

电子科技大学信息与软件工程学院

# 实 验 报 告

(三)

学 号 2018091618008

姓 名 袁昊男

(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：袁昊男      学号：2018091618008      指导教师：周帆

实验地点：信软楼 304

实验时间：2019. 11. 24、12. 01

**一、 实验名称：类的继承、接口及访问控制**

**二、 实验学时：2 学时**

**三、 实验目的：**

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

**四、 实验原理：**

1、面向对象

Java 语言具有真正的面向对象语言的特点，除了数值、布尔和字符三种基本的数据类型外，其它类型都是对象。它支持封装、多态性和继承。封装就是将对象内的数据和代码联编起来，形成一个对象；多态性是指一个接口，有多个内在实现形式表示；继承是指某一对象直接使用另一对象的所有属性和方法的过程。

对程序员来说，这意味着要注意应中的数据 and 操纵数据的方法（method），而不是严格地用过程来思考。在一个面向对象的系统中，类（class）是数据和操作数据的方法的集合。数据和方法一起描述对象（object）的状态和行为。每一对象是其状态和行为的封装。类是按一定体系和层次安排的，使得子类可以从超类继承行为。在这个类层次体系中有一个根类，它是具有一般行为的类。Java 程序是用类来组织的。

2、编译和解释性

Java 编译程序生成字节码（byte-code），而不是通常的机器码。Java 字节码提供对体系结构中性的目标文件格式，代码设计成可有效地传送程序到多个平台。Java 程序可以在任何实现了 Java 解释程序和运行系统（run-time system）的系统上运行。

在一个解释性的环境中，程序开发的标准“链接”阶段大大消失了。如果说 Java 还有一个链接阶段，它只是把新类装进环境的过程，

它是增量式的、轻量级的过程。因此, Java 支持快速原型和容易试验, 它将导致快速程序开发。这是一个与传统的、耗时的“编译、链接和测试”形成鲜明对比的精巧的开发过程。

## 五、 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。

## 六、 实验器材 (设备、元器件):

配置了 JDK 环境、安装有 Eclipse 软件的个人电脑一台。

## 七、 实验步骤:

- 1、按照题目要求分析需求、功能;
- 2、设计算法, 编写程序并进行测试;
- 3、结果分析, 撰写实验报告。

## 八、 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:

有两个类 `Person` 和 `Student`, 它们存在继承关系。在 `Person` 中, 有成员变量 `name`, `sex`, `age` 三个, 它们类型分别为 `String`、`char`、`int`, 有一个带参数的构造方法 `Person(String, char, int)`, 用来对成员变量初始化。有成员方法 `setData(String, char, int)` 设置成员变量 `name`, `sex`, `age` 的值。`getData()` 是不带参数且返回值由 `name`、`sex` 和 `age` 值构成的字符串的成员方法。

`Student` 类是 `Person` 的子类, 在 `Student` 中, 有 `int` 类型的 `sID` 和 `classNo` 成员变量用来表示学生的学号和班级号。它带有五个参数的方法 `setData()` 和不带参数的方法 `getData()`, `setData()` 方法设置成员变量的值, 而 `getData()` 则是返回五个成员变量的值组成的字符串。

(1) 代码

```

1. package pkg;
2.
3. class Person {
4.     public String name;
5.     public char sex;
6.     public int age;
7.
8.     public Per-
son(String name, char sex, int age) {
9.         this.name = name;
10.        this.sex = sex;
11.        this.age = age;
12.    }
13.    public void set-
Data(String name, char sex, int age) {
14.        this.name = name;
15.        this.sex = sex;
16.        this.age = age;
17.    }
18.
19.    public Person getData() {
20.        return this;
21.    }
22.
23.    public String toString() {
24.        re-
turn "Name: " + name + ", Sex: " + sex + ", Age: " +
age;
25.    }
26. }
27.
28. class Student extends Person {
29.     public Student() {
30.         super("Name", 'M', 0); // !!!!
31.     }
32.
33.     public int sID;
34.     public int classNo;
35.
36.     public void set-
Data(String name, char sex, int age, int sID, int cl
assNo) {
37.         setData(name, sex, age);
38.         this.sID = sID;
39.         this.classNo = classNo;
40.     }
41.
42.     public Student getData() {
43.         return this;
44.     }
45.
46.     public String toString() {
47.         re-
turn "Name: " + name + ", Sex: " + sex + ", Age: " +
age + ", ID: " + sID + ", Class: " + classNo;
48.     }
49. }
50.

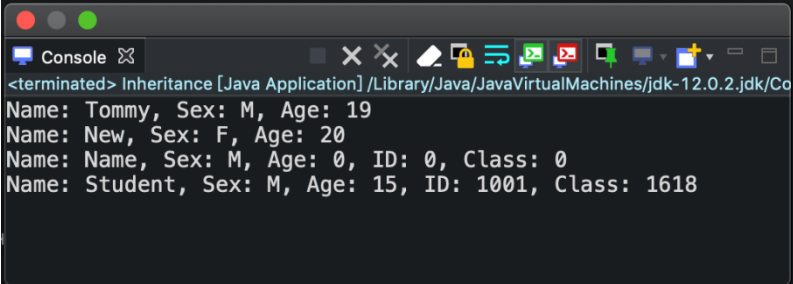
```

```

51. class Teacher extends Person {
52.     public Teacher() {
53.         super("Name", 'M', 0); // !!!
54.     }
55.     public int tID;
56.     public String department = "NULL";
57.
58.     public void setData(String name, char sex, int age, int tID, String
        department) {
59.         setData(name, sex, age);
60.         this.tID = tID;
61.         this.department = department;
62.     }
63. }
64. public class Inheritance {
65.     public static void main(String[] args) {
66.         Person person = new Per-
        son("Tommy", 'M', 19);
67.         System.out.println(person);
68.         person.setData("New", 'F', 20);
69.         System.out.println(person);
70.         Student student = new Student();
71.         System.out.println(student);
72.         student.setData("Stu-
        dent", 'M', 15, 1001, 1618);
73.         System.out.println(student);
74.     }
75.
76. }

```

(2) 运行截图



```

<terminated> Inheritance [Java Application] /Library/Java/JavaVirtualMachines/jdk-12.0.2.jdk/Co
Name: Tommy, Sex: M, Age: 19
Name: New, Sex: F, Age: 20
Name: Name, Sex: M, Age: 0, ID: 0, Class: 0
Name: Student, Sex: M, Age: 15, ID: 1001, Class: 1618

```

## 2、第五章习题 10:

抽象类 Person 如下定义:

```

Abstract class Person {
    String name;
    char sex;
    int age;
    abstract void setData(String name, char sex, int age);
    abstract String getDetail();
}

```

```
}
```

类 Student 和类 Teacher 均是 abstract 类 Person 的子类。类 Student 有成员变量 name, sex, age, sID, speciality, 其中 sID 表示学生的学号, speciality 表示学生专业。类 Teacher 有成员变量 name, sex, age, tID, department, 其中 tID 表示教师的编号, department 表示教师所在的部门。请编写出类 Student 和类 Teacher 所需要的最基本构成。

(1) 代码

```
1. package pkg;
2.
3. public class NewPerson {
4.     String nameString;
5.     char sex;
6.     int age;
7.     abstract void setData(String name, char sex, int age);
8.     abstract String getDetail();
9. }
10.
11. class Student extends NewPerson {
12.     int sID;
13.     String speciality;
14. }
15.
16. class Teacher extends NewPerson {
17.     int tID;
18.     String departmentString;
19. }
```

(2) 运行截图

无可运行程序实例。

### 3、第六章习题 7:

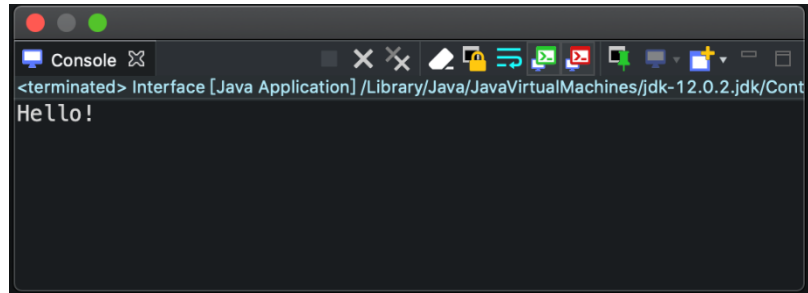
创建一个接口 Print, 在其中定义一个打印方法 Print()。再创建两个类实现这个接口。

(1) 代码

```
1. package pkg;
2.
3. interface Print {
4.     void print();
5. }
6.
7. public class Interface implements Print {
8.     public void print() {
9.         System.out.println("Hello!");
10.    }
11.
12.    public static void main(String[] args) {
13.        Interface test = new Interface();
14.        test.print();
15.    }
16. }
```

```
15.    }
16. }
```

## (2) 运行截图



## 4、第六章习题 8:

创建 Person 接口（即“人”），它有 setData()和 getData()方法对“人”属性 name、sex 和 birthday 赋值和获得这些属性组成的字符串信息。创建类 Student 实现 Person 接口，并对自己的“学生”属性的成员变量 sID、speciality 设置值和获得它们值所组成的字符串信息。

### (1) 代码

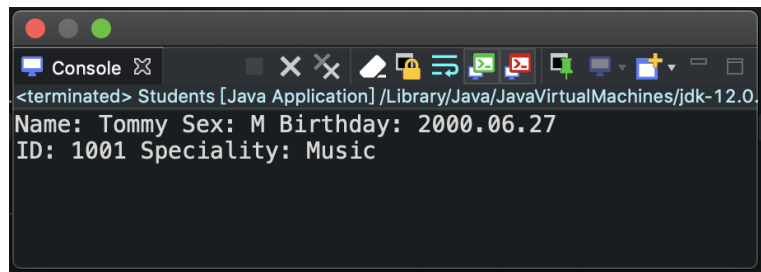
```
1. package pkg;
2.
3. interface Person1 {
4.     public void set-
       Data(String name, char sex, String birthday);
5.
6.     public String getData();
7. }
8.
9. public class Students implements Person1 {
10.     private String name;
11.     private char sex;
12.     private String birthday;
13.     private int sID;
14.     private String speciality;
15.
16.     public void set-
       Data(String name, char sex, String birthday) {
17.         this.name = name;
18.         this.sex = sex;
19.         this.birthday = birthday;
20.     }
21.
22.     public String getData() {
23.         re-
       turn "Name: " + name + " Sex: " + sex + " Birth-
       day: " + birthday + "\n" + "ID: " + sID + " Special-
       ity: "
24.             + speciality;
25.     }
26. }
```

```

27.     public void set-
        Data(String name, char sex, String birth-
            day, int sID, String speciality) {
28.         this.name = name;
29.         this.sex = sex;
30.         this.birthday = birthday;
31.         this.sID = sID;
32.         this.speciality = speciality;
33.     }
34.
35.     public static void main(String[] args) {
36.         Students person = new Students();
37.         person.set-
            Data("Tommy", 'M', "2000.06.27", 1001, "Music");
38.         System.out.println(person.getData());
39.     }
40. }

```

## (2) 运行截图



```

<terminated> Students [Java Application] /Library/Java/JavaVirtualMachines/jdk-12.0.
Name: Tommy Sex: M Birthday: 2000.06.27
ID: 1001 Speciality: Music

```

- 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。

## (1) 代码

```

1. package pkg2;
2.
3. public class PillarTest {
4.     public static void main(String[] args) {
5.         Pillar pillar1 = new Pil-
            lar(3, 3.5, 4.5, 3.3);
6.         Pillar pillar2 = new Pil-
            lar(4, 3.5, 4.5, 3.3);
7.         System.out.println("V-cuboid = " + pil-
            lar1.getV());
8.         System.out.println("V-triangu-
            lar prism = " + pillar2.getV());
9.     }
10. }
11.
12. class Pillar {
13.     int n;
14.     double a, b, h;
15.

```

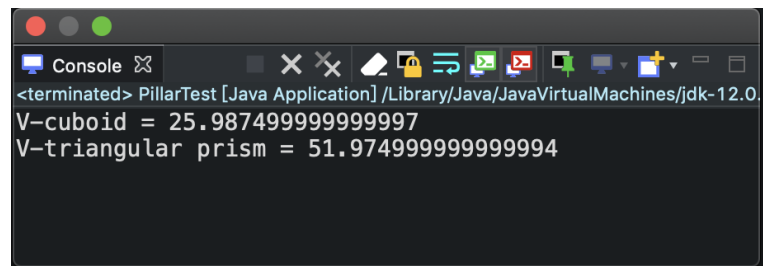


```

16.     public Pillar(int n, double a, double b, dou-
17.         ble h) {
18.         this.n = n;
19.         this.a = a;
20.         this.b = b;
21.         this.h = h;
22.     }
23.     public double getV() {
24.         if (n == 3) {
25.             Triangle tri = new Triangle(n, a, b);
26.             return h * tri.getArea();
27.         } else if (n == 4) {
28.             Rectangle rec = new Rectan-
29.                 gle(n, a, b);
30.             return h * rec.getArea();
31.         }
32.         return 0;
33.     }

```

## (2) 运行截图



```

<terminated> PillarTest [Java Application] /Library/Java/JavaVirtualMachines/jdk-12.0.
V-cuboid = 25.987499999999997
V-triangular prism = 51.974999999999994

```

- 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。

## (1) 代码

```

1. package pkg2;
2.
3. public class GradeTest {
4.     public static void main(String[] args) {
5.         Grade grade = new Grade();
6.         Student student = new Student(2018001, "张三", "男", "20000627", "软件学院", "软工");
7.         Teacher teacher = new Teacher(1001, "李老师", "软件", "网安");
8.         Course course = new Course(888001, "Java", 2, 48);
9.         grade.stu = student;
10.        grade.tea = teacher;
11.        grade.cour = course;
12.        grade.setNo(grade.stu.getNo());
13.        grade.setStudentName(grade.stu.getName());
14.        grade.setCourseName(grade.cour.getName());
15.        grade.setTeacherName(grade.tea.getName());

```

```

16.         grade.setPoint(100);
17.         grade.setGNo(66660001);
18.         System.out.print(grade.toString());
19.     }
20. }
21.
22. class Student {
23.     private int sNo;
24.     private String sName;
25.     private String sSex;
26.     private String sBirth; // YYYY.MM.DD
27.     private String sSchool;
28.     private String sMajor;
29.     public Student() {
30.     }
31.     public Stu-
dent(int sNo, String sName, String sSex, String sBir
th, String sSchool, String sMajor) {
32.         super();
33.         this.sNo = sNo;
34.         this.sName = sName;
35.         this.sSex = sSex;
36.         this.sBirth = sBirth;
37.         this.sSchool = sSchool;
38.         this.sMajor = sMajor;
39.     }
40.     public void setNo(int no) {
41.         this.sNo = no;
42.     }
43.     public int getNo() {
44.         return sNo;
45.     }
46.     public void setName(String name) {
47.         this.sName = name;
48.     }
49.     public String getName() {
50.         return sName;
51.     }
52.     public void setSex(String sex) {
53.         this.sSex = sex;
54.     }
55.     public String getSex() {
56.         return sSex;
57.     }
58.     public void setBirth(String birth) {
59.         this.sBirth = birth;
60.     }
61.     public String getBirth() {
62.         return sBirth;
63.     }
64.     public void setSchool(String school) {
65.         this.sSchool = school;
66.     }
67.     public String getSchool() {
68.         return sSchool;
69.     }
70.     public void setMajor(String major) {
71.         this.sMajor = major;

```

```

72.     }
73.     public String getMajor() {
74.         return sMajor;
75.     }
76.     public String toString() {
77.         re-
turn sNo + "\t" + sName + "\t" + sSex + "\t" + sBirt
h + "\t" + sSchool + "\t" + sMajor;
78.     }
79. }
80.
81. class Teacher {
82.     private int tNo;
83.     private String tName;
84.     private String tSchool;
85.     private String tDepartment;
86.     public Teacher() {
87.     }
88.     pub-
lic Teacher(int tNo, String tName, String tSchool, S
tring tDepartment) {
89.         super();
90.         this.tNo = tNo;
91.         this.tName = tName;
92.         this.tSchool = tSchool;
93.         this.tDepartment = tDepartment;
94.     }
95.     public void setNo(int no) {
96.         this.tNo = no;
97.     }
98.     public int getNo() {
99.         return tNo;
100.    }
101.    public void setName(String name) {
102.        this.tName = name;
103.    }
104.    public String getName() {
105.        return tName;
106.    }
107.    public void setSchool(String school) {
108.        this.tSchool = school;
109.    }
110.    public String getSchool() {
111.        return tSchool;
112.    }
113.    public void setDepartment(String depart-
ment) {
114.        this.tDepartment = department;
115.    }
116.    public String getDepartment() {
117.        return tDepartment;
118.    }
119.    public String toString() {
120.        re-
turn tNo + "\t" + tName + "\t" + tSchool + "\t" + tD
epartment;
121.    }
122. }

```

```

123.
124. class Course {
125.     private int cNo;
126.     private String cName;
127.     private int cCredit;
128.     private int cHour;
129.     public Course() {
130.     }
131.     public
132.     lic Course(int cNo, String cName, int cCredit, int c
133.     Hour) {
134.         super();
135.         this.cNo = cNo;
136.         this.cName = cName;
137.         this.cCredit = cCredit;
138.         this.cHour = cHour;
139.     }
140.     public void setNo(int no) {
141.         this.cNo = no;
142.     }
143.     public int getNo() {
144.         return cNo;
145.     }
146.     public void setName(String name) {
147.         this.cName = name;
148.     }
149.     public String getName() {
150.         return cName;
151.     }
152.     public void setCredit(int credit) {
153.         this.cCredit = credit;
154.     }
155.     public int getCredit() {
156.         return cCredit;
157.     }
158.     public void setHour(int hour) {
159.         this.cHour = hour;
160.     }
161.     public int getHour() {
162.         return cHour;
163.     }
164.     public String toString() {
165.         re-
166.         turn cNo + "\t" + cName + "\t" + cCredit + "\t" + cH
167.         our;
168.     }
169. }
170.
171. class Grade {
172.     public Student stu;
173.     public Student getStu() {
174.         return stu;
175.     }
176.     public void setStu(Student stu) {
177.         this.stu = stu;
178.     }
179.     public Teacher tea;
180.     public Teacher getTea() {

```

```

177.         return tea;
178.     }
179.     public void setTea(Teacher tea) {
180.         this.tea = tea;
181.     }
182.     public Course cour;
183.     public Course getCour() {
184.         return cour;
185.     }
186.     public void setCour(Course cour) {
187.         this.cour = cour;
188.     }
189.     private int gNo;
190.     private int sNo;
191.     private String cName;
192.     private String tName;
193.     private String sName;
194.     private int gPoint;
195.     public Grade() {
196.     }
197.     public
198.     lic Grade(int gNo, int sNo, String sName, String cName, String tName, int gPoint) {
199.         super();
200.         this.gNo = gNo;
201.         this.sNo = sNo;
202.         this.cName = cName;
203.         this.tName = tName;
204.         this.sName = sName;
205.         this.gPoint = gPoint;
206.     }
207.     public void setNo(int no) {
208.         this.sNo = no;
209.     }
210.     public int getSNo() {
211.         return sNo;
212.     }
213.     public void setGNo(int no) {
214.         this.gNo = no;
215.     }
216.     public int getGNo() {
217.         return gNo;
218.     }
219.     public void setCourseName(String name) {
220.         this.cName = name;
221.     }
222.     public String getCourseName() {
223.         return cName;
224.     }
225.     public void setTeacherName(String name) {
226.         this.tName = name;
227.     }
228.     public String getTeacherName() {
229.         return tName;
230.     }
231.     public void setStudentName(String name) {
232.         this.sName = name;
233.     }

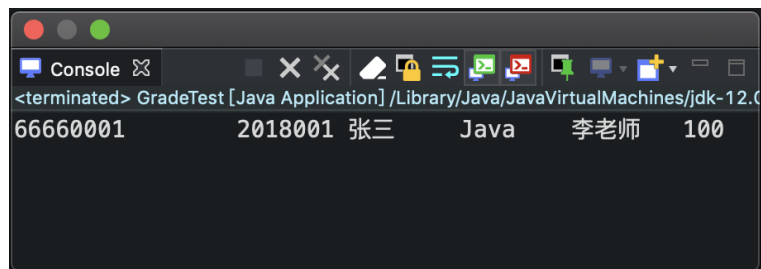
```

```

233.     public String getStudentName() {
234.         return sName;
235.     }
236.     public void setPoint(int point) {
237.         this.gPoint = point;
238.     }
239.     public int getPoint() {
240.         return gPoint;
241.     }
242.     public String toString() {
243.         re-
turn gNo + "\t" + sNo + "\t" + sName + "\t" + cName
+ "\t" + tName + "\t" + gPoint;
244.     }
245. }

```

(2) 运行截图



## 九、 总结及心得体会：

本实验目的是让学生熟悉 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

通过本实验，我体会到了面向对象程序设计的基本思想，掌握了 Java 对类的继承的基本操作，巩固了对类和对象的基本操作方法。提高了我对问题的分析能力以及编码能力，受益良多。

## 十、 对本实验过程及方法、手段的改进建议：

实验内容存在重复的题目，且部分题目的时效性不强。可以采用一些新颖的题目背景，激发学生的编程兴趣、提高实验能力。

**报告评分：**

**指导教师签字：**