

Bài 4: THỰC HÀNH LẬP TRÌNH JAVA HƯỚNG ĐỐI TƯỢNG

Bài tập 1: Xây dựng lớp: Bài toán hình chữ nhật.....	2
Bài tập 2: Xây dựng lớp : Bài toán hình tròn	3
Bài tập 3: Xây dựng lớp: Bài toán phương trình bậc 2.	3
Bài tập 4: Xây dựng lớp: Bài toán quản lý đăng ký xe.	4
Bài tập 5: Xác định và xây dựng lớp - mối quan hệ lớp	5
Bài tập 6: Xây dựng lớp: Bài toán Nhân viên và hàm thành phần.....	7
Bài tập 7: Xây dựng lớp: Bài toán Sinh viên	8
Bài tập 8: Quan hệ kế thừa- lớp trừu tượng: Bài toán hình phẳng.....	9
Bài tập 9: Quan hệ kế thừa: Bài toán quản lý tiền điện.....	9
Bài tập 10: Quan hệ kế thừa- lớp trừu tượng : Bài toán quản lý nhân viên	10
Bài tập 11: Cài đặt quan hệ kết tập: Bài toán hóa đơn bán hàng.	11
Bài tập 12: Quan hệ kế thừa: Bài toán Quản lý Kho.....	12

Bài tập 1: Xây dựng lớp: Bài toán hình chữ nhật.**Đề bài:**

Hình chữ nhật mô tả thông qua bằng chiều dài, chiều rộng. Qua hai tham số này ta có thể tính được chu vi và diện tích theo công thức toán học. Viết chương trình nhập và in thông tin hình chữ nhật có các thông tin: chiều dài, chiều rộng, chu vi, diện tích..

Yêu cầu chuẩn bị:

Chỉnh sửa lại các thông nhất với nhóm và giáo viên về thuộc tính, phương thức của lớp Hình chữ nhật và lớp chứa hàm main sau khi đã

Gợi ý cài đặt:

Cài đặt bài toán thực hiện theo yêu cầu.

File HCN.java

```
public class HCN {
    public static void main(String[] args) {
        float dai, rong, chuVi, dienTich;
        Random random = new Random();
        dai = random.nextFloat () + 10;
        rong = random.nextFloat () + 10;
        chuVi=(dai+rong)*2;
        dienTich=dai*rong;
        //in kết quả
        DecimalFormat d=new DecimalFormat("###,###.0#");
        System.out.printf("%15.1f%15.1f%15s%15s%n",
            dai,rong,d.format(tinhChuVi()),d.format(tinhDienT
            i ch()));
        //in tiêu đề
        System.out.printf("%15s%15s%15s%15s%n",
            "Chiều dài ", "Chiều rộng ", " chu vi", "Diện tích");
    } }
    -
```

void nhap()

float tinhCV()
float tinhDT()

voi xuất()

voi

Gợi ý đóng gói :

```
public class HìnhChuNhat {
    private float dai,rong;
    public float tinhDienTich(){
        return dai*rong;
    }
    public float tinhChuVi() {
        return (dai+rong)*2;
    }
    void nhap() {
        Random random = new Random();
        dai = random.nextFloat() + 1;
        rong = random.nextFloat() + 1;
    }
    public static void inTieuDe(){
        System.out.printf("%15s%15s%15s%15s%n",
            "Chiều dài ", "Chiều rộng ", " chu vi", "Diện tích");
    }
}
```

```

public void xuat() {
    System.out.printf("%15.1f%15.1f%15.1f%15.1f%n",
        dai,rong,tinhChuVi(),tinhDienTich());
}

public static void main(String[] args) {
    HinhChuNhat h=new HinhChuNhat();
    h.nhap();
    HinhChuNhat.inTieuDe();
    h.xuat();
}

```

Bài tập 2: Xây dựng lớp : Bài toán hình tròn

Đề bài:

Hình tròn có thể tính được chu vi và diện tích khi biết bán kính hoặc đường kính hình tròn. Các hình tròn in ra đều có cùng tiêu đề và dữ liệu cụ thể biểu diễn bán kính ; chu vi và diện tích. Thực hiện bài toán nhập và in mảng hình tròn có kết quả như bảng dữ liệu.

Yêu cầu chuẩn bị:

- Thống nhất các đóng gói với nhóm và giáo viên
- **Chuẩn bị phương án cài đặt** : xác định tên lớp, thuộc tính, tên các phương thức và câu lệnh phù hợp và cài đặt bài toán thực hiện theo yêu cầu.

Bài tập 3: Xây dựng lớp: Bài toán phương trình bậc 2.

Đề bài :

Phương trình bậc hai $ax^2 + bx + c = 0$ có các hệ số a,b,c là các số thực được nhập vào từ bàn phím hoặc sinh số ngẫu nhiên. Thực hiện xây dựng nhập mảng phương trình bậc 2 và hiển thị thông tin mảng sau khi nhập có quy cách theo bảng minh họa.

Yêu cầu chuẩn bị:

- Thống nhất với nhóm và giáo viên về đóng gói cho lớp.
- Chuẩn bị phương án cài đặt.

Yêu cầu thực hành: Cài đặt bài toán thực hiện theo yêu cầu.

Gợi ý thực hiện:

Lớp: PTB2 có :

- (1) Các thuộc tính: Là các hệ số của phương trình
- (2) Các phương thức:
 - Phương thức nhập : nhập các hệ số, trong đó yêu cầu nhập a khác không
 - Phương thức giải phương trình
 - Phương thức in tiêu đề
 - Phương thức in kết quả
- (3) Tạo lớp PTB2demo thực hiện khai báo minh họa 1 đối tượng phương trình bậc 2. Yêu cầu đối tượng nhập thông tin các hệ số và in kết quả

Bài tập 4: Xây dựng lớp: Bài toán quản lý đăng ký xe.**Đề bài:**

Sở giao thông cần theo dõi việc đăng ký xe của người dân. Dựa vào thông tin trị giá xe và dung tích xy lanh của xe, sở giao thông cũng tính mức thuế phải đóng trước bạ khi mua xe như sau:

- Dưới 100cc, 1% trị giá xe.
- Từ 100 đến 200cc, 3% trị giá xe.
- Trên 200cc, 5% trị giá xe.

Yêu cầu bài toán:

1. Thiết kế và cài đặt lớp phương tiện với các thuộc tính và phương thức phù hợp bảo đảm tính đóng gói theo hướng đối tượng.
2. Xây dựng lớp chứa hàm main. Hàm main in ra menu lựa chọn các công việc:
 - a.. Nhập thông tin và tạo danh sách các xe
 - b. Xuất bảng kê khai tiền thuế trước bạ của các xe.
 - c. Thoát.

Mẫu thiết kế xuất của chương trình:

Tên chủ xe	Loại xe	Dung tích	Trị giá	Thuế phải nộp
Nguyễn Thu Loan	Future Neo	100	35000000.00	1050000.00
Lê Minh Tính	Ford Ranger	3000	250000000.00	12500000.00
Nguyễn Minh Triết	Landscape	1500	1000000000.00	50000000.00

Yêu cầu chuẩn bị:

- Thống nhất đóng gói thuộc tính và phương thức cho lớp. Cài đặt theo gợi ý.

Yêu cầu thực hành: Cài đặt bài toán thực hiện theo yêu cầu.

Gợi ý thực hiện:

Căn cứ vào mẫu kết xuất của chương trình ta tạo lớp Xe có :

- Danh từ thuộc tính mô tả: Tên chủ xe, loại xe, dung tích, trị giá
- Danh từ phát sinh: thuế
- Động từ thực hiện:
 - tính thuế dựa vào dung tích xi lanh vậy ta viết phương thức tính thuế thuộc lớp xe theo công thức đã cho.
- Nhập thông tin xe; in thông tin xe; in có tiêu đề: xây dựng thêm phương thức nhập thông tin xe; phương thức xuất tiêu đề; phương thức in dữ liệu
- Căn cứ yêu cầu hàm main ta thực hiện:
 - Tạo lớp Xedemo có 1 thuộc tính là danh sách xe. Vì danh sách sẽ đc dùng trong hàm main nên nó có thêm chỉ thị static.
 - Xây dựng phương thức nhập danh sách xe cũng có chỉ thị static vì thao tác trên ds xe
 - Xây dựng phương thức inDL cũng static gọi xe in tiêu đề và in ds xe dùng lặp for.

Xây dựng hàm main có cấu trúc lặp while hoặc do-while chọn công việc bằng cách gọi hàm tương ứng thông qua biến chọn.

Bài tập 5: Xác định và xây dựng lớp - mối quan hệ lớp

Đề bài:

- Thông tin nhân viên trong công ty được quản lý thông qua họ đệm, tên nhân viên, ngày sinh và ngày vào làm.
- Thông tin về ngày tháng được mô tả thông qua ngày, tháng, năm. Yêu cầu tháng phải nằm trong khoảng từ 1-12 và ngày phải phù hợp với tháng tương ứng.
- Cài đặt minh họa nhập thông tin 1 nhân viên và in ra nhân viên đó sau khi nhập
- Mở rộng: Nhập vào 1 mảng nhân viên và in thông tin mảng sau khi nhập có mẫu như sau

DS nhan vien sau khi nhap:

Ho dem	Ten	Ngày sinh	Ngày vào làm
vu van h	f	7/8/8	6/7/8
uy	uj	8/8/9	7/7/7
98	98	9/9/9	8/8/8

Yêu cầu chuẩn bị:

- Thống nhất các nội dung về thuộc tính, giá trị trả về, nội dung phương thức với nhóm và giáo viên

Yêu cầu thực hành:

- Cài đặt bài toán thực hiện theo yêu cầu. Gợi ý sử dụng mảng.

Gợi ý cài đặt:

NgàyThang.java

```
public class NgàyThang {
    private int thang; // 1-12
    private int ngay; // 1-31 based on month
    private int nam; // any year
    public NgàyThang( int thang1 , int ngay1, int nam1 ){
        if ( thang1 > 0 && thang1 <= 12 ) // tháng hợp lệ
            this.thang = thang1;
        else {
            this.thang = 1;
            System.out.println( "Tháng " + thang1 + " không hợp lệ. đặt lại tháng =1." );
        }
        nam = nam1; // could validate year
        ngay = checkDay( ngay1 ); // kiểm tra ngày hợp lệ
        System.out.println( ngày tháng " + toString());
    }
    private int checkDay( int ngayKiemTra ) {
        int
        ngayTrongThang[]={0,31,28,31,30,31,30,31,31,30,31,30,31};
        // kiểm tra nếu ngày tháng hợp lệ
        if (ngayKiemTra > 0 && ngayKiemTra<=ngayTrongThang [thang
        ])
            return ngayKiemTra;
    }
}
```

```
else
if (((thang==2) && (ngayKiemTra == 29)) &&
    ((nam % 400 == 0) || ((nam % 4 == 0) && (nam % 100
!= 0 ))))
    return ngayKiemTra;
else {
    System.out.println( "ngày " + ngayKiemTra + " không
hợp lệ, đặt lại =1." );
    return 1;
}
} //end of checkDay()
//ghi đè phương thức toString để mô tả tháng/ngày/năm
public String toString() {    return thang + "/" + ngay +
"/" + nam;    }
} // end of class NgayThang
```

NhanVien.java

```
public class NhanVien{
    private String hoDem;
    private String ten;
    private NgayThang ngaySinh;
    private NgayThang ngayVaoLam;
    public NhanVien () {
        hoDem="";
        ten="";
        ngaySinh =new NgayThang(0,0,0);
        ngayVaoLam=new NgayThang(0,0,0);
    }
    public NhanVien(String hoDem, String ten, NgayThang
        ngaySinh, NgayThang ngayVaoLam) {
        this.hoDem = hoDem;        this.ten = ten;
        this.ngaySinh = ngaySinh;
        this.ngayVaoLam = ngayVaoLam;
    }
    //sinh viên tự xây dựng các hàm cần thiết khác để thực
    hiện được kết quả như yêu cầu kết xuất trên
    public String toString() {
        return ten + ", " + hoDem + " ngày vào lam: " + ngayVaoLam
        + "    sinh ngày: " + ngaySinh;
    }
}
```

NhanVienDemo.java

```
public class NhanVienDemo {
    public static void main(String args[]) {
        NgayThang ngaySinh = new NgayThang ( 7, 24, 1991 );
        NgayThang ngayVaoLam = new NgayThang( 3, 12, 1998 );
        NhanVien nv= new NhanVien ( "Vu thi", "Lan anh", ngaySinh,
            ngayVaoLam
        );
        //gọi hàm hiển thị message dialog trong thư viện
        JOptionPane.showMessageDialog( null, nv.toString(), "minh
        họa lớp Employee", JOptionPane.INFORMATION_MESSAGE );
        System.exit( 0 ); //thoát khỏi chương trình
    }
}
```

```
}
```

Yêu cầu thực hành:

- Cài đặt bài toán thực hiện theo yêu cầu nhập vào 1 danh sách nhân viên từ bàn phím. In danh sách nhân viên sau khi nhập ra màn hình theo dạng bảng dữ liệu

Bài tập 6: Xây dựng lớp: Bài toán Nhân viên và hàm thành phần.**Đề bài:**

Xây dựng lớp NhanVien như sau: tên file NhanVien.java có các thuộc tính: mã nhân viên - maNV; số sản phẩm - soSP. Lớp có các phương thức:

- khởi dựng không đối: NhanVien()
- khởi dựng có đối: NhanVien (ma: String, sp : int)
- get/set thuộc tính mã, số sản phẩm : getMaNV() ;setMaNV(ma : String) ; getSoSP() ; setSoSP(sp : int). Lưu ý kiểm tra số sản phẩm lớn 0 trước khi gán cho số sản phẩm
- kiểm tra vượt chuẩn: coVuotChuan() : boolean
- tổng kết: getTongKet(): String
- lấy lương: getLuong(): double
- in tiêu đề: XuatTieuDe(): static void
- kết xuất đối tượng dạng xâu: toString(): String
- coVuotChuan(): trả về true nếu soSP > 500, ngược lại trả về false. hàm này dùng để kiểm tra xem số lượng sản phẩm của nhân viên có vượt quá số lượng chuẩn hay không.
- getTongKet(): trả về chữ "Vượt" khi soSP > 500, ngược lại để trống (có thể sử dụng hàm coVuotChuan() để kiểm tra).
- getLuong() : trả về lương của một nhân viên, lương ăn theo sản phẩm với đơn giá cơ bản cho 1 sản phẩm là 20.000, và nếu số sản phẩm của nhân viên vượt chuẩn thì phần vượt chuẩn được tính đơn giá là 30.000.
- xuatTieuDe(): xuất tiêu đề gồm: mã nhân viên, số sản phẩm, lương, tổng kết.
- inDL() : in dữ liệu tương ứng với tiêu đề để tạo khoảng cách phù hợp

Yêu cầu: Viết hàm main để kiểm tra lớp NhanVien theo yêu cầu sau:

- Tạo 2 nhân viên với các thuộc tính cho người dùng nhập vào.
- Xuất ra các thông tin của họ, gồm mã, số sản phẩm, lương, tổng kết và cho biết nhân viên có vượt chuẩn không.

Mở rộng bài toán:

- Nhập vào 1 mảng Nhân viên, in mảng sau khi nhập.
- In danh sách nhân viên vượt chuẩn.

Yêu cầu chuẩn bị:

- Xác định phạm vi, giá trị trả về và nội dung thực hiện của các phương thức.

Yêu cầu thực hành:

- Cài đặt bài toán thực hiện theo yêu cầu

Bài tập 7: Xây dựng lớp: Bài toán Sinh viên**Đề bài:**

Xây dựng lớp **SinhVien** mô tả thông qua

Các thuộc tính: mã sinh viên, họ tên, điểm lý thuyết, điểm thực hành.

Các phương thức:

- khởi tạo không đối- khởi gán các giá trị mặc định cho thuộc tính
- khởi có đầy đủ các đối- khởi gán giá trị như truyền vào
- nhập()- nhập thông tin Sinh viên và tính luôn điểm trung bình= $(\text{đlt} + \text{dth})/2$.
- toString()- kết xuất xâu mô tả Sinh Viên

Yêu cầu: Viết hàm main cho phép thực hiện:

- Nhập 1 danh sách sinh viên
- Xuất thông tin sinh viên và cho biết kết quả học tập của sinh viên (Đậu khi điểm trung bình ≥ 5 , ngược lại là Rớt).
- Sắp xếp danh sách sinh viên theo tên. Tên trùng nhau sắp xếp theo điểm lý thuyết
- Tìm sinh viên có điểm trung bình cao nhất, sinh viên có điểm trung bình thấp nhất

Yêu cầu chuẩn bị:

- Thông nhất các thuộc tính và phương thức, phạm vi, giá trị trả về, cấu trúc mã lệnh thực hiện nội dung các yêu cầu.

Yêu cầu thực hành:

- Cài đặt bài toán thực hiện theo yêu cầu

Gợi ý cài đặt:

Lớp sinh viên: các thuộc tính riêng tư (private):

- Mã sinh viên – số nguyên
- Họ tên: chuỗi ký tự
- Điểm LT, điểm TH: số thực

Phương thức:

- Hàm tạo mặc định khởi gán giá trị mặc định cho các thuộc tính
- Hàm tạo có đối gán đầy đủ thông cho các thuộc tính
- Hàm getter/setter các thuộc tính
- Phương thức static inTieuDe diễn tả tiêu đề in của đối tượng.
- Xây dựng phương thức inDuLieu(). In nội dung dữ liệu tương ứng với tiêu đề.
- Xây dựng hàm main tạo 3 đối tượng sinh viên
- Sv1 chứa thông tin của mình bằng hàm tạo đủ thông số, thông tin truyền vào cho hàm tạo nhập vào từ bàn phím
- Sv2 thông tin được tạo bằng hàm đủ thông số với các giá trị cố định do người dùng mình họa.
- Sv3 tạo bằng hàm tạo mặc định. Nhập các thông tin vào từ bàn phím và gọi hàm set để gán giá trị cho các thuộc tính
- Tạo và in bản danh sách sinh viên gồm 4 cột: mã sinh viên, họ tên, điểm lý

- thuyết, thực hành, điểm trung bình. Bảng có 3 dòng cho 3 sinh viên
- Đưa ra sinh viên có điểm trung bình lớn nhất trong danh sách.

Bài tập 8: Quan hệ kế thừa- lớp trừu tượng: Bài toán hình phẳng

Đề bài:

Hình Vuông, hình chữ nhật hay hình tròn là các hình phẳng được mô tả thông qua các thuộc tính riêng nhưng đều tính được chu vi và diện tích. Vậy hình phẳng nói chung có tính được chu vi và diện tích không.

Yêu cầu: Cài đặt lớp Hình phẳng, hình vuông, hình chữ nhật và hình tròn sao cho đảm bảo tính hướng đối tượng (chú ý tính đóng gói và sử dụng lại sao cho tối ưu) . Minh họa tạo vài đối tượng hình phẳng nhưng lại cụ thể hóa từng lớp cụ thể để thực hiện nhập và in thông tin của mình.

Yêu cầu chuẩn bị theo nhóm.:

- Vẽ sơ đồ mô hình hóa biểu diễn mô tả cài đặt bài toán.
- Mô tả các phương thức: phạm vi, kiểu trả về, tên lớp, đánh sách tham số nếu có và tóm tắt chức năng thực hiện

Yêu cầu thực hành:

- Cài đặt bài toán thực hiện theo yêu cầu

Gợi ý thực hiện

- Nhóm thực hiện có thể mô tả khai báo sơ bộ trong NetBeans và sinh sơ đồ cài đặt biểu diễn lớp như các phiếu bài mẫu.
- Nhóm cũng có thể sử dụng công cụ mô tả lớp theo chuẩn UML khác.

```
abstract Class HìnhPhang{
    abstract float tinhCV();
    abstract float tinhDT();
}
class HìnhVuong extends HìnhPhang {
    public float canh;
    public float tinhCV(){
        return 4*canh;
    }
    public float tinhDT(){
        return canh*canh;
    }
}
```

- Tương tự cài đặt các lớp Hình tròn, vuông, chữ nhật cũng là trừu tượng và cụ thể hóa phương thức tính chu vi và diện tích
- Viết hàm main minh họa: khai báo mỗi lớp con kế thừa một đối tượng, nhập, xuất dữ liệu của chúng.

Bài tập 9: Quan hệ kế thừa: Bài toán quản lý tiền điện.

Đề bài:

Xây dựng chương trình quản lý danh sách hoá đơn tiền điện của khách hàng. Thông tin bao gồm các loại khách hàng :

- Khách hàng Việt Nam: mã khách hàng, họ tên, ngày ra hoá đơn (ngày, tháng, năm), đối tượng khách hàng (sinh hoạt, kinh doanh, sản xuất): số lượng (sốKW tiêu thụ), đơn giá, định mức. Thành tiền được tính như sau:
- Nếu số lượng \leq định mức thì: thành tiền = số lượng * đơn giá.
- Ngược lại thì: thành tiền = số lượng * đơn giá * định mức + số lượng KW vượt định mức * Đơn giá * 2.5.
- Khách hàng nước ngoài: mã khách hàng, họ tên, ngày ra hoá đơn (ngày, tháng, năm), quốc tịch, số lượng, đơn giá. Thành tiền được tính = số lượng * đơn giá.

Yêu cầu:

- Xây dựng các lớp với chức năng thừa kế.
- Nhập xuất danh sách các hóa đơn khách hàng.
- Tính tổng số lượng cho từng loại khách hàng.
- Tính trung bình thành tiền của khách hàng người nước ngoài
- Xuất ra các hoá đơn trong tháng 09 năm 2013 (của cả 2 loại khách hàng).

Yêu cầu chuẩn bị theo nhóm:

- Vẽ sơ đồ mô hình hóa biểu diễn mô tả cài đặt bài toán.
- Mô tả các phương thức: phạm vi, kiểu trả về, tên lớp, đánh sách tham số nếu có và tóm tắt chức năng thực hiện
- Thống nhất các thuộc tính kế thừa, phương thức có mã nguồn dùng chung, phương thức dùng chung nhưng chưa cụ thể hóa đc mã lệnh.

Yêu cầu thực hành:

- Cài đặt bài toán thực hiện theo yêu cầu

Bài tập 10: Quan hệ kế thừa- lớp trừu tượng : Bài toán quản lý nhân viên**Đề bài:**

Yêu cầu: Một công ty sản xuất có 2 loại nhân viên là nhân viên sản xuất và nhân viên văn phòng. Hai loại nhân viên được quản lý thông qua các thông tin: họ tên; năm vào làm, lương, phụ cấp. Lương của hai loại nhân viên được tính như sau:

- Nhân viên sản xuất: lương = sản phẩm x 10000.
- Nhân viên văn phòng: lương = mức lương – ngày nghỉ x 10000.

Ngoài lương được tính như trên, mỗi nhân viên còn được phụ cấp một khoảng tiền là 100000. Và khoảng tiền này cứ tăng thêm 20000 cho mỗi năm công tác ở công ty.

Yêu cầu: Viết chương trình: Nhập vào danh sách nhân viên công ty. Tính tổng số tiền công ty phải trả cho nhân viên mỗi tháng.

Yêu cầu chuẩn bị theo nhóm:

- Vẽ sơ đồ mô hình hóa biểu diễn mô tả cài đặt bài toán.
- Mô tả các phương thức: phạm vi, kiểu trả về, tên lớp, đánh sách tham số nếu có và tóm tắt chức năng thực hiện
- Thống nhất các thuộc tính kế thừa, phương thức có mã nguồn dùng chung và phương thức dùng chung nhưng chưa cụ thể hóa đc mã lệnh.

Yêu cầu thực hành: Cài đặt bài toán thực hiện theo yêu cầu

Bài tập 11: Cài đặt quan hệ kết tập: Bài toán hóa đơn bán hàng.

Đề bài: Cho mô tả của hóa đơn bán hàng như hình sau. Hãy xây dựng các lớp phù hợp thực hiện in hóa đơn bán hàng có mẫu như trên.

Mã HD: 001		Ngày bán: 12/2/2020	
Khách hàng: Vũ văn A		Địa chỉ: Hà Nội	
Tên hàng	Đơn giá	Số lượng	Thành tiền
Áo	200	152	352
Tất	10	256	266
....			
Tổng tiền		408	618

Yêu cầu chuẩn bị:

- Tìm các cụm danh từ, và các danh từ mô tả cụm danh từ phát hiện các lớp
- Tìm công việc cần thực hiện trên các danh từ để phát hiện các phương thức phù hợp
- Vẽ sơ đồ liên kết các lớp
- Cài đặt sơ đồ theo hướng đối tượng.
- Thực hiện nhập vào 1 danh sách hóa đơn
- In danh sách hóa đơn theo mẫu (không kẻ khung)
- Sắp xếp danh sách hàng hóa trong hóa đơn theo chiều tăng dần tên hàng. Tên hàng trùng nhau sắp tăng dần theo thành tiền

Gợi ý cài đặt:

- **Khách hàng** sở hữu các thông tin: họ tên và địa chỉ. Vậy ta đóng gói 2 thuộc tính này vào lớp khách hàng
- **Hàng hóa** sở hữu thông tin: tên hàng, số lượng, đơn giá. Thành tiền = số lượng * đơn giá
- **Hóa đơn** bao gồm: mã hóa đơn, ngày lập, khách mua hàng và danh sách hàng hóa.
- **Gợi ý mã hóa:** Xây dựng 2 lớp **khách hàng**, **hàng hóa** trước. sau đó xây dựng lớp **hóa đơn** ngoài các thuộc tính của nó là: mã hóa đơn và ngày bán, nó còn có thuộc tính khách hàng là đối tượng lớp Khách hàng, ds hàng hóa là tập hợp gồm các đối tượng lớp Hàng Hóa.

KhachHang.java.

```
public class KhachHang {
    private String hoten, diaChi;
    public void nhap()
    {
    }
    public String toString() {
    }
}
```

Hang.java

```
public class Hang {
```

```
private String tenHang;
private float soLuong,donGia;
public void nhap() {
    //sv cài đặt các lệnh nhập dữ liệu
}
public float tinhTongTien() {
    //sv cài đặt các lệnh tính tổngtienf
}
static void inTieuDe() {
    //sv cài đặt các lệnh in tiêu đề
}
public void inDL() {
    //sv cài đặt các lệnh in dữ liệu
}
}
} //end of class
```

HoaDon.java

```
public class HoaDon {
    private String soHD,ngayHD;
    private int soMatHang;
    private KhachHang kh= new KhachHang();
    private Hang dsHang[];
    private float tongTienHang=0;
public void nhap() {
    //sinh viên tự triển khai
}
public void inHD() {
    //sinh viên tự triển khai
}
public void sapXep() {
    // sinh viên tự triển khai
}
}
} //end of classs
```

HoaDonDemo.java

```
public class HoaDonDemo {
    public static void main(String[] args) {
        //sinh viên tự triển khai
    }
}
```

Yêu cầu thực hành: Cài đặt minh họa bài toán trên.

Bài tập 12: Quan hệ kế thừa: Bài toán Quản lý Kho

Đề bài: Hàng hóa quản lý trong kho của một siêu thị gồm có hàng thực phẩm, hàng sành sứ và hàng điện máy. Mỗi loại hàng đều có:

- Mã hàng (không được sửa, không được để trống),
- Tên hàng (không được rỗng),
- Số lượng tồn (≥ 0),
- Đơn giá (> 0).
- Hàng thực phẩm thì cần quan tâm đến thông tin ngày sản xuất, ngày hết hạn

(ngày hết hạn phải sau hoặc là ngày sản xuất) và nhà cung cấp.

- Hàng điện máy cần biết thời gian bảo hành bao nhiêu tháng (≥ 0), công suất bao nhiêu KW (> 0).
- Hàng sành sứ thì cần biết thông tin về nhà sản xuất và ngày nhập kho.
- Ngoài ra, người quản lý cần quan tâm đến số lượng tồn kho và các yếu tố khác của từng loại hàng hóa để đánh giá mức độ bán buôn, tiền VAT từng loại hàng hóa. Biết rằng VAT của hàng điện máy và sành sứ là 10%, VAT của hàng thực phẩm là 5%.

Yêu cầu chuẩn bị:

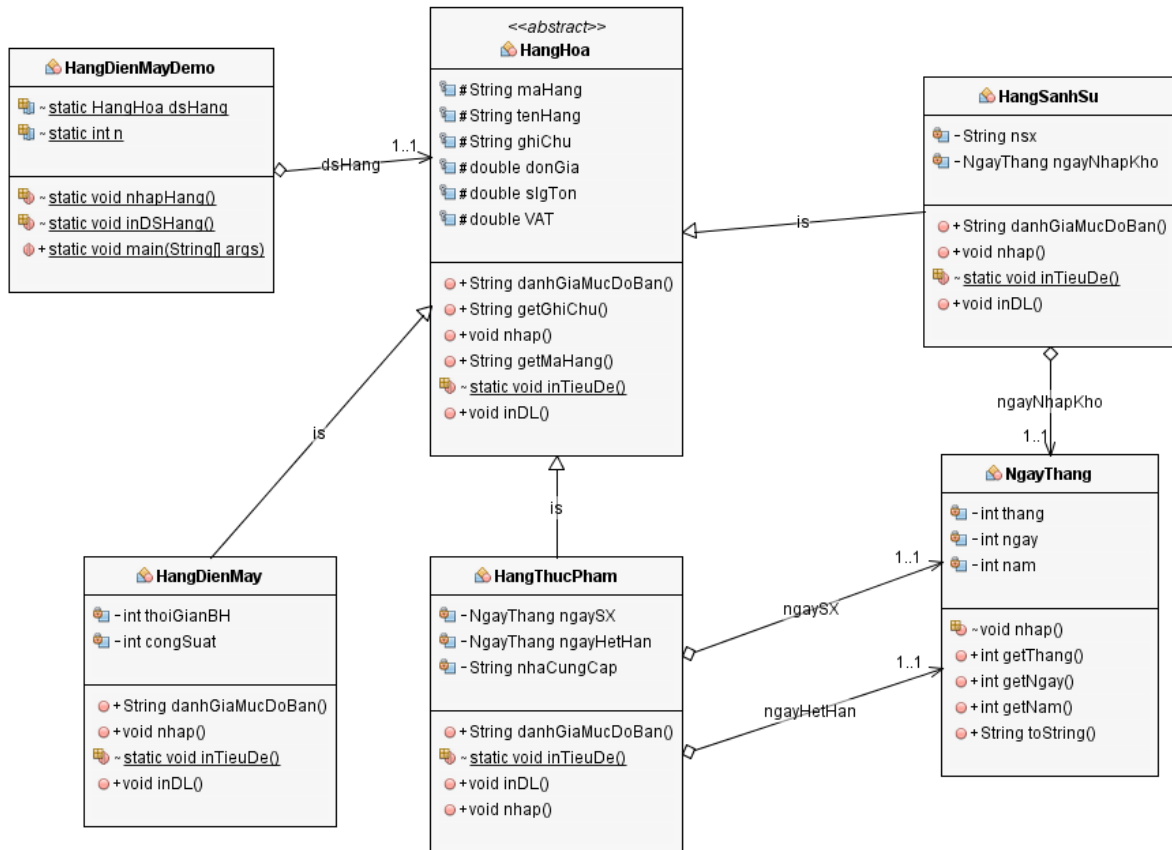
- Xác định: Các lớp có thể có. Lớp nào là lớp trừu tượng (abstract class), lớp nào là lớp cụ thể
- Các thuộc tính cho từng lớp.
- Các phương thức cho từng lớp (phương thức nào là phương thức trừu tượng (abstract method), danh sách các tham số có thể có cho từng phương thức và kiểu trả về của phương thức).
- Thiết kế mô hình quan hệ lớp (xây dựng cây thừa kế, các giao diện nếu có).

Yêu cầu thực hành:

- Cài đặt tường minh cho mỗi loại hàng cụ thể trên. Trong đó, để đánh giá mức độ bán buôn thì:
 - o Hàng điện máy, nếu số lượng tồn kho < 3 thì được đánh giá là bán được.
 - o Hàng thực phẩm, nếu vẫn còn tồn kho và bị hết hạn thì đánh giá là khó bán.
 - o Hàng sành sứ, nếu số lượng tồn kho > 50 và thời gian lưu kho > 10 ngày thì đánh giá là bán chậm.
 - o Các trường hợp còn lại xem như không đánh giá.
- Hãy viết lớp quản lý danh sách hàng hóa.
 - o Dùng mảng hoặc danh sách để lưu trữ danh sách hàng hóa.
 - o Viết phương thức thêm một hàng hóa vào danh sách (thêm thành công nếu không trùng mã)
 - o Viết phương thức in toàn bộ danh sách các hàng hóa theo từng loại.
 - o Tạo lớp cho phần thử nghiệm, với menu lựa chọn theo yêu cầu nhập xuất hàng hóa như thiết kế.

Gợi ý cài đặt.

- Bước 1: liệt kê các thuộc tính, phương thức giống nhau diễn đạt ở các lớp.
- Bước 2: Gom các thuộc tính, phương thức chung tạo thành lớp cha kế thừa
- Bước 3: Thiết kế các lớp con kế thừa trong lớp Cha
- Bước 4: Xây dựng hàm main thực hiện bài toán.
- **Sơ đồ lớp**



NgayThang.java

```
//lớp ngày tháng đơn giản. Sau khi thành thạo sinh viên  
sử dụng thao tác với lớp ngày tháng cung cấp bởi java  
//khai báo lớp:
```

```
public class NgayThang {  
    private int thang; // 1-12  
    private int ngay; // 1-31 based on month  
    private int nam; // any year
```

```
//xây dựng hàm nhập  
    void nhap() {  
        Scanner s=new Scanner (System.in);  
        ngay=s.nextInt();  
        thang=s.nextInt();  
        nam=s.nextInt();  
    }
```

```
//xây dựng các phương thức get/set  
    public int getThang() {  
        return thang;  
    }  
    public int getNgay() {  
        return ngay;
```



```
    }  
    public int getNam() {  
        return nam;  
    }  
    public String toString(){  
        return nam+"/"+ngay + "/" +thang;  
    }  
}
```

```
}
```

HangHoa.java

```
public abstract class HangHoa {
```

```
    protected String maHang;  
    protected String tenHang;  
    protected String ghiChu;  
    protected double donGia;  
    protected double slgTon;  
    protected double VAT;
```

```
    public abstract String danhGiaMucDoBan();
```

```
    public String getGhiChu() {  
        return ghiChu;  
    }  
}
```

```
    public void nhap() {  
        Scanner sc=new Scanner(System.in);  
        System.out.print(" Mã hàng:");  
        maHang=sc.nextLine();  
        System.out.print(" Tên hàng:");  
        tenHang=sc.nextLine();  
        System.out.println("Đơn giá:");  
        donGia=sc.nextDouble();  
        System.out.print("Số lượng tồn kho");  
        slgTon=sc.nextDouble();  
    }  
}
```

```
    public String getMaHang() {  
        return maHang;  
    }  
  
    static void inTieuDe(){  
        System.out.printf("%10s %10s %10s %10s %10s",  
            "Mã hàng","tên hàng","đơn giá","số lượng","VAT");  
    }  
}
```

```
    public void inDL(){  
        System.out.printf("%10s %10s %10.1f %10.1f %10.1f",  
            maHang,tenHang,donGia,slgTon,VAT);  
    }  
}
```



```
}  
} //end off class
```

HangSanhSu.java

```
class HangSanhSu extends HangHoa{  
    private String nsx;  
    private NgayThang ngayNhapKho=new NgayThang();  
    public String danhGiaMucDoBan () {  
        Date d=new Date();  
        //lấy ngày hiện tại của hệ thống  
        Calendar c = Calendar.getInstance();  
        d=c.getTime();  
        c.set(ngayNhapKho.getNam(), ngayNhapKho.getThang(),  
            ngayNhapKho.getNgay());  
        //hết hạn nếu nhập kho hơn 10 ngày  
        c.roll(Calendar.DAY_OF_MONTH,10);  
        if ( (slgTon>50 )  
            &&(d.compareTo(c.getTime()))>0)  
            return "Bán chậm";  
        else return "Bình thường";  
    }  
    public void nhap() {  
        super.nhap();  
        Scanner sc=new Scanner(System.in);  
        System.out.print("Năm sx:");  
        nsx=sc.nextLine();  
        System.out.print("Ngày nhập kho:");  
        ngayNhapKho.nhap();  
        ghiChu=danhGiaMucDoBan();  
        VAT=0.1;  
    }  
    static void inTieuDe() {  
        HangHoa.inTieuDe();  
        System.out.printf("%15s %20s %20s %n",  
            "Nhà sx", "Ngày nhập kho", " Ghi chú");  
    }  
    public void inDL() {  
        super.inDL();  
        System.out.printf("%15s %20s %20s %n",  
            nsx, ngayNhapKho, ghiChu);  
    }  
}
```




```
}
```

HangThucPham.java

```
public class HangThucPham extends HangHoa{
```

```
    private NgayThang ngaySX;  
    private NgayThang ngayHetHan;  
    private String nhaCungCap;
```

```
public String danhGiaMucDoBan() {
```

```
    Calendar c=Calendar.getInstance();  
    Date d= c.getTime();  
    c.set(ngayHetHan.getNam(), ngayHetHan.getThang() -  
        1, ngayHetHan.getNgay());  
    if ( (slgTon!=0 ) && (d.compareTo(c.getTime()))>0)  
        return "Kho ban";  
    else return "Bình thường";  
}
```

```
static void inTieuDe() {
```

```
    HangHoa.inTieuDe();  
    System.out.printf("%15s %10s %15s %20s %n",  
        "Ngày sx", "Ngày hết hạn", "Nhà cc", " Ghi chú");  
}
```

```
public void inDL() {
```

```
    super.inDL();  
    System.out.printf("%15s %10s %15s %20s %n",  
        ngaySX, ngayHetHan, nhaCungCap, ghiChu);  
}
```

```
public void nhap() {
```

```
    super.nhap();  
    Scanner sc=new Scanner(System.in);  
    System.out.print("Nhà cung cấp:");  
    nhaCungCap=sc.nextLine();  
    System.out.print("Ngày sx:");  
    ngaySX=new NgayThang(); ngaySX.nhap();  
    System.out.print("Ngày hết hạn");  
    ngayHetHan=new NgayThang(); ngayHetHan.nhap();  
    ghiChu=danhGiaMucDoBan();  
    VAT=0;
```

```
}
```

```
}
```

QLSieuThiDemo.java

```
public class QLSieuThiDemo {
    //khai bao danh sach mat hang
    static HangHoa dsHang[];
    static int n;

    static void nhapHang() {
        int chon, j, i=0;
        System.out.println("Nhập số lượng hàng trong siêu thị=");
        Scanner s=new Scanner(System.in);
        n=s.nextInt();
        dsHang=new HangHoa[n];
        do{
            System.out.println("Nhập: 1- điện máy; 2- sành sứ; 3.thực phẩm");
            chon=s.nextInt();
            HangHoa h;
            if(chon==1)
                h=new HangDienMay();
            else if (chon==2)
                h=new HangSanhSu();
            else
                h=new HangThucPham();
            h.nhap();
            for( j=0;j<i;j++)
            {
                if
(h.getMaHang().equalsIgnoreCase(dsHang[j].getMaHang())) {
                    System.out.println("Hàng đã tồn tại");
                    break;
                }
            }
            if (i==j) {
                dsHang[i]=h;
                i=i+1;
            }
        }
        while (i<n);
    }
}
```

```
static void inDSHang() {
    //in danh sach sau khi nhap
    System.out.println("Danh sach hang sanh su:");
    HangSanhSu.inTieuDe();
    for(int i=0;i<n;i++)
        if (dsHang[i] instanceof HangSanhSu)
            dsHang[i].inDL();
    System.out.println("Danh sách hàng điện máy:");
    HangDienMay.inTieuDe();
    for(int i=0;i<n;i++)
        if (dsHang[i] instanceof HangDienMay)
            dsHang[i].inDL();
    System.out.println("Danh sách hàng thực phẩm:");
    HangThucPham.inTieuDe();
    for(int i=0;i<n;i++)
        if (dsHang[i] instanceof HangThucPham)
            dsHang[i].inDL();
}

public static void main(String[] args) {
    //nhap hang
    nhapHang();
    //in danh sach sau khi nhap
    inDSHang();
}
```