电子科技大学信息与软件工程学院

**实 验 报 告**

学 号 2017221132009

姓 名 陆圣珩

（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：陆圣珩 学号：2017221302009 指导教师：周帆**

**实验地点：信软实验楼303 实验时间：2018.12.9**

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：**

Java的类、对象的基本操作，类的构造方法与作用，类的声明和各部分的声明、作用和编写方法，实例化对象和成员访问方法。

1. **实验内容：**
2. 完成第五章习题9、10编程。
3. 完成第六章习题7、8编程。
4. 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。
5. 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。
6. **实验器材（设备、元器件）：**

PC机

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）**

实验一

代码：

|  |
| --- |
| class Person{  String name;  char sex;  int age;  Person(String name,char sex,int age){  this.name = name;  this.sex = sex;  this.age = age;  }  void setDate(String name,char sex,int age) {  this.name = name;  this.sex = sex;  this.age = age;  }  String getDate() {  String date = " name: " + name + " sex: " + sex + " age: " + age;  return date;  } } class Student extends Person{  long sID;int classNO;  Student(String name,char sex,int age,long sID,int classNO){  super(name,sex,age);  this.sID = sID;  this.classNO = classNO;  }  void setDate(String name,char sex,int age,long sID,int classNO){  this.name = name;  this.sex = sex;  this.age = age;  this.sID = sID;  this.classNO = classNO;  }  String getDate() {  String date = " name: " + name + " sex: " + sex + " age: " + age;  date += '\n' + " sID: " + sID +" classNO: " + classNO;  return date;  } }  public class Test {  public static void main(String []args) {  Student people = new Student("lsh",'m',19,1302009,1302);  String date = people.getDate();  System.*out*.print(date);  } } |

测试结果：

|  |
| --- |
|  |

代码：

|  |
| --- |
| abstract class Person{  String name;  char sex;  int age;  abstract void setDate(String name,char sex,int age);  abstract String getDetail(); }  class Student extends Person{  int sID;  String speciality;  Student(String name,char sex,int age,int sID,String speciality){  setDate(name,sex,age);  this.sID = sID;  this.speciality = speciality;  }  void setDate(String name,char sex,int age) {  this.name = name;  this.sex = sex;  this.age = age;  }  String getDetail() {  String date = " name: " + name + " sex: " + sex + " age: " + age;  date += '\n' + " sID: " + sID +" speciality: " + speciality;  return date;  }  }  class Teacher extends Person{  int tID;  String department;  Teacher(String name,char sex,int age,int tID,String department){  setDate(name,sex,age);  this.tID = tID;  this.department = department;  }  void setDate(String name,char sex,int age) {  this.name = name;  this.sex = sex;  this.age = age;  }  String getDetail() {  String date = " name: " + name + " sex: " + sex + " age: " + age;  date += '\n' + " tID: " + tID +" department: " + department;  return date;  }  }  public class Test {  public static void main(String []args) {  Student people = new Student("lsh",'m',19,130209,"software");  String date = people.getDetail();  System.*out*.print(date);  } } |

测试结果：

|  |
| --- |
|  |

实验二

代码：

|  |
| --- |
| package com.allenx555.test;  public interface Print {  void print(); } |
| package com.allenx555.test;  class PrintTest implements Print{  public void print() {  System.*out*.println("test1");  } }  class PrintEnd implements Print{  public void print() {  System.*out*.println("test2");  } } public class Test {  public static void main(String []args) {  PrintTest start = new PrintTest();  PrintEnd end = new PrintEnd();  start.print();  end.print();  } } |

测试结果：

|  |
| --- |
|  |

代码：

|  |
| --- |
| package com.allenx555.test;  public interface Person {  void setDate(String name, char sex, long birthday);  String getData(); } |
| package com.allenx555.test;  class Student implements Person{  String name; char sex; long birthday;  long sID; String speciality;  void init(long sID, String speciality){  this.sID = sID;  this.speciality = speciality;  }  String getInit() {  return String.*valueOf*(sID)+' '+speciality;  }  public void setDate(String name, char sex, long birthday) {  this.name = name;  this.sex = sex;  this.birthday = birthday;  }  public String getData() {  return name+' '+sex+' '+birthday;  } } |

测试结果：

|  |
| --- |
|  |

实验三

代码：

|  |
| --- |
| abstract class Shape{  int n;  Shape(int n){  this.n= n;  }  abstract double getArea(); }  class Triangle extends Shape{  double a, b, c;  Triangle(double a,double b,double c){  super(3);  this.a = a;  this.b = b;  this.c = c;  }  double getArea() {  return (1.0/4.0)\*Math.*sqrt*((a+b+c)\*(a+b-c)\*(a+c-b)\*(b+c-a));  } }  class Rectangle extends Shape{  double a, b;  Rectangle(double a, double b){  super(4);  this.a = a;  this.b = b;  }  double getArea() {  return a\*b;  } }  class Pillar{  int n;  double a, b, c;  double s, h;  Pillar(double a, double b, double c, double h){  this.n = 3;  this.a = a;  this.b = b;  this.c = c;  this.h = h;  }  Pillar(double a, double b, double h){  this.n = 4;  this.a = a;  this.b = b;  this.h = h;  }  double getVolume() {  if(n==3) {  Triangle pillar = new Triangle(a, b, c);  s = pillar.getArea();  }  if(n==4) {  Rectangle pillar = new Rectangle(a, b);  s = pillar.getArea();  }  return s\*h;  } }  public class Test {  public static void main(String []args) {  Pillar pillar = new Pillar(3,4,5);  System.*out*.println("n is :"+String.*valueOf*(pillar.n));  System.*out*.println("a is :"+String.*valueOf*(pillar.a));  System.*out*.println("b is :"+String.*valueOf*(pillar.b));  System.*out*.println("Height is :"+String.*valueOf*(pillar.h));  System.*out*.println("Volume is :"+String.*valueOf*(pillar.getVolume()));  } } |

测试结果：

|  |
| --- |
|  |

实验四

代码：

|  |
| --- |
| class Student{  int studentNumber;  Student(int n){  studentNumber = n;  } }  class Teacher{  int teacherNumber;  Teacher(int n){  teacherNumber = n;  } }  class Course{  int courseNumber;  String[] name;  Course(int n){  courseNumber = n;  } }  class Grade{  int n;  int studentNumber;  int teacherNumber;  int courseNumber;  Grade(int n,int a,int b,int c){  this.n = n;  Student gradeStudent = new Student(a);  this.studentNumber = gradeStudent.studentNumber;  Teacher gradeTeacher = new Teacher(b);  this.teacherNumber = gradeTeacher.teacherNumber;  Course gradeCourse = new Course(c);  this.courseNumber = gradeCourse.courseNumber;  } }  public class Test{  public static void main(String []args) {  Grade grade1 = new Grade(100,1,2,3);  System.*out*.println("grade1.studentNumber:\t"+grade1.studentNumber);  System.*out*.println("grade1.teacherNumber:\t"+grade1.teacherNumber);  System.*out*.println("grade1.courseNumber:\t"+grade1.courseNumber);  System.*out*.println("grade1.grade:\t"+grade1.n);  } } |

测试结果：

|  |
| --- |
|  |

**九．总结及心得体会：**

相对于前两次实验，本次实验更加贴近实用性而非单纯Java特性语言。通过本次实验可以了解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。在本次实验结束之后可以对于Java有了一个基础的大致的了解，并可以上手一些小的项目进行开发。

**十．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：**