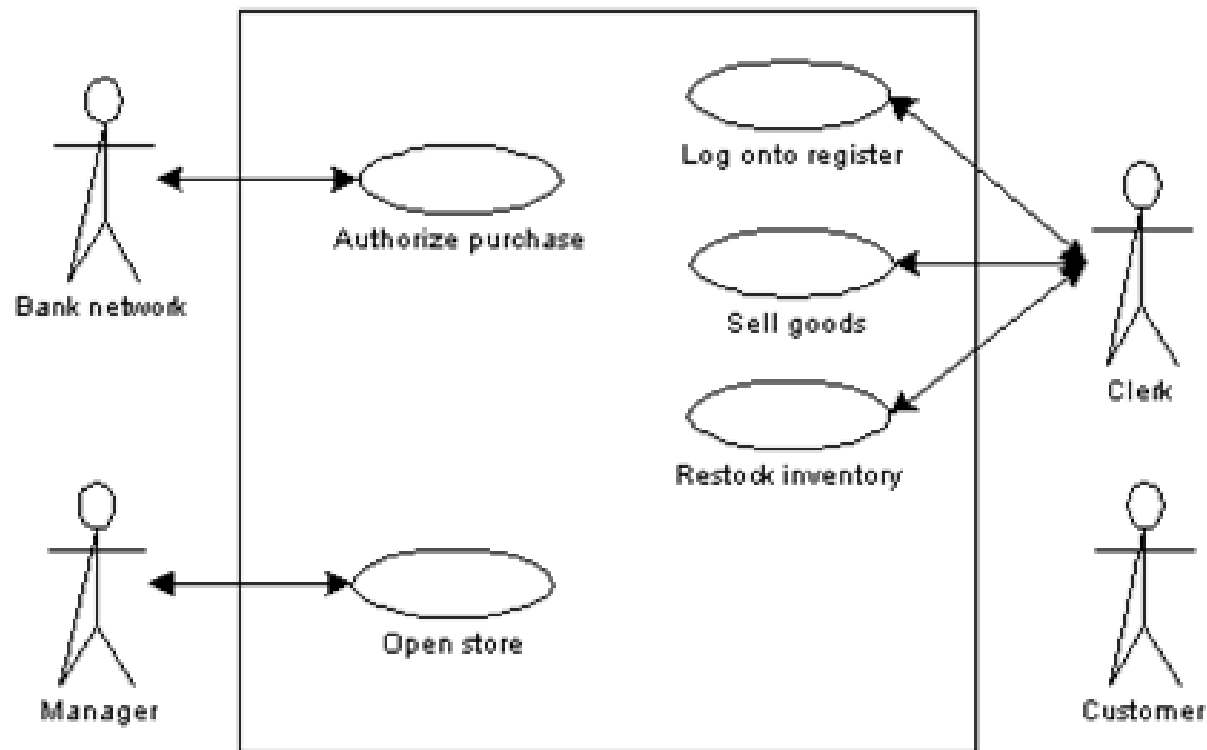




Bài tập tổng hợp (tiếp theo)

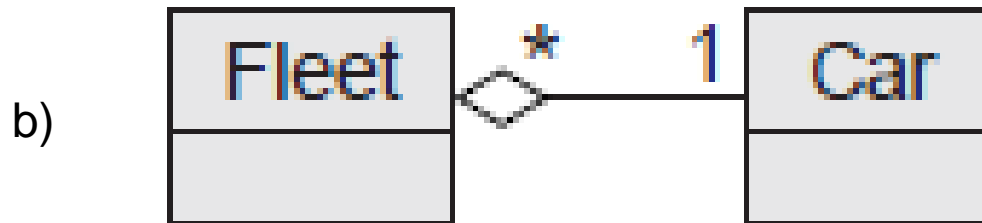
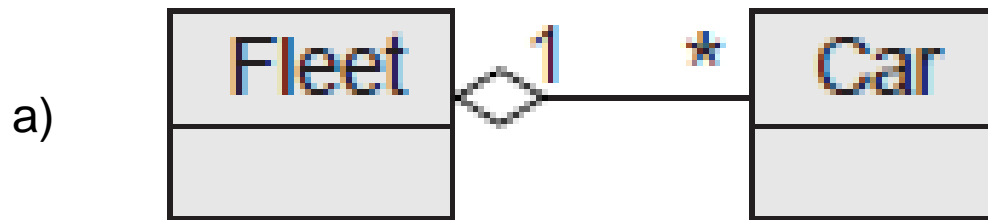
Quiz 0

- Cho biểu đồ use case như sau, xác định các tác nhân của hệ thống:



Quiz 1

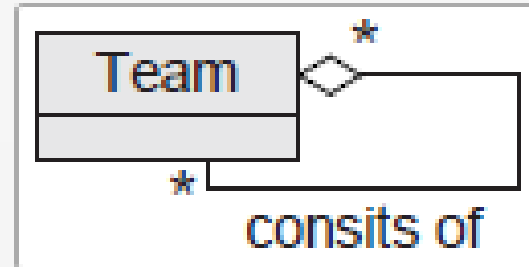
- Sơ đồ nào mô tả chính xác phát biểu sau:
"Đội xe cho thuê xe có nhiều xe, một xe thuộc đúng một đội xe"



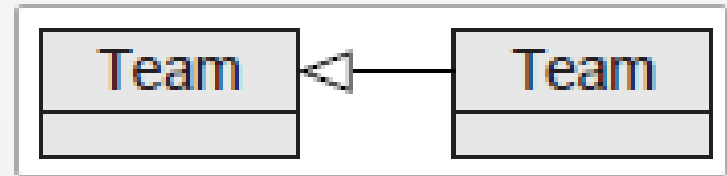
Quiz 2

- Sơ đồ nào mô tả chính xác phát biểu sau: "Một đội có thể bao gồm một số đội khác"

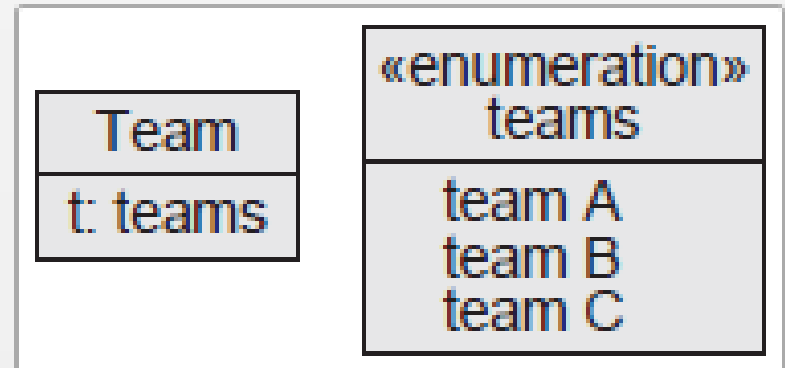
a)



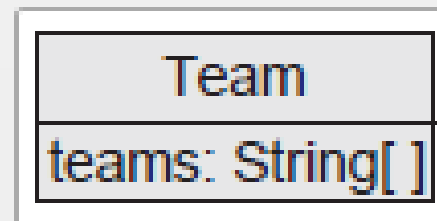
b)



c)



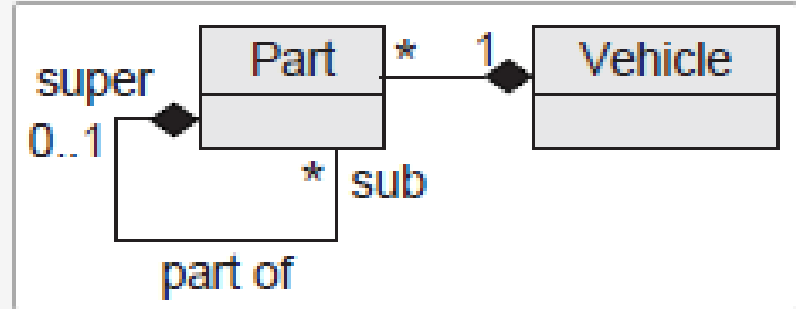
d)



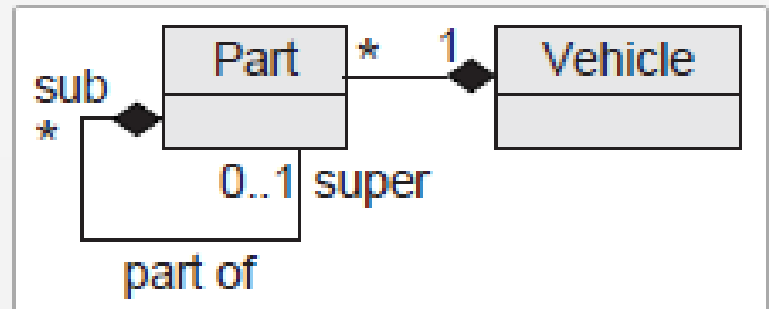
Quiz 3

- Sơ đồ nào mô tả chính xác phát biểu sau:
 - "Một chiếc xe bao gồm nhiều bộ phận. Các bộ phận lại bao gồm lẫn nhau."
 - Mỗi phần có thể được nằm trong tối đa một super-part.
 - Một super-part bao gồm nhiều phần phụ (sup-part)."

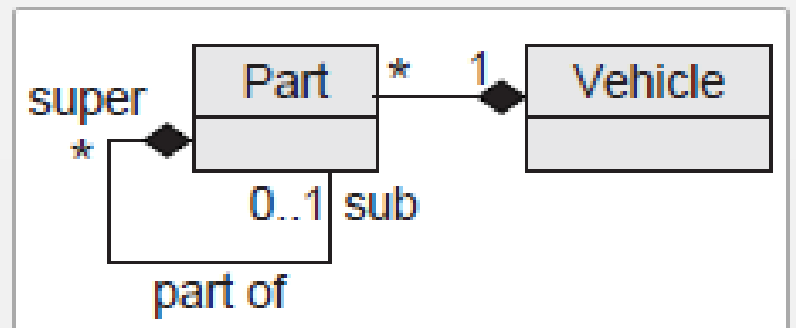
a)



b)

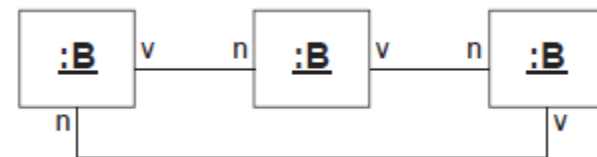
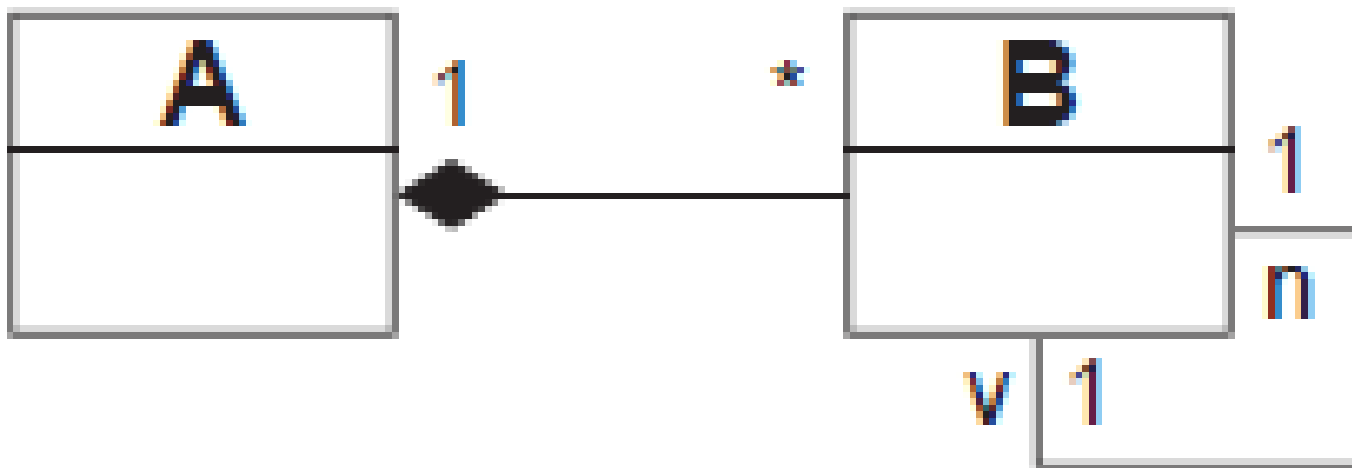


c)



Quiz 4

- Sơ đồ sau mô tả thông tin gì?





Quiz 5

- Cho biết kết quả khi thực hiện đoạn mã sau:

```
public class MyClass {  
    public static void main(String[] args) {  
        B b = new B("Test");  
    }  
}  
  
class A {  
    A() { this("1", "2"); }  
    A(String s, String t) { this(s + t); }  
    A(String s) { System.out.println(s); }  
}  
  
class B extends A {  
    B(String s) { System.out.println(s); }  
    B(String s, String t) { this(t + s + "3"); }  
    B() { super("4"); };  
}
```



Quiz 6

- a. Đoạn mã dưới đây có lỗi gì không?

```
abstract class ABC {  
    void firstMethod() {  
        System.out.println("First Method");  
    }  
    void secondMethod() {  
        System.out.println("Second Method");  
    }  
}
```

- b. Lớp nào là lớp trừu tượng, lớp nào có thể tạo đối tượng?

```
abstract class A {  
    }  
  
class B extends A {  
    }
```




Quiz 7

- 1. Khai báo nào là hợp lệ trong một interface?
 - a. `public static int answer = 42;`
 - b. `int answer;`
 - c. `final static int answer = 42;`
 - d. `public int answer = 42;`
 - e. `private final static int answer = 42;`
- 2. Một lớp có thể kế thừa chính bản thân nó không?
- 3. Chuyện gì xảy ra nếu lớp cha và lớp con đều có thuộc tính trùng tên?
- 4. Phát biểu “Các phương thức khởi tạo cũng được thừa kế xuống các lớp con” là đúng hay sai?
- 5. Có thể xây dựng các phương thức khởi tạo cho lớp trừu tượng không?
- 6. Có thể khai báo phương thức `protected` trong một giao diện không?



Quiz 8

- Kết quả khi biên dịch và thực thi đoạn chương trình sau:

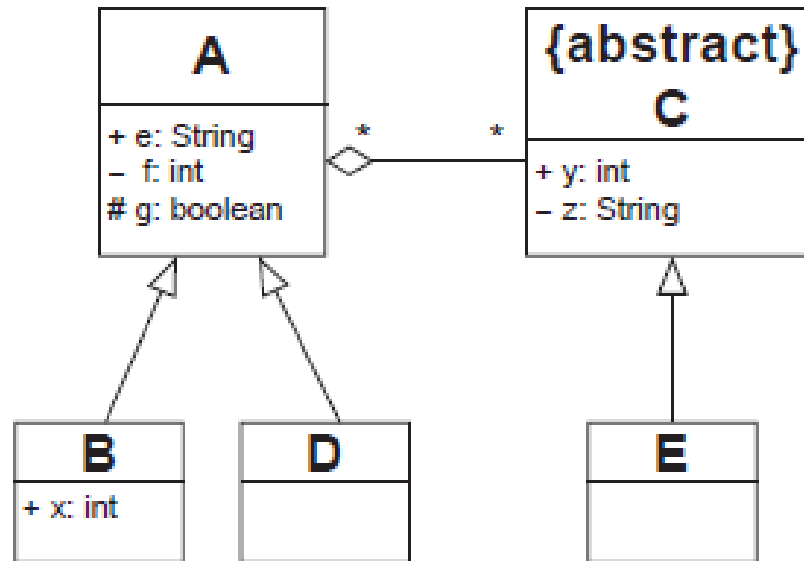
```
class A {  
    int i = 10;  
}
```

```
class B extends A {  
    int i = 20;  
}
```

```
public class MainClass {  
    public static void main(String[] args) {  
        A a = new B();  
        System.out.println(a.i);  
    }  
}
```

Quiz 9

Cho biểu đồ lớp như hình dưới đây, hãy xác định các phát biểu đi kèm là đúng hay sai?

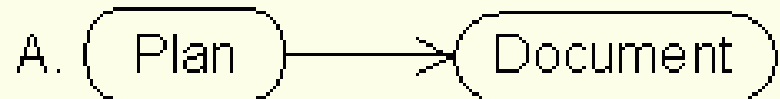


- a) Một đối tượng của lớp A có thể truy xuất thuộc tính y
- b) Các đối tượng của lớp B và C có thể truy xuất thuộc tính f
- c) Một đối tượng của lớp A có thể chứa hoặc không các đối tượng của lớp C
- d) Các đối tượng của lớp B và D có thể truy xuất thuộc tính g

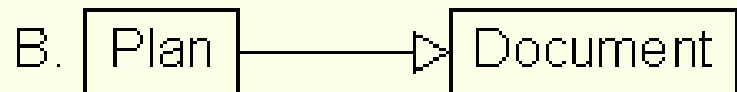
Quiz 10

Hãy gắn mô hình UML với giải thích phù hợp?

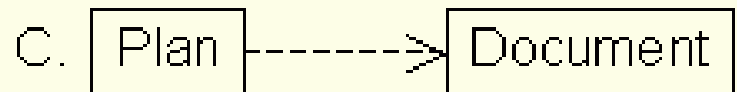
___ A plan is dependent on a document.



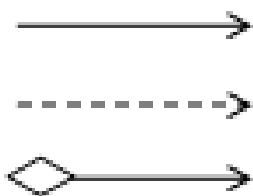
___ Plan first, then document.



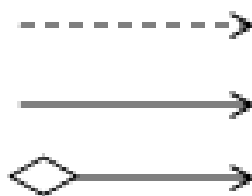
___ A plan is a special type of document.



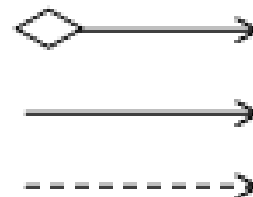
Phương án nào dưới đây cho thứ tự các loại liên kết trong UML đi từ tổng quát đến cụ thể?



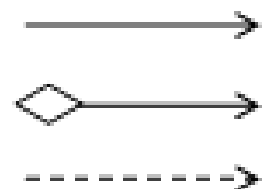
a.



b.



c.

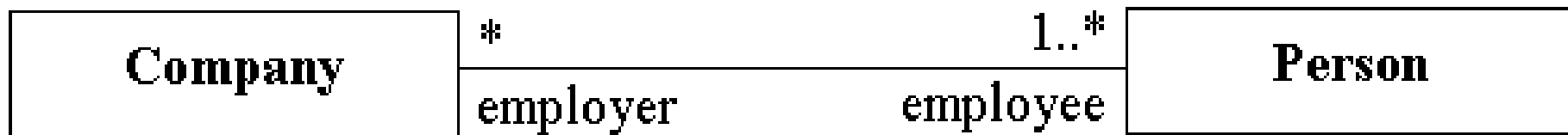


d.



Quiz 11

Cho biểu đồ lớp như dưới đây, hãy trả lời các câu hỏi đi kèm:

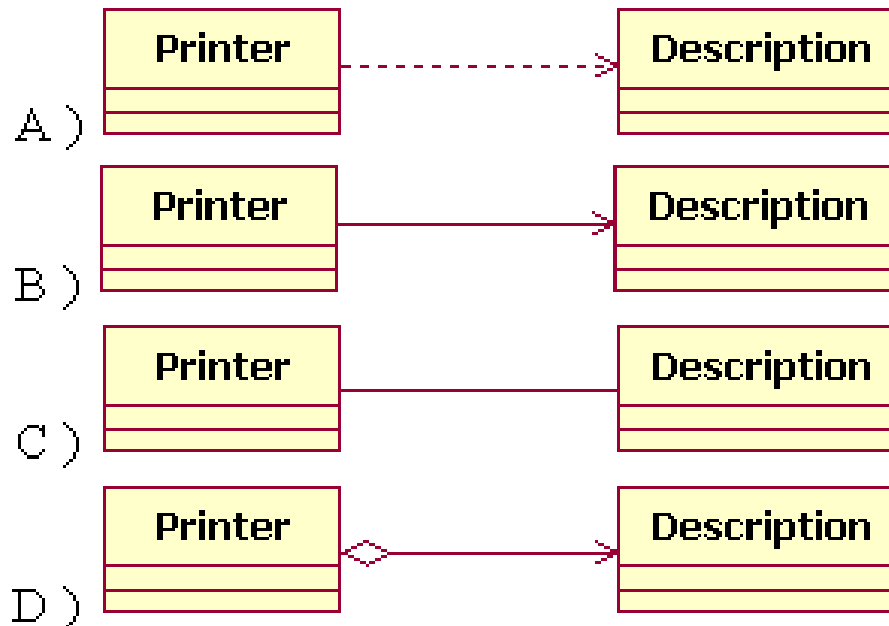


- a. Có thể có công ty mà không có nhân viên nào không?
- b. Một người có thể bị thất nghiệp (không làm cho công ty nào cả) không?

Quiz 12

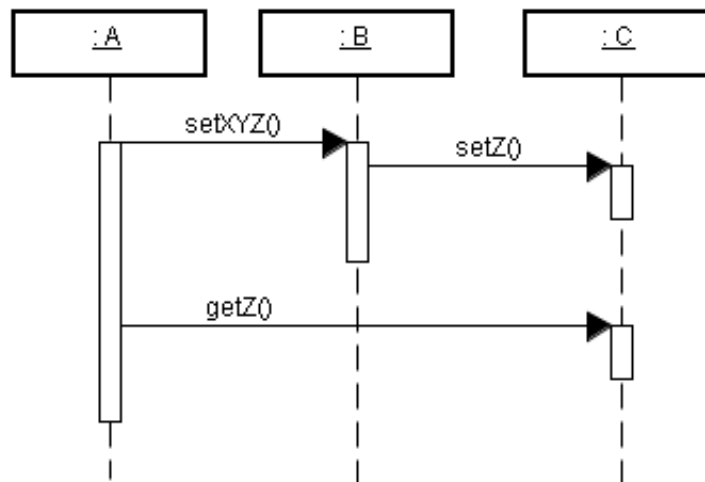
- Cho đoạn mã nguồn dưới đây, hãy cho biết mối quan hệ giữa hai lớp "Printer" và "Description"

```
public class Printer {  
    public void printDescription(Description desc) {  
        print(desc.getName());  
        print(desc.getOrganization());  
    }  
}
```

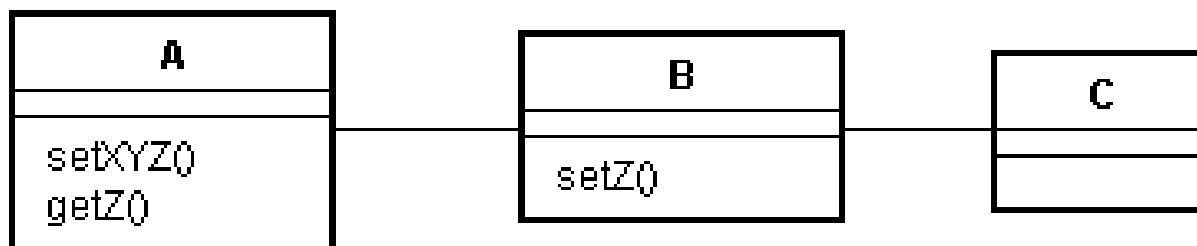


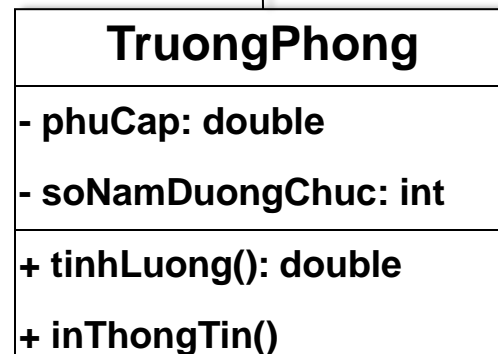
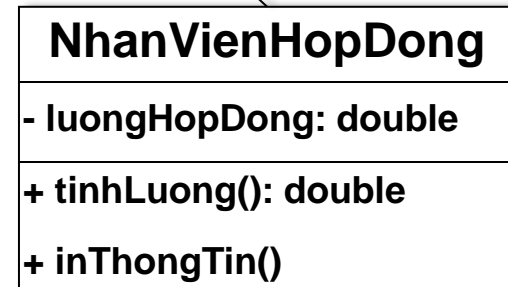
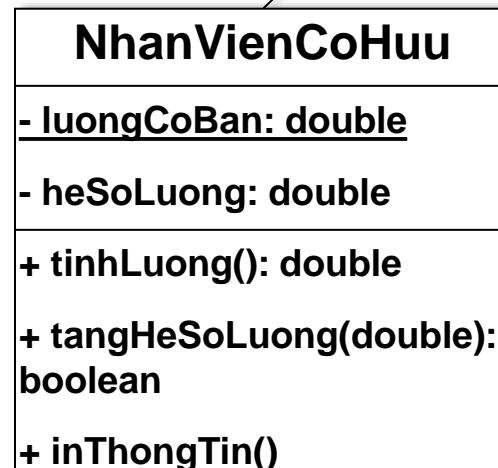
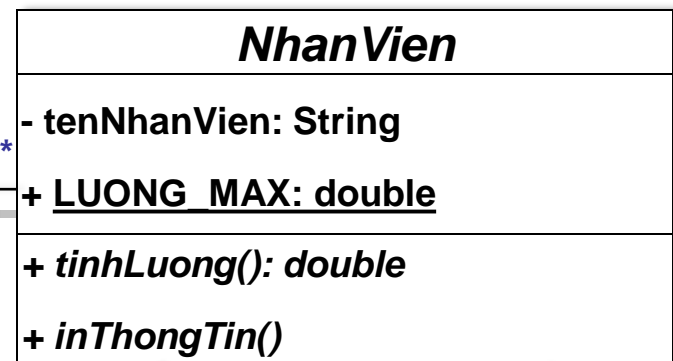
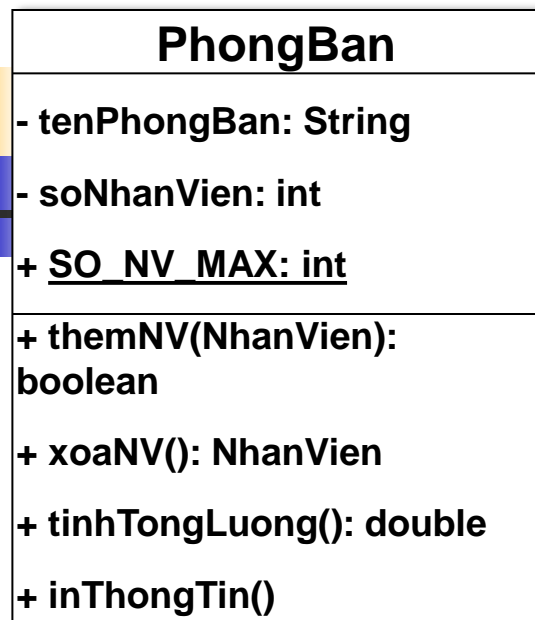
Quiz 13

- Cho biểu đồ tuần tự như dưới đây:



- Từ biểu đồ trên người ta xây dựng biểu đồ lớp tương ứng như sau, hãy cho biết biểu đồ lớp này có chính xác không? Nếu không hãy sửa lại cho đúng:





Bài tập 1

- Xây dựng các lớp như biểu đồ ở hình bên
 - Sửa lớp NhanVien thành lớp CanBoCoHuu
 - Cho lớp CanBoCoHuu thừa kế lớp abstract NhanVien
 - Tính tổng lương của tất cả nhân viên trong phòng ban



Bài tập 2

Xây dựng biểu đồ lớp cho 1 trò chơi như sau (trên 1 máy tính duy nhất):

- Số lượng người chơi: luôn là 4. Nếu có < 4 người chơi thật tham gia, máy tính sẽ tự động thêm 1 số người chơi ảo cho đủ 4 người.
- Có 4 quân súc sắc. Mỗi quân súc sắc, xác suất cho 1 mặt là 20%, các mặt còn lại là 16% (lần lượt 4 quân cho các mặt 1, 2, 3, 4 chấm)
- Mỗi người chơi ít nhất có các thuộc tính và hành vi cơ bản sau:
 - Tên
 - Số điểm đang có
 - Riêng người chơi ảo có thêm cách thức biểu lộ thất bại khác nhau. (Lưu ý có tối đa 4 người chơi ảo)
- Có 1 trọng tài, điều khiển cuộc chơi. Trọng tài có nhiệm vụ:
 - Chỉ định người chơi tiếp theo.
 - Tính điểm cho người chơi. Nếu tổng điểm cũ và điểm vừa gieo của 1 người chơi là 21 → thắng cuộc, kết thúc cuộc chơi. Nếu điểm cũ + điểm vừa gieo lớn hơn 21 → tính điểm của người chơi là 0. Trường hợp còn lại, cộng điểm bình thường.
 - Tuyên bố người thắng cuộc. Những người chơi ảo thua cuộc sẽ lần lượt thực hiện cách thức biểu lộ thất bại của mình.
- Mỗi người chơi, khi đến lượt, sẽ nhận ngẫu nhiên 1 quân súc sắc và gieo¹⁸

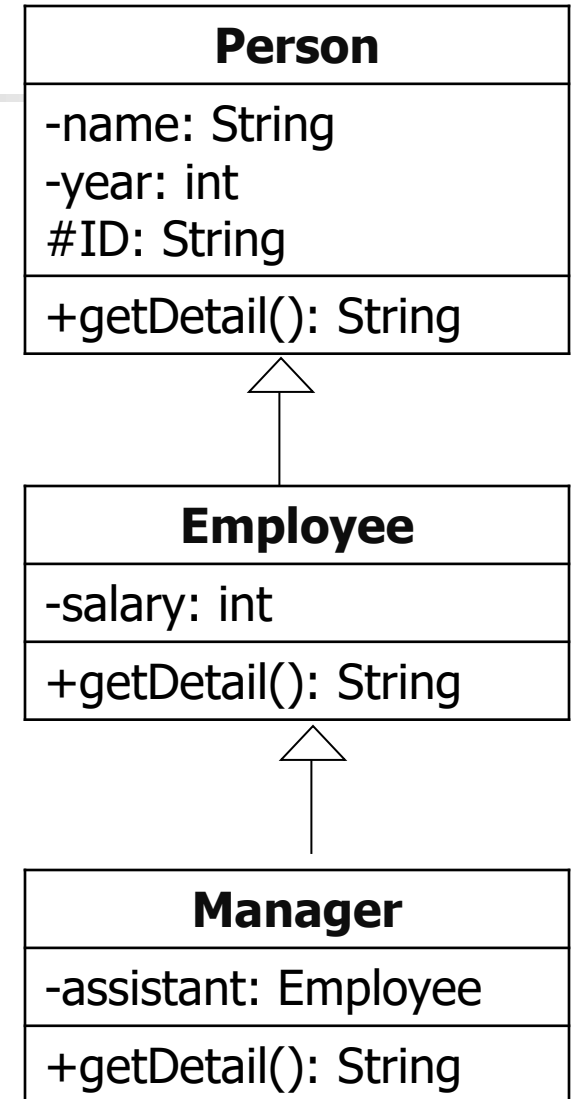


Bài tập 3

- Cần xây dựng HT (máy) làm bánh Pizza tự động ở 1 công ty. Pizza có các thuộc tính: tên loại bột (dough), tên loại nước sốt (sauce), tên loại hương vị (flavour).
- Có 5 loại bánh Pizza (Hawaii Pizza, NewYork Pizza, Berlin Pizza, London Pizza, Hanoi Pizza). Mỗi loại có công thức riêng và có thùng chế biến tương ứng (Hawaii Pizza Builder, NewYork Pizza Builder, Berlin Pizza Builder, London Pizza Builder, Hanoi Pizza Builder). Mỗi thùng chế biến ít nhất cung cấp các phương thức tạo bột, tạo hương vị, tạo nước sốt cho bên ngoài gọi đến.
- Mỗi máy làm bánh được cài đặt tối đa 4 thùng chế biến (có thể ít hơn, và có thể tháo bớt, thêm các thùng dễ dàng). Tại 1 thời điểm, mỗi thùng chế biến chỉ tạo được 1 bánh Pizza, nhưng 4 thùng có thể cùng lúc tạo 4 bánh Pizza)
- Nhân viên trong công ty chỉ cần chọn 1 loại bánh trong danh sách (tùy cài đặt các thùng chế biến), chọn loại nguyên liệu làm bánh, đợi, nhận bánh ở khay ra (OutputTray) & thưởng thức. Khi nhận yêu cầu từ nhân viên, máy sẽ kích hoạt thùng chế biến tương ứng, thực hiện lần lượt tạo bột, tạo hương vị và tạo nước sốt, trả lại bánh Pizza mong muốn cho nhân viên.
- Lưu ý: do việc làm bánh mất thời gian, nên phải giải quyết được nhiều yêu cầu làm bánh gửi liên tục (ví dụ một nhân viên đặt bánh, chưa nhận được, một nhân viên khác lại đặt bánh tiếp).

Bài tập 4

- Cài đặt 3 lớp như trên biểu đồ (Tự tạo phương thức getter và setter **ít nhất có thể**, sao cho đảm bảo nguyên lý đóng gói & che dấu dữ liệu. Tạo các phương thức khởi dựng **ít nhất có thể**)
- Cài đặt lớp Test:
 - Nhập vào **một công ty** gồm ba mảng: 20 Person, 10 Employee, 5 Manager (đầy đủ dữ liệu). Khi nhập dữ liệu cho Manager, lưu ý assistant phải nằm trong số 10 Employee nói trên
 - Tính xem có bao nhiêu người làm trợ lý (assistant) trong công ty (Một Employee được làm trợ lý cho nhiều Manager)
 - Nhập vào 1 Employee, kiểm tra xem trong công ty đã có Employee này chưa
 - Tính tổng lương toàn công ty, in ra người có lương cao nhất
- Lưu ý:
 - Có thể tạo thêm các phương thức tiện ích





Bài tập 5

Cần XD hệ thống ĐHNĐ cho một tòa nhà 10 tầng (80 phòng):

- Có hệ thống điều hòa tổng, cho mọi hành lang, phòng, ...
- 10 phòng đặc biệt có hệ thống điều hòa riêng
- Điều hòa tổng và điều hòa riêng đều có chức năng:
 - Tăng/Giảm nhiệt độ, Bật/Tắt, Thông báo về nhiệt độ hiện thời, Tăng/Giảm tốc độ quạt gió, Bật/Tắt chế độ tuyết.
- Có máy điều khiển trung tâm cho người điều hành.
 - Thực hiện các chức năng kể trên cho hệ thống điều hòa tổng
 - Khóa/mở khóa từng điều hòa riêng



Bài tập 5(2)

- Tại 10 phòng đặc biệt:
 - có điều khiển riêng cho hệ thống điều hòa của phòng đó.
 - Người dùng có thể thực hiện các chức năng nói trên.
 - Tuy nhiên, khi điều hòa riêng của phòng bị khóa, bất kỳ chức năng nào được yêu cầu đều không được thực hiện, (thông báo là đang bị khóa)
 - Nếu hệ thống điều hòa riêng được mở khóa, cửa chặn thông gió (của phòng này với hệ thống điều hòa tổng) sẽ tự động được đóng vào. Ngược lại, nếu khóa, cửa chặn thông gió lại mở ra.
- Người điều hành được HT hỗ trợ dự toán tiền điện cho từng tháng:
 - Giá tiền điện cho điều hòa tổng = $\sum (30 - \text{Nhiệt độ điều hòa}) * \Delta t * 1000$ (nghìn VNĐ)
 - Giá tiền điện cho điều hòa riêng = $\sum (30 - \text{Nhiệt độ điều hòa}) * \Delta t * 20$ (nghìn VNĐ)
 - Trong đó, **Δt** tính theo đơn vị giờ, là thời gian **Nhiệt độ điều hòa** được duy trì.



Bài tập 5(3)

Yêu cầu:

- Vẽ biểu đồ use case
- Thiết kế biểu đồ lớp để xây dựng hệ thống. Giải thích
- Viết code Java cho lớp ứng với đối tượng Điều hòa tổng. Nếu lớp này kế thừa/Thực thi 1 lớp/giao diện khác, viết code đầy đủ cho lớp/giao diện đó.
- Viết code cho phương thức dự toán tiền điện của toàn bộ hệ thống



Bài tập 6

- Giả thiết có lớp TimeLogger như sau

```
public class TimeLogger{  
    public void confirmTwelveAM(){  
        // log thông báo  
    }  
    public void confirmTwelvePM(){  
        // log thông báo  
    }  
    public void someMethod(){  
        // log thông tin gì đó  
    }  
}
```

- Khi TimeLogger thực thi, có thể cần ghi ra file x.txt, y.txt, z.txt, ... hoặc hiển thị kết quả ra giao diện GUI nào đó, ... tùy hoàn cảnh sử dụng.
- **Yêu cầu: thiết kế lớp TimeLogger đảm bảo tính năng trên.**



Bài tập 7

- Một ngân hàng (**Bank**) có 2 loại tài khoản (**Account**) : tài khoản thường (**NormalAccount**) và tài khoản cao cấp (**VipAccount**) với thuộc tính: tên TK (**name**), định danh (**id**), số dư (**balance**). Riêng TK thường có thuộc tính **percent**: % lãi suất tháng và phương thức dự tính lãi suất (**getInterest**) có được theo số tháng đưa vào (tính lũy tiến theo tháng).
- Chủ TK được gửi (**credit**) và rút (**debit**) một lượng nguyên tiền. Với TK thường, số tiền rút ra không lớn hơn số dư hiện có, và rút ra tối đa là 10 triệu. Với TK VIP được rút tối đa 50 triệu và lượng tiền âm không quá 1 tỷ.
- Nhờ thực thi giao diện chuyển tiền (**Transferable**), 2 TK đều cung cấp phương thức chuyển tiền (**transferMoney**). có tham số là đối tượng TK đích & số tiền cần chuyển. Khi thực thi, TK nguồn bị trừ, TK đích được cộng. Riêng TK nguồn là TK thường, số dư bị trừ 5% số tiền cần chuyển. Muốn thành công, sau khi tính toán, số dư TK nguồn phải lớn hơn 0. Khi đó, một đối tượng giao dịch (**Transaction**) chứa đối tượng TK nguồn, đích & số tiền đã chuyển sẽ được tạo và lưu vào danh sách các giao dịch của ngân hàng
- Ngân hàng cần có phương thức tính tổng số tiền ngân hàng đang có (**getTotalMoney**) và tổng số tiền đã chuyển đổi qua lại giữa các TK từ lúc bắt đầu hoạt động (**getTotalMoneyTransfer**).
- Câu hỏi:
 - Vẽ thiết kế biểu đồ lớp (UML class diagram) để xây dựng phần mềm quản lý ngân hàng trên.
 - Viết mã nguồn cho lớp VipAccount, lớp Bank.