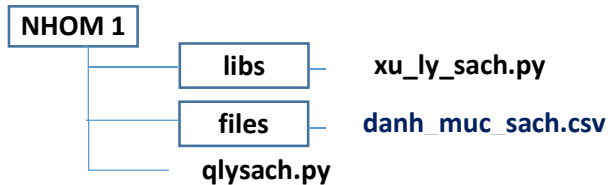
Viết lại file **gly\_can\_bo\_v2.py** với các chức năng cần thực hiện được gọi từ các function đã viết trong module **xu\_ly\_thong\_tin\_cb.py**

## Nhóm 4

Để quản lý lý sách của một nhà xuất bản với thông tin của mỗi cuốn sách bao gồm: *mã sách (MaS: kiểu xâu)* , *tên sách (tenS: kiểu xâu)*; *tác giả (tacgia: kiểu xâu)*; *năm xuất bản (namxb: số nguyên)*; *thành tiền (thanhtien: số thực)*; *thuế (thue: số thực)*; *tiền thực thu (thucthu: số thực)*

Trong đó {mã sách; tên sách; tác giả; năm XB; thành tiền} được nhập vào từ bàn phím.

Tạo một project (thư mục) **NHOM04**, có cấu trúc như sau:



**I.** Viết chương trình quản lý sách version1 đặt tên là **qlysach.py** thực hiện các chức năng sau đây:

- Tạo menu chọn chức năng cần thực hiện
- Mở file danh mục sách
- Nhập thông tin các cuốn sách từ bàn phím, không nhập trường thuế.
- Tính thuế cho các cuốn sách biết:  $\text{Biết thuế} = 10\% \times \text{thành tiền}$ .  
Tính thực thu với  $\text{thực thu} = \text{thành tiền} - \text{thuế}$
- Lưu danh sách các cuốn sách với đầy đủ thông tin vào tệp **danh\_muc\_sach.csv** trong thư mục files.
- Sắp xếp các quyển sách theo thứ tự giảm dần của năm xuất bản. In ra màn hình danh sách các cuốn sách trước và sau khi sắp xếp.

**II.** Tổ chức viết lại chương trình ở phần I với nội dung như sau:

Trong thư mục **libs** viết module **xu\_ly\_sach.py** chứa các hàm (function) thực hiện các chức năng sau:

- Mở file danh mục sách (**danh\_muc\_sach.csv**)
- Thêm thông tin các cuốn sách  
(*MaS, tenS, tacgia, namxb, thanhtien, thue, thucthu*)
- In danh mục sách với đầy đủ thông tin.
- Lưu file danh mục sách  
(lưu vào thư mục files với tên **danh\_muc\_sach.csv**)
- Tìm quyển sách theo mã sách (maS).
- Xóa quyển sách có mã là ms, với ms nhập từ bàn phím.
- Thống kê các cuốn sách có năm xuất bản = năm, với năm là một số nguyên được nhập từ bàn phím.
- Thêm vào danh mục sách một cuốn sách với thông tin nhập từ bàn phím.
- Quyển sách có tiền thực thu cao nhất

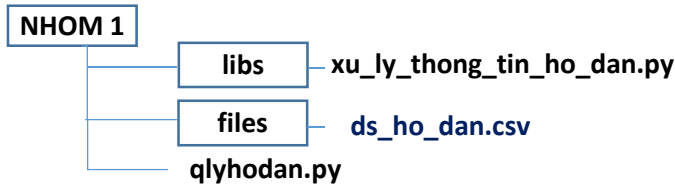
Viết lại file **qlysachv2.py** với các chức năng cần thực hiện được gọi từ các function đã viết trong module **xu\_ly\_sach.py**

## Nhóm 5

Để quản lý quản lý hộ dân trong một phường với thông tin của mỗi hộ bao gồm: mã hộ (kiểu xâu); tên chủ hộ (kiểu xâu); số thành viên (số nguyên); mức thu nhập (số thực); hộ nghèo (kiểu bool, true là hộ nghèo, false: không); trợ cấp (số thực).

Trong đó: mã hộ; tên chủ hộ; số thành viên; mức thu nhập được nhập vào từ bàn phím.

Tạo một project (thư mục) **NHOM05**, có cấu trúc như sau:



**I.** Viết chương trình quản lý cửa hàng version1 đặt tên là **qlyhodan.py** thực hiện các chức năng sau đây:

- Tạo menu chọn chức năng cần thực hiện
- Mở file **ds\_ho\_dan.csv**
- Nhập thông tin các cửa hộ dân từ bàn phím, không nhập trường trợ cấp.
- Tính trợ cấp cho các hộ biết:  
Nếu là hộ nghèo và số thành viên  $\geq 5$  thì trợ cấp là 1triệu/1 người  
Nếu là hộ nghèo và  $3 \leq$  số thành viên  $< 5$  thì trợ cấp là 800nghìn/1 người  
Nếu là hộ nghèo và  $1 \leq$  số thành viên  $< 3$  thì trợ cấp là 500nghìn/1 người  
Nếu không là hộ nghèo trợ cấp = 0
- Lưu danh sách các hộ dân với đầy đủ thông tin vào tệp **ds\_ho\_dan.csv** trong thư mục files.
- Sắp xếp các hộ dân theo thứ tự giảm dần của số thành viên. In ra màn hình danh sách trước và sau khi sắp xếp.

**II.** Tổ chức viết lại chương trình ở phần I với nội dung như sau:

Trong thư mục **libs** viết module **xu\_ly\_thong\_tin\_ho\_dan.py** chứa các hàm (function) thực hiện các chức năng sau:

- Mở file **ds\_ho\_dan.csv** (**ds\_ho\_dan.csv**)
- Thêm thông tin các hộ dân  
(maho, chuho, so\_tv, muc\_tn, ho\_ngheo, tro\_cap)
- Inra màn hình danh sách hộ dân với đầy với đầy đủ thông tin.
- Lưu file danh sách hộ dân  
(lưu vào thư mục files với tên ds\_ho\_dan.csv)
- In thông tin các hộ dân thu nhập nhỏ nhất.
- Xóa khỏi danh sách hộ dân, hộ có tên chủ hộ là ten, với ten nhập vào từ bàn phím
- Thống kê mức thu nhập
- Lọc hộ dân theo mức thu nhập
- Hộ dân có trợ cấp cao nhất

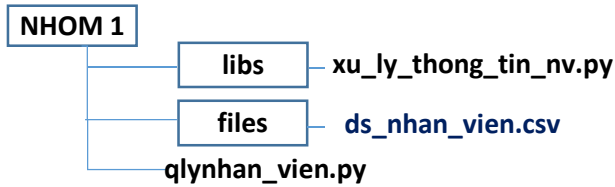
Viết lại file **qly\_ho\_dan\_v2.py** với các chức năng cần thực hiện được gọi từ các function đã viết trong module **xu\_ly\_thong\_tin\_ho\_dan.py**

## Nhóm 6

Để quản lý lương nhân viên trong một công ty với thông tin của mỗi nhân viên bao gồm: mã nhân viên (manv: kiểu xâu); tên nhân viên (tennv: kiểu xâu); năm công tác; lương cơ bản (luongcb: số thực); phụ cấp (phucap: số thực); thu nhập.

Trong đó mã nv; tên nv; lương cơ bản; phụ cấp được nhập vào từ bàn phím.

Tạo một project (thư mục) **NHOM06**, có cấu trúc như sau:



I. Viết chương trình quản lý cửa hàng version1 đặt tên là **qlynhanvien.py** thực hiện các chức năng sau đây:

- Tạo menu chọn chức năng cần thực hiện
- Mở file danh sách nhân viên
- Nhập thông tin nhân viên từ bàn phím, không nhập trường thu nhập.
- Tính thu nhập của nhân viên với thu nhập = lương cơ bản + phụ cấp  
Biết năm công tác  $\geq 20$ : phụ cấp = 10000000  
 $15 \leq$  Năm công tác  $< 20$ : phụ cấp = 8000000  
 $10 \leq$  Năm công tác  $< 15$ : phụ cấp = 5000000  
 $5 \leq$  Năm công tác  $< 10$ : phụ cấp = 3000000  
Năm công tác  $< 5$ : phụ cấp = 1000000
- Lưu danh sách các nhân viên với đầy đủ thông tin vào tệp **ds\_nhan\_vien.csv** trong thư mục files.
- Sắp xếp các nhân viên theo thứ tự tăng dần theo hệ số lương. In ra màn hình danh sách trước và sau khi sắp xếp.

II. Tổ chức viết lại chương trình ở phần I với nội dung như sau:

Trong thư mục **libs** viết module **xu\_ly\_thong\_tin\_nv.py** chứa các hàm (function) thực hiện các chức năng sau:

- Mở file danh sách nhân viên (**ds\_nhan\_vien.csv**)
- Thêm thông tin các nhân viên (manv, tennhanvien, namcongtac, luong\_cb, phucap, thunhap)
- In danh sách nhân viên với đầy đủ thông tin.
- Lưu file quản lý nhân viên (lưu vào thư mục files với tên **qlynhan\_vien.csv**)
- Tìm nhân viên có thu nhập cao nhất.
- Xóa nhân viên có mã là ma, với ma nhập vào từ bàn phím.
- Lọc nhân viên theo thu nhập

Sắp xếp danh sách nhân viên theo thứ tự tăng dần về hệ số lương. In ds nhân viên sau sắp xếp

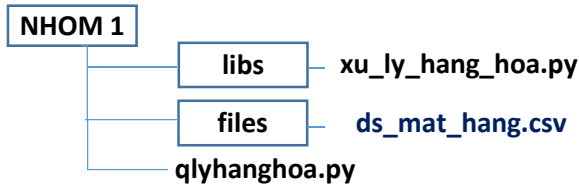
Viết lại file **qlynhanvien\_v2.py** với các chức năng cần thực hiện được gọi từ các function đã viết trong module **xu\_ly\_thong\_tin\_nv.py**

## Nhóm 7

Để quản lý lý hàng hóa trong một siêu thị với thông tin của mỗi mặt hàng bao gồm: mã hàng (kiểu xâu); tên hàng (kiểu xâu); đơn vị tính (kiểu xâu); đơn giá (số thực); số lượng (số nguyên); thành tiền (số thực);

Trong đó: mã hàng; tên hàng; đơn vị tính; đơn giá; số lượng được nhập vào từ bàn phím.

Tạo một project (thư mục) **NHOM07**, có cấu trúc như sau:



**I.** Viết chương trình quản lý cửa hàng version1 đặt tên là **qlyhanghoa.py** thực hiện các chức năng sau đây:

- Tạo menu chọn chức năng cần thực hiện
- Mở file **ds\_mat\_hang.csv**
- Nhập thông tin các mặt hàng từ bàn phím, không nhập trường “thành tiền”
- Tính thuế biết: Biết thuế = 10% doanh thu với doanh nghiệp có số vốn đầu tư >= 50 triệu, còn lại thuế = 5% doanh thu.
- Lưu danh sách các mặt hàng với đầy đủ thông tin vào tệp **ds\_mat\_hang.csv** trong thư mục files.
- Sắp xếp các mặt hàng theo thứ tự giảm dần của số lượng. In ra màn hình danh sách trước và sau khi sắp xếp.

**II.** Tổ chức viết lại chương trình ở phần I với nội dung như sau:

Trong thư mục **libs** viết module **xu\_ly\_hang\_hoa.py** chứa các hàm (function) thực hiện các chức năng sau:

- Mở file danh sách mặt hàng (**ds\_mat\_hang.csv**)
- Nhập thông tin các mặt hàng từ bàn phím (mahang, tenhang, dvt, dongia, so\_luong, thanh tien)
- In danh sách hàng hóa với đầy đủ thông tin.
- Lưu file danh sách mặt hàng (lưu vào thư mục files với tên **ds\_mat\_hang.csv**)
- Thêm vào danh sách mặt hàng có thông tin mặt hàng được nhập vào từ bàn phím sao cho vẫn đảm bảo thứ tự sắp xếp
- Xóa các mặt hàng có đơn vị tính là dv. Biết dv là một xâu nhập vào từ bàn phím (Thùng, Hộp, Gói, ...)
- Lọc hàng hóa theo vốn đầu tư
- Cửa hàng có doanh thu cao nhất

Viết lại file **qlyhanghoa\_v2.py** với các chức năng cần thực hiện được gọi từ các function đã viết trong module **xu\_ly\_hang\_hoa.py**

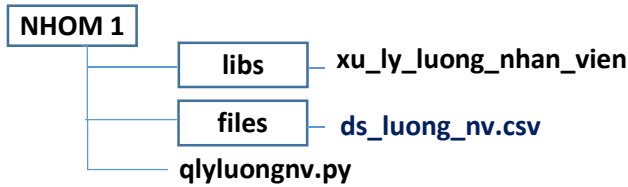


## Nhóm 8

Để quản lý lương nhân viên trong một phân xưởng với thông tin của mỗi nhân viên bao gồm: mã nv (kiểu xâu); tên nv (kiểu xâu); lương ngày (số thực); ngày công (số nguyên); lương tháng (số thực); thưởng;

Trong đó: mã nv; tên nv; lương ngày; ngày công được nhập vào từ bàn phím.

Tạo một project (thư mục) **NHOM08**, có cấu trúc như sau:



**I.** Viết chương trình quản lý lương nhân viên version1 đặt tên là **qlyluongnv.py** thực hiện các chức năng sau đây:

- Tạo menu chọn chức năng cần thực hiện
- Mở file danh sách lương nhân viên
- Nhập thông tin lương nhân viên từ bàn phím, không nhập trường lương tháng.
- Tính lương tháng của nhân viên với lương tháng=lương ngày\*ngày công.  
Tính thưởng cho các nhân viên, biết:  
Nếu ngày công  $\geq 26$ : Thưởng = 5000000  
Nếu  $24 \leq$  ngày công  $< 26$ : Thưởng = 3000000  
Nếu  $22 \leq$  ngày công  $< 24$ : Thưởng = 2000000  
Nếu ngày công  $< 22$ : Thưởng = 1000000
- Lưu danh sách lương nhân viên với đầy đủ thông tin vào tệp **ds\_luong\_nv.csv** trong thư mục files.
- Sắp xếp danh sách nhân viên theo thứ tự giảm dần của trường thưởng. In ra màn hình danh sách trước và sau khi sắp xếp.

**II.** Tổ chức viết lại chương trình ở phần I với nội dung như sau:

Trong thư mục **libs** viết module **xu\_ly\_luong\_nhan\_vien.py** chứa các hàm (function) thực hiện các chức năng sau:

- Mở file danh sách lương nhân viên (**ds\_luong\_nv.csv**)
- Thêm thông tin lương nhân viên (*manv, tennv, luongngay, ngaycong, luongthang, thuong*)
- In danh sách lương các nhân viên với đầy đủ thông tin.
- Lưu file danh sách lương nhân viên (lưu vào thư mục files với tên ds\_luong\_nv.csv)
- Tìm nhân viên có ngày công cao nhất.
- Xóa khỏi danh sách nhân viên có mã là manv, với manv nhập vào từ bàn phím.
- Thống kê doanh thu
- Lọc cửa hàng theo vốn đầu tư
- Cửa hàng có doanh thu cao nhất

Viết lại file **qlycuahang\_v2.py** với các chức năng cần thực hiện được gọi từ các function đã viết trong module **xu\_ly\_cua\_hang.py**

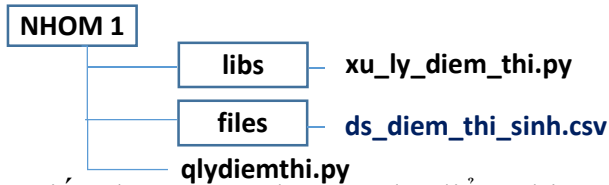


## Nhóm 9

Để quản lý quản lý Điểm của thí sinh gồm: mã thí sinh; tên thí sinh, khu vực, điểm môn 1, điểm môn 2, điểm môn 3, điểm ưu tiên, tổng điểm

Trong đó: mã thí sinh; họ tên thí sinh, khu vực, điểm môn 1, điểm môn 2, điểm môn 3 được nhập vào từ bàn phím.

Tạo một project (thư mục) **NHOM09**, có cấu trúc như sau:



I. Viết chương trình quản lý điểm thi version1 đặt tên là **qlydiemthi.py** thực hiện các chức năng sau đây:

- Tạo menu chọn chức năng cần thực hiện
- Mở file danh sách điểm thí sinh.
- Nhập thông tin các thí sinh từ bàn phím, không nhập trường tổng điểm.
- Tính tổng điểm theo khu vực, biết:
  - Nếu thí sinh thuộc khu vực 1: cộng 0.5 điểm
  - Nếu thí sinh thuộc khu vực 2: cộng 1.0 điểm
  - Nếu thí sinh thuộc khu vực 3: không được cộng điểm ưu tiênTính tổng điểm của các thí sinh với tổng điểm = điểm môn1 + điểm môn2 + điểm môn3 + điểm ưu tiên
- Lưu danh sách thí sinh với đầy đủ thông tin vào tệp **ds\_diem\_thi\_sinh.csv** trong thư mục files.
- Sắp xếp các cửa hàng theo thứ tự giảm dần của tổng điểm. In ra màn hình danh sách trước và sau khi sắp xếp.

II. Tổ chức viết lại chương trình ở phần I với nội dung như sau:

Trong thư mục **libs** viết module **xu\_ly\_diem\_thi.py** chứa các hàm (function) thực hiện các chức năng sau:

- Mở file danh sách điểm thí sinh (**ds\_diem\_thi\_sinh.csv**)
- Thêm thông tin các thí sinh  
(*ma\_thi\_sinh, ten\_thi\_sinh, khu\_vuc, diemmon1, diemmon2, diemmon3, tong\_diem*)
- In danh sách các thí sinh với đầy đủ thông tin.
- Lưu file danh sách điểm thí sinh  
(lưu vào thư mục files với tên **ds\_diem\_thi\_sinh.csv**)
- Tìm thí sinh có tổng điểm cao nhất
- Xóa khỏi danh sách các thí sinh có tổng điểm < điểm chuẩn. Biết rằng: điểm chuẩn là một số thực nhập vào từ bàn phím.
- Sắp xếp danh sách thí sinh theo thứ tự giảm dần của tổng điểm. In ra màn hình danh sách trước và sau khi sắp xếp.

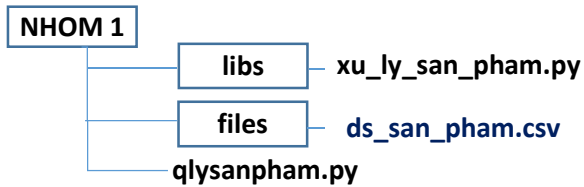
Viết lại file **qlydiemthi\_v2.py** với các chức năng cần thực hiện được gọi từ các function đã viết trong module **xu\_ly\_diem\_thi.py**

## Nhóm 10

Để quản lý quản lý các sản phẩm của một công ty bao gồm: mã sản phẩm, tên sản phẩm, số lượng, đơn giá, giảm giá, tổng tiền

Trong đó {mã sản phẩm, tên sản phẩm, số lượng, đơn giá} được nhập vào từ bàn phím.

Tạo một project (thư mục) **NHOM10**, có cấu trúc như sau:



**I.** Viết chương trình quản lý sản phẩm version1 đặt tên là **qlysanpham.py** thực hiện các chức năng sau đây:

- Tạo menu chọn chức năng cần thực hiện
- Mở file danh sách sản phẩm
- Nhập thông tin các sản phẩm từ bàn phím, không nhập trường tổng tiền.
- Tính giảm giá cho các sản phẩm, biết:
  - Nếu  $50 \leq \text{Số lượng} < 100$ : giảm giá 500000
  - Nếu  $100 \leq \text{Số lượng} < 150$ : giảm giá 1000000
  - Nếu  $\text{Số lượng} \geq 150$ : giảm giá 2000000Tính Tổng tiền của sản phẩm với tổng tiền = số lượng \* đơn giá – giảm giá
- Lưu danh sách các sản phẩm với đầy đủ thông tin vào tệp **ds\_san\_pham.csv** trong thư mục files.
- Sắp xếp các sản phẩm theo thứ tự giảm dần của tổng tiền. In ra màn hình danh sách trước và sau khi sắp xếp.

**II.** Tổ chức viết lại chương trình ở phần I với nội dung như sau:

Trong thư mục **libs** viết module **xu\_ly\_san\_pham.py** chứa các hàm (function) thực hiện các chức năng sau:

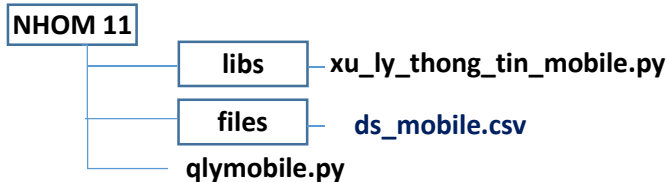
- Mở file danh sách sản phẩm (**ds\_san\_pham.csv**)
- Thêm thông tin các sản phẩm (*masp, tensp, soluong, dongia, giamgia, tong\_tien*)
- In danh sách các sản phẩm với đầy đủ thông tin ra màn hình.
- Lưu file danh sách sản phẩm (lưu vào thư mục files với tên **ds\_san\_pham.csv**)
- Tìm sản phẩm theo mã sản phẩm (masp).
- Xóa Xóa khỏi danh sách các sản phẩm có số lượng > x với x là một số nguyên được nhập vào từ bàn phím
- Thêm vào danh sách sản phẩm có thông tin được nhập vào từ bàn phím, sao cho vẫn đảm bảo thứ tự sắp xếp

Viết lại file **qlysanpham\_v2.py** với các chức năng cần thực hiện được gọi từ các function đã viết trong module **xu\_ly\_san\_pham.py**

## Nhóm 11

Để quản lý trình quản lý doanh thu mặt hàng thiết bị điện thoại của một cửa hàng với thông tin của mỗi thiết bị bao gồm: mã điện thoại (kiểu xâu); tên điện thoại (kiểu xâu); đơn giá (số thực); số lượng (số nguyên); chất lượng (số nguyên: 1, 2); thuế VAT (số thực); tổng tiền (số thực) Trong đó {mã điện thoại; tên điện thoại; đơn giá; số lượng; chất lượng} được nhập vào từ bàn phím.

Tạo một project (thư mục) **NHOM011**, có cấu trúc như sau:



**I.** Viết chương trình quản lý cửa hàng version1 đặt tên là **qlymobile.py** thực hiện các chức năng sau đây:

- Tạo menu chọn chức năng cần thực hiện
- Mở file danh sách mobile
- Nhập thông tin điện thoại từ bàn phím, không nhập trường tổng tiền và thuế VAT.
- Tính thuế VAT của các mặt hàng, biết:  
Nếu chất lượng loại 1:  $VAT = 30\% * \text{đơn giá} * \text{số lượng}$   
Nếu chất lượng loại 2:  $VAT = 10\% * \text{đơn giá} * \text{số lượng}$   
Tính tổng tiền với tổng tiền = đơn giá \* số lượng + thuế VAT
- Lưu danh sách các điện thoại với đầy đủ thông tin vào tệp **ds\_mobile.csv** trong thư mục files.
- Sắp xếp các cửa hàng theo thứ tự giảm dần của tổng tiền. In ra màn hình danh sách trước và sau khi sắp xếp.

**II.** Tổ chức viết lại chương trình ở phần I với nội dung như sau:

Trong thư mục **libs** viết module **xu\_ly\_thong\_tin\_mobile.py** chứa các hàm (function) thực hiện các chức năng sau:

- Mở file danh sách mobile (**ds\_mobile.csv**)
- Thêm thông tin các điện thoại
  - (*madt, tendt, dongia, soluong, chatluong, VAT, tong\_tien*)
- In danh sách điện thoại với đầy đủ thông tin.
- Lưu file danh sách điện thoại
  - (lưu vào thư mục files với tên **dsmobile.csv**)
- Tìm điện thoại theo mã điện thoại (madt).
- Xóa khỏi danh sách các thiết bị có thuế VAT > vat với vat là một số thực được nhập vào từ bàn phím

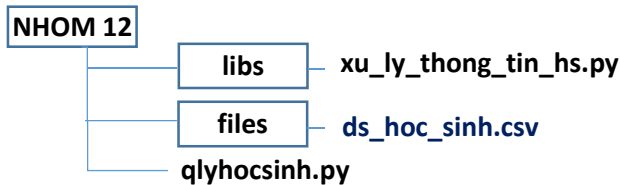
Viết lại file **qlymobile\_v2.py** với các chức năng cần thực hiện được gọi từ các function đã viết trong module **xu\_ly\_thong\_tin\_mobile.py**

## Nhóm 12

Để quản lý quản lý học sinh lưu trữ nội dung các thông tin dạng cấu trúc gồm: Mã HS, Tên HS, Điểm Toán, Điểm Lý, Điểm Hoá, Điểm TB, Xếp loại.

Các thông tin Mã HS, Tên HS, Điểm Toán, Điểm Lý, Điểm Hoá được nhập vào từ bàn phím.

Tạo một project (thư mục) **NHOM012**, có cấu trúc như sau:



I. Viết chương trình quản lý học sinh version1 đặt tên là **qlyhocsinh.py** thực hiện các chức năng sau đây:

- Tạo menu chọn chức năng cần thực hiện
- Mở file danh sách học sinh
- Nhập thông tin các học sinh từ bàn phím, không nhập trường điểm TB.
- Tính điểm trung bình cho các học sinh biết:  
$$\text{ĐTB} = (\text{Điểm Toán} + \text{Điểm Lý} + \text{Điểm Hoá}) / 3.$$
  
Điền thông tin Xếp loại cho học sinh biết:  
Nếu  $\text{ĐTB} \geq 9$ , xếp loại "Xuất sắc".  
Nếu  $\text{ĐTB} \geq 8$ , xếp loại "Giỏi".  
Nếu  $\text{ĐTB} \geq 7$ , xếp loại "Khá".  
Nếu  $\text{ĐTB} \geq 5$ , xếp loại "Trung bình".  
Nếu  $\text{ĐTB} < 5$ , xếp loại "Yếu".
- Lưu danh sách các học sinh với đầy đủ thông tin vào tệp **ds\_hoc\_sinh.csv** trong thư mục files.
- Sắp xếp các học sinh theo thứ tự giảm dần của điểm trung bình. In ra màn hình danh sách trước và sau khi sắp xếp.

II. Tổ chức viết lại chương trình ở phần I với nội dung như sau:

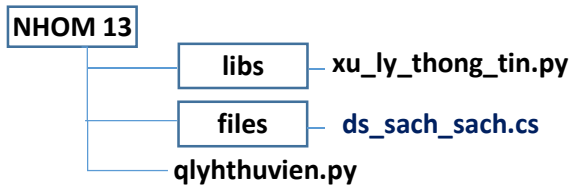
Trong thư mục **libs** viết module **xu\_ly\_thong\_tin\_hs.py** chứa các hàm (function) thực hiện các chức năng sau:

- Mở file danh sách học sinh (**ds\_hoc\_sinh.csv**)
- Thêm thông tin các học sinh  
(*mahs, tenhs, diemtoan, diemly, diemhoa, diemTB, xeploai*)
- In danh sách học sinh với đầy đủ thông tin.
- Lưu file danh sách học sinh  
(lưu vào thư mục files với tên **ds\_hoc\_sinh.csv**)
- In ra màn hình danh sách các học sinh có điểm Hoá lớn nhất.
- Xóa khỏi danh sách các học sinh có điểm Lý < diem\_ly với diem\_ly là một số được nhập vào từ bàn phím.

Viết lại file **qlyhocsinh\_v2.py** với các chức năng cần thực hiện được gọi từ các function đã viết trong module **xu\_ly\_thong\_tin\_hs.py**

## Nhóm 13

Tạo một project (thư mục) **NHOM13**, có cấu trúc như sau:



Trong đó các tập tin:

- **xu\_ly\_thu\_vien.py**: Chứa các hàm để xử lý thông tin sách (nhập, cập nhật, tính toán).
- **danh\_sach\_sach.csv**: Lưu trữ thông tin sách với các trường *Mã Sách*, *Tên Sách*, *Tác Giả*, *Thể Loại*, *Giá Sách*, *Số Lượng*, *Năm Xuất Bản*.
- **qlythuvien.py**: File chính để chạy ứng dụng.

**I.** Viết chương trình quản lý sách thư viện version1 đặt tên là **qlythuvien.py** thực hiện các chức năng sau đây:

- Menu Chức Năng: Giao diện dòng lệnh cho các chức năng quản lý thư viện.
- Mở File Danh Sách Sách: Đọc và hiển thị thông tin từ danh\_sach\_sach.csv.
- Nhập Thông Tin Sách: Nhập thông tin sách mới từ bàn phím.
- Tính Toán Giá Trị Tổng Của Sách: Tự động tính tổng giá trị dựa trên giá và số lượng sách.
- Lưu Thông Tin Sách: Ghi thông tin mới hoặc cập nhật vào danh\_sach\_sach.csv.
- Sắp Xếp và Hiển Thị Danh Sách Sách: Sắp xếp sách theo tiêu chí như năm xuất bản

**II.** Tổ chức viết lại chương trình ở phần I với nội dung như sau:

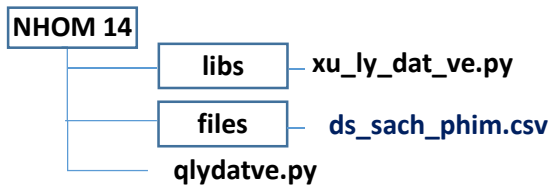
Trong thư mục **libs** viết module **xu\_ly\_thong\_tin\_hs.py** chứa các hàm (function) thực hiện các chức năng sau:

- Mở file danh sách học sinh (**ds\_sach\_sach.csv**)
- Tính Giá Trị Tổng Của Mỗi Thể Loại Sách: Tính tổng giá trị sách dựa trên giá và số lượng cho mỗi thể loại.
- In danh sách sách với đầy đủ thông tin.
- Lưu file danh sách sách
  - (lưu vào thư mục files với tên **ds\_sach\_sach.csv**)
- In ra màn hình danh sách sách có giá trị cao nhất và thấp nhất trong thư viện..
- Xóa khỏi danh sách các sách cũ (có Năm Xuất Bản trước 1993) với năm xuất bản là một số được nhập vào từ bàn phím.

Viết lại file **qlyhocsinh\_v2.py** với các chức năng cần thực hiện được gọi từ các function đã viết trong module **xu\_ly\_thong\_tin.py**

## NHÓM 14.

Tạo một project (thư mục) **NHOM14**, có cấu trúc như sau:



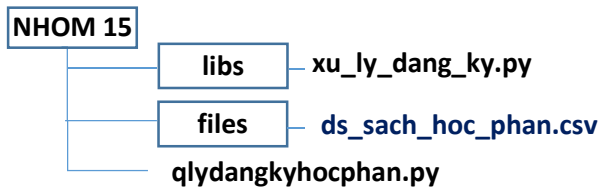
Trong đó các tập tin:

- xu\_ly\_dat\_ve.py: Chứa các hàm xử lý thông tin đặt vé (nhập, cập nhật, tính toán).
  - danh\_sach\_phim.csv: Lưu trữ thông tin phim với các trường: *Mã Phim*, *Tên Phim*, *Thời Lượng*, *Thẻ Loại*, *Giá Vé*, *Số lượng ghế*, *Ngày chiếu*.
  - qlydatve.py: File chính để chạy ứng dụng.
- I. Viết chương trình quản lý đặt vé xem phim version 1 đặt tên là **qlydatve.py** thực hiện các chức năng sau đây:
- a. Menu Chức Năng: Giao diện dòng lệnh cho các chức năng đặt vé.
  - b. Mở File Danh Sách Phim: Đọc và hiển thị thông tin từ danh\_sach\_phim.csv.
  - c. Nhập Thông Tin Phim: Nhập thông tin phim mới từ bàn phím.
  - d. Tính Toán Tổng Doanh Thu: Tính tổng doanh thu tự động tính trên giá vé và số lượng vé bán ra.
  - e. Lưu Thông Tin Phim: Ghi thông tin mới hoặc cập nhật vào danh\_sach\_phim.csv.
  - f. Sắp Xếp và Hiển Thị Danh Sách Phim: Sắp xếp phim theo tiêu chí như ngày chiếu, tên phim.
- II. Tổ chức viết lại chương trình ở phần I với nội dung như sau:
- Trong thư mục libs viết module **xu\_ly\_dat\_ve.py** chứa các hàm (function) thực hiện các chức năng sau:
- a. Mở file danh sách phim (danh\_sach\_phim.csv).
  - b. Tính tổng doanh thu từ bán vé.
  - c. In danh sách phim với đầy đủ thông tin.
  - d. Lưu file danh sách phim.
  - e. In ra màn hình phim có doanh thu cao nhất và thấp nhất.
  - f. Xóa khỏi danh sách các phim cũ (với ngày chiếu là một số được nhập từ bàn phím).

Viết lại file qlydatve\_v2.py sẽ gọi các chức năng từ xu\_ly\_dat\_ve.py.

## NHÓM 15.

Tạo một project (thư mục) **NHOM15**, có cấu trúc như sau:



Trong đó các tập tin:

- xu\_ly\_dang\_ky.py: Chứa các hàm xử lý thông tin đăng ký khóa học (nhập, cập nhật, tính toán).
- danh\_sach\_hoc\_phan.csv: Lưu trữ thông tin học phần với các trường: Mã Học phần, Tên Học phần, Giảng Viên, Loại học phần, Học Phí, Số tín chỉ, Ngày Bắt Đầu.
- qlydatve.py: File chính để chạy ứng dụng.

I. Viết chương trình quản lý đăng ký học phần version1 đặt tên là **qlydangkyhocphan.py** thực hiện các chức năng sau đây:

- Menu Chức Năng: Giao diện dòng lệnh cho các chức năng đăng ký học phần.
- Mở File Danh Sách Khóa Học: Đọc và hiển thị thông tin từ danh\_sach\_hoc\_phan.csv.
- Nhập Thông Tin Học phần: Nhập thông tin học phần mới từ bàn phím.
- Tính toán học phí: tự động tính theo công thức sau:
  - Học phần cơ bản/đại cương = số tín chỉ x 300.000 VNĐ
  - Học phần nâng cao = số tín chỉ x 350.000 VNĐ
  - Học phần chuyên sâu = số tín chỉ x 400.000 VNĐ
  - Học phần ONLINE = số tín chỉ x 250.000 VNĐ.
- Lưu thông tin học phần: Ghi thông tin mới hoặc cập nhật vào danh\_sach\_hoc\_phan.csv.
- Sắp Xếp và Hiển Thị Danh Sách Khóa Học: Sắp xếp khóa học theo tiêu chí như ngày bắt đầu, tên khóa học.

II. Tổ chức viết lại chương trình ở phần I với nội dung như sau:

Trong thư mục libs viết module xu\_ly\_dang\_ky\_hp.py chứa các hàm (function) thực hiện các chức năng sau:

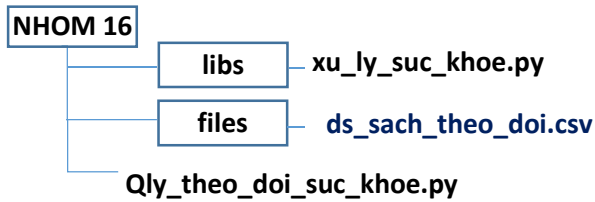
- Mở file danh sách khóa học (danh\_sach\_hoc\_phan.csv).
- Kiểm tra sự hợp lệ của thông tin đăng ký.
- In danh sách khóa học với đầy đủ thông tin.
- Lưu file danh sách khóa học.
- In ra màn hình danh sách khóa học phổ biến nhất.

Viết lại file qlydangkyhocphan.py version 2, sẽ gọi các chức năng từ xu\_ly\_dang\_ky.py.



## NHÓM 16.

Tạo một project (thư mục) **NHOM16**, có cấu trúc như sau:



Trong đó các tập tin:

- xu\_ly\_suc\_khoe.py: Chứa các hàm xử lý thông tin sức khỏe (nhập, cập nhật, phân tích).
- danh\_sach\_theo\_doi.csv: Lưu trữ thông tin theo dõi sức khỏe với các trường như *Mã Người Dùng, Tên, Tuổi, Cân Nặng, Chiều Cao, Mức Hoạt Động* (giá trị tổng hợp biểu diễn mức độ hoạt động hàng ngày (nhẹ, trung bình, nặng)., *Calo Tiêu Thụ ước lượng* (Lượng calo tiêu thụ hàng ngày ước lượng dựa trên mức hoạt động), *Ngày Theo Dõi*.
- qlydatve.py: File chính để chạy ứng dụng.

I. Viết chương trình quản lý đăng ký học phần version1 đặt tên là **qlydangkyhocphan.py** thực hiện các chức năng sau đây:

- Menu Chức Năng: Giao diện để chọn các chức năng như nhập thông tin, phân tích dữ liệu, xem báo cáo.
- Nhập Thông Tin Theo Dõi: Người dùng nhập thông tin cá nhân (chiều cao, cân nặng, hoạt động hàng ngày).
- Phân Tích Dữ Liệu Sức Khỏe: Tính chỉ số BMI từ chiều cao và cân nặng, ước lượng lượng calo tiêu thụ dựa trên mức độ hoạt động, biết rằng ước lượng calo tiêu thụ được tính như sau:
  - Mức Hoạt Động Nhẹ (như đi bộ nhẹ):  $\text{Calo tiêu thụ} = \text{Cân nặng (kg)} \times \text{Thời gian (giờ)} \times 2.3$
  - Mức Hoạt Động Trung Bình (như đi bộ nhanh, tập thể dục nhẹ):  $\text{Calo tiêu thụ} = \text{Cân nặng (kg)} \times \text{Thời gian (giờ)} \times 3.8$
  - Mức Hoạt Động Nặng (như chạy, tập thể dục nặng):  $\text{Calo tiêu thụ} = \text{Cân nặng (kg)} \times \text{Thời gian (giờ)} \times 6.0$
- Lưu Thông Tin Theo Dõi: Ghi và cập nhật dữ liệu theo dõi vào danh\_sach\_theo\_doi.csv.
- Hiển Thị và Sắp Xếp Dữ Liệu: Sắp xếp theo chỉ số BMI hoặc calo tiêu thụ, và hiển thị xu hướng sức khỏe.

II. Tổ chức viết lại chương trình ở phần I với nội dung như sau:

Trong thư mục libs viết module xu\_ly\_suc\_khoe.py chứa các hàm (function) thực hiện các chức năng sau:

- nhap\_thong\_tin\_nguoi\_dung(): Nhập và lưu trữ thông tin cơ bản của người dùng như tên, tuổi, cân nặng, chiều cao.
- tinh\_bmi(): Tính chỉ số BMI dựa trên cân nặng và chiều cao của người dùng.
- nhap\_hoat\_dong\_hang\_ngay(): Cho phép người dùng nhập thông tin về hoạt động hàng ngày và thời gian thực hiện.

- `tinh_calor_tieu_thu()`: Tính lượng calo tiêu thụ dựa trên hoạt động hàng ngày và thông tin cá nhân của người dùng.
- `xuat_bao_cao_suc_khoe()`: Hiển thị báo cáo sức khỏe tổng hợp, bao gồm thông tin về BMI, calo tiêu thụ và các hoạt động hàng ngày.