## 实验4：继承与多态（1）

### 1.实验目的

（1）掌握类的继承的实现方式。

（2）理解类的继承中方法的覆盖。

### 2.实验内容

（1）设计一个Circle类，成员有半径r，有求面积和周长的方法。再设计一个继承Circle类的Sphere球类，增加求球体表面积和体积的方法。 编写测试类测试该球体类。

1. 源代码：

/\*Circle类\*/

import java.util.\*;

import java.lang.\*;

public class Circle {

public Circle(double tmp) {

// TODO Auto-generated constructor stub

r = tmp;

}

public double getlen() {

return 2 \* Math.PI \* r;

}

public double getArea() {

return Math.PI\*r\*r;

}

public double getr() {

return r;

}

private double r;

}

/\*Sphere类\*/

import java.util.\*;

import java.lang.\*;

public class Sphere extends Circle

{

public Sphere(double r) {

// TODO Auto-generated constructor stub

super(r);

}

public double get3Area() {

return 4\*getArea();

}

public double get3volume() {

return (3.0/4)\*getArea()\*getr();

}

}

/\*测试类\*/

import java.util.\*;

import java.lang.\*;

public class test1 {

public static void main(String[] args) {

// TODO Auto-generated method stub

Sphere Sp1 = new Sphere(4.0);

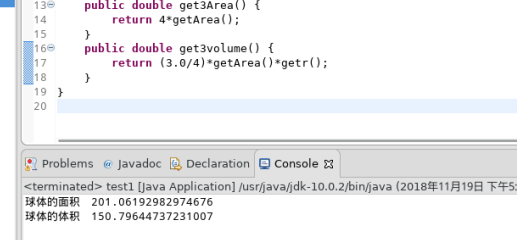
System.out.println("球体的面积　" + Sp1.get3Area());

System.out.println("球体的体积　" + Sp1.get3volume());

}

}

2.　运行截图：



（2）设计一个学生类Student，类有成员变量姓名、年龄，有“学习”方法。Student类派生出本科生类，本科生类派生出研究生类，本科生类增加专业和学位属性，覆盖学习方法。研究生类增加研究方向属性，覆盖学习方法。每个类都有显示方法，用于输出属性信息。编写测试类测试这三个类。

1.　源代码：

/\*Student 类\*/

import java.util.\*;

import java.lang.\*;

public class Student {

public Student(String na, int ag) {

name = na;

age = ag;

}

public void study() {

System.out.println("I'am a student ,I'am studying !!!!!");

}

public void print() {

System.out.print("name= " + name + " ,age= " + age);

}

private String name;

private int age;

}

/\*benke类\*

import java.util.\*;

import java.lang.\*;

public class benke extends Student {

public benke(String name,int age,String ma,String deg) {

super(name,age);

major = ma ;

degree = deg;

}

@Override

public void study() {

System.out.println("I'am a benke ,I'am studying !!!!!");

}

public void print() {

super.print();

System.out.print(" major= " + major + ", degree= " + degree);

}

private String major;

private String degree;

}

/\*Graduate　类\*/

import java.util.\*;

import java.lang.\*;

public class Graduate extends benke {

public Graduate(String name, int age, String ma, String deg, String dir) {

super(name, age, ma, deg);

research\_dir = dir;

}

@Override

public void study() {

System.out.println("I'am a Graduate ,I'am studying !!!!!");

}

public void print() {

super.print();

System.out.print(" research\_dir= " + research\_dir);

}

private String research\_dir;

}

/\*测试类\*/

import java.util.\*;

import java.lang.\*;

public class test {

public static void main(String[] args) {

Student st = new Student("小王",26);

st.print();

System.out.println();

st.study();

System.out.println();

benke ben = new benke("小张",28,"CS","CS\_degree");

ben.print();

System.out.println();

ben.study();

System.out.println();

Graduate gg = new Graduate("小刘",36,"艺术","艺术\_degree","movie");

gg.print();

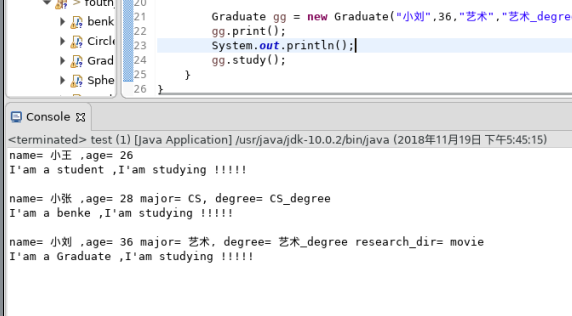
System.out.println();

gg.study();

}

}

2.　运行截图：



### 实验总结

这次实验主要练习了掌握类的继承的实现方式和类的继承中方法的覆盖。基本都比较简单．以后继续加油！！！！！！！