

Bài 1: Xây dựng chương trình quản lý kết quả học tập của sinh viên tại một trường đại học. Có 2 loại sinh viên là sinh viên chính quy và sinh viên tại chức với các thông tin giống nhau: mã sinh viên, họ tên, ngày tháng năm sinh, năm vào học, điểm đầu vào và danh sách kết quả học tập. Sinh viên tại chức có thêm thông tin nơi liên kết đào tạo (Đồng Nai, Cà Mau, ...). Khoa gồm có các thông tin: tên khoa và danh sách sinh viên đang theo học. Kết quả học tập gồm có tên học kỳ, điểm trung bình học kỳ đó.

Hiện thực các yêu cầu sau:

Sinh viên hãy thiết kế và viết code thể hiện các class của các **sinh viên**, ... của chương trình làm sao để tuân thủ theo mô hình **OOP** đã học, áp dụng 4 tính chất : bao đóng (*encapsulation*) , kế thừa (*inheritance*) , đa hình (*polymorphism*), trừu tượng (*abstraction*). Lưu ý: *Bất kì một sinh viên nào cũng cần có một phương thức có tên là **ShowMyInfor** để hiển thị thông tin của sinh viên đó ra màn hình cosole, yêu cầu này là bắt buộc với các thành viên xây dựng code cho chương trình này. Bạn hãy lưu ý khi thiết kế code để thỏa mãn yêu cầu này.*

1. Hiện thực các lớp cần thiết cho bài toán trên
2. Phương thức khởi tạo (constructor) cho các lớp: constructor chuẩn, constructor có tham số, constructor sao chép
3. Phương thức input và output để cho phép người dùng nhập thông tin cho các loại sinh viên
4. Phương thức xác định sinh viên có phải là chính quy hay không?
5. Xử lý Exception khi người dùng nhập sai dữ liệu
6. Phương thức lấy điểm trung bình các môn học của sinh viên chính quy dựa vào học kỳ cho trước
7. Phương thức xác định tổng số sinh viên chính quy của khoa?
8. Tìm ra sinh viên có điểm đầu vào cao nhất ở mỗi khoa
9. Ở mỗi khoa, lấy ra danh sách các sinh viên tại chức tại nơi liên kết đào tạo cho trước
10. Ở mỗi khoa, lấy ra danh sách sinh viên có điểm trung bình ở học kỳ gần nhất (là học kỳ cuối cùng trong danh sách kết quả học tập của sinh viên) từ 8.0 trở lên

Bài 2: Xây dựng chương trình quản lý kết quả học tập của sinh viên tại một trường đại học. Có 2 loại sinh viên là sinh viên chính quy và sinh viên tại chức với các thông tin giống nhau: mã sinh viên, họ tên, ngày tháng năm sinh, năm vào học, điểm đầu vào và danh sách kết quả học tập. Sinh viên tại chức có thêm thông tin nơi liên kết đào tạo (Đồng Nai, Cà Mau, ...). Khoa gồm có các thông tin: tên khoa và danh sách sinh viên đang theo học. Kết quả học tập gồm có tên học kỳ, điểm trung bình học kỳ đó.

Hiện thực các yêu cầu sau:

Sinh viên hãy thiết kế và viết code thể hiện các class của các **sinh viên**, ... của chương trình làm sao để tuân thủ theo mô hình **OOP** đã học, áp dụng 4 tính chất : bao đóng (*encapsulation*) , kế thừa (*inheritance*) , đa hình (*polymorphism*), trừu tượng (*abstraction*). Lưu ý: *Bất kì một sinh viên nào cũng cần có một phương thức có tên là **ShowMyInfor** để hiển thị thông tin của sinh viên đó ra màn hình cosole, yêu cầu này là bắt buộc với các thành viên xây dựng code cho chương trình này. Bạn hãy lưu ý khi thiết kế code để thỏa mãn yêu cầu này.*

1. Hiện thực các lớp cần thiết cho bài toán trên
2. Phương thức khởi tạo (constructor) cho các lớp: constructor chuẩn, constructor có tham số, constructor sao chép
3. Phương thức input và output để cho phép người dùng nhập thông tin cho các loại sinh viên
4. Phương thức xác định sinh viên có phải là tại chức hay không?
5. Xử lý Exception khi người dùng nhập sai dữ liệu?
6. Phương thức lấy điểm trung bình các môn học của sinh viên tại chức dựa vào học kỳ cho trước
7. Phương thức xác định tổng số sinh viên tại chức của khoa?
8. Ở mỗi khoa, tìm ra sinh viên có điểm trung bình học kỳ cao nhất (ở bất kỳ học kỳ nào)
9. Ở mỗi khoa, sắp xếp danh sách sinh viên tăng dần theo loại và giảm dần theo năm vào học (sử dụng interface Comparable hoặc Comparator)
10. Ở mỗi khoa, thống kê số lượng sinh viên theo năm vào học. Ví dụ 2020: 100, 2019: 90, 2018: 120.

Bài 3 Viết chương trình quản lý việc tuyển sinh viên sau khi tốt nghiệp đại học. Có 2 loại sinh viên bao gồm **sinh viên tốt nghiệp loại khá giỏi (GoodStudent)** và **sinh viên tốt nghiệp loại trung bình (NormalStudent)**.

Cả 2 loại sinh viên trên đều phải cung cấp các thông tin sau khi nộp hồ sơ xin việc: Họ tên (fullName), Ngày tháng năm sinh (doB), Giới tính (sex), Số điện thoại (phoneNumber), Tên trường đã học (universityName), Xếp loại tốt nghiệp (gradeLevel).

Riêng sinh viên loại khá giỏi phải có thêm thông tin: điểm trung bình học tập (**gpa**) theo thang điểm 10, tên của loại học bổng (hoặc giải thưởng) cao nhất đã từng giành được (**bestRewardName**).

Riêng sinh viên loại trung bình thì phải có thêm các thông tin: điểm TOEIC (**englishScore**), điểm thi đầu vào chuyên môn do công ty tổ chức thi (**entryTestScore**) theo thang điểm 10.

Yêu cầu 1: Sinh viên hãy thiết kế và viết code thể hiện các class của các **sinh viên** và **lớp học** của chương trình làm sao để tuân thủ theo mô hình **OOP** đã học, áp dụng 4 tính chất : bao đóng (*encapsulation*), kế thừa (*inheritance*), đa hình (*polymorphism*), trừu tượng (*abstraction*). Lưu ý: *Bất kì một sinh viên nào cũng cần có một phương thức có tên là **ShowMyInfor** để hiển thị thông tin của sinh viên đó ra màn hình cosole, yêu cầu này là bắt buộc với các thành viên xây dựng code cho chương trình này. Bạn hãy lưu ý khi thiết kế code để thỏa mãn yêu cầu này.*

1. Hiện thực các lớp cần thiết cho bài toán trên
2. Phương thức khởi tạo (constructor) cho các lớp: constructor chuẩn, constructor có tham số, constructor sao chép

Yêu Cầu 2 : Viết phương thức input và output để cho phép người dùng nhập thông tin cho sinh viên. Ngoài ra nếu có bất cứ một exception nào khác trong quá trình thực thi chương trình, sinh viên hãy thông báo ra màn hình nội dung *“Input files have unknow errors !!!”*.

Yêu cầu 3: Chương trình cần có chức năng lựa chọn ứng viên trúng tuyển vào công ty theo nguyên tắc sau:

- Người dùng sẽ nhập vào số lượng sinh viên cần tuyển dụng (tối thiểu là 11, tối đa là 15).
- Chương trình sẽ tự động chọn ra ứng viên phù hợp cho công ty theo các bước như sau:
 - Nếu số lượng ứng viên là sinh viên khá giỏi có nhiều hơn số lượng cần tuyển thì xét ưu tiên theo điểm GPA. Nếu *xuất hiện ứng viên khá giỏi có cùng điểm GPA thì xét ưu tiên theo họ tên*. Vd: nếu họ tên là Nguyễn Văn A và Nguyễn Văn B thì ứng viên Nguyễn Văn A được chọn. (*Giả sử không bao giờ có sinh viên có trùng họ tên nhau*).
 - Sau khi tuyển hết ứng viên khá giỏi, nếu vẫn chưa đủ số lượng cần tuyển, chương trình sẽ lấy ứng viên trung bình. Các ứng viên trung bình được xét ưu tiên theo điểm thi đầu vào, nếu điểm thi đầu vào bằng nhau thì xét đến điểm TOEIC. *Nếu xuất hiện ứng viên trung bình có cùng điểm TOEIC thì xét ưu tiên theo họ tên*.

Bài 4 Viết chương trình quản lý việc tuyển sinh viên sau khi tốt nghiệp đại học. Có 2 loại sinh viên bao gồm **sinh viên tốt nghiệp loại khá giỏi (GoodStudent)** và **sinh viên tốt nghiệp loại trung bình (NormalStudent)**.

Cả 2 loại sinh viên trên đều phải cung cấp các thông tin sau khi nộp hồ sơ xin việc: Họ tên (fullName), Ngày tháng năm sinh (doB), Giới tính (sex), Số điện thoại (phoneNumber), Tên trường đã học (universityName), Xếp loại tốt nghiệp (gradeLevel).

Riêng sinh viên loại khá giỏi phải có thêm thông tin: điểm trung bình học tập (**gpa**) theo thang điểm 10, tên của loại học bổng (hoặc giải thưởng) cao nhất đã từng giành được (**bestRewardName**).

Riêng sinh viên loại trung bình thì phải có thêm các thông tin: điểm TOEIC (**englishScore**), điểm thi đầu vào chuyên môn do công ty tổ chức thi (**entryTestScore**) theo thang điểm 10.

Yêu cầu 1: Sinh viên hãy thiết kế và viết code thể hiện các class của các **sinh viên** và **lớp học** của chương trình làm sao để tuân thủ theo mô hình **OOP** đã học, áp dụng 4 tính chất : bao đóng (*encapsulation*), kế thừa (*inheritance*), đa hình (*polymorphism*), trừu tượng (*abstraction*). Lưu ý: *Bất kì một sinh viên nào cũng cần có một phương thức có tên là **ShowMyInfor** để hiển thị thông tin của sinh viên đó ra màn hình cosole, yêu cầu này là bắt buộc với các thành viên xây dựng code cho chương trình này. Bạn hãy lưu ý khi thiết kế code để thỏa mãn yêu cầu này.*

1. Hiện thực các lớp cần thiết cho bài toán trên
2. Phương thức khởi tạo (constructor) cho các lớp: constructor chuẩn, constructor có tham số, constructor sao chép

Yêu Cầu 2 : Viết phương thức input và output để cho phép người dùng nhập thông tin cho sinh viên. Ngoài ra nếu có bất cứ một exception nào khác trong quá trình thực thi chương trình, sinh viên hãy thông báo ra màn hình nội dung *“Input files have unknow errors !!!”*.

Yêu cầu 3: Chương trình cần có chức năng lựa chọn ứng viên trúng tuyển vào công ty theo nguyên tắc sau:

- Người dùng sẽ nhập vào số lượng sinh viên cần tuyển dụng (>0 và <5)
- Chương trình sẽ tự động chọn ra ứng viên phù hợp cho công ty theo các bước như sau:
 - Nếu số lượng ứng viên là sinh viên khá giỏi ít hơn hoặc bằng số lượng cần tuyển thì nhận hết sinh viên khá giỏi.

Yêu cầu 3: Chương trình có thể hiển thị được thông tin họ tên và số điện thoại của tất cả các sinh viên đã nhập vào hệ thống (yêu cầu dùng collection sort để sắp xếp giảm dần theo **fullName** và tăng dần theo **phoneNumber** đối với sinh viên trước khi hiển thị).

Bài 5: Ngành công an cần quản lý các phương tiện giao thông gồm: ô tô, xe máy, xe tải. Mỗi loại gồm các thông tin: ID, Hãng sản xuất, năm sản xuất, giá bán và màu xe, thời gian sử dụng.

Các ô tô có các thuộc tính riêng: số chỗ ngồi, kiểu động cơ.

Các xe máy có các thuộc tính riêng: công suất.

Xe tải cần quản lý thêm: Trọng tải.

Yêu cầu 1: Sinh viên hãy thiết kế và viết code thể hiện các class để quản lý các phương tiện trên của chương trình làm sao để tuân thủ theo mô hình **OOP** đã học, áp dụng 4 tính chất: bao đóng (*encapsulation*), kế thừa (*inheritance*), đa hình (*polymorphism*), trừu tượng (*abstraction*). Lưu ý: *Bất kì một phương tiện nào cũng cần có một phương thức có tên là **ShowThatInfor** để hiển thị thông tin của phương tiện đó ra màn hình console, yêu cầu này là bắt buộc với các thành viên xây dựng code cho chương trình này. Bạn hãy lưu ý khi thiết kế code để thỏa mãn yêu cầu này.*

1. Hiện thực các lớp cần thiết cho bài toán trên
2. Phương thức khởi tạo (constructor) cho các lớp: constructor chuẩn, constructor có tham số, constructor sao chép

Yêu Cầu 2 : Viết phương thức input và output để cho phép người dùng nhập thông tin cho các phương tiện. Ngoài ra nếu có bất cứ một exception nào khác trong quá trình thực thi chương trình, phương tiện hãy thông báo ra màn hình nội dung “*Input files have unknow errors !!!*”.

Yêu cầu 3: Xây dựng lớp QLPTGT có các chức năng (phương thức):

- Viết phương thức thêm, xóa(theo ID) các phương tiện thuộc các loại trên.
- Viết phương thức tìm phương tiện theo hãng sản xuất,
- Viết phương thức tìm phương tiện theo màu xe,
- Viết phương thức cập nhật lại giá bán của các loại phương tiện
- Viết phương thức kiểm tra những loại phương tiện quá hạn sử dụng, tính tới thời gian hiện tại.

Bài 6: Một thư viện cần quản lý các tài liệu bao gồm Sách, Tạp chí, Báo. Mỗi tài liệu gồm có các thuộc tính sau: Mã tài liệu(Mã tài liệu là duy nhất), Tên nhà xuất bản, số bản phát hành.

Các loại sách cần quản lý thêm các thuộc tính: tên tác giả, số trang.

Các tạp chí cần quản lý thêm: Số phát hành, tháng phát hành.

Các báo cần quản lý thêm: Ngày phát hành.

Yêu cầu 1: Sinh viên hãy thiết kế và viết code thể hiện các class để quản lý tài liệu cho thư viện của chương trình làm sao để tuân thủ theo mô hình **OOP** đã học, áp dụng 4 tính chất: bao đóng (*encapsulation*), kế thừa (*inheritance*), đa hình (*polymorphism*), trừu tượng (*abstraction*). Lưu ý: *Bất kì một tài liệu nào cũng cần có một phương thức có tên là **ShowThatInfor** để hiển thị thông tin của tài liệu đó ra màn hình cosole, yêu cầu này là bắt buộc với các thành viên xây dựng code cho chương trình này. Bạn hãy lưu ý khi thiết kế code để thỏa mãn yêu cầu này.*

1. Hiện thực các lớp cần thiết cho bài toán trên
2. Phương thức khởi tạo (constructor) cho các lớp: constructor chuẩn, constructor có tham số, constructor sao chép

Yêu Cầu 2 : Viết phương thức input và output để cho phép người dùng nhập thông tin cho các tài liệu. Ngoài ra nếu có bất cứ một exception nào khác trong quá trình thực thi chương trình, tài liệu hãy thông báo ra màn hình nội dung “*Input files have unknow errors !!!*”.

Yêu cầu 3: Xây dựng lớp *QuanLySach* có các chức năng/ phương thức sau

- Thêm mới tài liệu: Sách, tạp chí, báo.
- Xoá tài liệu theo mã tài liệu.
- Cập nhật thông tin về tài liệu dựa vào mã tài liệu.
- Tìm kiếm tài liệu theo loại: Sách, tạp chí, báo.