

TÀI LIỆU THỰC TẬP LẬP TRÌNH MẠNG: LAB 03

DANH MỤC THUẬT NGỮ TIẾNG ANH

Từ	Nghĩa của từ
abstract	Trừu tượng
break	Dừng vòng lặp
catch	Từ khóa đầu của một khối bắt ngoại lệ
continue	Bỏ qua phần cuối vòng lặp, tiếp tục sang bước tiếp theo
default	Giá trị mặc định của phương thức switch()
extends	Kế thừa
final	Một hằng số, phương thức hay một lớp không được ghi đè
finally	Một phần của khối xử lý ngoại lệ try luôn được thực hiện
implements	Thực hiện giao diện
import	Khai báo một gói thư viện
instanceof	Kiểm tra một đối tượng là một thể hiện của lớp
interface	Giao diện
new	Tạo một đối tượng mới của lớp
null	Tham chiếu rỗng
package	Gói
private	Tiền tố chỉ được truy cập bởi phương thức của lớp
protected	Tiền tố được truy cập bởi phương thức của lớp, lớp con của và các lớp khác trong cùng một gói
public	Tiền tố có thể được truy cập bởi phương thức của tất cả các lớp
return	Trả về của một phương thức
super	Gọi phương thức của lớp cha
synchronized	Đồng bộ
this	Tham chiếu đến đối tượng hiện tại

DANH MỤC CHỮ VIẾT TẮT

Chữ viết tắt	Ý nghĩa
UDP	User Datagram Protocol
TCP	Transmission Control Protocol
IP	Internet Protocol
URL	<i>Uniform Resource Locator</i>
CSDL	Cơ Sở Dữ Liệu
JDBC	Java Database Connectivity
CNTT	Công Nghệ Thông Tin
HĐH	Hệ Điều Hành
MVC	Model-View-Control
DNS	Domain Name System
API	Application Programming Interface
FTP	File Transfer Protocol
JDK	Java Development Kit
GB	GigaByte
UCLN	Ước Chung Lớn Nhất
BCNN	Bội Chung Nhỏ Nhất
RAM	Random Access Memory
RMI	Remote Method Invocation
JVM	Java Virtual Machine
NIC	Network Interface Card
ĐH KTKT CN	Đại học Kinh tế Kỹ thuật Công nghiệp

LỜI NÓI ĐẦU

Ngày nay do nhu cầu thực tế và do sự phát triển mạnh mẽ của nhiều công nghệ tích hợp, dẫn đến các chương trình ứng dụng hầu hết đều có khả năng thực hiện trên môi trường mạng. Ngôn ngữ JAVA là ngôn ngữ phù hợp để viết các ứng dụng mạng. So với lập trình thông thường, lập trình mạng đòi hỏi người lập trình hiểu biết và có kỹ năng tốt để viết các chương trình giao tiếp và trao đổi dữ liệu giữa các máy tính với nhau.

Để hỗ trợ sinh viên chuyên ngành CNTT trong nhà trường tiếp cận với kỹ thuật lập trình mới này, tiếp theo cuốn tài liệu học tập lý thuyết “**Công nghệ JAVA**”, chúng tôi xây dựng cuốn “**Bài tập lập trình mạng**”, nhằm cung cấp cho sinh viên những kiến thức và kỹ thuật cơ bản nhất để phát triển các chương trình ứng dụng mạng, thông qua các dạng bài tập từ cơ bản đến nâng cao qua các chủ đề: lập trình cơ bản, lập trình hướng đối tượng, lập trình CSDL JDBC, lập trình mạng dùng socket, lập trình phân tán với RMI. Sinh viên sẽ thực hiện các bài thực hành này trên phòng máy nhà trường.

Nội dung cuốn tài liệu bao gồm 12 bài lab chia thành các chủ đề khác nhau. Trong mỗi chủ đề chúng tôi đưa ra tóm tắt lý thuyết, bài tập mẫu, sau đó là bài tập tương tự, và bài tập tổng hợp. Kết quả qua những bài lab, sinh viên được rèn và thành thạo các kỹ năng lập trình hướng đối tượng, lập trình CSDL, lập trình với giao thức truyền thông có sẵn và khả năng tích hợp trong các ứng dụng khác nhau, nhất là các giao thức truyền thông thời gian thực, từ đó sinh viên có thể viết được các phần mềm quản lý theo mô hình MVC, xây dựng được các ứng dụng mạng, các ứng dụng tích hợp và triệu gọi lẫn nhau trên mạng Intranet (mạng cục bộ), mạng Internet (mạng toàn cầu), các hệ thống xử lý truy xuất dữ liệu phân tán hoàn chỉnh. Nội dung biên soạn phù hợp với chuẩn đầu ra của ngành CNTT và ngành mạng máy tính và truyền thông dữ liệu về kỹ năng và kiến thức. Sau khi học xong học phần này sinh viên có thể viết phần mềm quản lý, truyền thông.

Chúng tôi xin chân thành cảm ơn Thầy Nguyễn Hoàng Chiến, phó chủ nhiệm khoa, phụ trách khoa CNTT trường ĐH KTKT CN cùng với các đồng nghiệp đã đóng góp ý kiến cho cuốn tài liệu này. Vì tài liệu được biên soạn lần đầu, chúng tôi đã cố gắng hoàn chỉnh, song không tránh khỏi thiếu sót. Rất mong nhận được sự góp ý của bạn đọc để tài liệu học tập được hoàn thiện hơn.

Xin trân trọng cảm ơn!

Nhóm tác giả

LAB 3. GIAO DIỆN, LỚP TRỪU TƯỢNG [7,13]

A. MỤC TIÊU

- Trang bị cho sinh viên kỹ năng lập trình với giao diện (**interface**), lớp trừu tượng (**abstract class**)
- Triển khai code **interface** và **abstract class** trong JAVA.

B. NỘI DUNG

- Sử dụng NETBEAN để tạo **interface**.
- Thực hành các bài toán dùng **interface**, lớp trừu tượng

C. YÊU CẦU PHẦN CỨNG, PHẦN MỀM

- Máy tính cài HĐH windows, RAM tối thiểu 1GB.
- NETBEAN IDE 8.0, JDK 1.8.

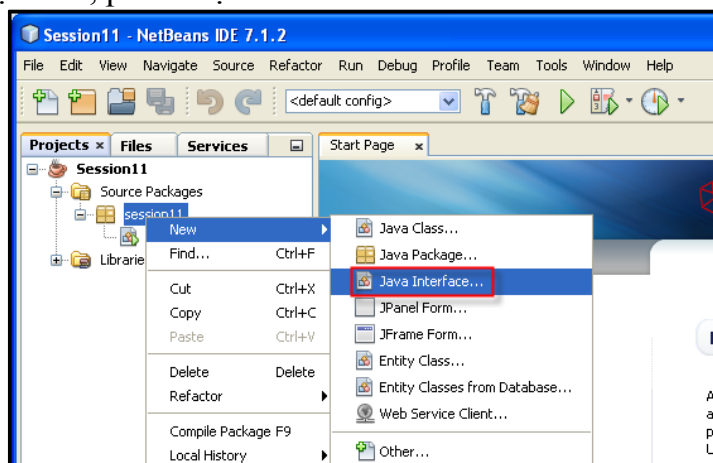
D. KẾT QUẢ SAU KHI HOÀN THÀNH

Sử dụng giao diện, lớp trừu tượng và phương thức trừu tượng giải quyết được các bài toán quản lý.

E. HƯỚNG DẪN CHI TIẾT

1. interface (giao diện)

- Khai báo các phương thức chung (không cài đặt cụ thể)
- Nếu có thuộc tính, phải được khai báo là **final**



Hình 1. Tạo interface

Bài 1.

- Tạo một giao diện IXe nằm trong gói xeco.info gồm có 2 phương thức:

```
public void nhap();  
public void hienthi();
```

- Tạo Lớp XeMay cũng nằm trong gói xeco.info thực thi giao diện IXe và có các thuộc tính sau:

```
String bienso;
String loaixe;
String mauxe;
float giatien;
```

Cài đặt Constructor, các phương thức set/get và Override các phương thức trong giao diện IXe.

- Tạo lớp XeMayHoaBinh nằm trong gói hoabinh.xemay kế thừa lớp XeMay. Bổ sung thêm thuộc tính:
 - n (kiểu **int**) : Dùng để nhập vào số lượng xe
 - Một mảng n phần tử kiểu XeMay dùng để lưu thông tin của n xe máy đang quản lý tại tỉnh Hòa Bình.

Cài đặt constructor, override tất cả các phương thức trong lớp XeMay. Phương thức nhập phải nhập vào số lượng xe và gọi nhập thông tin của n xe đó.

- Tạo lớp XeMayHaNoi nằm trong gói hanoi.xemay kế thừa lớp XeMay. Bổ sung thêm thuộc tính:
 - n (kiểu **int**): Dùng để nhập vào số lượng xe
 - Một mảng n phần tử kiểu XeMay dùng để lưu thông tin của n xe máy đang quản lý tại tỉnh Hà Nội.

Cài đặt constructor, override tất cả các phương thức trong lớp XeMay. Phương thức nhập phải nhập vào số lượng xe và gọi nhập thông tin của n xe đó.

Tạo lớp QuanLyChung nằm trong gói quanlychung.xemay, có menu như sau:

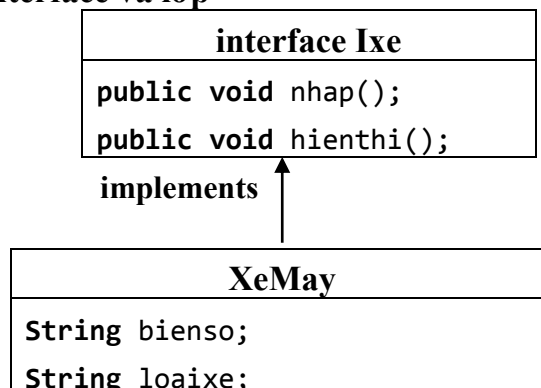
1. Nhập thông tin cho n xe máy tại tỉnh Hòa Bình.
2. Nhập thông tin cho n xe máy tại tỉnh Hà Nội.
3. Sắp xếp danh sách tăng theo biển số xe.
4. Tìm kiếm thông tin xe theo biển số xe.
5. Thống kê số lượng xe đang quản lý.
6. Thoát

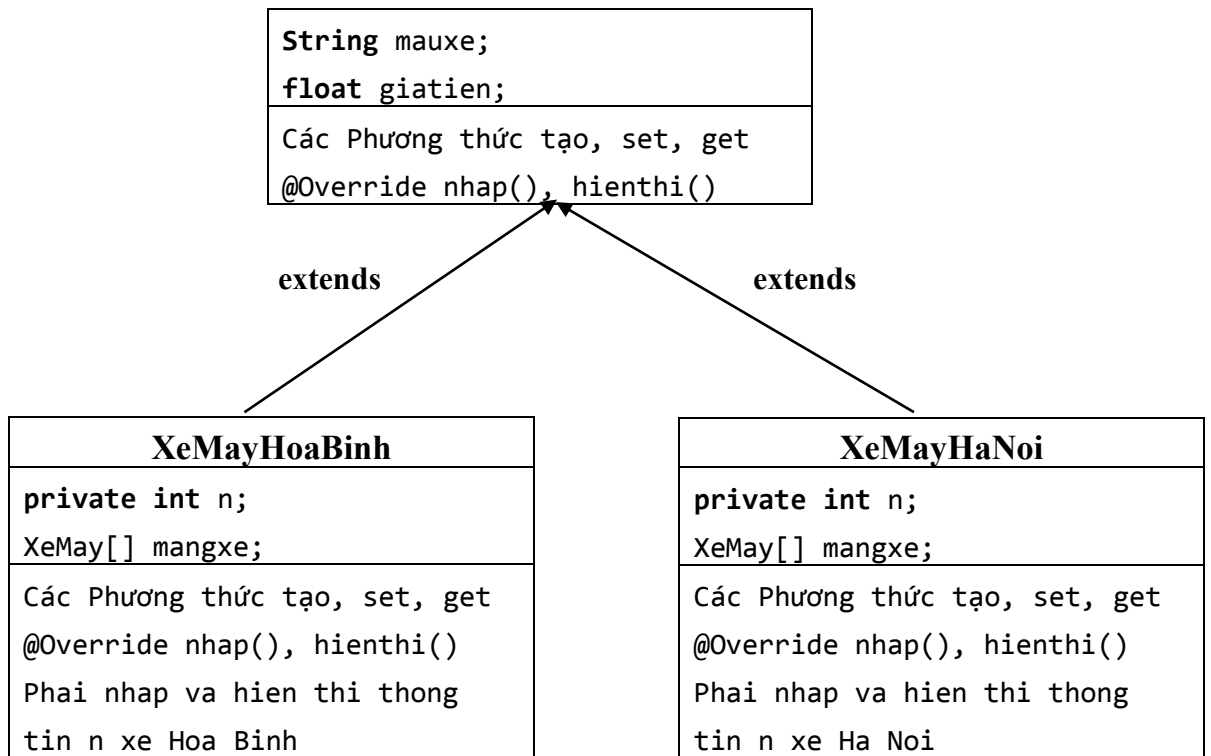
Yêu cầu:

Sau khi người sử dụng chọn chức năng 3, 4, 5 sẽ đưa ra thông báo nhập vào tỉnh cần thực hiện (Hòa Bình hoặc Hà Nội).

Hướng dẫn:

Sơ đồ phân cấp các interface và lớp





Bước 1: Tạo giao diện IXe

```

package bai2.xeco.info;
public interface IXe {
    public void nhap();
    public void hienthi();
}
  
```

Tạo lớp XeMay thực thi giao diện IXe, bổ sung các thuộc tính như đề bài

```

package bai2.xeco.info;
public class XeMay implements IXe{
    String bienso;
    String loaixe;
    String mauxe;
    float giatien;
    //Chèn các hàm tạo không tham số, 4 tham số vào đây
    //Sau đó thực hiện Override các phương thức nhap() và hienthi() để nhập
    và hiển thị thông tin xe máy.
}
  
```

Bước 2: Tạo lớp XeMayHoaBinh kế thừa lớp XeMay

```

public class XeMayHoaBinh extends XeMay{
    private int n;
    XeMay[] mangxe;
}
  
```

```

        public XeMayHoaBinh(String bienso, String loaixe, String mauxe,
float giatien, int n, XeMay[] mangxe) {
            super(bienso, loaixe, mauxe, giatien);
            this.n = n;
            this.mangxe = mangxe;
        }
        public XeMayHoaBinh() {
            super();
        }
        public int getN() {
            return n;
        }
        public void setN(int n) {
            this.n = n;
        }
        public XeMay[] getMangxe() {
            return mangxe;
        }
        public void setMangxe(XeMay[] mangxe) {
            this.mangxe = mangxe;
        }
    }
    //Ghi đề phương thức nhap()và hienthi ()
    @Override
    public void nhap() {
        Scanner sc=new Scanner(System.in);
        System.out.println("Nhap  vào số lượng xe máy ở hoa bình");
        n=Integer.parseInt(sc.nextLine());
        mangxe=new XeMay[n];
        for(int i=0;i<n;i++) {
            System.out.println("Nhap thông tin xe  máy thứ :"+(i+1));
            mangxe[i]=new XeMay();
            mangxe[i].nhap();
        }
    }
    @Override
    public void hienthi() {
        for(int i=0;i<n;i++) {
            System.out.println("Thông tin xe  máy thứ :"+(i+1));
            mangxe[i].hienthi();
        }
    }
}

```

```
}  
}
```

Bước 3: Tương tự tạo lớp XeMayHaNoi kế thừa từ lớp XeMay, thực hiện giống lớp XeMayHoaBinh.

Bước 4: Tạo lớp QuanLyChung có hàm main và sử dụng

Bài 2.

- Tạo **interface** Imark trong package đặt tên uneti có hai phương thức:

```
public void input()  
public void display()
```

- Tạo một lớp đặt tên StudentUneti trong package uneti **implements interface** IMark và bổ sung các thuộc tính sau:

```
private String StuId;  
private String StuName;  
private String gender;  
private String birthday;  
private String nativePlace;
```

Tạo hai constructors cho **class** này.

Tạo phương thức set/get cho các thuộc tính.

Ghi đè các phương thức được thừa kế trong **interface**.

- Tạo lớp đặt tên StudentMark trong package uneti.mark **implements interface** IMark và thêm vào các thuộc tính sau:

```
private String StuId;  
private String className;  
private String subjectName;  
private int semester;  
private float mark;
```

Tạo hai phương thức tạo cho **class** này.

Tạo phương thức set/get cho các thuộc tính.

Ghi đè các phương thức được thừa kế trong **interface**.

Tạo **class** tên StudentMarkTotal trong package uneti.mark kế thừa lớp StudentMark và thêm vào các thuộc tính sau:

```
private int totalExamSubject;  
private float everageMark;
```

Tạo hai phương thức tạo cho **class** này.

Tạo phương thức set/get cho các thuộc tính.

Ghi đè các phương thức được kế thừa

Thêm phương thức tạo:

```
public void getTotalExamSubject(StudentMarkTotal [] list); //Tính tổng  
số môn thi
```



```
public void calculateEverageMark(StudentMarkTotal[] list); //Tính điểm
trung bình các môn thi
```

Tạo **class** Manager trong package uneti.mark thực hiện tạo menu sau:

1. Nhập thông tin n sinh viên Uneti.
2. Nhập m điểm thi cho các sinh viên này
3. Sắp xếp sinh viên theo tên và hiển thị
4. Tìm thông tin điểm thi theo id của sinh viên
5. Exit

Chú ý:

Danh sách n sinh viên khai báo trong lớp StudentUneti

Danh sách n điểm thi được khai báo trong lớp StudentMarkTotal

Khi nhập điểm thi cho mỗi sinh viên, phải kiểm tra StuId có tồn tại trong danh sách thông tin sinh viên không.

Hướng dẫn:

Bước 1: Tạo **interface** Imark

Bước 2: Tạo lớp StudentUneti thực thi **interface** Imark

```
public class StudentUneti implements Imark{
    private String stuId;
    private String stuName;
    private String gender;
    private String birthday;
    private String nativePlace;
    //Chèn các Phương thức tạo không tham số, 5 tham số vào đây.
    //Sau đó thực hiện Override các phương thức nhap() và hienthi() để nhập
    và hiển thị thông tin sinh viên.
}
```

Tương tự tạo lớp StudentMark thực thi **interface** Imark

Bước 3: Tạo **class** StudentMarkTotal kế thừa lớp StudentMark

```
public class StudentMarkTotal extends StudentMark{
    private int totalExamSubject;
    private float everageMark;
    public StudentMarkTotal(int totalExamSubject, float everageMark) {
        super();
        this.totalExamSubject = totalExamSubject;
        this.everageMark = everageMark;
    }
    public StudentMarkTotal() {
        super();
    }
}
```

```

    public int getTotalExamSubject() {
        return totalExamSubject;
    }
    public int getTotalExamSubject(StudentMarkTotal[] arr) {
        return arr.length;
    }
    public void calculateEverageMark(StudentMarkTotal[] arr) {
        for (int i = 0; i < arr.length; i++) {
            everageMark+=arr[i].getMark();
        }
    }
    public void setTotalExamSubject(int totalExamSubject) {
        this.totalExamSubject = totalExamSubject;
    }
    public float getEverageMark() {
        return everageMark;
    }
    public void setEverageMark(float everageMark) {
        this.everageMark = everageMark;
    }
}

```

Bước 4: Tạo class Manager

```

public class Manager {
    public static void main(String[] args) {
        int n=0;
        StudentUneti[] std=null;
        StudentMark[] stdmark = null;
        do {
            Scanner sc=new Scanner(System.in);
            System.out.println("1 Nhap thong tin n sinh vien uneti");
            System.out.println("2 Nhap m diem cho cac sinh vien nay");
            System.out.println("3 sap xep sinh vien theo ten va hien thi ");
            System.out.println("4 Tim thong tin diem cua sinh vien theo id");
            System.out.println("Nhap vao lua chon cua ban");
            n=Integer.parseInt(sc.nextLine());
            switch (n) {
                case 1:
                    int m;
                    System.out.println("nhap vao so sinh vien :");
                    m=Integer.parseInt(sc.nextLine());

```

```

        std=new StudentUneti[m];
        for (int i = 0; i < m; i++) {
            std[i]=new StudentUneti();
            int dem=0;
            do {
                System.out.println("Nhap vao thong tin sinh vien thu "+(i+1));
                std[i].input();
                dem=0;
                for (int j = 0; j < i; j++) {
                    if(std[i].getStuId().equals(std[j].getStuId())&&std!=null) {
                        System.out.println("ma id trung vui long nhap lai");
                        dem++;
                    }
                }
            }while(dem!=0);
        }
        break;
    case 2:
        if(std!=null) {
            int k;
            System.out.println("nhap vao so sinh vien :");
            k=Integer.parseInt(sc.nextLine());
            stdmark=new StudentMark[k];
            for (int i = 0; i < stdmark.length; i++) {
                stdmark[i]=new StudentMark();
                int dem=0;
                do {
                    System.out.println("Nhap vao thong tin mon hoc thu "+(i+1));
                    stdmark[i].input();
                    dem=0;
                    for (int j = 0; j < std.length; j++) {
                        if(stdmark[i].getStuID().equals(std[j].getStuId())==false)
                            System.out.println("ma id sinh vien chua co vui long nhap lai");
                        dem++;
                    }
                }
            }while(dem!=0);
        }
    }
}

else {

```

```

System.out.println("can nhap thong tin cho sinh vien truoc");
    }
    break;
case 3:
    if(std==null) {
        System.out.println("ban chua nhap du lieu ");
    }
    else {
        for (int i = 0; i < std.length- 1; i++) {
            for (int j = i + 1; j <std.length; j++) {
                if(std[i].getStuName().compareTo(std[j].getStuName())>0)
                {
                    StudentUneti tg=std[i];
                    std[i]= std[j];
                    std[j]=tg;
                }
            }
        }
        for (int i = 0; i < std.length; i++) {
            System.out.println("thong tin sinh vien thu "+(i+1));
            std[i].display();
        }
    }
    break;
case 4:
    if(std==null||stdmark==null) {
        System.out.println("ban chua nhap du lieu ");
    }
    else {
        String id;
        System.out.println("Mhap vao id ban muon tim:");
        id= sc.nextLine();
        System.out.println("Thong tin sinh vien ");
        for (int i = 0; i < std.length; i++) {
            if(id.equals(std[i].getStuId())){
                stdmark[i].display();
            }
        }
        System.out.println("voi cac diem :");
        for (int i = 0; i < stdmark.length; i++) {

```

```

        if(id.equals(stdmark[i].getStuID())){
            stdmark[i].display();
        }
    }
    break;
case 5:
    break;
default:
    System.out.println("khong co lua chon cua ban");
    break;
}
}while(n!=5);
}
}

```

2. Lớp trừu tượng

- Lớp chứa phương thức trừu tượng bắt buộc phải khai báo là lớp trừu tượng.
- Phương thức trừu tượng không được phép định nghĩa cụ thể tại lớp cha
- Lớp con kế thừa lớp trừu tượng phải định nghĩa nội dung phương thức trừu tượng.

Bài 3.

Xây dựng lớp Employee có name và phương thức trừu tượng là earnings(). Xây dựng lớp Boss kế thừa từ Employee có cách tính lương là một khoản cố định hàng tháng. Xây dựng lớp PieceWorker có cách tính lương dựa trên số sản phẩm làm được, lương một sản phẩm là \$ 0.5. Xây dựng lớp CommissionWorker có cách tính lương là một khoản cố định + tiền hoa hồng trên số sản phẩm bán được, mỗi sản phẩm được \$ 0.1 hoa hồng. Viết chương trình có Phương thức main và sử dụng.

Hướng dẫn:

Bước 1: Tạo lớp trừu tượng Employee

```

public abstract class Employee {
    private String name;
    private float luong;
    public abstract float earnings();
    public abstract void hienthi();
    public float getLuong() {
        return luong;
    }
    public void setLuong(float luong) {
        this.luong = luong;
    }
    public String getName() {

```

```

        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
    public void nhap() {
        Scanner sc=new Scanner(System.in);
        System.out.println("Nhap vao ten ");
        name=sc.nextLine();
        earnings();
    }
    public void display() {
        System.out.println("Name :"+name);
        hienthi();
    }
}

```

Bước 2: Tạo lớp Boss kế thừa từ Employee

```

public class Boss extends Employee {
    private float luong;
    public float getLuong() {
        return luong;
    }
    public void setLuong(float luong) {
        this.luong = luong;
    }
    @Override
    public float earnings() {
        return luong=4500;
    }
    @Override
    public void hienthi() {
        System.out.println("luong :"+luong);
    }
}

```

Bước 3 : Xây dựng lớp PieceWorker kế thừa Employee

```

public class PieceWorker extends Employee {
    float soluongsanpham;
    float luong;
    @Override

```

```

    public float earnings() {
        Scanner sc=new Scanner(System.in);
        System.out.println("Nhap vao so luong san pham");
        soluongsanpham=Float.parseFloat(sc.nextLine());
        return luong=(float) (soluongsanpham*0.5);
    }
    @Override
    public void hienthi() {
        System.out.println("luong :"+luong);
    }
}

```

Bước 4: Xây dựng lớp CommissionWorker kế thừa Employee

```

public class CommissionWorker extends Employee{
    float luong;
    @Override
    public float earnings() {
        return luong=new Boss().earnings()+new PieceWorked().earnings();
    }
    @Override
    public void hienthi() {
        System.out.println("luong :"+luong);
    }
}

```

Bước 5: Tạo class Test có Phương thức main và sử dụng.

```

public class Test {
    public static void main(String[] args) {
        Employee[] arr=new Employee[3];
        System.out.println("Nhap Thong tin boss");
        arr[0]=new Boss();
        arr[0].nhap();
        System.out.println("Nhap Thong tin PieceWorked ");
        arr[1]=new PieceWorker();
        arr[1].nhap();
        System.out.println("Nhap Thong tin CommisstionWorked ");
        arr[2]=new CommissionWorker();
        arr[2].nhap();
        System.out.println("Thong tin boss");
        arr[0].display();
        System.out.println("Thong tin PieceWorked");
    }
}

```

```

arr[1].display();
System.out.println("Thông tin CommisstionWorked");
arr[2].display();
    }
}

```

Bài 4.

Tạo lớp SinhVienPoly gồm hai thuộc tính họ tên và ngành cùng phương thức trừu tượng là getDiem(). Thêm phương thức getHocLuc() để xếp loại học lực. Lớp cũng bao gồm một phương thức xuất để xuất họ tên, ngành, điểm và học lực ra màn hình.

Hướng dẫn:

Xây dựng lớp SinhVienPoly có mô hình sau

SinhVienPoly
+hoTen: String
+nganh: String
SinhVienPoly (hoTen , nganh)
+getDiem(): double
+getHocLuc(): String
+xuat():void

Vì chưa biết sinh viên này học những môn nào nên chưa tính được điểm vì vậy phương thức getDiem() phải là trừu tượng.

Chú ý: Lớp SinhVienPoly phải là lớp trừu tượng vì có phương thức getDiem() là phương thức trừu tượng.

Phương thức getHocLuc() viết bình thường vẫn sử dụng phương thức getDiem() để lấy điểm của sinh viên. Học lực tính như sau:

Yếu: Nếu điểm <5

Trung Bình: Nếu $5 \leq \text{điểm} \leq 7$

Khá: Nếu $7 < \text{điểm} < 8$

Giỏi: Nếu điểm ≥ 8

Bài 5. Tạo lớp SinhVienIT và SinhVienBiz kế thừa từ lớp SinhVienPoly

SinhVienIT gồm các thuộc tính điểm java, html, css. Ghi đè phương thức getDiem() để tính điểm cho sinh viên IT theo công thức $(2 * \text{java} + \text{html} + \text{css}) / 4$.

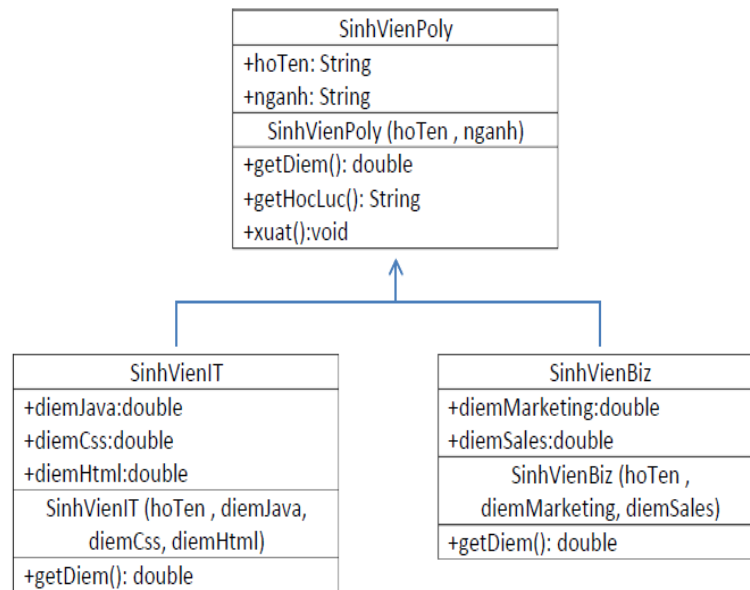
SinhVienBiz gồm các thuộc tính điểm marketing, sales. Ghi đè phương thức getDiem() để tính điểm cho sinh viên Biz theo công thức $(2 * \text{marketing} + \text{sales}) / 3$.

Sau đó viết chương trình quản lý sinh viên gồm các chức năng sau:

1. Nhập danh sách sinh viên
2. Xuất thông tin sinh viên
3. Xuất danh sách sinh viên có học lực giỏi
4. Sắp xếp danh sách sinh viên theo điểm
5. Kết thúc.

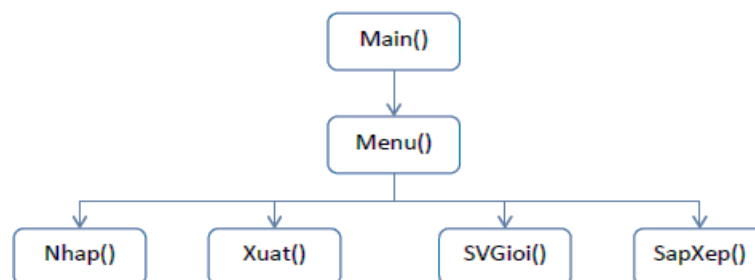
Hướng dẫn:

Bước 1: Tổ chức các lớp theo phân cấp kế thừa sau:



Ghi đè phương thức `getDiem()` trên 2 lớp **SinhVienIT** và **SinhVienBiz** theo yêu cầu của đề để tính điểm cho sinh viên của các ngành.

Bước 2: Tổ chức chương trình theo các chức năng sau



Bài 6. Tổng hợp

Một trung tâm tin học cần quản lý giảng viên cơ hữu và giảng viên thỉnh giảng.

Giảng viên cơ hữu ký hợp đồng lao động lớn hơn một năm được hưởng thu nhập hàng tháng bao gồm lương thỏa thuận cố định và lương cộng thêm trong trường hợp vượt giờ quy định trong tháng (số giờ quy định trong tháng là 40)

Giảng viên tham gia giảng dạy thỉnh giảng ký hợp đồng lao động theo từng lớp học được hưởng thu nhập hàng tháng theo số giờ đứng lớp. Biết rằng mỗi giờ dạy có giá 200.000đ.

Thông tin giảng viên cơ hữu: tên giảng viên, email, địa chỉ, điện thoại, số giờ giảng dạy trong tháng, lương thỏa thuận và số giờ quy định chung trong tháng.

Thông tin giảng viên thỉnh giảng: tên giảng viên, email, địa chỉ, điện thoại, cơ quan làm việc, số giờ giảng dạy trong tháng

Xây dựng chương trình cho phép nhân viên trong trung tâm thực hiện các chức năng sau:

1. Nhập vào thông tin của giảng viên
2. Xuất danh sách toàn bộ giảng viên
3. Xuất danh sách giảng viên cơ hữu

4. Xuất danh sách giảng viên thỉnh giảng
5. Tính tổng số tiền lương của toàn bộ giảng viên
6. Tìm giảng viên có tổng lương cao nhất.

Hướng dẫn:

Bước 1: Tạo lớp trừu tượng **GiangVien**

Do chưa biết **GiangVien** thuộc loại nào nên có các phương thức trừu tượng sau: Nhập giảng viên, hiển thị, tính lương.

```
public abstract class GiangVien {
    private String tenGiangVien;
    private String email;
    private String diachi;
    private String dienThoai;
    private float tongLuong;
    private int soGioGiang;

    Chèn Phương thức tạo không tham số, có tham số set, get
    public abstract GiangVien inputq();
    public abstract void displayq();
    public abstract void tinhluong();
    public void input(){
        Sử dụng Scanner nhập dữ liệu
    }
}

public void display(){
    Hiển thị toàn bộ các thuộc tính. Dùng System.out.print
}
}
```

Bước 2: Tạo lớp **GiangVienCoHuu** kế thừa từ lớp **GiangVien**

```
public class GiangVienCoHuu extends GiangVien{
    private float luongThoaThuan;
    private int soGioQuyDinh=40;
    public GiangVienCoHuu() {
    }
    public GiangVienCoHuu(float luongThoaThuan, int soGioQuyDinh) {
        this.luongThoaThuan = luongThoaThuan;
        this.soGioQuyDinh = soGioQuyDinh;
    }
    public GiangVienCoHuu(float luongThoaThuan, int soGioQuyDinh,
String tenGiangVien, String email, String diachi, String dienThoai, float
tongLuong, int soGioGiang) {
        super(tenGiangVien, email, diachi, dienThoai, tongLuong, soGioGiang);
    }
}
```

```

        this.luongThoaThuan = luongThoaThuan;
        this.soGioQuyDinh = soGioQuyDinh;
    }
    @Override
    public GiangVien inputq() {
        super.input();
        Scanner sc=new Scanner(System.in);
        System.out.println("Nhap luong thoa thuan:");
        this.luongThoaThuan=Float.parseFloat(sc.nextLine());
        return this;
    }
    @Override
    public void displayq() {
        super.display();
        System.out.println("luong thoa thuan :"+luongThoaThuan);
        System.out.println("so gio quy dinh :"+soGioQuyDinh);
    }
    @Override
    public void tinhluong() {
        float luong;
        luong=luongThoaThuan+(this.getSoGioGiang()-40)*200000;
        this.setTongLuong(luong);
    }
}

```

Bước 3: Tạo lớp GiangVienThinhGiang kế thừa GiangVien

```

public class GiangVienThinhGiang extends GiangVien{
    private String coQuanLamViec;
    @Override
    public GiangVien inputq() {
        super.input();
        Scanner sc=new Scanner(System.in);
        System.out.println("Nhap vao co quan lam viec :");
        coQuanLamViec=sc.nextLine();
        return this;
    }
    @Override
    public void displayq() {
        super.display();
        System.out.println("co quan lam viec :"+coQuanLamViec);
    }
}

```

```

@Override
public void tinhluong() {
    float luong;
    luong=this.getSoGioGiang()*200000;
    this.setTongLuong(luong);
}
}

```

Bước 4: Tạo lớp QuanLiGiaoVien có Phương thức main

```

public class QuanLiGiaoVien {
    public static void menu1(){
        System.out.println("1 Nhap thông tin giảng viên cơ hữu");
        System.out.println("2 nhập thông tin giảng viên thỉnh giảng");
        System.out.println("3 hiển thị toàn bộ thông tin");
        System.out.println("4 hiển thị thông tin giảng viên cơ hữu");
        System.out.println("5 hiển thị thông tin GV thỉnh giảng");
        System.out.println("6 tổng lương của toàn bộ giảng viên");
        System.out.println("7 lương cao nhất");
        System.out.println("8 thoát");
    }
    public static void main(String[] args) {
        int n = 0;
        ArrayList<GiangVien> list=new ArrayList<GiangVien>();
        do {
            menu1();
            System.out.println("Nhập vào lựa chọn của bạn :");
            Scanner sc = new Scanner(System.in);
            n = Integer.parseInt(sc.nextLine());
            switch (n) {
                case 1: {
                    System.out.println("nhập giảng viên cơ hữu :");
                    list.add(new GiangVienCoHuu().inputq());
                    break;
                }
                case 2: {
                    System.out.println("nhập giảng viên thỉnh giảng :");
                    list.add(new GiangVienThinhGiang().inputq());
                    break;
                }
                case 3: {
                    if(list==null){

```

```

        System.out.println("ban chua nhap du lieu");
    }
    else{
        System.out.println("thong tin giang vien vua nhap la :");
        int i=0;
        for (GiangVien e:list) {
            System.out.println("thong tin giang vien thu "+(i+1));
            i++;
            e.displayq();
        }
    }
    break;
}
case 4:
{
    if(list==null){
        System.out.println("ban chua nhap du lieu");
    }
    else{
        int dem=0;
        System.out.println("danh sach giang vien co huu");
        for (GiangVien i : list) {
            if (i instanceof GiangVienCoHuu) {
                i.displayq();
                dem++;
            }
        }
        if(dem==0){
            System.out.println("khong co san pham ban muon tim");
        }
    }
    break;
}
case 5:
{
    if(list==null){
        System.out.println("ban chua nhap du lieu");
    }
    else{
        int dem=0;

```

```

System.out.println("danh sach giang vien thinh giang");
        for (GiangVien i : list) {
            if (i instanceof GiangVienThinhGiang) {
                i.displayq();
                dem++;
            }
        }
        if(dem==0){
System.out.println("khong co san pham ban muon tim");
        }
    }
    break;
}
case 6:{
    if(list==null){
        System.out.println("ban chua nhap du lieu");
    }
    else{
        float tong=0;
System.out.println("tong luong cua toan bo giao vien la :");
        for (GiangVien i:list) {
            tong+=i.getTongLuong();
        }
        System.out.println(tong);
    }
    break;
}
case 7:{
    if(list==null){
        System.out.println("ban chua nhap du lieu");
    }
    else{
        System.out.println("giang vien co luong cao nhat");
        float max=0;
        for (GiangVien i:list) {
            if(max<i.getTongLuong()){
                max=i.getTongLuong();
            }
        }
        for (GiangVien i:list) {

```

```

        if(max==i.getTongLuong()){
            i.displayq();
        }
    }
}
case 8:
break;
default:{
    System.out.println("khong co lua chon cua ban ");
    break;
}
}
} while (n != 8);
}
}

```

TÀI LIỆU THAM KHẢO

- [1]. Cay S. Horstmann, *Core Java Volum I - Fundamentals, Tenth Edition*, NewYork : Prentice Hall, 2016.
- [2]. Cay S. Horstmann. *Core Java Volum II - Advanced Features, Tenth Edition*, New York : Prentice Hall, 2017.
- [3].Eng.haneen Ei-masry, *Java database connection*, Islamic University of Gaza Faculty of Engineering Department of Computer Engineering ECOM 4113: DataBase Lab, 2014.
- [4]. Angelos Stavrou, *Advanced Network Programming Lab using Java*, Network Security, ISA 656, Angelos Stavrou.
- [5]. Marenglen Biba, Ph.D, *Manual for Lab practices, Remote Method Invocation Three Tier Application with a Database Server*, Department of Comsputer Science, University of New York.
- [6].Elliotte Rusty Harold, *Java Network Programming, Fourth Edition*, O'Reilly Media, 2013.
- [7]. Đoàn Văn Ban, *Lập trình hướng đối tượng với JAVA*, Nhà xuất bản Khoa học và Kỹ thuật, 2005.
- [8]. ThS. Dương Thành Phết, *Bài tập thực hành Chuyên đề 1 CNPM- Java*, Khoa CNTT- Trường ĐH Công nghệ TP.HCM.
- [9]. <https://www.oracle.com/technetwork/java/socket-140484.html#>
- [10]. https://personales.unican.es/corcuerp/java/Labs/LAB_22.htm
- [11]. <http://www.nrcmec.org/pdf/Manuals/CSE/student/2-2%20java16-17.pdf>
- [12]. <http://cse.mait.ac.in/pdf/LAB%20MANUAL/JAVA.pdf>
- [13]. https://www.academia.edu/35283541/Bài_tập_môn_lập_trình_hướng_đối_tượng