**实 验 报 告**

**课程名称** 面向对象技术（Java）

**实验项目**  继承与接口

**实验仪器**  个人计算机 \_\_

**学 院** 计算机学院

**专 业** 计算机科学与技术

**班级/学号**\_\_\_计类1801/2018011140\_\_

**学生姓名** \_\_\_张京宇\_\_\_\_

**实验日期** 2019/4/17

**指导教师 徐 莉**

## 实验项目三 继承与接口

**一、实验目的**

掌握使用继承与接口开发Java应用程序的方法，培养学生针对特定软件需求，完成可重用软件模块的分析、设计、测试的能力，提高学生将相关知识和对象描述方法用于分析复杂工程问题、并提出解决方案的能力。

**二、实验要求**

1. 上机实验前，按照每一个实验题目的具体要求提前完成准备工作，如：查找资料、设计算法、写出全部源程序，做好预习笔记，做到有准备地上机。
2. 上机实验课内，积极调试及修改程序。得到正确运行结果后，上传程序到过程化考核平台/http://211.68.40.252:8008/
3. 上机实验课后，学生应根据实验数据及结果，完成实验报告。

**三、实验设备**

硬件环境：个人电脑一台

软件环境：JDK1.8及以上版本，NetBeans8.0及以上版本

**四、实验内容**

**任务1** 、（本题25分）根据给定的GeometricObject类层次图，分别实现GeometricObject、Circle和Triangle类。要求：

（1）按照类图3-1，定义各类的数据域属性和方法。

（2）默认的GeometricObject的颜色为white。默认的Circle对象的半径为10。默认的Triangle对象的三边为（3，4，5）。

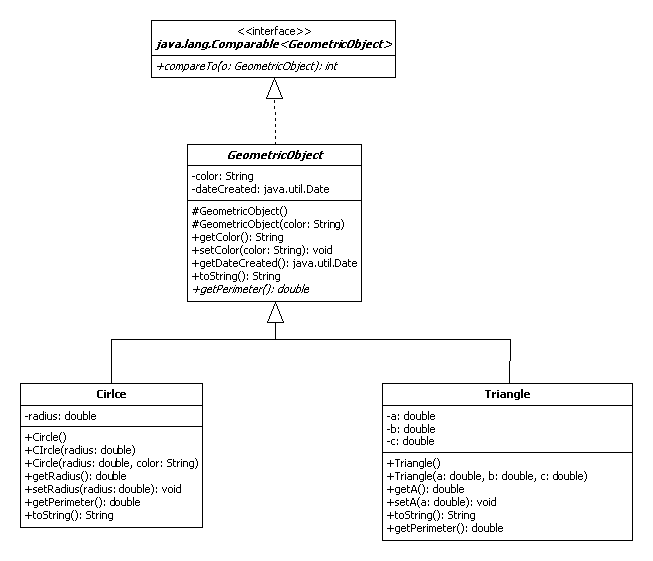
（3）GeometricObject类的toString()方法，返回对象的颜色信息。

（4）Circle的toString()方法，覆盖父类的toString()，分别返回颜色、半径的信息。

（5）Triangle的toString()方法，覆盖父类的toString()，分别返回颜色、边长的信息。

（6）GeometricObject类实现java.lang.Comparable接口，覆盖public int compareTo()方法，实现两个几何对象在周长上的比较。

（5）Circle和Triangle类分别覆盖实现getPerimeter()方法。



编写一个测试程序，在main()方法中实现下述功能：

（1）创建一个几何对象数组GeometricObject[] object；

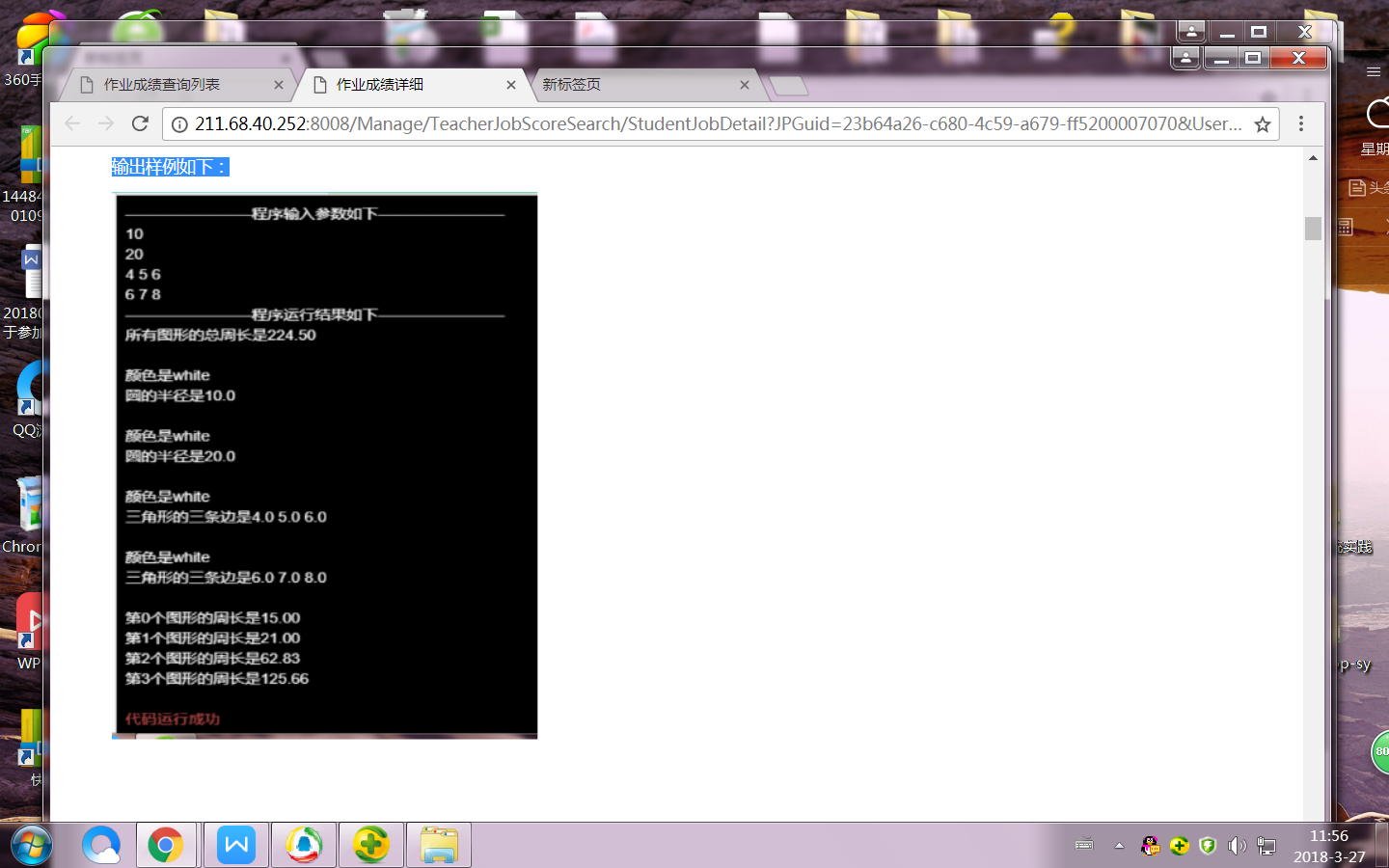
（2）分别存放两个圆和两个三角形，从键盘输入圆的半径和三角形的三条边长；

（3）计算这一组几何对象的总周长并输出；

（4）输出每个几何对象的相关信息；

（5）使用java.util.Arrays.sort()方法，对几何对象数组中的几何图形，按周长进行排序，并输出排序结果。

输出样例如下：



**程序清单：**

import java.util.Scanner;

import java.util.Date;

import java.util.Arrays;

/\*\*

\*

\* @author Zam90

\*/

public class TestGeometricObject {

/\*\*

\* @param args the command line arguments

\*/

public static void main(String[] args) {

Scanner input = new Scanner(System.in);

double sum = 0;

GeometricObject[] object = new GeometricObject[4];

Circle c1 = new Circle(input.nextDouble());

Circle c2 = new Circle(input.nextDouble());

Triangle t1 = new Triangle(input.nextDouble(),input.nextDouble(),input.nextDouble());

Triangle t2 = new Triangle(input.nextDouble(),input.nextDouble(),input.nextDouble());

object[0] = c1;

object[1] = c2;

object[2] = t1;

object[3] = t2;

for(int i = 0;i<object.length;i++)

{

sum += object[i].getPerimeter();

}

System.out.printf("所有图形的总周长是%.2f\n",sum);

for(int i = 0;i<object.length;i++)

{

System.out.println(object[i].toString());

}

double perimeter[] = {c1.getPerimeter(),c2.getPerimeter(),t1.getPerimeter(),t2.getPerimeter()};

java.util.Arrays.sort(perimeter);

for(int i = 0;i<perimeter.length;i++)

{

System.out.printf("第%d个图形的周长是%.2f\n",i,perimeter[i]);

}

}

}

abstract class GeometricObject implements Comparable<GeometricObject>

{

private String color;

private Date dateCreated = new Date();

protected GeometricObject()

{

color = "white";

}

protected GeometricObject(String c)

{

color = c;

}

public abstract int compareTo(GeometricObject o);

public String getColor()

{

return color;

}

public void setColor(String color)

{

this.color = color;

}

public Date getDateCreated()

{

return dateCreated;

}

@Override

public String toString()

{

return "颜色是"+color;

}

public abstract double getPerimeter();

}

class Circle extends GeometricObject

{

private double radius;

public Circle()

{

radius = 10;

setColor("white");

}

public Circle(double r)

{

radius = r;

setColor("white");

}

public Circle(double r,String c)

{

radius = r;

setColor(c);

}

public double getRadius()

{

return radius;

}

public void setRadius(double radius)

{

this.radius = radius;

}

@Override

public double getPerimeter()

{

return 2 \* Math.PI \* radius;

}

@Override

public String toString()

{

return "颜色是"+getColor()+"\n圆的半径是"+radius+"\n";

}

@Override

public int compareTo(GeometricObject o)

{

if(getPerimeter()>o.getPerimeter())

return 1;

else if(getPerimeter()>o.getPerimeter())

return -1;

else

return 0;

}

}

class Triangle extends GeometricObject

{

private double a,b,c;

public Triangle()

{

a = 3;

b = 4;

c = 5;

setColor("white");

}

public Triangle(double a,double b,double c)

{

this.a = a;

this.b = b;

this.c = c;

setColor("white");

}

public double getA()

{

return a;

}

public void setA(double a)

{

this.a = a;

}

@Override

public double getPerimeter()

{

return a + b + c;

}

@Override

public String toString()

{

return "颜色是"+getColor()+"\n三角形的三条边是"+a+" "+b+" "+c+"\n";

}

@Override

public int compareTo(GeometricObject o)

{

if(getPerimeter()>o.getPerimeter())

return 1;

else if(getPerimeter()>o.getPerimeter())

return -1;

else

return 0;

}

}

**任务1实验小结：**

主要谈谈你在该实验中用到了哪些具体技术或知识点？该技术或知识点解决了什麽问题？你的设计的优缺点？你在实验中遇到的问题，你是如何解决的？

本实验中我应用了接口的相关知识，利用implements关键字使 GeometricObject类实现java.lang.Comparable接口，并覆盖public int compareTo()方法，从而实现两个几何对象在周长上的比较。在设计程序中，我使用了java.util.Arrays.sort()方法，对几何对象数组中的几何图形，按周长进行排序，并输出排序结果。在实验中，一开始我并不会使用java.util.Arrays.sort()方法，后参考java手册获取使用方法。

**任务2** 、（本题25分）编写一个计算图形面积的程序，程序应当能够计算并输出矩形、圆的面积。考虑到程序的未来扩展，设计一个接口ShapeInterface，在此基础上设计实现类Rectangle（矩形类）和Circle（圆形 类），具体要求如下：

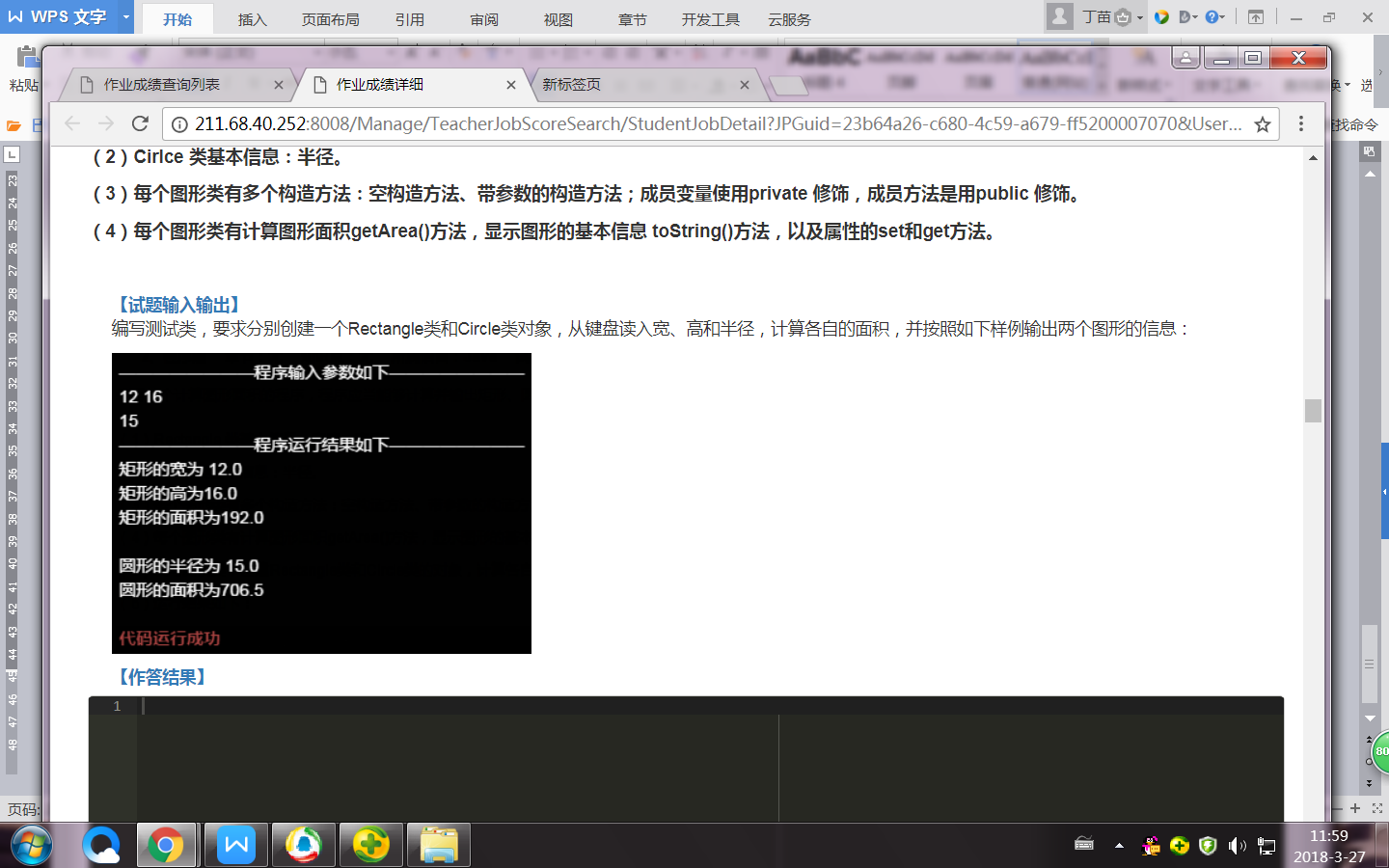
（1）Rectangle 类基本信息：宽度、高度。

（2）Cirlce 类基本信息：半径。

（3）每个图形类有多个构造方法：空构造方法、带参数的构造方法；成员变量使用private 修饰，成员方法是用public 修饰。

（4）每个图形类有计算图形面积getArea()方法，显示图形的基本信息 toString()方法，以及属性的set和get方法。

编写测试类，要求分别创建一个Rectangle类和Circle类对象，从键盘读入宽、高和半径，计算各自的面积，并按照如下样例输出两个图形的信息：



**程序清单：**

import java.util.Scanner;

/\*\*

\*

\* @author Zam90

\*/

public class Test {

public static void main(String[] args) {

Scanner input = new Scanner(System.in);

Rectangle r1 = new Rectangle(input.nextDouble(),input.nextDouble());

Circle c1 = new Circle(input.nextDouble());

System.out.print(r1.toString()+"\n\n"+c1.toString());

}

}

public interface ShapeInterface{}

class Rectangle

{

private double width,length;

public Rectangle()

{

width = 0;

length = 0;

}

public Rectangle(double w,double h)

{

width = w;

length = h;

}

public double getWidth()

{

return width;

}

public void setWidth(double width)

{

this.width = width;

}

public double getHeight()

{

return length;

}

public void setHeight(double length)

{

this.length = length;

}

public double getArea()

{

return width \* length;

}

@Override

public String toString()

{

return "矩形的宽为 "+width+"\n矩形的高为"+length+"\n矩形的面积为"+getArea();

}

}

class Circle

{

private double radius;

public Circle()

{

radius = 0;

}

public Circle(double r)

{

radius = r;

}

public double getRadius()

{

return radius;

}

public void setRadius(double radius)

{

this.radius = radius;

}

public double getArea()

{

return 3.14 \* radius \* radius;

}

@Override

public String toString()

{

return "圆形的半径为 "+radius+"\n圆形的面积为"+getArea();

}

}

**任务2实验小结：**

主要谈谈你在该实验中用到了哪些具体技术或知识点？该技术或知识点解决了什麽问题？你的设计的优缺点？你在实验中遇到的问题，你是如何解决的？

本实验中我应用了接口的相关知识，利用interface关键字创建ShapeInterface接口,并创建Rectangle（矩形类）和Circle（圆形 类）实现该接口。在设计程序中，我在类中均有计算图形面积getArea()方法，显示图形的基本信息 toString()方法，以及属性的set和get方法。在实验中，一开始我并不会创建接口，后参考java手册获取创建方法。

**任务3** 、（本题25分）一个小型公司，职员有2类，第一类是按月领取薪水，第二类按照周领取薪水，请计算公司每年需要支付的员工薪水的总额。

根据上述需求，程序设计公司类Company、职员类Employee、按月领取工资的员工类MonthWorker以及按周领取工资的员工类WeekWorker类，具体说明如下：

（1）Employee类：是从所有的员工抽象得到的abstract类。

   a）一个用protected修饰的String类型的name属性，描述员工的姓名；

   b）一个无参的构造方法；

   c）一个用于设定name属性的带参数的构造方法 Employee(String name)；

   d) name属性的修改器和访问器；

   e）一个公共的计算员工年薪的方法earnings()，返回值类型为double；

   f）一个公共的toString()方法，用于输出员工的信息，返回值类型为String。

（2）MonthWorker类：按月领取薪水的员工，父类是Employee。

   a）一个私有的double类型的monthSal属性，描述员工每月的薪水；

   b）一个无参的构造方法；

   c) 一个用于设定姓名和每月薪水的带参数的构造方法 MonthWorker(String name,double monthSal)；

   d) monthSal属性的修改器和访问器；

 e）重写父类的earnings()方法，计算员工的年薪；

   f）重写父类的toString()：输出员工的信息。

(3) WeekWorker类：按周领取薪水的员工，父类是Employee。

   a）一个私有的double类型的weekSal属性，描述员工每周的薪水；

   b）一个无参的构造方法；

   c) 一个用于设定姓名和每周薪水的带参数的构造方法 WeekWorker(String name,double weekSal)；

   d) weekSal属性的修改器和访问器；

   e）重写父类的earnings()方法，计算员工的年薪，假设一年52个周；

   f）重写父类的toString()：输出员工的信息。

(4) Company类：公司类.

   a）一个私有的Employee对象数组，用于存放公司员工，数组中的元素可以是MonthWorker对象，也可以是WeekWorker对象；

   b）一个私有的用于记录员工人数的数据域numberOfWorkers,初始值为0；

   c)一个无参的构造方法；

   d)一个公共的用于计算公司每年需要支付的员工薪水总额的方法 computeTotalSal()，返回值为double；

   e）一个公共的用于添加单个员工的方法 addEmployee(Employee e)，无返回值；

   f）一个公共的toString()方法，返回公司员工信息和每年需要支付的员工薪水总额。

要求：

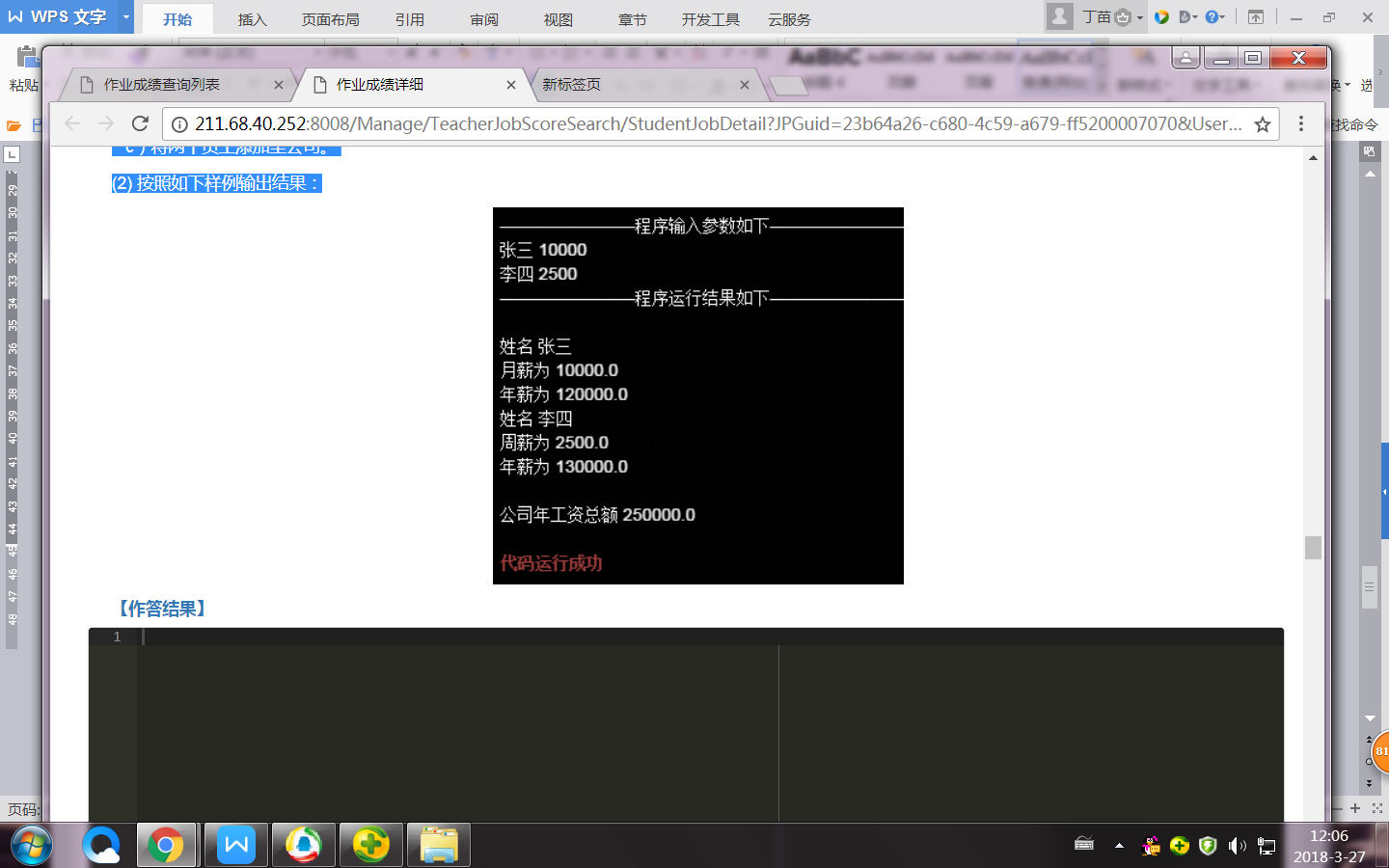
(1) 编写测试类TestCompany，要求如下：

   a) 创建公司对象；

   b）创建一个MonthWorker对象和一个WeekWorker对象，从键盘读入两个员工的姓名和薪水；

   c）将两个员工添加至公司。

(2) 按照如下样例输出结果：



**程序清单：**

**import java.util.Scanner;**

**/\*\***

**\***

**\* @author Zam90**

**\*/**

**public class Testcompany {**

**/\*\***

**\* @param args the command line arguments**

**\*/**

**public static void main(String[] args) {**

**Scanner input = new Scanner(System.in);**

**MonthWorker m = new MonthWorker(input.next(),input.nextDouble());**

**WeekWorker w = new WeekWorker(input.next(),input.nextDouble());**

**m.getNearning();**

**m.Print();**

**w.getEarning();**

**w.Printl();**

**System.out.println("");**

**Company c = new Company();**

**double z;**

**z = m.getNearning()+w.getEarning();**

**System.out.println("公司年工资总额"+z);**

**}**

**}**

**abstract class Employee{**

**Employee(){**

**}**

**protected String name;**

**public Employee(String name) {**

**this.name = name;**

**}**

**public String getName() {**

**return name;**

**}**

**public void setName(String name) {**

**this.name = name;**

**}**

**public abstract Double earning ();**

**}**

**class MonthWorker extends Employee{**

**private double monthSal;**

**double nearning;**

**public double getNearning() {**

**nearning = monthSal\*12;**

**return nearning;**

**}**

**public void setNearning(double earning) {**

**this.nearning = earning;**

**}**

**public MonthWorker() {**

**}**

**public MonthWorker(double monthSal) {**

**this.monthSal = monthSal;**

**}**

**public MonthWorker( String name,double monthSal) {**

**super(name);**

**this.monthSal = monthSal;**

**}**

**public double getMonthSal() {**

**return monthSal;**

**}**

**public void setMonthSal(double monthSal) {**

**this.monthSal = monthSal;**

**}**

**public void Print() {**

**System.out.println("姓名 "+getName());**

**System.out.println("月薪为 "+getMonthSal());**

**System.out.println("年薪为 "+getNearning());**

**}**

**@Override**

**public Double earning() {**

**double earning;**

**earning = monthSal\*12;**

**return earning;**

**}**

**}**

**class WeekWorker extends Employee{**

**private double weekSal;**

**double earning;**

**public WeekWorker() {**

**}**

**public WeekWorker(String name,double weekSal ) {**

**super(name);**

**this.weekSal = weekSal;**

**}**

**public double getWeekSal() {**

**return weekSal;**

**}**

**public void setWeekSal(double weekSal) {**

**this.weekSal = weekSal;**

**}**

**public double getEarning() {**

**earning = weekSal\*52;**

**return earning;**

**}**

**public void setEarning(double earing) {**

**this.earning = weekSal\*52;**

**}**

**@Override**

**public Double earning() {**

**double earning ;**

**earning = weekSal\*52;**

**return earning;**

**}**

**public void Printl(){**

**System.out.println("姓名 "+getName());**

**System.out.println("周薪为 "+getWeekSal());**

**System.out.println("年薪为 "+getEarning());**

**}**

**}**

**class Company{**

**private int n = 0;**

**Employee[] a = new Employee[10];**

**MonthWorker m = new MonthWorker();**

**WeekWorker w = new WeekWorker();**

**public Company() {**

**}**

**public void addEmployee(Employee e)**

**{**

**Employee a = new Employee() {**

**int i =0;**

**@Override**

**public Double earning() {**

**throw new UnsupportedOperationException("Not supported yet.");**

**}**

**};**

**}**

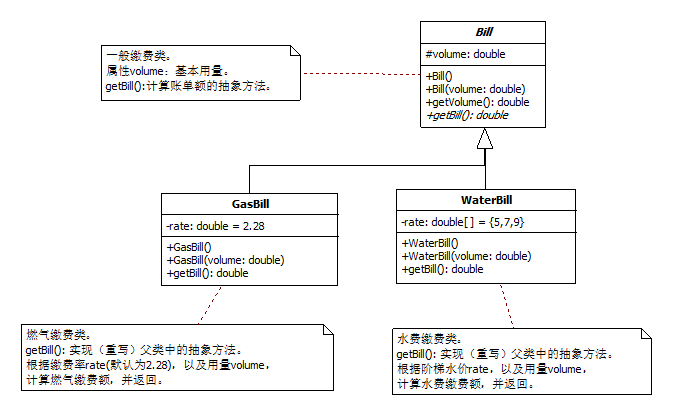
**}**

**任务3实验小结：**

主要谈谈你在该实验中用到了哪些具体技术或知识点？该技术或知识点解决了什麽问题？你的设计的优缺点？你在实验中遇到的问题，你是如何解决的？

本实验中我应用了抽象类继承的相关知识，利用extends关键字使类MonthWorker、 WeekWorker继承抽象类employee,并创建Testcompany类测试。在实验中，有少量语法问题，已经借助NetBeans解决。

**任务4**  、（本题25分）根据UML图，定义抽象缴费单类Bill，实现GasBill类，定义居民燃气缴费单类。实现WaterBill类，定义居民用水缴费单类。

****

说明：燃气费（率）按本地普通居民燃气费2.28元/立方米计算。

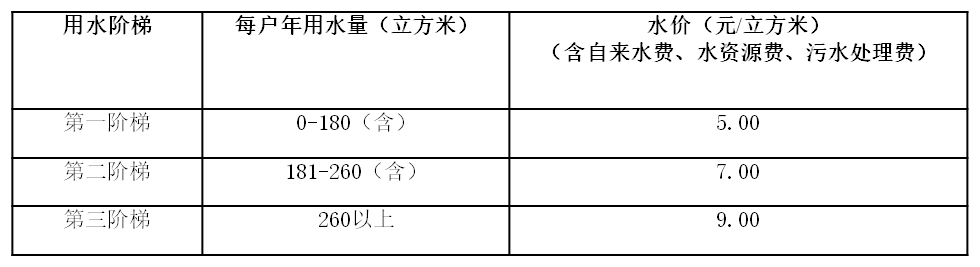
             燃气缴费总额 = 燃气费率(rate) \* 燃气总量(volume)。

      水费（率）按本地阶梯水价计算,如表1所示。

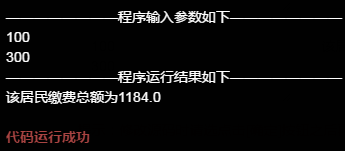
               水缴费总额 = 各阶梯水价之和

           每阶梯水价额 = 每阶梯水价(rate[i]) \* 本阶梯水量(volume)

下表 本地居民用水阶梯水价表



编写测试程序，从键盘输入某居民的用水量和用气量，计算该居民应该缴纳的水费和燃气费的总额，按照如下样例输出结果。



**程序清单：**

**import java.util.Scanner;**

**/\*\***

**\***

**\* @author Zam90**

**\*/**

**public class Testbill {**

**/\*\***

**\* @param args the command line arguments**

**\*/**

**public static void main(String[] args) {**

**Scanner input = new Scanner(System.in);**

**WaterBill w = new WaterBill(input.nextDouble());**

**GasBill g = new GasBill(input.nextDouble());**

**double sum = g.getBill() + w.getBill();**

**System.out.print("该居民缴费总额为"+sum);**

**}**

**}**

**abstract class Bill**

**{**

**protected double volume;**

**public Bill()**

**{**

**volume = 0;**

**}**

**public Bill(double volume)**

**{**

**this.volume = volume;**

**}**

**public double getVolume()**

**{**

**return volume;**

**}**

**public abstract double getBill();**

**}**

**class GasBill extends Bill**

**{**

**private double rate = 2.28;**

**public GasBill()**

**{**

**volume = 0;**

**}**

**public GasBill(double volume)**

**{**

**this.volume = volume;**

**}**

**@Override**

**public double getBill()**

**{**

**return volume \* rate;**

**}**

**}**

**class WaterBill extends Bill**

**{**

**private double[] rate = {5,7,9};**

**public WaterBill()**

**{**

**volume = 0;**

**}**

**public WaterBill(double volume)**

**{**

**this.volume = volume;**

**}**

**@Override**

**public double getBill()**

**{**

**if(volume<=180&&volume>0)**

**return rate[0] \* volume;**

**else if(volume<=260)**

**return 900 + rate[1] \* (volume-180);**

**else**

**return 1460 + rate[2] \* (volume-260);**

**}**

**}**

**任务4实验小结：**

主要谈谈你在该实验中用到了哪些具体技术或知识点？该技术或知识点解决了什麽问题？你的设计的优缺点？你在实验中遇到的问题，你是如何解决的？

本实验中我应用了抽象类继承的相关知识，我根据UML图，定义抽象缴费单类Bill，并使GasBill类、WaterBill类继承Bill类，实现GasBill类，定义居民燃气缴费单类。实现WaterBill类，定义居民用水缴费单类。在实验中，有少量语法问题，已经借助NetBeans解决

**五、实验总结**

通过本次实验，我掌握了使用继承与接口开发Java应用程序的方法，培养了我针对特定软件需求，完成可重用软件模块的分析、设计、测试的能力，使我对面向对象技术的意义与作用有了更深一层的了解，提高了我将相关知识和对象描述方法用于分析复杂工程问题、并提出解决方案的能力。