**实验四（2）接口和继承、多态**

**一、实验目的**

1、掌握接口的定义方法

2、掌握接口的实现方法

3、掌握接口回调的使用

4. 了解面向接口编程思想

**二、实验要求**

**1. 验证并运行书上第6章例6，将运行结果截图，作为实验结果提交。并回答：**

**（1）面向接口编程的主要思想是什么？（2）接口和抽象类的异同点？什么情况下适合面向接口编程？**

**2. 定义接口Shape，其中包括area方法。类Circle、Square和Triangle均实现了接口Shape。定义主函数，创建元素个数为3的Shape类型的一维数组，分别为数组元素创建Circle、Square和Triangle类型的对象，最后分别调用各数组元素的Area方法，输出相关信息。**

**下面是程序代码，根据注释，将缺失部分补全。**

**//**【代码1】**，**将Shape定义为interface，其中应包含area方法

//定义Circle类实现Shape

class Circle implements Shape {

final double pi = 3.14159;

private double r;

public Circle(double r){

this.r = r;

}

public double area(){//求圆的面积

return pi \* r \* r;

}

}

//定义Square类实现Shape

class Square implements Shape {

private double s;

public Square(double s){

this.s = s;

}

**//**【代码2】**，**重写area方法，实现求正方形面积

}

//定义Triangle类实现Shape

class Triangle implements Shape {

private double b,h;

public Triangle(double b,double h){

this.b = b;

this.h = h;

}

**//**【代码3】**，**重写area方法，实现求三角形面积

}

//包含main()的测试类

public class Ex2\_2{

static void drawOneShape(Shape s){

if (s instanceof Square)

System.out.println("正方形面积为：" + s.area());

else if (s instanceof Circle)

System.out.println("圆面积为：" + s.area());

else if (s instanceof Triangle)

System.out.println("三角形面积为：" + s.area());

}

static void drawShapes(Shape[] ss){

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

drawOneShape(ss[i]);

}

public static void main(String[] args){

Shape[] s = new Shape[3];

s[0] = new Circle(1.0);

s[1] = new Square(2.0);

s[2] = new Triangle(1.0,2.0);

drawShapes(s);

}

}

**3. 下面是关于天气预报的接口使用的例子，根据注释，将缺失部分补全。**

**WeatherState.java源文件**

public interface WeatherState {   //接口

    public void showState();

}

**Weather.java源文件**

public class Weather {

   WeatherState  state;

   public void show() {

      state.showState();

   }

   public void setState(WeatherState s) {

      state = s;

   }

}

**WeatherForecast.java源文件**

public class WeatherForecast {  //主类

   public static void main(String args[]) {

      Weather weatherBeijing =new Weather();

      System.out.print(" 今天白天:");

      weatherBeijing.setState(new CloudyDayState());

      weatherBeijing.show();

      System.out.print(" 今天夜间:");

      weatherBeijing.setState(new LightRainState());

      weatherBeijing.show();

      System.out.print("转:");

      weatherBeijing.setState(new HeavyRainState());

      weatherBeijing.show();

      System.out.print(" 明天白天:");

      weatherBeijing.setState(new LightRainState());

      weatherBeijing.show();

      System.out.print(" 明天夜间:");

      weatherBeijing.setState(new CloudyLittleState());

      weatherBeijing.show();

   }

}

**CloudyLittleState.java源文件**

public class CloudyLittleState implements WeatherState {

    public void showState() {

       System.out.print("少云,有时晴.");

    }

}

**CloudyDayState.java源文件**

public class CloudyDayState implements WeatherState {

    【代码1】 //重写public void showState()

}

**HeavyRainState.java源文件**

public class HeavyRainState implements WeatherState{

    【代码2】 //重写public void showState()

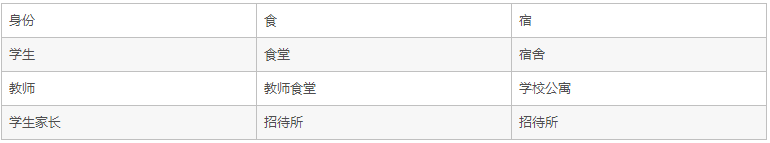
}

**LightRainState.java源文件**

public class LightRainState implements WeatherState {

    【代码3】 //重写public void showState()方法

**4. 假设一个学校接待方面的程序，招待不同身份的人的食宿问题，其对应规则如下：**



**理论上，当然可以对每个不同身份的人各定义一个对应的类，并实现各自的方法，但是观察这写类，可以归纳出其有一个共同的模板，即“人”的“食、宿”问题。这时候，就可以发挥接口的功能了。请同学们应用接口编写程序，实现如下输出：**

**程序执行结果：**

学生去食堂吃饭！

学生回寝室睡觉！

教师去教工餐厅吃饭！

教师回学校公寓睡觉！

家长去招待所饭馆吃饭！

家长回招待所睡觉！

三、提交各实验程序源代码及实验报告