电子科技大学信息与软件工程学院

**实 验 报 告**

学 号2017221103014

姓 名冯世维

（实验） 课程名称 面向对象程序设计Java

理论教师周帆

实验教师何中海

**电 子 科 技 大 学**

**实 验 报 告**

**学生姓名：冯世维 学号：2017221103014 指导教师：何中海**

**实验地点：信软学院楼西304 实验时间：2018.12.16**

1. **实验名称：**类的继承、接口及访问控制
2. **实验学时：**2
3. **实验目的：**

熟悉Java的类的继承基本操作，接口及访问控制，理解面向对象程序设计的基本思想。

1. **实验原理：**

1.继承就是子类继承父类的特征和行为，使得子类对象（实例）具有父类的实例域和方法，或子类从父类继承方法，使得子类具有父类相同的行为。

2.在 Java 中通过 extends 关键字可以申明一个类是从另外一个类继承而来的，一般形式如下：

class 父类 {

}

class 子类 extends 父类 {

}

3.继承的特性有：

1. 子类拥有父类非private的属性，方法。
2. 子类可以拥有自己的属性和方法，即子类可以对父类进行扩展。
3. 子类可以用自己的方式实现父类的方法。
4. Java的继承是单继承，但是可以多重继承，单继承就是一个子类只能继承一个父类，多重继承就是，例如A类继承B类，B类继承C类，所以按照关系就是C类是B类的父类，B类是A类的父类，这是java继承区别于C++继承的一个特性。
5. 提高了类之间的耦合性（继承的缺点，耦合度高就会造成代码之间的联系越紧密，代码独立性越差）。

4.继承关键字

继承可以使用 extends 和 implements 这两个关键字来实现继承，而且所有的类都是继承于 java.lang.Object，当一个类没有继承的两个关键字，则默认继承object（这个类在 java.lang 包中，所以不需要 import）祖先类。

5. 构造器

子类是不继承父类的构造器（构造方法或者构造函数）的，它只是调用（隐式或显式）。如果父类的构造器带有参数，则必须在子类的构造器中显式地通过 super 关键字调用父类的构造器并配以适当的参数列表。

如果父类构造器没有参数，则在子类的构造器中不需要使用 super 关键字调用父类构造器，系统会自动调用父类的无参构造器。

6. Java有四种访问权限， 其中三种有访问权限修饰符，分别为private，public和protected，还有一种不带任何修饰符。

default：即不加任何访问修饰符，通常称为“默认访问模式“。该模式下，只允许在同一个包中进行访问。

protect: 介于public 和 private 之间的一种访问修饰符，一般称之为“保护形”。被其修饰的类、属性以及方法只能被类本身的方法及子类访问，即使子类在不同的包中也可以访问。

public： Java语言中访问限制最宽的修饰符，一般称之为“公共的”。被其修饰的类、属性以及方法不仅可以跨类访问，而且允许跨包（package）访问。

7. 抽象类和接口中成员或方法访问权限

关于抽象类中成员访问权限，其基本上继承了类的特性，但由于抽象类之所以为抽象类，是因为它是作为父类来使用的，是等待子类去实现的，而类中 private的权限只能是自个访问自个，所以在抽象类中方法为abstract时只有public,protected,default三种访问权限。

1. **实验内容：**

1. 完成第五章习题9、10编程。

2. 完成第六章习题7、8编程。

3. 有几何形状边数为n及可计算面积area的Shape类，其子类Triangle类及Rectangle类实现几何形状三角形和矩形面积area计算，利用前三个形状类实现柱体Pillar类的体积计算，并在PillarTest类中实现对某一柱体的体积计算。

4. 创建学生成绩中所涉及的类：Student类、Teacher类、Course类，并由Grade类将Student类、Teacher类和Course类关联起来，由GradeTest类对以上四个类进行测试。

1. **实验器材（设备、元器件）：**

一台配置好Java环境的电脑，eclipse

1. **实验步骤：**

1、创建工程

2、编辑程序

3、编译程序

4、调试程序

5、运行程序，分析结果

本实验4个小题目都分别按上述步骤进行。

1. **实验结果与分析（含重要数据结果分析或核心代码流程分析）**

**import** java.util.Scanner;

**public** **class** sa {

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

Person person1=**new** Person("冯世维",'男',19);

System.***out***.println("person1:"+person1.getDate());

Person person2=person1;

person2.setDate("李四",'男',34);

System.***out***.println("person2:"+person2.getDate());

Student student1=**new** Student("王五",'女',17);

System.***out***.println("student1:"+student1.getDate());

Student student2=student1;

student2.setDate("翠花",'女',17,001,1103);

System.***out***.println("student2:"+student2.getDate());

}

}

**class** Person{

String name;

**char** sex;

**int** age;

Person(String n,**char** s,**int** a){

name=n;

sex=s;

age=a;

}

**void** setDate(String n,**char** s,**int** a){

name=n;

sex=s;

age=a;

}

String getDate(){ **return** "name:"+name+" sex:"+sex+" age:"+age;

}

}

**class** Student **extends** Person{

**int** sID,classNo;

Student(String n,**char** s,**int** a){

**super**(n,s,a);

}

**void** setDate(String n,**char** s,**int** a,**int** i,**int** c){

**super**.setDate(n,s,a);

sID=i;

classNo=c;

}

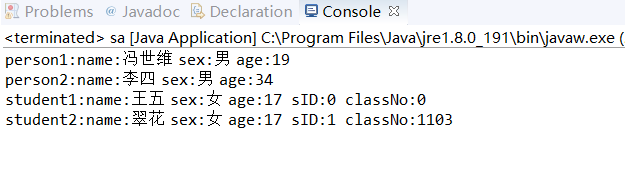
String getDate(){

**return** **super**.getDate()+" sID:"+sID+" classNo:"+classNo;

}

}

实验结果：



第五章习题10代码如下：

**public** **class** test3 {

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

Teacher teacher=**new** Teacher();

teacher.setDate("王老师",'女',29);

System.***out***.println("teacher:"+teacher.getDetail());

Student student=**new** Student();

student.setDate("张三",'男',14);

System.***out***.println("student:"+student.getDetail());

}

}

**abstract** **class** Person{

String name;

**char** sex;

**int** age;

**abstract** **void** setDate(String name,**char** sex,**int** age);

**abstract** String getDetail();

}

**class** Teacher **extends** Person{

String name;

**char** sex;

**int** age;

**int** tID;

String department;

**void** setDate(String n,**char** s,**int** a){

name=n;

sex=s;

age=a;

}

String getDetail(){

**return** "name:"+name+" sex:"+sex+" age:"+age+" tID:"+tID+" department:"+department;

}

}

**class** Student **extends** Person{

String name;

**char** sex;

**int** age;

**int** sID;

String speciality;

**void** setDate(String n,**char** s,**int** a){

name=n;

sex=s;

age=a;

}

String getDetail(){

**return** "name:"+name+" sex:"+sex+" age:"+age+" sID:"+sID+" speciality:"+speciality;

}

}

实验结果为：



第六章习题7代码如下所示：

**public** **class** test3 {

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

classA person=**new** classA();

person.setDate("Sam",'m',19);

person.print();

classB student=**new** classB();

student.setDate("Jim","math",95.5);

student.print();

}

}

**interface** Print{

**void** print();

}

**class** classA **implements** Print{

String name;

**char** sex;

**int** age;

**void** setDate(String n,**char** s,**int** a){

name=n;

sex=s;

age=a;

}

**public** **void** print(){

System.***out***.println("name:"+name+" sex:"+sex+" age:"+age);

}

}

**class** classB **implements** Print{

String name;

String subject;

**double** grade;

**void** setDate(String n,String sub,**double** g){

name=n;

subject=sub;

grade=g;

}

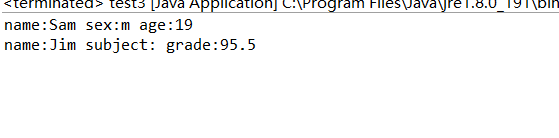
**public** **void** print(){

System.***out***.println("name:"+name+" subject:"+" grade:"+grade);

}

}

实验结果如下：



第六章习题8代码如下：

**public** **class** test34 {

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

Student student=**new** Student();

student.setDate(015,"software");

System.***out***.println(student.getDate());

}

}

**interface** Person{

String ***name***="Jimmy";

**char** ***sex***='w';

String getDate();

}

**class** Student **implements** Person{

**int** sID;

String speciality;

**public** **void** setDate(**int** s,String spec){

sID=s;

speciality=spec;

}

**public** String getDate(){

**return** "name:"+***name***+" sex:"+***sex***+" birthday:"+birthday+" sID:"+sID+" speciality:"+speciality;

}

}

实验结果如下：



实验内容3代码如下所示：

**import** java.util.Scanner;

**public** **class** test33 {

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

String str;

**double** h;

System.***out***.println("请输入柱体的高：");

h=**new** Scanner(System.***in***).nextDouble();

System.***out***.println("请输入柱体底面的形状（三角形/矩形）：");

str=**new** Scanner(System.***in***).nextLine();

**if** (str.equals("三角形")) {

Pillar pillar=**new** Pillar(3,h);

pillar.volume=pillar.getVolume();

System.***out***.println("该三棱柱的体积为："+pillar.volume);

}**else** **if** (str.equals("矩形")) {

Pillar pillar=**new** Pillar(4,h);

pillar.volume=pillar.getVolume();

System.***out***.println("该四棱柱的体积为："+pillar.volume);

}**else**{

System.***out***.println("抱歉，输入错误，您只能选择三角形或者矩形。");

System.*exit*(1);

}

}

}

**abstract** **class** Shape{

**double** area;

**abstract** **double** getArea();

}

**class** Triangle **extends** Shape{

**double** width,length;

Triangle(**double** w ,**double** l){

width=w;

length=l;

}

**public** **double** getArea(){

**return** width\*length;

}

}

**class** Rectangle **extends** Shape{

**double** length1,length2,length3;

Rectangle(**double** l1,**double** l2,**double** l3){

length1=l1;

length2=l2;

length3=l3;

}

**public** **double** getArea(){

**double** p=0.5\*(length1+length2+length3);

**double** area\_2=p\*(p-length1)\*(p-length2)\*(p-length3);

**return** java.lang.Math.*sqrt*(area\_2);

}

}

**class** Pillar{

**int** n;

**double** highth;

Shape shape;

**double** volume;

Pillar(**int** n,**double** highth){

**this**.n=n;

**this**.highth=highth;

}

**double** getVolume(){

**if**(n==3){

System.***out***.println("请输入底面三角形的三边长：");

**double** l1=**new** Scanner(System.***in***).nextDouble();

**double** l2=**new** Scanner(System.***in***).nextDouble();

**double** l3=**new** Scanner(System.***in***).nextDouble();

shape=**new** Rectangle(l1,l2,l3);

shape.area=shape.getArea();

**return** shape.area\*highth;

}**else**{

System.***out***.println("请输入底面矩形的长和宽：");

**double** l1=**new** Scanner(System.***in***).nextDouble();

**double** l2=**new** Scanner(System.***in***).nextDouble();

shape=**new** Triangle(l1,l2);

shape.area=shape.getArea();

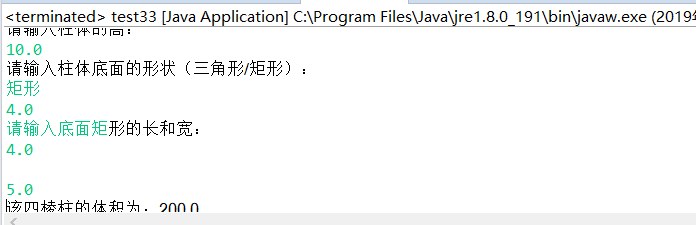
**return** shape.area\*highth;

}

}

}

实验结果如下：



实验内容4代码如下：

**import** java.util.Scanner;

**public** **class** test34 {

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

System.***out***.println("开始录入成绩信息：");

Grade grade=**new** Grade();

grade.begin();

System.***out***.println("录入完毕\n录入的成绩信息如下：\n课程名称："+grade.course.course+" 学生："+grade.student.name+" 任课教师："+grade.teacher.name+" 成绩："+grade.grade);

}

}

**class** Grade{

**double** grade;

Student student=**new** Student();

Teacher teacher=**new** Teacher();

Course course=**new** Course();

**void** begin(){

System.***out***.println("请输入课程成绩：");

grade=**new** Scanner(System.***in***).nextDouble();

student.setDate(teacher,course);

teacher.setDate(student,course);

course.setDate(student,teacher);

}

}

**class** Student{

Teacher teacher;

String name;

Course course;

**void** setDate(Teacher t,Course course){

teacher=t;

**this**.course=course;

System.***out***.println("请输入学生姓名：");

name=**new** Scanner(System.***in***).nextLine();

}

}

**class** Teacher{

Student student;

String name;

Course course;

**void** setDate(Student stu,Course cour){

student=stu;

course=cour;

System.***out***.println("请输入教师姓名：");

name=**new** Scanner(System.***in***).nextLine();

}

}

**class** Course{

String course;

Teacher teacher;

Student student;

**void** setDate(Student stu,Teacher tea){

student=stu;

teacher=tea;

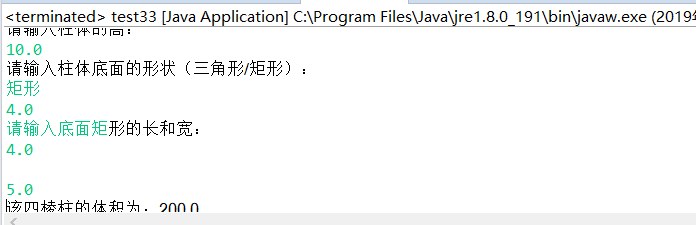
System.***out***.println("请输入课程名称:");

course=**new** Scanner(System.***in***).nextLine();

}

}

实验结果如下：



1. **总结及心得体会：**

该实验是Java编程方法实现实验

第五个通过对问题的需求模型创建类、对象及它们之间关系，设计与实现其相关类、对象等的代码编写，实现实际问题的功能性。

1. **对本实验过程及方法、手段的改进建议：**

无。

**报告评分：**

**指导教师签字：**