

## HƯỚNG DẪN THỰC HÀNH OOP

Ngày 30/10/2022- PM3 -T7- A1

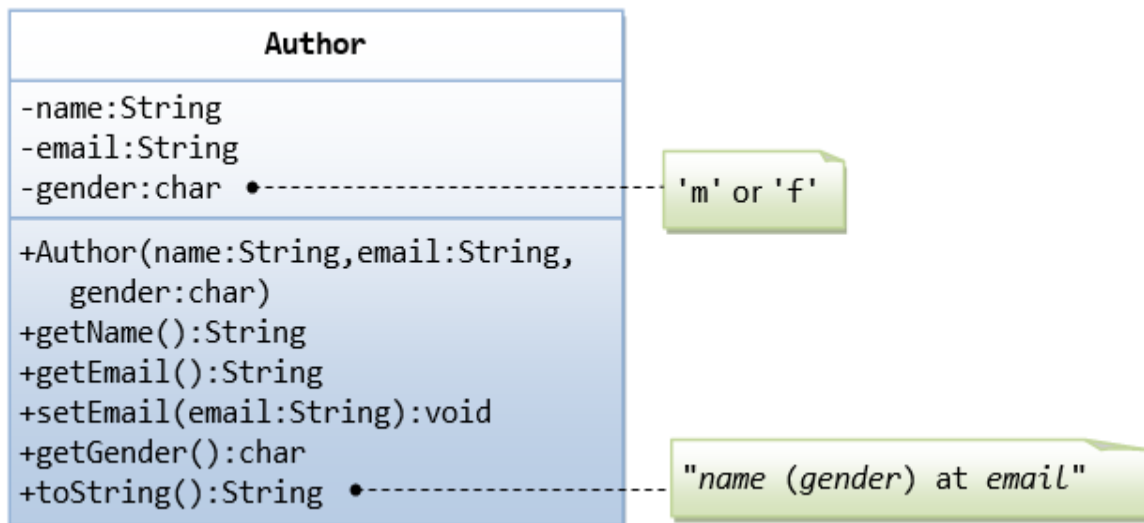
**Yêu cầu:** Nộp sản phẩm từ bài 5 gồm code+ báo cáo kết quả chạy chương trình.

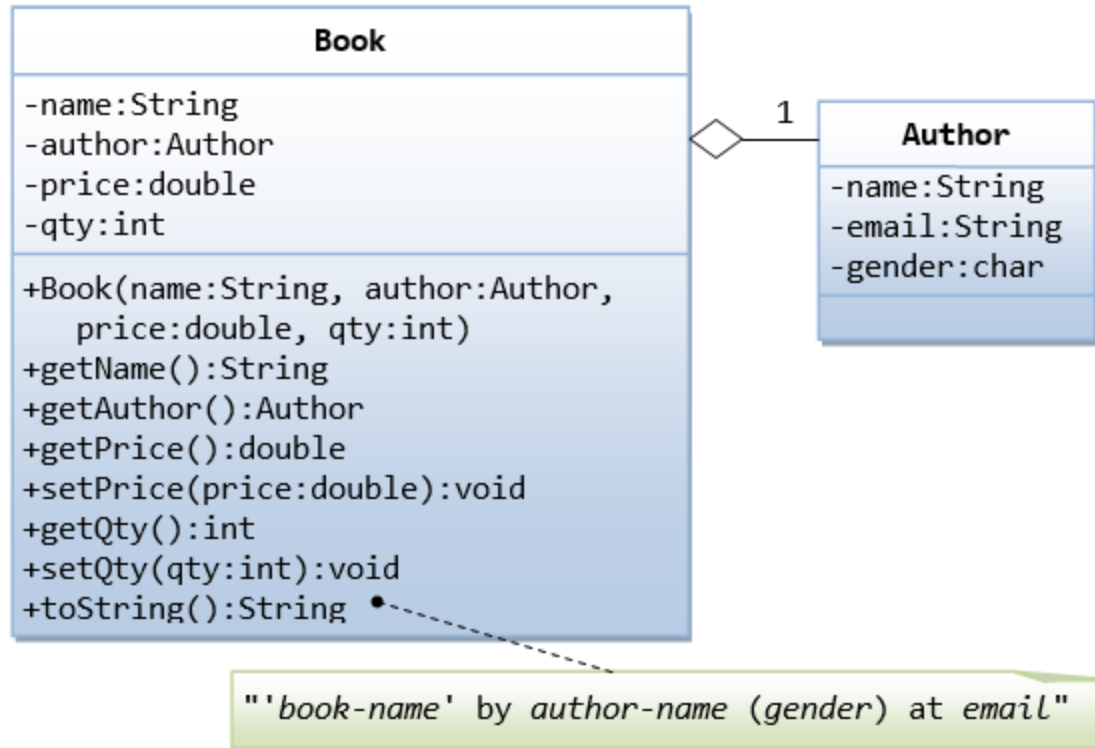
Bài 1: (Kế thừa, đa hình, quan hệ Has-A) Viết chương trình theo các cấu trúc sau đây:

(Tham khảo:

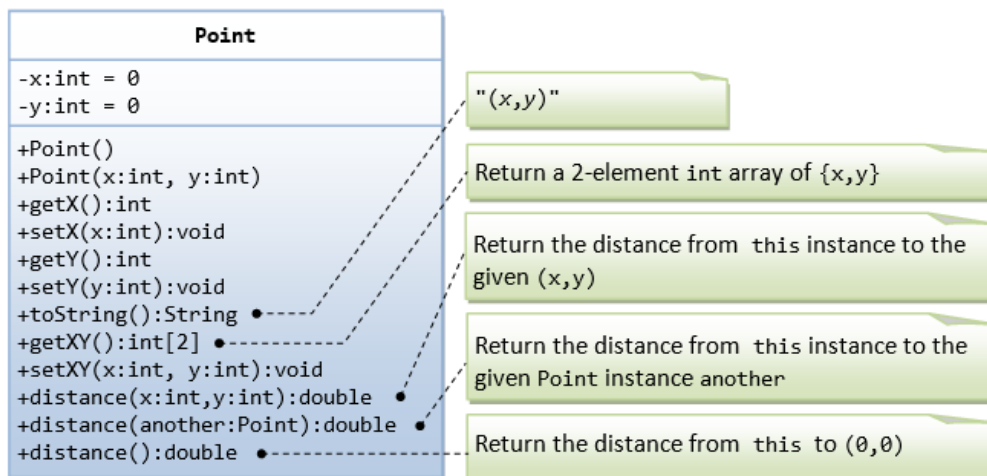
[https://www3.ntu.edu.sg/home/ehchua/programming/java/J3b\\_OOPinheritancePolymorphism.html](https://www3.ntu.edu.sg/home/ehchua/programming/java/J3b_OOPinheritancePolymorphism.html))

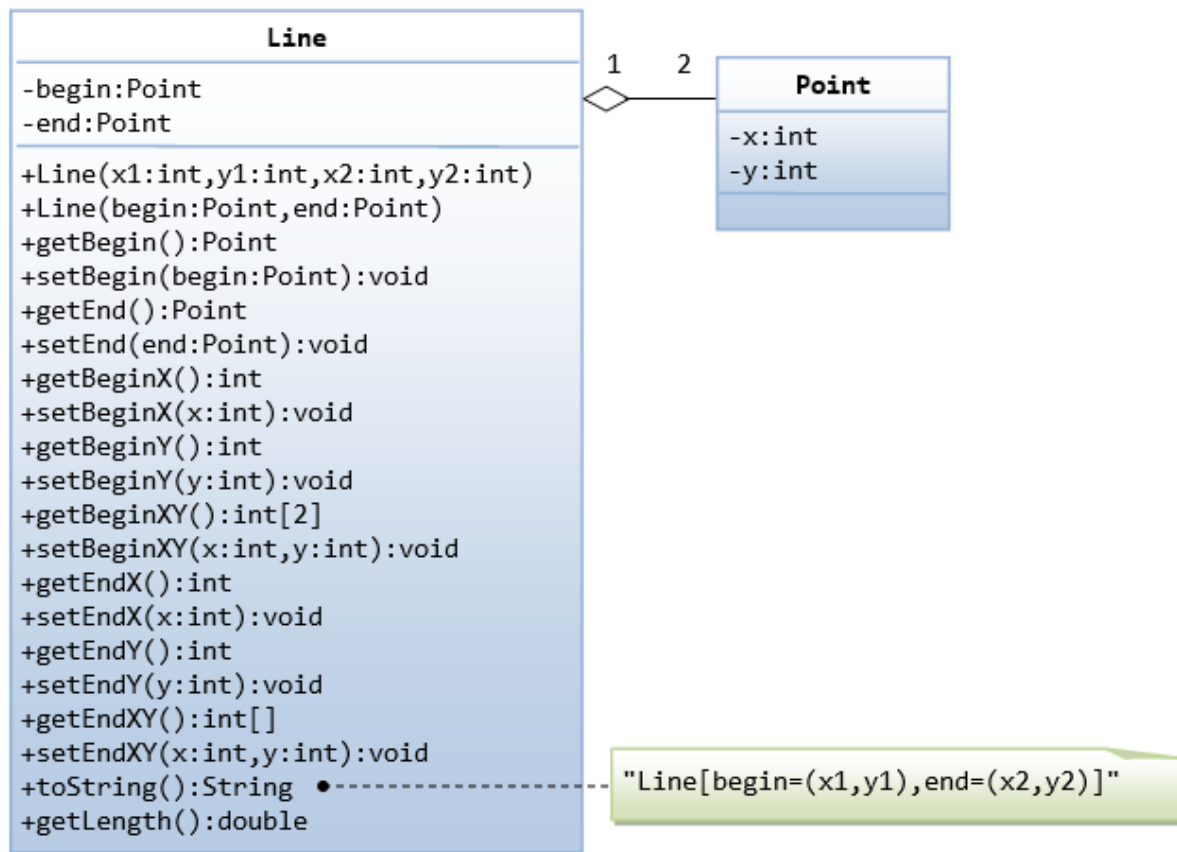
a)



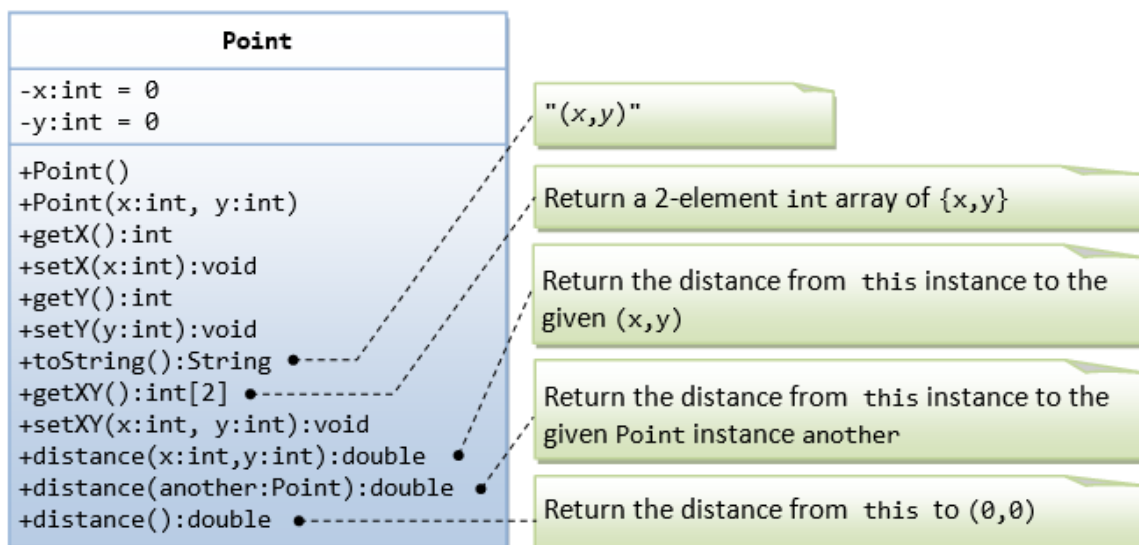


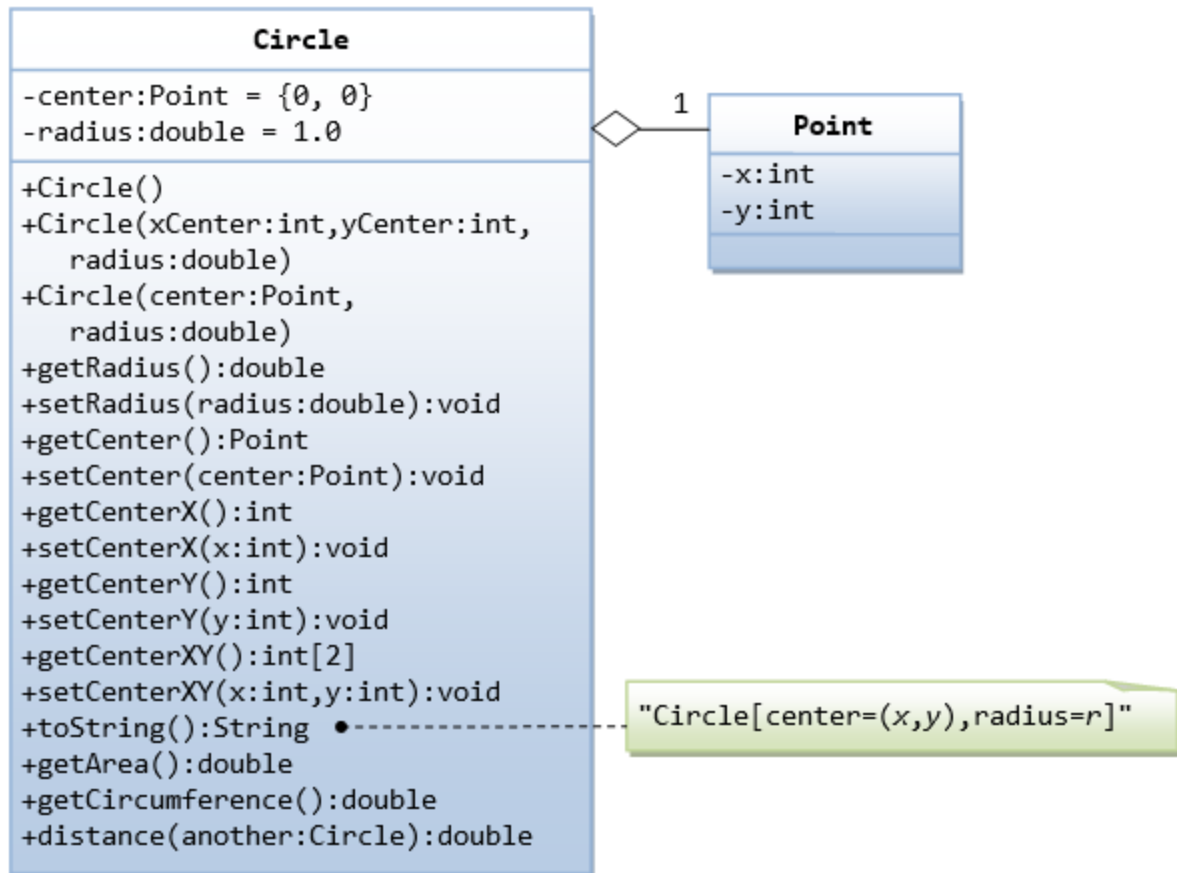
b)





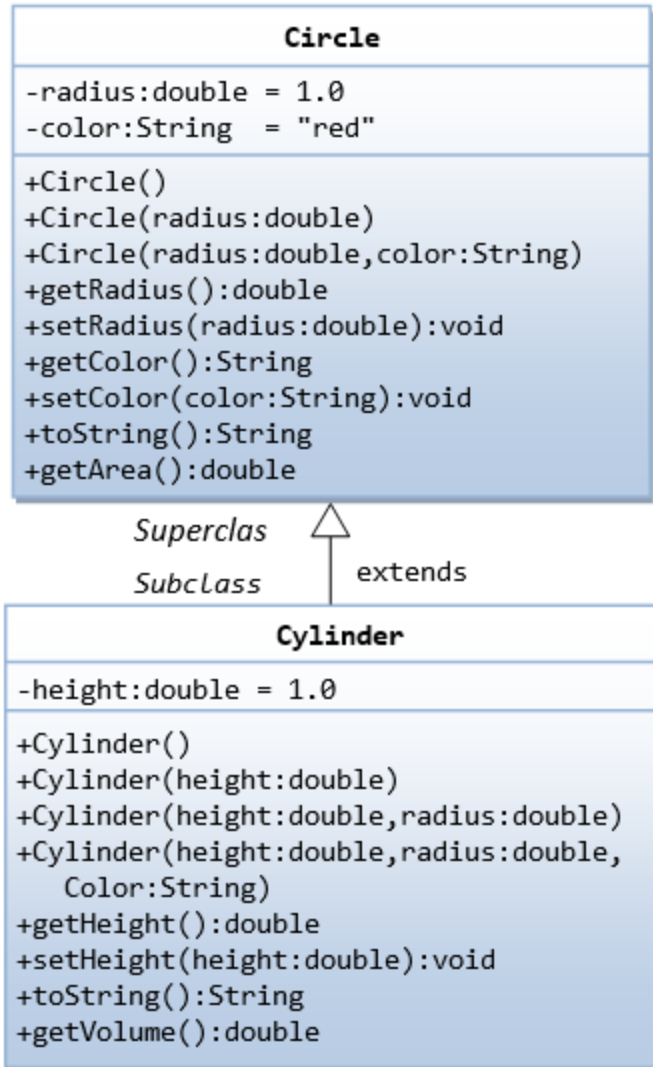
c)



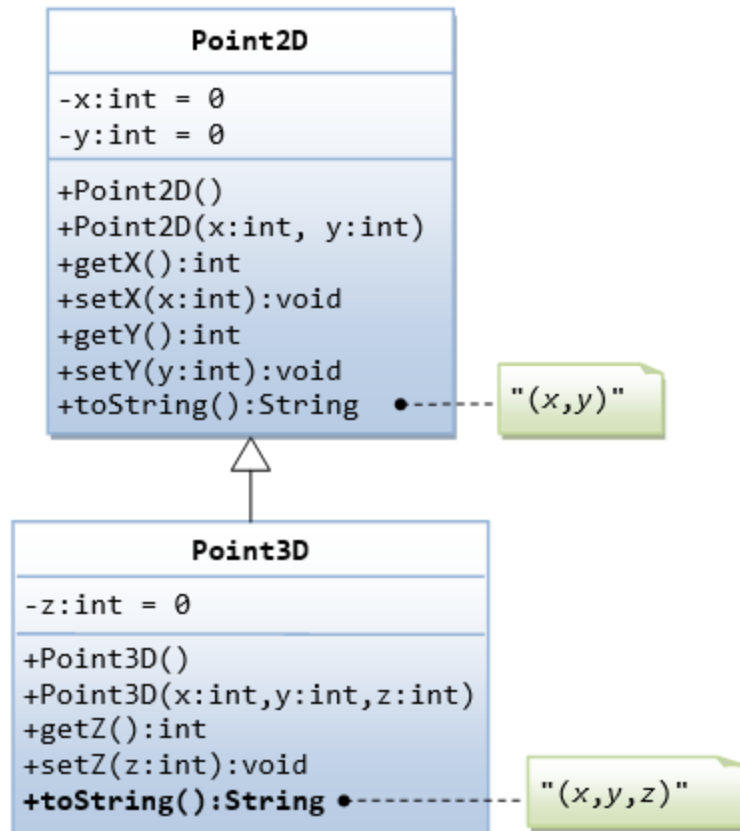


Bài 2: (Quan hệ kế thừa) Viết chương trình thực hiện theo cấu trúc sau:

a)

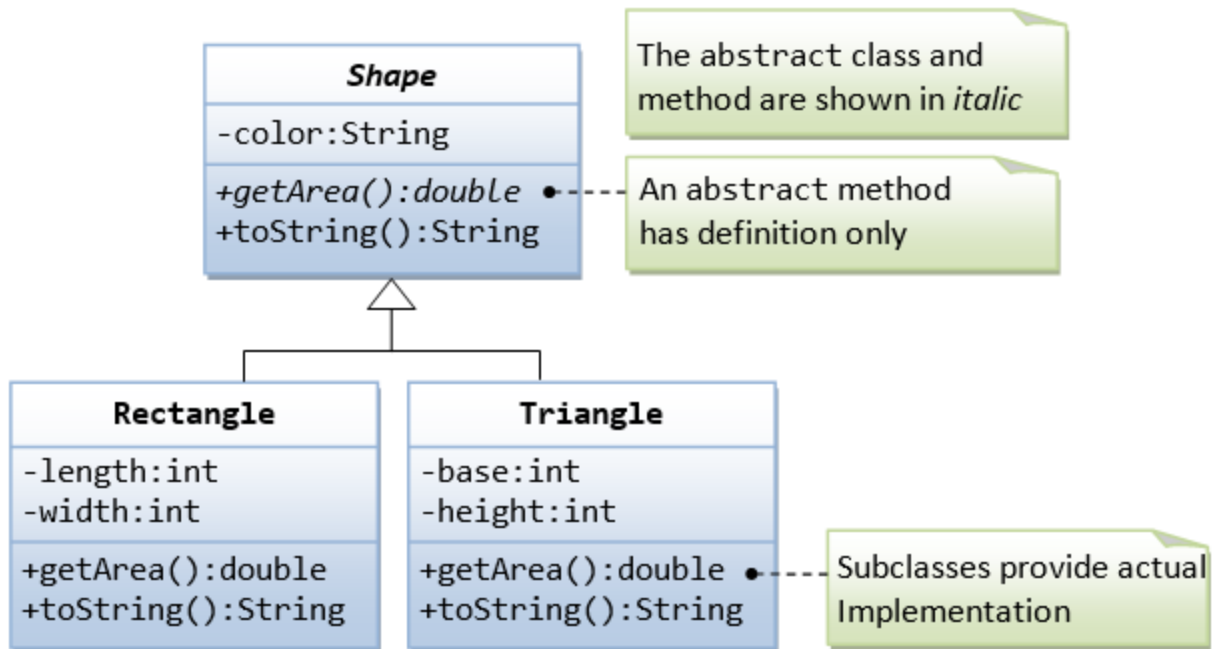


b)

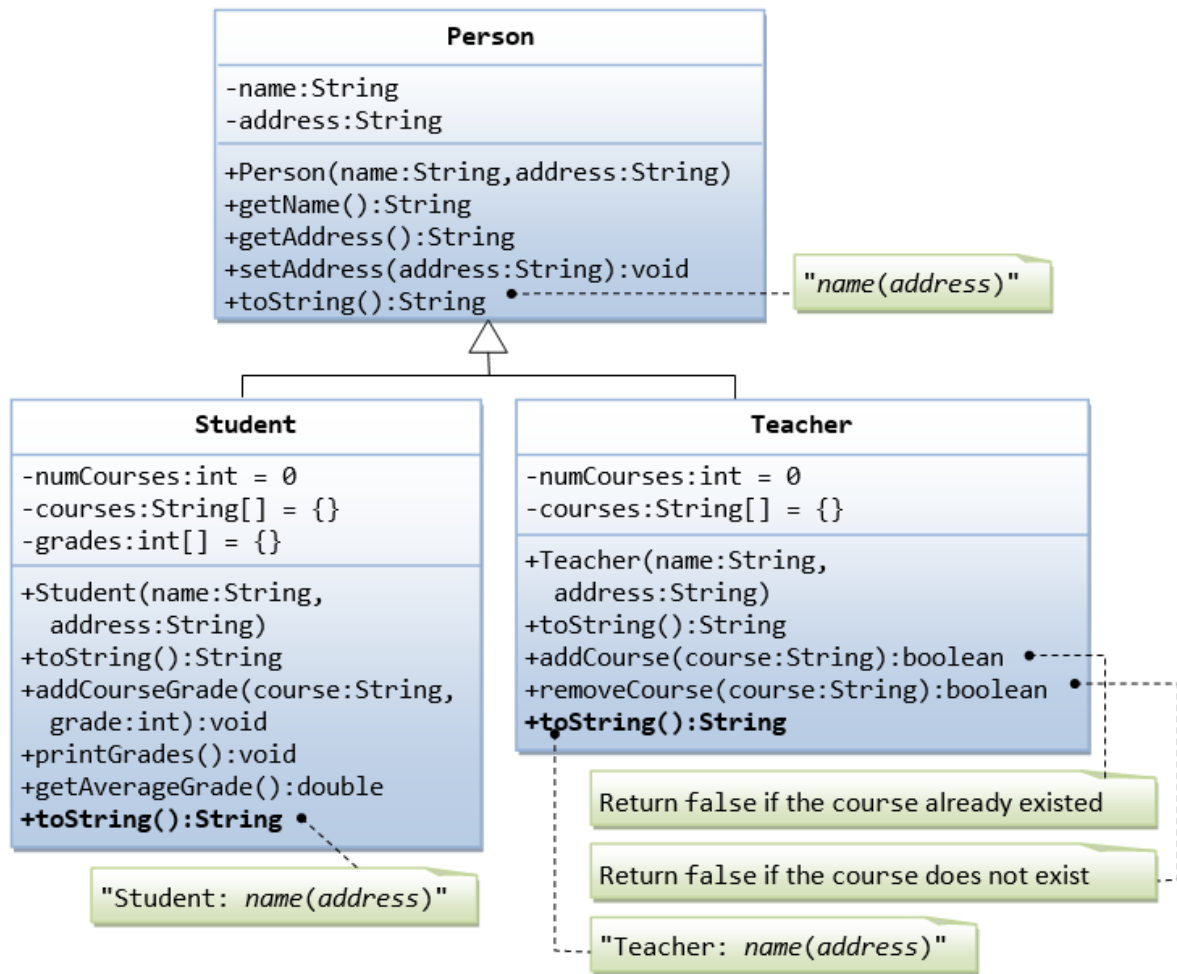


Bài 3: ( Lớp trừu tượng và giao diện) Viết chương trình với các cấu trúc sau

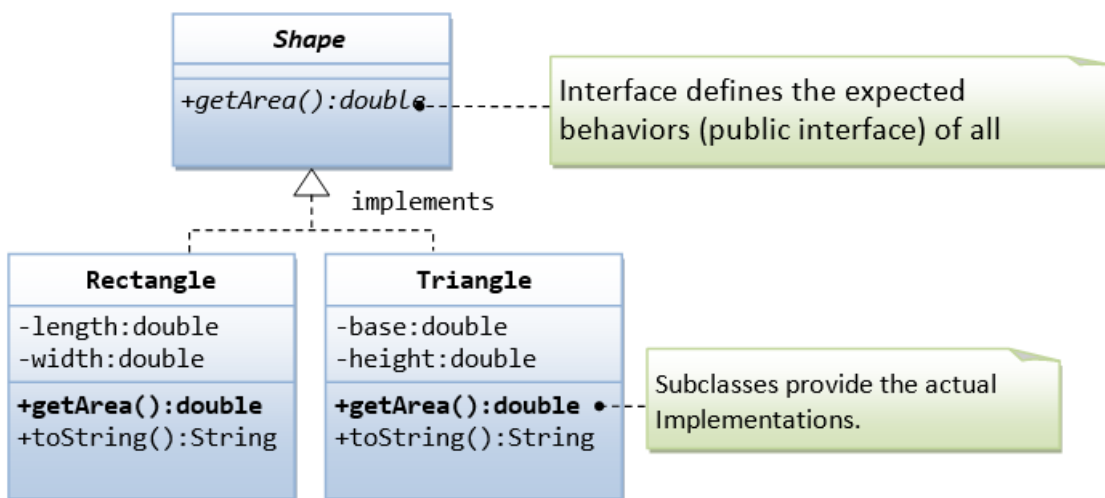
a)



b)

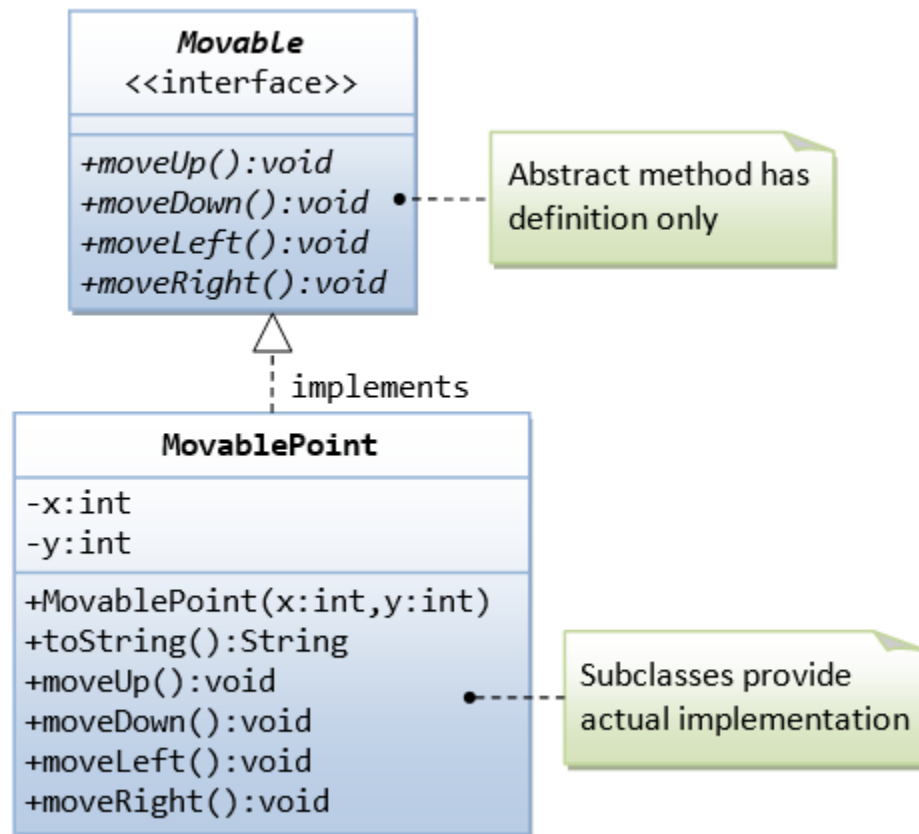


b)





c)



Bài 5: Các bài tự thiết kế, cài đặt

5.1. Chương trình Quản lý tài khoản ngân hàng.

Ngân hàng ABC muốn lưu trữ thông tin của mỗi tài khoản như sau:

Mỗi tài khoản chứa các thông tin:

Bài viết này được đăng tại [free tuts .net]

Số tài khoản ( Kiểu long)

Tên tài khoản (kiểu chuỗi)

Số tiền trong tài khoản (kiểu double)

Thiết kế lớp Account để lưu trữ các thông tin, lớp bao gồm các phương thức sau:

- Constructor: Có 2 constructor ( mặc định và đầy đủ tham số)
- Các phương thức get, set cho từng thuộc tính

- Phương thức toString để trả về chuỗi chứa toàn bộ thông tin tài khoản, yêu cầu định dạng tiền tệ.

Thêm các thông tin sau vào lớp Account:

- Hằng số lãi suất có giá trị khởi tạo 0.035
- Constructor có 2 đối số: số tài khoản, tên tài khoản. Constructor này sẽ khởi tạo số tiền mặc định là 50
- Phương thức nạp tiền vào tài khoản: Lấy số tiền hiện tại trong tài khoản + số tiền nạp vào
- Phương thức rút tiền: Lấy số tiền hiện tại trong tài khoản – (số tiền muốn rút + phí rút tiền)
- Phương thức đáo hạn: Mỗi lần đến kỳ đáo hạn thì số tiền trong tài khoản = số tiền trong tài khoản + số tiền trong tài khoản \* LAISUAT
- Phương thức chuyển khoản từ tài khoản này sang tài khoản khác

Chú ý: Mỗi thao tác phải kiểm tra số tiền nạp, rút, chuyển có hợp lệ hay không? (VD: tiền nhập vào < 0, tiền rút nhiều hơn tiền trong tài khoản thì thông báo không hợp lệ và yêu cầu nhập lại)

### Hướng dẫn:

Chúng ta sẽ lần lượt thực hiện theo các yêu cầu của bài toán:

- Đầu tiên chúng ta sẽ tạo class Account với các thuộc tính soTK, tenTK, soTienTrongTK.
- Tạo các constructor mặc định và constructor có tham số.
- Tạo các phương thức getter và setter.
- khởi tạo phương thức toString để hiển thị kết quả dưới dạng chuỗi.
- Khởi tạo phương thức nạp tiền với điều kiện số tiền phải lớn hơn hoặc bằng 0.
- Khởi tạo phương thức rút tiền với điều kiện số tiền rút phải bé hơn hoặc bằng số tiền có trong tài khoản + phí.
- Khởi tạo phương thức đáo hạn.
- Khởi tạo phương thức in kết quả theo Format.
- Tiếp đến chúng ta sẽ tạo class Main với các Menu theo yêu cầu của đề bài và gọi các phương thức đã tạo ở class Account.

## 5.2. Chương trình quản lý sinh viên.

Viết chương trình xây dựng đối tượng Sinh viên gồm có các thuộc tính sau:

- Mã sinh viên là số nguyên.
- Họ tên: chuỗi ký tự.
- Địa chỉ: chuỗi ký tự.
- Số điện thoại: là số bao gồm 7 chữ số.

Các thuộc tính khai báo private, định nghĩa các phương thức get/set cho từng thuộc tính.

Viết các constructor để khởi tạo đối tượng (constructor mặc định, constructor có tham số).

Override phương thức toString của lớp Object.

Xây dựng lớp cho phép nhập vào N sinh viên (dùng mảng). Xuất danh sách trên ra màn hình theo thứ tự tăng dần của mã sinh viên.

### **Hướng dẫn:**

Chúng ta sẽ đi lần lượt các yêu cầu của đề bài:

- Tạo class SinhVienNhap với các thuộc tính: maSV, hoTen, diaChi, sTD.
- Khởi tạo các constructor mặc định và constructor có tham số.
- Khởi tạo các phương thức getter và setter.
- Khởi tạo phương thức toString để hiển thị kết quả dưới dạng chuỗi.
- Tạo class Main để nhập vào số lượng sinh viên và các thông tin của sinh viên.
- Xuất thông tin của sinh viên theo thứ tự tăng dần của mã sinh viên.

5.3. 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:**

Vẽ class diagram cho phát biểu bài toán trên OOP:

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 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
11. Ở mỗi khoa, tìm ra sinh viên có điểm trung bình học kỳ cao nhất (ở bất kỳ học kỳ nào)
12. 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)
13. Ở mỗi khoa, thống kê số lượng sinh viên theo năm vào học. Ví dụ 2020: 100, 2019: 90, 2018: 120.

5.4. Phần mềm quản lý nhân viên của công ty được mô tả nghiệp vụ như sau:

Mỗi nhân viên (**Employee**) được công ty chia thành 3 loại sau: Nhân viên có kinh nghiệm lâu năm (**Experience**), nhân viên mới ra trường (**Fresher**), Nhân viên thực tập (**Intern**). Tất cả các **Employee** đều có các thuộc tính là: **ID**, **FullName**, **BirthDay**, **Phone**, **Email**, **Employee\_type**, **Employee\_count** và phương thức là **ShowInfo** để hiển thị thông tin của nhân viên đó (hiển thị thông tin nhân viên ra màn hình console).

**Trong đó :**

- **Employee\_type** có giá trị tương ứng là 0: Experience, 1: Fresher , 2: Intern (tùy vào người dùng nhập vào ứng viên loại nào)
- **Employee\_count** dùng để người dùng đếm số lượng nhân viên trong một đợt người dùng nhập nhân viên mới vào cơ sở dữ liệu. (**mỗi lần người dùng nhập thêm mới nhân viên thì thuộc tính Employee\_count của class Employee sẽ tăng lên 1**)

**Ngoài ra :**

- Đối với nhân viên **Experience** có thêm thuộc tính: Số năm kinh nghiệm (ExpInYear), Kỹ năng chuyên môn (ProSkill)

- Đối với nhân viên **Fresher** có thêm thuộc tính: Thời gian tốt nghiệp(Graduation\_date), Xếp loại tốt nghiệp (Graduation\_rank) , Trường tốt nghiệp (Education)
- Đối với nhân viên **Intern** có thêm thuộc tính: Chuyên ngành đang học (Majors), Học kì đang học (Semester), Tên trường đang học (University\_name)

**Lưu ý:** Tùy mỗi loại nhân viên, phương thức **showMe** sẽ được bổ sung thêm các thuộc tính của riêng loại nhân viên đó.

\*\* Ngoài ra mỗi nhân viên khi vào làm cần phải nộp bằng cấp nghề nghiệp đi kèm, bộ phận tuyển dụng cần quản lý các bằng cấp này. **Một nhân viên có thể có nhiều bằng cấp (Certificate).** Với mỗi bằng cấp có các thông tin bao gồm: **CertificatedID, CertificateName, CertificateRank, CertificatedDate.**

**Yêu cầu:**

1. Hãy thiết kế và viết code của chương trình trên làm sao để tuân thủ theo đúng mô hình **OOP** đã học, áp dụng đầy đủ 4 tính chất : bao đóng (*encapsulation*) , kế thừa (*inheritance*) , đa hình (*polymorphism*) , trừu tượng (*abstraction*).
2. Xác định và viết code constructor cho tất cả các class.
3. Xác định và viết code cho các **abstract method, abstract class, override/overload method, static field.**
4. Xác định IS- A, HAS- A relationship trong phần thiết kế code đã viết ở trên.
5. Sử dụng và giải thích được ý nghĩa của 2 keyword: **super, this** trong phần thiết kế code ở trên.
6. Hãy giải thích ý nghĩa của toán tử **instanceof**, hãy ứng dụng instanceof để **downcasting** một object **Employee** trở thành các object **Experience, Fresher hoặc intern** trong các yêu cầu 2, yêu cầu 3, yêu cầu 4 bên dưới.
7. Viết chức năng cho phép thêm, sửa (Nhập ID để xác định một nhân viên, nếu tồn tại cho phép người dùng chỉnh sửa thông tin của nhân viên), xóa(xóa theo ID) các loại nhân viên trên.
8. Viết các hàm kiểm tra tính hợp lệ của ngày sinh, email, tên và số điện thoại của nhân viên. Áp dụng các hàm này vào chức năng số 7.
9. Viết chương trình tìm tất cả các nhân viên intern.
10. Viết chương trình tìm tất cả các nhân viên experience .
11. Viết chương trình tìm tất cả các nhân viên fresher.
12. Tạo ra các BirthDayException, PhoneException, EmailException, FullNameException để ném ra ngoại lệ trong trường hợp birthday, email, và phone người dùng nhập vào không hợp lệ.

## PHỤ LỤC

### I. Mẫu báo cáo

#### **BÁO CÁO THỰC HÀNH BUỔI 2**

NỘI DUNG: PHẦN LẬP TRÌNH JAVA HƯỚNG ĐỐI TƯỢNG

##### **1. Thông tin chung**

*Họ tên:*

*Mã sinh viên:*

*Ngày thực hành:*

*Lớp:*

*Phòng máy:*

*Thời gian:*

##### **2. Nội dung báo cáo**

###### **Bài 1:**

.....

###### **Bài 2:**

.....

.....

##### **3. Ý kiến (nếu có)**