1. Viết chương trình nhập vào một mảng các số nguyên từ bàn phím. Thực hiện:

- In lại các phần từ mảng đã nhập.

- Tính tổng các phần từ mảng.

- In các phần tử lẻ có trong mảng.

- Cho biết giá trị và vị trí phần từ nhỏ nhất trong mảng.

1. Viết chương trình nhập vào số nguyên n và thực hiện xuất ra màn hình n số đầu tiên của chuỗi Fibonacci (có hai giá trị đầu là 1 và 1).
2. Lập trình tính diện tích tam giác theo công thức Heron.
3. Cho thông tin của **HỌC SINH** gồm: mã học sinh, họ tên, lớp, số buổi ăn bán trú. Viết chương trình theo hướng đối tượng để quản lí danh sách **HỌC SINH** với các yêu cầu sau:

- Nhập danh sách học sinh.

- In lại danh sách học sinh.

- Tính tổng tiền bán trú của tất cả các học sinh (tiền bán trú = số buổi ăn bán trú \* 30000)

- Sắp xếp danh sách học sinh theo thứ tự tăng dần của tiền ăn bán trú.

- Thực hiện in danh sách tên các HỌC SINH có số buổi ăn bán trú trên 20 buổi.

1. Viết chương trình tính diện tích, chu vi hình chữ nhật.

- Hãy viết lớp HCN gồm có:

+ Các thuộc tính: chiều dài, chiều rộng.

+ Phương thức thiết lập (**set**), và lấy (**get**) thông tin chiều dài, chiều rộng.

+ Phương thức tính diện tích, chu vi.

+ Phương thức **toString** hiển thị các thông tin dài, rộng, diện tích, chu vi.

- Xây dựng lớp **HCNTest** có chứa phương thức main để thực hiện test. Chiều dài, chiều rộng của hình chữ nhật được nhập từ bàn phím.

1. Cho mô tả lớp **Congnhan** (công nhân) như sau:

|  |
| --- |
| **Congnhan** |
| + lcb: float static//lương cơ bản  - hsl: float//hệ số lương  - hoten: String//họ tên |
| + setHsl(h:float):void  + getHsl(): float  + setHoten(ht:String):void  + getHoten():String  +tinhLuong():float//tính lương |

Trong đó, phương thức **tinhLuong** hoạt động theo công thức:

*Lương = lcb\*hsl*

Viết chương trình chính:

1. Xây dựng lớp Congnhan theo mô tả trên.

2. Cài đặt lương cơ bản của công nhân = 1150

3. Nhập danh sách n Công nhân (0<n<20).

Hiện danh sách vừa nhập ra màn hình với thông tin mỗi công nhân gồm:

Họ tên, Hệ số lương, Lương.

4. Nhập vào một số thực S.

Hiện danh sách họ tên các công nhân có lương >S.

1. Cho lớp **DIEM(điểm)** để mô tả một điểm trong không gian gồm:

- Thuộc tính x, y (tọa độ trong không gian)

- Phương thức: khởi tạo (không có tham số - có tham số), nhập, xuất,…

Xây dựng lớp **DOANTHANG** gồm:

- Thuộc tính: A(Điểm đầu), B(Điểm cuối)

- Phương thức: Khởi tạo (không có tham số - có tham số), nhập, xuất, tính chiều dài,…

Áp dụng:

- Nhập vào danh sách các **DOANTHANG.**

**-** In lại danh sách các **DOANTHANG** vừa nhập.

- Cho biết thông tin **DOANTHANG** có độ dài lớn nhất.

- Tính tổng độ dài tất cả các đoạn thẳng đã nhập

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

- Số tài khoản

- Tên tài khoản

- Số tiền trong tài khoản

a) Thiết kế lớp ***TaiKhoan*** để lưu trữ các thông tin, lớp bao gồm các phương thức sau:

+ Phương thức khởi tạo (Constructor): có 2 phương thức khởi tạo (mặc định và đẩy 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.

b) Thêm các thông tin sau vào lớp ***TaiKhoan***

+ Hằng số lãi suất có giá trị khởi tạo là **0.05**

+ Phương thức khởi tạo có 2 đối số: số tài khoản, tên tài khoản.

Phương thức khởi tạo này sẽ khởi tạo số tiền mặc định là 100

+ Phương thức gửi tiền vào tài khoản: *số tiền hiện tại trong tài khoản + số tiền nạp vào.*

+ Phương thức rút tiền: *Số tiền hiện tại trong tài khoản – (số tiền muốn rút + phí rút tiền).*

+ Phương thức đáo hạn: Mỗi lần đến kì rút đáo hạn thì số tiền trong tài khoản = *Số tiền trong tài khoản + số tiền trong tài khoản \* LaiSuat*

Chú ý: Mỗi thao tác phải kiểm tra số tiền gửi, rút có hợp lệ hay không. Ví dụ nhập vào < 0, tiền rút nhiều hơn tiền trong tài khoản thì thông báo không hợp lệ và yêu cầu nhập lại.

c) Xây dựng lớp ***TaiKhoanTest*** có chứa phương thức main để thực hiện test các chức năng trên.

1. Thêm vào lớp ***Congnhan*** đã cho ở Bài 6 phương thức:

|  |
| --- |
| **+ tinhLuong** (pc: float):float |
| INPUT: pc:float//% phụ cấp được hưởng |
| OUTPUT: lương của công nhân |
| XỬ LÝ: Tính lương theo công thức:  *Lương = lcb\*hsl\*(1+pc)* |

Viết chương trình:

1. Cài đặt lương cơ bản của công nhân bằng 1150, nhập vào danh sách N công nhân (0<n<=20)

2. Nhập mức phụ cấp P.

3. Hiện danh sách công nhân với các thông tin: *Họ tên, Hệ số lương, LươngT(chưa gồm phụ cấp), LươngS(gồm phụ cấp).*

4. Tính tổng số tiền chênh lệch do chi trả phụ cấp

1. Tạo các lớp theo sơ đồ phân cấp sau:

|  |
| --- |
| MATHANG |
| - tenHang: String  - maHang: int  - nuocSX: String |
| + nhap()  + hienThi()  + … |

|  |
| --- |
| TULANH |
| - dungTich: int  - mauSac: String  - hangSX: String  - soLuong: int  - donGia: float |
| + nhap()  + hienThi()  + thanhTien()  + … |

**Yêu cầu:**

**-** Xây dựng mỗi lớp ở một file riêng biệt, ngoài các thành phần đã liệt kê, có thể bổ sung thêm các thuộc tính và phương thức cho phù hợp

- Các lớp đều có phương thức khởi tạo không có tham số và có tham số.

**Áp dụng:**

- Nhập danh sách các tủ lạnh

- In danh sách các tủ lạnh đã nhập

- Liệt kê danh sách tủ lạnh theo hãng sản xuất nào đó được nhập từ bàn phím

- Tính tổng tiền các tủ lạnh có trong danh sách đã được nhập

- In các tủ lạnh có dung tích trên 200 lít

- SX danh sách tủ lạnh theo thứ tự giảm dần của số lượng.

*(Yêu cầu chung: xác định access modifier (private, protected, public) cho từng thuộc tính/phương thức mỗi lớp, cài đặt getter/setter, cài đặt constructor không có tham số và constructor có thông số đầy đủ).*

1. Giả sử cần xây dựng chương trình quản lý dùng cho một học viện nghiên cứu giảng dạy và ứng dụng. Đối tượng quản lý bao gồm các sinh viên đang theo học, các nhân viên đang làm việc tại học viện, các khách hàng đến giao dịch mua bán sản phẩm ứng dụng. Dựa vào một số đặc tính của từng đối tượng, người quản lý cần đưa ra cách thức đánh giá khác nhau.

- Lớp **Person:** bao gồm các thuộc tính họ tên, địa chỉ, phương thức toString.

- Các lớp **Student, Employee, Customer** (mô tả dưới đây) thừa kế lớp **Person.**

+ Lớp **Student:** Bao gồm các thuộc tính điểm môn học 1, điểm môn học 2, và các phương thức tính DTB, đánh giá, overriding phương thức toString trả về bảng điểm sinh viên (gồm thông tin thuộc tính và điểm TB).

+ Lớp **Employee:** Bao gồm thuộc tính heSoLuong, và các phương thức: tính lương, đánh giá, overriding phương thức toString trả về bảng lương cho nhân viên (gồm thông tin thuộc tính đối tượng và tiền lương).

+ Lớp **Customer:** Bao gồm thuộc tính tên công ty, trị giá hóa đơn, đánh giá, phương thức toString trả về thông tin hóa đơn cho khách hàng (gồm các thuộc tính của đối tượng).

- Lớp có 1 biến danh sách để lưu các sinh viên, nhân viên, khách hàng (dùng 1 biến array Person), biến lưu tổng số người có trong danh sách, constructor mặc định khởi tạo array với dung lượng cho trước, phương thức thêm một người vào danh sách (thông số Person), xóa 1 người khỏi danh sách (nhận thông số là họ tên của người cần xóa), sắp xếp danh sách theo thứ tự họ tên, phương thức xuất danh sách. Khi danh sách đầy thì tự động tăng dung lượng dãy lên 50%.

- Viết lớp với phương thức main cho phần kiểm nghiệm. Giao tiếp với người dùng bằng menu (thể hiện tính đa hình – polymorphism bằng cách cho phép lựa chọn nhập thông tin là sinh viên, nhân viên hay khách hàng).

1. Cho sơ đồ các lớp như sau:

|  |
| --- |
| Người |
| - Hoten  - Gioitinh |
| + Nhap()  + Hien () |

|  |
| --- |
| Học sinh |
| - Tenlop |
| + Nhap()  + Hien () |

1. Xây dựng lớp Người với các thuộc tính Hoten, Gioitinh và phương thức: Nhap(), Hien().

2. Xây dựng lớp Học sinh thừa kế lớp Người với thuộc tính: Tenlop và phương thức Nhap(), Hien().

3. Viết chương trình:

+ Nhập vào N học sinh của các lớp khác nhau.

+ Sắp xếp danh sách học sinh theo tên lớp, hiện danh sách ra màn hình.

Cho một tập danh sách các cán bộ cần quản lý của một cơ quan nhà nước:

- CHUYÊN VIÊN: cán bộ làm tại các phòng ban.

- GIẢNG VIÊN: cán bộ làm công tác giảng dạy tại các khoa

- QUẢN LÝ: cán bộ làm công tác quản lý (trưởng – phó) tại các khoa và phòng ban

**Yêu cầu quản lý:**

- Thông tin từng cán bộ

- Tính tiền lương và phụ cấp cho từng cán bộ như sau:

+ CHUYÊN VIÊN: tiền lương = Hệ số lương \*1350000, không phụ cấp.

+ GIẢNG VIÊN: tiền lương = (Hệ số lương \* 1350000) + phụ cấp đặc biệt ngành (20% lương)

+ QUẢN LÝ: tiền lương = (hệ số lương + hệ số phụ cấp) \* 1350000

**Yêu cầu sinh viên:**

**-** Dùng kiến thức mô hình hóa dữ liệu trong lập trình hướng đối tượng để xây dựng các lớp.

- Thiết kế mô hình cây phân cấp các lớp. Mối quan hệ giữa các lớp và interface.

- Chuyển từ mô hình trên sang ngôn ngữ lập trình Java. (Chuyển sang thiết kế các lớp và interface tương ứng bằng ngôn ngữ lập trình Java)

- Cài đặt test chương trình áp dụng nhập vào một danh sách đối tượng Cán bộ bất kỳ (có thể là CHUYÊN VIÊN hoặc GIẢNG VIÊN hoặc QUẢN LÝ). Sau đó in lại danh sách thông tin các đối tượng Cán bộ đã nhập.

1. Với 1 tập mini các loại xe trong thế giới cho bên dưới:

**Yêu cầu quản lý:**

- Thông tin từng loại xe.

- Tính tiền thuế cho từng chiếc xe dựa trên giá trị xe như sau:

+ **Xe đạp:** Không đóng thuế.

+ **Xe máy:** VAT=10% và thuế trước bạ 5%

+ **Xe ô tô khách:** Thuế tiêu thụ đặc biệt 30% (số chỗ >=5); 50% (số chỗ<5), thuế VAT=10%, thuế trước bạ 20%.

+ **Xe ô tô tải:** VAT=10%, thuế trước bạ 2%.

**Yêu cầu sinh viên:**

- Dùng kiến thức mô hình hóa dữ liệu trong lập trình hướng đối tượng để xây dưng các lớp.

- Thiết kế mô hình cây phân cấp các lớp. Mối quan hệ giữa các lớp và các interface.

- Chuyển từ mô hình trên sang ngôn ngữ lập trình Java. (chuyển sang thiết kế các lớp và interface tương ứng bằng ngôn ngữ lập trình Java)

- Cài đặt test chương trình.

|  |
| --- |
| **Nguoi** |
| - Hoten: String  - Namsinh: int  - Gioitinh: bool |
| + getHoten(): String  + getNamSinh(): int  + getGioiTinh():int  + Nguoi(in hoten: String, in namsinh: int, in gioitinh: bool) |

iHoatdong

|  |
| --- |
| **Hocsinh** |
| - tenlop: String |
| + gioithieu: String  + getTenLop(): String  + Hocsinh(in hoten: String, in namsinh: int, in gioitinh: bool, in tenlop: String) |

1. Tạo lớp Nguoi như trên sơ đồ

2. Tạo giao diện (interface) **IHoatdong** với phương thức ***string gioithieu***().

3. Tạo lớp Hocsinh thừa kế từ Nguoi và thi hành giao diện IHoatDong (cho biết học sinh học lớp nào)

4. Viết chương trình:

a. Nhập vào N học sinh.

b. Hiện N học sinh ra màn hình.

c. Nhập vào 1 năm sinh X, hiển thị ra màn hình tên từng học sinh sinh năm đó và thông tin tự giới thiệu của từng học sinh.

d. Lưu danh sách học sinh vào file.

e. Đọc danh sách học sinh có từ file và in lên màn hình.