**《面向对象程序设计(Java)》**

**实验报告**

姓 名 石家晖

学 号 2019081238

学 院 软件工程学院

专 业 软件工程

年 级 2019级

指导教师 文立玉

**成都信息工程大学软件工程学院**

**2020年9月**

# 实验4：

1. **实验目的**

1. 掌握Java的继承与多态，能编写体现类的继承性（成员变量、成员方法、成员变量隐藏）、类的多态性（成员方法重载、构造方法重载）的程序。

2. 掌握抽象类与接口

3. 掌握Object类的equals方法和toString方法的使用

1. **实验要求**

1. 实验在2学时内完成。

2. 1人1组独立完成。

1. **实验设备与环境**

JDK13, NEOVIM，Linux

1. **设计思路与具体实现**

1. 设计一个简单的学校人事管理系统,管理教师和学生信息。实现教师与学生基本信息的添加、删除、修改、查询。其中编号、姓名、性别、出生日期为共有的，教师包括部门、职称、工资；学生包括高考分数、专业field、班级等。

(1) 设计思路

设计一个简单的学校人事管理系统,管理教师和学生信息。实现教师与学生基本信息的添加、删除、修改、查询。其中编号、姓名、性别、出生日期为共有的，教师包括部门、职称、工资；学生包括高考分数、专业field、班级等。

(2) 实验步骤

①、创建父类Person和它的子类Student和Teacher

②、实现教师与学生基本信息的添加、删除、修改、查询。其中编号、姓名、性别、出生日期为共有的，教师包括部门、职称、工资；学生包括高考分数、专业field、班级等。

③、 在主函数中输入一些测试用数据进行测试

(3) 关键代码

class Person {

private int num;

private String name;

private String sex;

private int birthYear, birthMonth, birthDay;

public void setNum (int num) {

this.num = num;

}

public void setName (String name) {

this.name = name;

}

public void setSex (String sex) {

this.sex = sex;

}

public void setbirth (int y, int m, int d) {

this.birthYear = y;

this.birthMonth = m;

this.birthDay = d;

}

public int getNum() {

return num;

}

public String getSex() {

return sex;

}

public String getName() {

return name;

}

public int getBirthYear() {

return birthYear;

}

public int getBirthMonth() {

return birthMonth;

}

public int getbirthDay() {

return birthDay;

}

public void check() {

System.out.println ("编号：" + num);

System.out.println ("姓名：" + name);

System.out.println ("性别：" + sex);

System.out.println ("出生日期：" + birthYear + "年" + birthMonth + "月" + birthDay + "日");

}

public Person (int num,

String name,

String sex,

int birthYear,

int birthMonth,

int birthDay) {

this.num = num;

this.name = name;

this.sex = sex;

this.birthYear = birthYear;

this.birthMonth = birthMonth;

this.birthDay = birthDay;

}

}

class Teacher extends Person {

private String department, title;

private int salary;

public String getDepartment() {

return department;

}

public String getTitle() {

return title;

}

public int getSalary() {

return salary;

}

public void setDepartment (String department) {

this.department = department;

}

public void setTitle (String title) {

this.title = title;

}

public void setSalary (int salary) {

this.salary = salary;

}

public void check() {

System.out.println ("编号：" + getNum() );

System.out.println ("姓名：" + getName() );

System.out.println ("性别：" + getSex() );

System.out.println ("出生日期：" + getBirthYear() + "年" + getBirthMonth() + "月" + getbirthDay() + "日");

System.out.println ("部门：" + department);

System.out.println ("职称：" + title);

System.out.println ("薪资：" + salary);

}

public Teacher (int num,

String name,

String sex,

int birthYear,

int birthMonth,

int birthDay,

String department,

String title,

int salary) {

super (num, name, sex, birthYear, birthMonth, birthDay);

this.department = department;

this.title = title;

this.salary = salary;

}

}

class Student extends Person {

private String field, sClass;

private int score;

public void setScore (int score) {

this.score = score;

}

public void setField (String field) {

this.field = field;

}

public void setSClass (String sClass) {

this.sClass = sClass;

}

public int getScore() {

return score;

}

public String getField() {

return field;

}

public String getSClass() {

return sClass;

}

public void check() {

System.out.println ("编号：" + getNum() );

System.out.println ("姓名：" + getName() );

System.out.println ("性别：" + getSex() );

System.out.println ("出生日期：" + getBirthYear() + "年" + getBirthMonth() + "月" + getbirthDay() + "日");

System.out.println ("高考分数：" + score);

System.out.println ("专业：" + field);

System.out.println ("班级：" + sClass);

}

public Student (int num,

String name,

String sex,

int birthYear,

int birthMonth,

int birthDay,

int score,

String field,

String sClass) {

super (num, name, sex, birthYear, birthMonth, birthDay);

this.score = score;

this.field = field;

this.sClass = sClass;

}

}

(4) 程序测试过程

在主函数中分别对类Teacher和Student进行输入输出测试。

(5) 运行结果（运行效果）

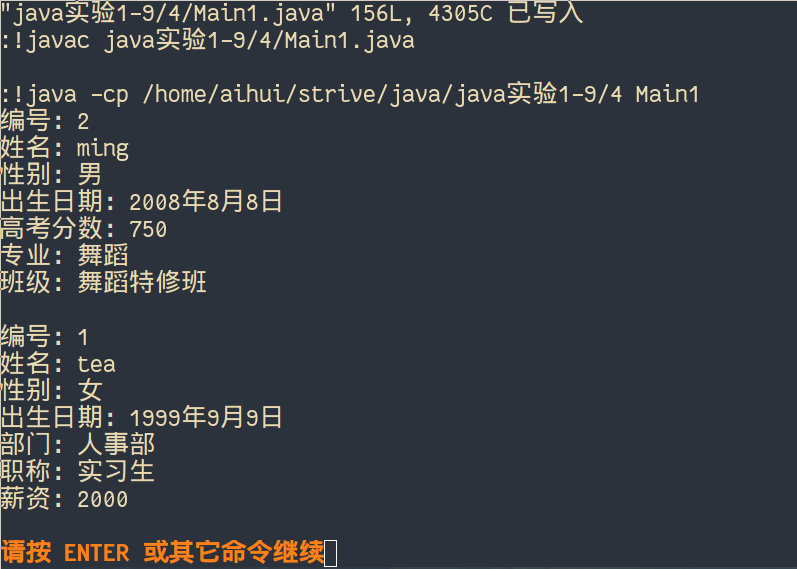


图4-1

2. 设计 3 个类 , 分别是学生类 Student, 本科生类 Undergaduate, 研究生类 Postgraduate, 其中 Student 类是一个抽象类 , 它包含一些基本的学生信息如姓名、所学课程、课程成绩等 , 而 Undergraduate 类和 Postgraduate 都是 Student 类的子类 , 它们之间的主要差别是计算课程成绩等级的方法有所不同 , 研究生的标准要比本科生的标准高一些 , 如下表所示。

|  |  |  |  |
| --- | --- | --- | --- |
| 本科生标准 | | 研究生标准 | |
| 80-100 | 优秀 | 90-100 | 优秀 |
| 70-80 | 良好 | 80-90 | 良好 |
| 60-70 | 一般 | 70-80 | 一般 |
| 50-60 | 及格 | 60-70 | 及格 |
| 50以下 | 不及格 | 60以下 | 不及格 |

假设某班级里既有本科生也有研究生 , 请编写程序统计出全班学生的成绩等级并显示出来。

(1) 设计思路

设计 3 个类 , 分别是学生类 Student, 本科生类 Undergaduate, 研究生类 首先创建抽象学生父类Student，编写构造方法，打印方法等方法，创建本科生类Undergaduate, 研究生类Postgraduate继承自Student，在两个子类中分别实现其抽象方法。在主函数中通过数组储存学生信息，通过判断学生学历，用父类引用指向子类对象，区别构造对象是本科生还是研究生。再调用输出方法对信息进行遍历输出。

(2) 实验步骤

①、创建父类Student和它的子类本科生类 Undergaduate和 研究生类 Postgraduate

②、Student 类是一个抽象类 , 填充它包含一些基本的学生信息如姓名、所学课程、课程成绩等 。而 Undergraduate 类和 Postgraduate 都是 Student 类的子类 , 分别实现本科生类 Undergaduate和 研究生类 Postgraduate所涉及的功能

③、在主函数中输入一些测试用数据进行测试。设计一个学生数组 , 实现其既能存放本科生对象 , 又能存放研究生对象。

(3) 关键代码

abstract class Student {

String name;

String sClass;

int score;

public void setName (String name) {

this.name = name;

}

public void setsClass (String sClass) {

this.sClass = sClass;

}

public void setScore (int score) {

this.score = score;

}

public String getName() {

return name;

}

public String getsClass() {

return sClass;

}

public int getScore() {

return score;

}

public void print() {}

public Student (String name, String sClass, int score) {

this.name = name;

this.sClass = sClass;

this.score = score;

}

}

class Undergaduate extends Student {

public void print() {

System.out.print ("本科生");

System.out.print (" " + name);

System.out.print (" " + sClass);

System.out.print (" ");

if (score >= 80)

System.out.println ("优秀");

else if (score >= 70)

System.out.println ("良好");

else if (score >= 60)

System.out.println ("一般");

else if (score >= 50)

System.out.println ("及格");

else

System.out.println ("不及格");

}

public Undergaduate (String name, String sClass, int score) {

super (name, sClass, score);

}

}

class Postgraduate extends Student {

public void print() {

System.out.print ("研究生");

System.out.print (" " + name);

System.out.print (" " + sClass);

System.out.print (" ");

if (score >= 90)

System.out.println ("优秀");

else if (score >= 80)

System.out.println ("良好");

else if (score >= 70)

System.out.println ("一般");

else if (score >= 60)

System.out.println ("及格");

else

System.out.println ("不及格");

}

public Postgraduate (String name, String sClass, int score) {

super (name, sClass, score);

}

}

(4) 程序测试过程

在主函数中创建数组对类进行输入输出测试。

(5) 运行结果（运行效果）

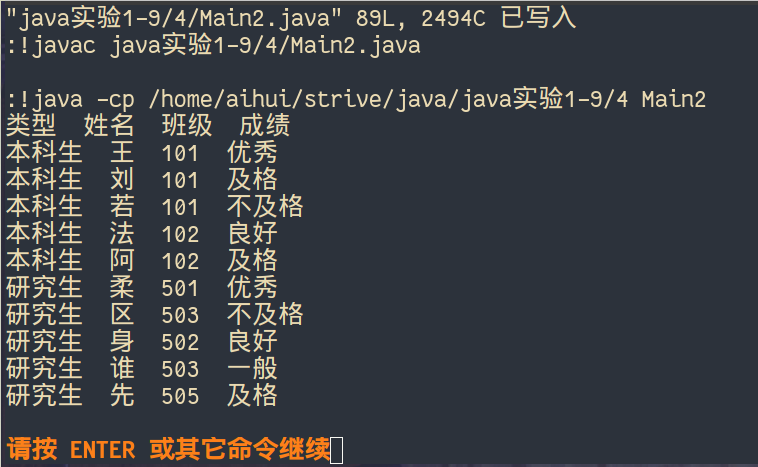


图4-2

1. **实验总结**

通过本次实验，我掌握了Java的继承与多态，能编写体现类的继承性（成员变量、成员方法、成员变量隐藏）、类的多态性（成员方法重载、构造方法重载）的程序。

另外，我还掌握了抽象类与接口并且掌握了Object类的equals方法和toString方法的使用

1. **附录（程序代码）**

1.public class Main1 {

public static void main (String[] args) {

Student s = new Student(2,"ming","男",2008,8,8,750,"舞蹈","舞蹈特修班");

Teacher t = new Teacher(1,"tea","女",1999,9,9,"人事部","实习生",2000);

s.check();

System.out.println();

t.check();

}

}

class Person {

private int num;

private String name;

private String sex;

private int birthYear, birthMonth, birthDay;

public void setNum (int num) {

this.num = num;

}

public void setName (String name) {

this.name = name;

}

public void setSex (String sex) {

this.sex = sex;

}

public void setbirth (int y, int m, int d) {

this.birthYear = y;

this.birthMonth = m;

this.birthDay = d;

}

public int getNum() {

return num;

}

public String getSex() {

return sex;

}

public String getName() {

return name;

}

public int getBirthYear() {

return birthYear;

}

public int getBirthMonth() {

return birthMonth;

}

public int getbirthDay() {

return birthDay;

}

public void check() {

System.out.println ("编号：" + num);

System.out.println ("姓名：" + name);

System.out.println ("性别：" + sex);

System.out.println ("出生日期：" + birthYear + "年" + birthMonth + "月" + birthDay + "日");

}

public Person (int num,

String name,

String sex,

int birthYear,

int birthMonth,

int birthDay) {

this.num = num;

this.name = name;

this.sex = sex;

this.birthYear = birthYear;

this.birthMonth = birthMonth;

this.birthDay = birthDay;

}

}

class Teacher extends Person {

private String department, title;

private int salary;

public String getDepartment() {

return department;

}

public String getTitle() {

return title;

}

public int getSalary() {

return salary;

}

public void setDepartment (String department) {

this.department = department;

}

public void setTitle (String title) {

this.title = title;

}

public void setSalary (int salary) {

this.salary = salary;

}

public void check() {

System.out.println ("编号：" + getNum() );

System.out.println ("姓名：" + getName() );

System.out.println ("性别：" + getSex() );

System.out.println ("出生日期：" + getBirthYear() + "年" + getBirthMonth() + "月" + getbirthDay() + "日");

System.out.println ("部门：" + department);

System.out.println ("职称：" + title);

System.out.println ("薪资：" + salary);

}

public Teacher (int num,

String name,

String sex,

int birthYear,

int birthMonth,

int birthDay,

String department,

String title,

int salary) {

super (num, name, sex, birthYear, birthMonth, birthDay);

this.department = department;

this.title = title;

this.salary = salary;

}

}

class Student extends Person {

private String field, sClass;

private int score;

public void setScore (int score) {

this.score = score;

}

public void setField (String field) {

this.field = field;

}

public void setSClass (String sClass) {

this.sClass = sClass;

}

public int getScore() {

return score;

}

public String getField() {

return field;

}

public String getSClass() {

return sClass;

}

public void check() {

System.out.println ("编号：" + getNum() );

System.out.println ("姓名：" + getName() );

System.out.println ("性别：" + getSex() );

System.out.println ("出生日期：" + getBirthYear() + "年" + getBirthMonth() + "月" + getbirthDay() + "日");

System.out.println ("高考分数：" + score);

System.out.println ("专业：" + field);

System.out.println ("班级：" + sClass);

}

public Student (int num,

String name,

String sex,

int birthYear,

int birthMonth,

int birthDay,

int score,

String field,

String sClass) {

super (num, name, sex, birthYear, birthMonth, birthDay);

this.score = score;

this.field = field;

this.sClass = sClass;

}

}

2.

public class Main2 {

public static void main (String[] args) {

Student []stu = new Student[10];

stu[0] = new Undergaduate ("王", "101", 80);

stu[1] = new Undergaduate ("刘", "101", 50);

stu[2] = new Undergaduate ("若", "101", 30);

stu[3] = new Undergaduate ("法", "102", 75);

stu[4] = new Undergaduate ("阿", "102", 55);

stu[5] = new Postgraduate ("柔", "501", 95);

stu[6] = new Postgraduate ("区", "503", 49);

stu[7] = new Postgraduate ("身", "502", 85);

stu[8] = new Postgraduate ("谁", "503", 75);

stu[9] = new Postgraduate ("先", "505", 65);

System.out.println ("类型 姓名 班级 成绩");

for(Student i:stu)

i.print();

}

}

abstract class Student {

String name;

String sClass;

int score;

public void setName (String name) {

this.name = name;

}

public void setsClass (String sClass) {

this.sClass = sClass;

}

public void setScore (int score) {

this.score = score;

}

public String getName() {

return name;

}

public String getsClass() {

return sClass;

}

public int getScore() {

return score;

}

public void print() {}

public Student (String name, String sClass, int score) {

this.name = name;

this.sClass = sClass;

this.score = score;

}

}

class Undergaduate extends Student {

public void print() {

System.out.print ("本科生");

System.out.print (" " + name);

System.out.print (" " + sClass);

System.out.print (" ");

if (score >= 80)

System.out.println ("优秀");

else if (score >= 70)

System.out.println ("良好");

else if (score >= 60)

System.out.println ("一般");

else if (score >= 50)

System.out.println ("及格");

else

System.out.println ("不及格");

}

public Undergaduate (String name, String sClass, int score) {

super (name, sClass, score);

}

}

class Postgraduate extends Student {

public void print() {

System.out.print ("研究生");

System.out.print (" " + name);

System.out.print (" " + sClass);

System.out.print (" ");

if (score >= 90)

System.out.println ("优秀");

else if (score >= 80)

System.out.println ("良好");

else if (score >= 70)

System.out.println ("一般");

else if (score >= 60)

System.out.println ("及格");

else

System.out.println ("不及格");

}

public Postgraduate (String name, String sClass, int score) {

super (name, sClass, score);

}

}