

Java: Bài tập phần Class

Bài tập 1:

Viết **class NhanVien** gồm các thuộc tính:

- + Tên
- + Tuổi
- + Địa chỉ
- + Tiền lương (kiểu double)
- + Tổng số giờ làm (kiểu int)

Constructor không tham số. **Constructor** đầy đủ tham số. Các hàm **get/set** và các phương thức:

- **void inputInfo()** : Nhập các thông tin cho nhân viên từ bàn phím
- **void printInfo()** : in ra tất cả các thông tin của nhân viên
- **double tinhThuong()**: Tính toán và trả về số tiền thưởng của nhân viên theo công thức sau:

Nếu tổng số giờ làm của nhân viên ≥ 200 thì thưởng = lương * 20%

Nếu tổng số giờ làm của nhân viên < 200 và ≥ 100 thì thưởng = lương * 10%

Nếu tổng số giờ làm của nhân viên < 100 thì thưởng = 0

Bài tập 2:

Tạo lớp **Student**, lưu trữ các thông tin một sinh viên:

- Mã sinh viên: chứa 8 kí tự
- Điểm trung bình: từ 0.0 – 10.0
- Tuổi: Phải lớn hơn hoặc bằng 18
- Lớp: Phải bắt đầu bởi kí tự 'A' hoặc kí tự 'C'

Constructor không tham số. **Constructor** đầy đủ tham số. Các hàm **get/set**

- phương thức **inputInfo()**, nhập thông tin Student từ bàn phím
- phương thức **showInfo()**, hiển thị tất cả thông tin Student
- Viết phương thức xét xem Student có được học bổng không? Điểm trung bình trên 8.0 là được học bổng

Bài tập 3:

Trung tâm Aptech thực hiện trao học bổng cho các học viên xuất sắc và đáp ứng đủ các yêu cầu sau:

- Là học viên đăng ký khóa học HDSE
- Có điểm tổng kết $\geq 75\%$
- Không vi phạm nội quy của trung tâm

d/ Các kì thi chỉ thi lần đầu tiên

Các dữ liệu 1 2 3 4 của 1 học viên được nhập từ bàn phím.

Viết chương trình xem học viên đó có được học bổng không.

Bài tập 4:

Tạo **class SoNguyenTo** gồm:

- biến **a** lưu trữ 1 số nguyên tố.

Constructor không tham số. **Constructor** 1 tham số:

public SoNguyenTo(int x){} Nếu **x** là số nguyên tố thì lưu **x** vào biến **a**. Nếu không thì in ra màn hình: **x không phải là số nguyên tố, không lưu trữ.**

Hàm **boolean isSoNguyenTo(int x){}** kiểm tra số **x** có phải số nguyên tố không.

Hàm **int timSoNguyenToTiepTheo(){}** tìm và trả về số nguyên tố liền sau số nguyên tố **a**.

Hàm **get/set** biến **a**; Nếu tham số truyền vào hàm **set** là 1 số nguyên tố thì mới gán vào **a**. Nếu không thì hiển thị thông báo: không set.

Ở hàm **Main**. Khai báo 1 đối tượng thuộc class **SoNguyenTo** và test các hàm đã viết.

Bài tập 5:

Xây dựng lớp tam giác (Triangle) có các thành phần:

* Các thuộc tính: 3 cạnh **a, b, c** của tam giác.

* Các phương thức:

- Nhập độ dài 3 cạnh
- Xác định kiểu tam giác
- Tính chu vi tam giác
- Tính diện tích tam giác

Bài tập 6:

Xây dựng lớp hình chữ nhật (Rectangle) có các thành phần sau:

* Các thuộc tính: chiều dài, chiều rộng.

* Các phương thức:

- Nhập chiều dài, chiều rộng
- Tính diện tích
- Tính chu vi

Bài tập 7:

Xây dựng lớp Phân số (Fraction) có các thành phần sau:

* Các thuộc tính: Tử số, mẫu số.

* Hàm tạo không đối số và hai đối số để khởi tạo giá trị cho tử số và mẫu số.

- * Các phương thức setter và getter.
- * Các phương thức:
 - Nhập phân số
 - In Phân số
 - Rút gọn phân số
 - Nghịch đảo phân số
 - add(), sub(), mul(), div() tương ứng để thực hiện cộng, trừ, nhân, chia hai phân số cho nhau.

Bài tập 8:

Xây dựng lớp số phức có các thành phần sau:

+ Các thuộc tính:

- Phần thực
- Phần ảo

+ Các hàm thành phần:

- Cộng hai số phức
- Trừ hai số phức
- Nhân hai số phức
- Chia hai số phức

Nhập vào 2 số phức và thực hiện các phép toán trên hai số phức đó.

Bài tập 9:

Xây dựng lớp Vector có các thành phần sau:

+ Các thuộc tính:

- Hoành độ đầu
- Tung độ đầu
- Hoành độ cuối
- Tung độ cuối

+ Các phương thức

- Kiểm tra hai vector có bằng nhau không?
- Tính góc giữa 2 vector
- Tính module của 2 vector
- Kiểm tra hai vector có cùng phương không?
- Cộng hai vector
- Trừ hai vector
- Nhân hai vector

Nhập vào 2 vector và thực hiện các phép toán trên hai vector đó.

Bài tập 10:

Xây dựng lớp đa thức (polynomial) và các phép toán trên đa thức. Thực hiện nhập vào 2 đa thức và tính tổng, tích của nó.

Bài tập 11:

Xây dựng lớp ma trận và các phép toán trên ma trận. Thực hiện nhập vào 2 ma trận và tính tổng, tích của nó.

Bài tập 12:

Xây dựng lớp đa giác, hình bình hành thừa kế từ đa giác, hình chữ nhật thừa kế từ hình bình hành và hình vuông thừa kế từ hình chữ nhật. Nhập vào các thuộc tính cần thiết của mỗi hình và tính chu vi, diện tích của hình đó.

Bài tập 13:

Xây dựng lớp điểm, lớp elip thừa kế từ lớp điểm, lớp đường tròn thừa kế từ lớp elip. Nhập vào các thuộc tính cần thiết của elip và tính diện tích.

Bài tập 14:

Xây dựng lớp tam giác, lớp tam giác vuông, tam giác cân thừa kế từ lớp tam giác. Lớp tam giác đều thừa kế từ lớp tam giác cân.

Bài tập 15:

Mô phỏng sự hoạt động của một chiếc đèn pin. Với hai nhóm đối tượng cơ bản là Đèn (FlashLamp) và Pin (Battery). Đối tượng pin mang trong mình thông tin về trạng thái năng lượng, đối tượng đèn sử dụng một đối tượng pin để cung cấp năng lượng cho hoạt động chiếu sáng.

Mô tả chi tiết lớp các đối tượng pin và đèn như sau:

FlashLamp

- status: boolean
- battery: Battery
- + FlashLamp()

- + setBattery(Battery): void
- + setBatteryInfo(): int
- + turnOn(): void
- + turnOff(): void

Battery

- energy: int
- + Battery()
- + setEnergy(int): void
- + getEnergy(int): int
- + decreaseEnergy(): void

1. Xây dựng lớp Pin (Battery) với các thuộc tính và các phương thức đã cho như trong sơ đồ trên. Trong đó:

- Thuộc tính: energy có kiểu số nguyên thể hiện năng lượng của Pin.

- Phương thức:

- + Battery(): Khởi tạo mặc định giá trị năng lượng Pin (energy) là 10

- + void setEnergy(int energy): Thiết đặt lại giá trị mới cho pin (thực hiện việc sạc pin)

- + int getEnergy(): Trả về thông tin năng lượng pin đang có

- + void decreaseEnergy(): mỗi lần Pin được sử dụng, năng lượng của Pin sẽ giảm đi 2 đơn vị.

2. Xây dựng lớp FlashLamp với các thuộc tính và phương thức như trong sơ đồ trên. Trong đó:

- Thuộc tính:

- + boolean status: trạng thái của đèn, nếu status = true tức đèn được bật, ngược lại đèn tắt.

- + Battery battery: pin của đèn

- Phương thức:

+ FlashLamp(): khởi tạo trạng thái đèn tắt và chưa có pin

+ void setBattery(Battery b): nạp pin cho đèn

+ int getBatteryInfo(): lấy về năng lượng Pin của đèn

+ void turnOn(): In ra thông tin đèn có sáng hay không (trạng thái đèn là true và năng lượng > 0)

+ turnOff(): In ra thông tin đèn không sáng.

3. Xây dựng lớp TestFlashLamp chứa phương thức main() với kịch bản như sau:

- Khai báo và khởi tạo một đối tượng Battery

- Khai báo và khởi tạo một đối tượng FlashLamp

- Lắp pin cho đèn

- Bật và tắt đèn pin 10 lần

- Hiện thị ra màn hình mức năng lượng còn lại trong pin

Bài tập 16:

Tạo class SoNguyenTo gồm:

biến a lưu trữ 1 số nguyên tố.

Constructor không tham số.

Constructor 1 tham số:

public SoNguyenTo(int x){} Nếu x là số nguyên tố thì lưu x vào biến a. Nếu không thì in ra màn hình: x không phải là số nguyên tố, không lưu trữ.

Hàm boolean isSoNguyenTo(int x){} kiểm tra số x có phải số nguyên tố không.

Hàm int timSoNguyenToTiepTheo(){} tìm và trả về số nguyên tố liền sau số nguyên tố a.

Hàm get/set biến a; Nếu tham số truyền vào hàm set là 1 số nguyên tố thì mới gán vào a. Nếu không thì hiển thị thông báo: không set.

Ở hàm Main: Khai báo 1 đối tượng thuộc class SoNguyenTo và test các hàm đã viết.

Bài tập 17:

Câu hỏi 1:

Một lớp gọi là MyPoint, thể hiện một mô hình điểm hai chiều (2D) gồm hai tọa độ x và y, được thiết kế theo dạng sơ đồ lớp. Lớp MyPoint bao gồm:

- Hai biến thể hiện x (kiểu int) và y (int).
- Hàm tạo không đối số dùng để khởi tạo một điểm có tọa độ (0,0).
- Một hàm tạo dùng để khởi tạo một điểm với tọa độ đã cho theo người dùng.
- Các phương thức getter và setter cho các biến thể hiện x và y.
- Một phương thức setXY() để set giá trị cho cả x và y.
- Phương thức toString() trả về mô tả chuỗi theo định dạng "(x, y)".
- Một phương thức gọi là distance(int x, int y) trả về khoảng cách từ điểm này tới điểm khác tại các tọa độ (x, y) đã cho.
- Một phương thức tải chồng distance(MyPoint point1) trả về khoảng cách từ điểm hiện thời tới điểm point1.

Yêu cầu dành cho bạn: Viết mã lệnh xây dựng lớp MyPoint ở trên. Rồi viết một lớp có tên TestMyPoint để kiểm thử chương trình của bạn.

Gợi ý:

```
// Tải chồng phương thức distance()
public double distance(int x, int y) {
    int xDiff = this.x - x;
    int yDiff = ...
```

```

return Math.sqrt(xDiff * xDiff + yDiff * yDiff);
}
public double distance(MyPoint amother) {
int xDiff = this.x - another.x;
...
}
// Chương trình kiểm thử:
MyPoint p1=new MyPoint(3,0);
MyPoint p2=new MyPoint(0,4);
...
// Kiểm thử việc tải chồng phương thứcdistance():
System.out.println(p1.distance(p2));
System.out.println(p1.distance(5,6));

```

Câu hỏi 2:

Một lớp gọi là MyCircle, nó miêu tả một hình tròn có tâm là (x,y) và một bán kính. Lớp MyCircle sử dụng một thể hiện của lớp MyPoint (đã tạo ở câu 1) làm tâm của nó. Lớp MyCircle bao gồm:

- Hai biến thể hiện private: tâm (một thể hiện của MyPoint) và bán kính (có kiểu int).
- Một hàm tạo để khởi tạo hình tròn với tâm có tọa độ (x,y) và bán kính được đưa ra từ người dùng.
- Một hàm tạo khác để khởi tạo một hình tròn với tâm là thể hiện của lớp MyPoint và bán kính tương ứng.
- Các phương thức getter và setter.
- Phương thức toString() trả về một chuỗi mô tả hình tròn hiện thời theo dạng "Hình tròn @ (x,y) bán kính=r".
- Phương thức getArea() trả về diện tích của hình tròn.

Bạn hãy xây dựng lớp MyCircle. Ngoài ra, bạn viết lớp TestMyCircle để kiểm thử chương trình của bạn.

Bài tập 18: Hệ thống quản lý sản phẩm

1. Tạo một lớp có tên Product bao gồm các thuộc tính và phương thức sau:

- String Name
- String Description
- double Price // $0 < \text{Price} \leq 100$
- int[] rate // lưu các đánh giá của người dùng cho sản phẩm, giá trị từ 1 - 5
- void viewInfo() // hiển thị tên, giá và mô tả về sản phẩm

2. Tạo lớp Shop gồm các thuộc tính và phương thức sau:

- ArrayList ProductList // lưu danh sách các sản phẩm của shop
- void addProduct() // yêu cầu người dùng nhập thông tin của sản phẩm rồi lưu vào ProductList
- void removeProduct() // yêu cầu người dùng nhập vào tên sản phẩm sau đó tìm và xóa sản phẩm có tên tương ứng trong ProductList
- void iterateProductList() // hiển thị các sản phẩm trong ProductList, gọi phương thức viewInfo() của lớp Product, tính trung bình cộng đánh giá cho từng sản phẩm và hiển thị thông tin ra màn hình.
- void searchProduct() // yêu cầu người dùng nhập vào 2 số, sau đó tìm và hiển thị thông tin của những sản phẩm có giá nằm giữa hai số đó.

3. Tạo menu:

PRODUCT MANAGEMENT SYSTEM

Add new product
Remove product
Iterate product list
Search product
Exit

và thực thi các phương thức tương ứng trong lớp Shop với mỗi mục chọn.

Câu hỏi thêm:

Tạo thêm một mục trong Menu ứng với phương thức gọi là SortProduct() đặt trong lớp Shop để sắp xếp các sản phẩm trong ProductList theo giá.