电子科技大学信息与软件工程学院

**实 验 报 告**

学 号 张超魁

姓 名 2017221103017

（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：张超魁 学号：2017221103017 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：JDK使用及Java基础**
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. 类的继承：

类继承另一个类，这个类除了创建自己的成员外，还能够继承或扩展另一个类的成员。运用继承，你能够创建一个通用类，它定义了一系列相关项目的一般特性。该类可以被更具体的类继承，每个具体的类都增加一些自己特有的东西。被继承的类叫超类（superclass），继承超类的类叫子类（subclass）。

子类格式：

[abstract|final] class SubCls extends SupCls{

SubClassBody

}

abstract是可选项，修饰的类叫抽象类，指示其对象引用的必须是其子类实例；

final是可选项，修饰的类叫终结类，指示其不能被继承，不能有子类；

SubCls是创建的类，称为子类；

extends是关键字，指示两个类存在的继承关系；

SuperCls是SubCls类的超类。

在创建类时，若缺省extends，则该类就为Object类的直接子类。Object类是Java语言中所有类的直接或间接超类。Object类存放在java.lang包中。

子类继承了超类的所有成员，但是无法访问某些受限的成员（private修饰）。

2. 方法覆盖与运行时的多态

在子类的创建中，具有与超类中有相同的方法名、相同的参数以及相同的返回数据类型。

超类的对象可以引用子类的实例（向上转型），该对象仍然只能够调用超类中已定义的方法和变量。

3. final和abstract类和方法

由final修饰的类称终结类，不能被继承。由于安全性的原因或者是面向对象设计的考虑，限定一些类不能被继承。final类不能被继承，保证了该类的唯一性。对于一个类的定义已经很完善，不需再创建它的子类，也可以将其修饰为final类。

当一个类的定义完全表示抽象概念时，它不能够被实例化为一个对象。抽象类本身存在未实现的方法(abstract方法)，这些方法不具备实际功能；抽象类只能衍生出子类，抽象方法则由衍生子类时所覆盖。abstract方法必须是在abstract类中，但是abstract类中也可以有非abstract方法。

4. 成员访问控制：如表1所示：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **修饰符** | **public** | **protect** | **无修饰符** | **private** |
| 类 | Y | Y | Y | Y |
| 包 | Y | Y | Y | N |
| 子类 | Y | Y | N | N |
| 所有环境 | Y | N | N | N |

表1 访问控制权限表

5. 接口：

Java是通过接口使得处于不同类层次，甚至互不相关的类可以具有相同的行为。接口是方法定义（没有实现，Java8之后可以有static方法和default 方法）、常量的集合。用接口，你可以指定一个类必须做什么，而不是规定它如何去做。在类层次的任何地方都可以使用接口定义一个行为的协议实现它。

接口具有成员变量和成员方法。但是接口中的所有方法都是abstract方法，这些方法是没有语句的。（Java8之后可以有static方法和default方法）

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）：**

环境：Windows 10、JDK 1.8.0\_191、

IDE：IntelliJ IDEA 2018.3.2

1. **实验步骤：**

1. 明确项目要求

2. 编写代码

3. 编译代码

4. 测试程序

5. 根据测试结果对程序进行调试改进

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）**

1. 第五章习题9：

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** setData(String name, **char** sex, **int** age) {  
 **this**.**name** = name;  
 **this**.**sex** = sex;  
 **this**.**age** = age;  
 }  
  
 String getData() { **return "Name: "** + **name** + **" Sex: "** + **sex** + **" Age: "** + **age**; }  
}  
  
**class** Student **extends** Person {  
 **private int sID**;  
 **private int classNo**;  
  
 Student(String name, **char** sex, **int** age, **int** sID, **int** classNo) {  
 **super**(name, sex, age);  
 **this**.**sID** = sID;  
 **this**.**classNo** = classNo;  
 }  
  
 **void** setData(String name, **char** sex, **int** age, **int** sID, **int** classNo) {  
 **super**.setData(name, sex, age);  
 **this**.**sID** = sID;  
 **this**.**classNo** = classNo;  
 }  
  
 String getData() {  
 **return "Name: "** + **super**.**name** + **" Sex: "** + **super**.**sex** + **" Age: "** + **super**.**age** + **" SID: "** + **sID** + **" ClassNo: "** + **classNo**;  
 }  
}  
  
**public class** C5Q9 {  
 **public static void** main(String[] agrs) {  
 Person person1 = **new** Person(**"aaa"**, **'f'**, 26);  
 person1.setData(**"p1"**, **'f'**, 50);  
 System.***out***.println(person1.getData());  
  
 Student student1 = **new** Student(**"Tadokoro"**, **'m'**, 24, 114514, 1919810);  
 System.***out***.println(student1.getData());  
 student1.setData(**"koji"**, **'f'**, 24, 889464, 364364);  
 System.***out***.println(student1.getData());  
 }  
}

实验结果如图1所示：

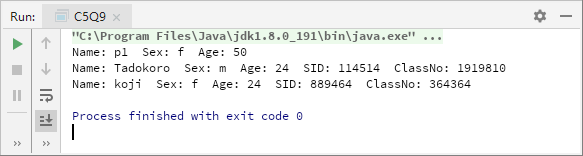


图1 第五章习题9实验结果

2. 第5章习题10：

**abstract class** Person {  
 String **name**;  
 **char sex**;  
 **int age**;  
  
 **void** setData(String name, **char** sex, **int** age) {  
 **this**.**name** = name;  
 **this**.**sex** = sex;  
 **this**.**age** = age;  
 }  
  
 **abstract** String getDetail();  
}  
  
**class** Student **extends** Person {  
 **private int sID**;  
 **private** String **speciality**;  
  
 Student(String name, **char** sex, **int** age, **int** sID, String speciality) {  
 **super**.setData(name, sex, age);  
 **this**.**sID** = sID;  
 **this**.**speciality** = speciality;  
 }  
  
 **void** setData(String name, **char** sex, **int** age, **int** sID, String speciality) {  
 **super**.setData(name, sex, age);  
 **this**.**sID** = sID;  
 **this**.**speciality** = speciality;  
 }  
  
 @Override  
 String getDetail() {  
 **return "Name: "** + **super**.**name** + **" Sex: "** + **super**.**sex** + **" Age: "** + **super**.**age** + **" SID: "** + **sID** + **" Speciality: "** + **speciality**;  
 }  
}  
  
**class** Teacher **extends** Person {  
 **private int tID**;  
 **private** String **department**;  
  
 Teacher(String name, **char** sex, **int** age, **int** tID, String department) {  
 **super**.setData(name, sex, age);  
 **this**.**tID** = tID;  
 **this**.**department** = department;  
 }  
  
 **void** setData(String name, **char** sex, **int** age, **int** tID, String department) {  
 **super**.setData(name, sex, age);  
 **this**.**tID** = tID;  
 **this**.**department** = department;  
 }  
  
 @Override  
 String getDetail() {  
 **return "Name: "** + **super**.**name** + **" Sex: "** + **super**.**sex** + **" Age: "** + **super**.**age** + **" TID: "** + **tID** + **" Department: "** + **department**;  
 }  
}  
  
**public class** C5Q10 {  
 **public static void** main(String[] args) {  
 Person student1 = **new** Student(**"Tadokoro"**, **'m'**, 24, 114514, **"Swimming"**);  
 System.***out***.println(student1.getDetail());  
 ((Student) student1).setData(**"koji"**, **'f'**, 24, 1919810, **"karate"**);  
 System.***out***.println(student1.getDetail());  
 }  
}

实验结果如图2所示：

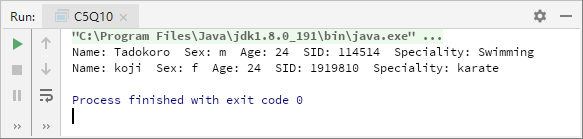


图2 第5章习题10实验结果

3. 第六章习题7：

**interface** Print {  
 **void** print();  
}  
  
**class** PrintInt **implements** Print {  
 **private int toPrint**;  
  
 PrintInt(**int** i) {  
 **toPrint** = i;  
 }  
  
 **public void** print() {  
 System.***out***.println(**toPrint**);  
 }  
}  
  
**class** PrintString **implements** Print {  
 **private** String **toPrint**;  
  
 PrintString(String str) {  
 **toPrint** = str;  
 }  
  
 **public void** print() {  
 System.***out***.println(**toPrint**);  
 }  
}  
  
**public class** C6Q7 {  
 **public static void** main(String[] args) {  
 PrintInt i = **new** PrintInt(123);  
 PrintString str = **new** PrintString(**"abc"**);  
  
 i.print();  
 str.print();  
 }  
}

实验结果如图3所示：

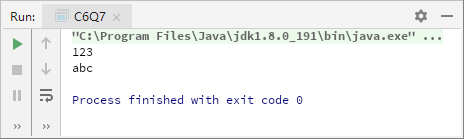


图3 第六章习题7实验结果

4. 第六章习题8：

**interface** Person {  
 **void** setData(String name, **char** sex, String birthday);  
  
 **void** getData();  
}  
  
**class** Student **implements** Person {  
 **private** String **name**;  
 **private char sex**;  
 **private** String **birthday**;  
 **private int sID**;  
 **private** String **speciality**;  
  
 **public void** setData(String name, **char** sex, String birthday) {  
 **this**.**name** = name;  
 **this**.**sex** = sex;  
 **this**.**birthday** = birthday;  
 }  
  
 **void** setData(String name, **char** sex, String birthday, **int** sID, String speciality) {  
 **this**.**name** = name;  
 **this**.**sex** = sex;  
 **this**.**birthday** = birthday;  
 **this**.**sID** = sID;  
 **this**.**speciality** = speciality;  
 }  
  
 **public void** getData() {  
 System.***out***.println(**"Name: "** + **name** + **" Sex: "** + **sex** + **" Brithday: "** + **birthday** + **" sID: "** + **sID** + **" Speciality: "** + **speciality**);  
 }  
}  
  
**public class** C6Q8 {  
 **public static void** main(String[] args) {  
 Student stu = **new** Student();  
 stu.setData(**"bbb"**, **'m'**, **"2000-2-3"**, 114514, **"ccc"**);  
 stu.getData();  
 }  
}

实验结果如图4所示：

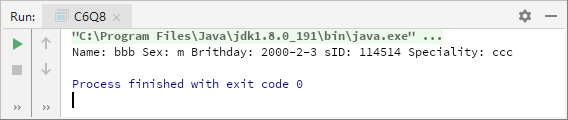


图4 第六章习题8实验结果

5. 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。

**abstract class** Shape {  
 **public abstract double** area();  
}  
  
**class** Triangle **extends** Shape {  
 **private double a**, **h**;  
  
 Triangle(**double** a, **double** h) {  
 **this**.**a** = a;  
 **this**.**h** = h;  
 }  
  
 **public double** area() { **return a** \* **h** / 2; }  
}  
  
**class** Rectangle **extends** Shape {  
 **private double a**, **b**;  
  
 Rectangle(**double** a, **double** b) {  
 **this**.**a** = a;  
 **this**.**b** = b;  
 }  
  
 **public double** area() { **return a** \* **b**; }  
}  
  
**class** Pillar {  
 **private double h**;  
 **private** Shape **down**;  
  
 Pillar(Shape down, **double** h) {  
 **this**.**down** = down;  
 **this**.**h** = h;  
 }  
 Pillar(Triangle down, **double** h) {  
 **this**.**down** = down;  
 **this**.**h** = h;  
 }  
 Pillar(Rectangle down, **double** h) {  
 **this**.**down** = down;  
 **this**.**h** = h;  
 }  
  
 **double** vec() { **return down**.area() \* **h**; }  
}  
  
**public class** PillarTest {  
 **public static void** main(String[] args) {  
 Shape t = **new** Triangle(3, 4);  
 Rectangle d = **new** Rectangle(2, 2);  
 Pillar v1 = **new** Pillar(t, 3);  
 Pillar v2 = **new** Pillar(d, 5);  
 System.***out***.println(v1.vec());  
 System.***out***.println(v2.vec());  
 }  
}

实验结果如图5所示：

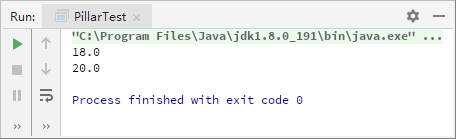


图5 题目5实验结果

6. 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。

**package** Q4;  
  
**class** Person {  
 **private** String **name**;  
 **private char sex**;  
 **private int age**;  
  
 **void** setData(String name, **char** sex, **int** age) {  
 **this**.**name** = name;  
 **this**.**sex** = sex;  
 **this**.**age** = age;  
 }  
 String getData() { **return "name: "** + **name** + **" sex: "** + **sex** + **" age: "** + **age**; }  
}  
  
**class** Student **extends** Person {  
 **private** String **sID**;  
  
 **void** setData(String id, String n, **char** s, **int** a) {  
 **super**.setData(n, s, a);  
 **sID** = id;  
 }  
 String getData() { **return "sID: "** + **sID** + **" "** + **super**.getData(); }  
}  
  
**class** Teacher **extends** Person {  
 **private** String **tID**;  
  
 **void** setData(String id, String n, **char** s, **int** a) {  
 **super**.setData(n, s, a);  
 **tID** = id;  
 }  
 String getData() { **return "tID: "** + **tID** + **" "** + **super**.getData(); }  
}  
  
**class** Course {  
 **private** String **cID**;  
 **private** String **name**;  
  
 **void** setData(String c, String n) {  
 **this**.**cID** = c;  
 **this**.**name** = n;  
 }  
 String getData() { **return "cID: "** + **cID** + **" name: "** + **name**; }  
}  
  
**class** Grade {  
 **private int score**;  
 Student **stu** = **new** Student();  
 Teacher **tea** = **new** Teacher();  
 Course **cou** = **new** Course();  
  
 **void** setData(**int** s) { **score** = s; }  
 String getScore() { **return "score: "** + **score**; }  
}  
  
**public class** GradeTest {  
 **public static void** main(String[] args) {  
 Grade gra = **new** Grade();  
 gra.**stu**.setData(**"01"**, **"Tom"**, **'m'**, 21);  
 gra.**tea**.setData(**"02"**, **"Divid"**, **'m'**, 35);  
 gra.**cou**.setData(**"12"**, **"Java"**);  
 gra.setData(85);  
 System.***out***.println(gra.**stu**.getData() + **"\n\t"** + gra.**tea**.getData() + **"\n\t"** + gra.**cou**.getData() + **" "** + gra.getScore());  
 }  
}

实验结果如图6所示：

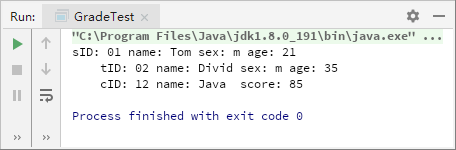


图6 习题6实验结果

1. **总结及心得体会：**

本实验引入了类的继承与接口的实现，深入了“面向对象”的概念。子类的成员隐藏和覆盖超类中相同的成员、超类的对象可以对子类的实例引用都体现了面向对象的多态性，final以及访问权限控制都体现了封装性，继承性自然不用说了。

子类通过对父类中方法与成员变量的继承，大大缩减了编码量。而父类的方法并没有限制子类的多样性，通过过载的方法可以对原有方法进行修改或补充，使类之间的关系更加明确。

Java语言中不允许多重继承，但是可以实现多个接口。但接口不能实现方法，只能由类来实现。

本次实验的类以及变量、方法比较多，为了避免对象对某些类的成员访问时带来危害，必须尽可能严格限制类以及变量、方法的访问权限。

1. **对本实验过程及方法、手段的改进建议：**

本次实验的类以及变量、方法众多，因此保持良好的命名以及编码习惯非常重要。

在实验中可以尽可能多得尝试问题的不同解决方法，如体会Java的多态性，使用超类或接口来对子类的实例引用等。

**报告评分：**

**指导教师签字：**