

因为要期末考试了，在加上复习一下上机的题目。于是就进行了分析

# java之面向对象详解

## 第一个题

考察 封装类，类的继承

- 定义一个类Employee表示职员，定义Employee类的要求如下：
  - 一个私有的成员变量 name，表示职员的名字；
  - 一个私有的成员变量 monthlySalary，表示职员的月薪；
  - 一个带参数的构造器，实例化Employee对象时初始化名字和月薪；
  - name变量的getter方法；
  - monthlySalary变量的setter和getter方法；
  - 一个公有的成员方法：double getAnnualSalary(int months) 计算并返回该职员的年收入，年收入的计算公式为：月薪(monthlySalary) \* 发放月份数(months)
- 再定义一个类Manager表示经理，且Manager类继承于Employee类，经理是职员的一种，Manger类的定义要求如下：
  - 新增一个私有的成员变量 bonus，表示经理的奖金；
  - 一个带参数的构造器，实例化Manager对象时初始化其名字、月薪和奖金；
  - bonus变量的setter和getter方法；
  - 重写父类的成员方法：double getAnnualSalary(int months) 计算并返回该经理的年收入，经理年收入的计算公式为：月薪(monthlySalary) \* 发放月份数(months) + 奖金(bonus)

```
1 public class Main {
2     public static void main(String[] args) {
3         Scanner scanner = new Scanner(System.in);
4         System.out.println("请分别输入员工的姓名、月薪和经理的姓名、月薪和奖金:");
5         String name = scanner.next();
6         double monthlySalary = scanner.nextDouble();
7         String managerName = scanner.next();
8         double managerMonthlySalary = scanner.nextDouble();
9         double bonus = scanner.nextDouble();
10        Employee employee = new Employee(name,monthlySalary);
11        System.out.println("员工:"+employee.getName()+"的年薪
是:"+employee.getAnnualSalary(12));
12        Employee manager = new
Manager(managerName,managerMonthlySalary,bonus);
13        System.out.println("经理:"+manager.getName()+"的年薪
是:"+manager.getAnnualSalary(12));
14    }
15 }
```

上面给了Main类，我们就需要写一个Employee类和Manager类，并且Manager是继承在Employee类上面的

```
1 class Employee {
2     private String name;//名字 进行封装类的属性
3     private double monthlySalary;//月薪
4     public Employee(String name,double monthlySalary) {//有参数的构造方法。实例
化的时候进行调用
5         this.name=name;
6         this.monthlySalary=monthlySalary;
7     }
8     public String getName() {//get方法进行封装
9         return name;
10    }
11    public double getMonthlySalary() {
```

```

12         return monthlySalary;
13     }
14     public void setMonthlySalary(double monthlySalary) { //set方法进行封装
15         this.monthlySalary = monthlySalary;
16     }
17     public double getAnnualSalary(int months) { //在类里面定义了其他方法
18         double mon=0.0;
19         mon=this.monthlySalary*months;
20         return mon;
21     }
22
23 }

```

而下面要写的是继承类，也就是 Manager 是继承在 Employee类

```

1 class Manager extends Employee { //通过extends关键词进行继承
2     private double bonus; //奖金
3     public Manager(String name, double monthlySalary, double bonus) {
4         super(name, monthlySalary); //继承父类的属性关键词是super
5         this.bonus=bonus; //重写类的属性
6     }
7     public double getBonus() { //get方法进行封装
8         return bonus;
9     }
10    public void setBonus(double bonus) {
11        this.bonus = bonus;
12    }
13    public double getAnnualSalary(int months) { //定义了类方法对父类的方法进行了重写
14        double mon=0.0;
15        mon=getMonthlySalary()*months+this.bonus; //使用getMonthlySalary()方法
        进行调用，也就是调用父类的方法(直接调用)
16        return mon;
17    }
18
19 }

```

## 第二个题

考察:接口的定义，类去实现接口的方法

1. 定义一个接口Luminescent，表示能发光的东西，该接口有一个抽象方法: void light(); 输出该光源发光的方式;
2. 定义一个类Candle表示蜡烛，Candle类要实现Luminescent接口，Candle类实现Luminescent接口的抽象方法 void light(); 在方法中输出: "燃烧发光"
3. 定义一个类Lamp表示电灯，Lamp类要实现Luminescent接口，Lamp类实现Luminescent接口的抽象方法 void light(); 在方法中输出: "电能发光"
4. 定义一个类Sun表示太阳，Sun类要实现Luminescent接口，Sun类实现Luminescent接口的抽象方法 void light(); 在方法中输出: "太阳能发光"

Main类

```

1 public class Main {
2     public static void main(String[] args) {
3         System.out.println("请输入整数1、2或3,以选择不同的发光体。1表示蜡烛，2表示电
        灯，3表示太阳:");
4         Scanner scanner = new Scanner(System.in);
5         int select = scanner.nextInt();
6         Luminescent luminescent = null;
7         switch (select){
8             case 1:

```

```

9         luminescent = new Candle();
10        luminescent.light();
11        break;
12    case 2:
13        luminescent = new Lamp();
14        luminescent.light();
15        break;
16    case 3:
17        luminescent = new Sun();
18        luminescent.light();
19        break;
20    default:
21        System.out.println("error choice");
22    }
23 }
24 }

```

我们先声明一个接口，定义抽象方法

```

1  /**
2   * 定义一个接口Luminescent，表示能发光的东西，
3   * 该接口有一个抽象方法：void light(); 输出该光源发光的方式，
4   */
5  interface Luminescent{//定义一个接口Luminescent
6      void light();//抽象方法
7      //public abstract void light();
8  }
9  //注意接口的关键词是 interface

```

```

1  /**
2   * 定义一个类Candle表示蜡烛，Candle类要实现Luminescent接口，
3   * Candle类实现Luminescent接口的抽象方法 void light(); 在方法中输出："燃烧发光"
4   *
5   */
6  class Candle implements Luminescent{//类Candle表示蜡烛
7      public void light() {//实现的代码
8          System.out.print("燃烧发光");
9      }
10
11 }
12 //注：implements关键词用来类继承接口

```

```

1  /**
2   * 定义一个类Lamp表示电灯，Lamp类要实现Luminescent接口，
3   * Lamp类实现Luminescent接口的抽象方法 void light(); 在方法中输出："电能发光"
4   *
5   */
6  class Lamp implements Luminescent{
7      //一个类通过关键字implements声明自己使用一个或者多个接口。
8      public void light() {//实现接口方法的代码
9          System.out.print("电能发光");
10     }
11 }
12

```

```
1  /**
2   * 定义一个类Sun表示太阳，Sun类要实现Luminescent接口，
3   * Sun类实现Luminescent接口的抽象方法 void light(); 在方法中输出: "太阳能发光"
4   *
5   */
6  class Sun implements Luminescent{
7      public void light() {
8          System.out.print("太阳能发光");
9      }
10 }
```

[extends和implements区别](#)