

面向对象Java—实验报告

****

**实验二：面向对象（一）**.

**姓 名：**  於俊涛

**学 号： 202007020625**

**专业班级： 计算机203**

**系 别： 计算机系**

**学 院： 电子信息与人工智能学院**

**实验二 面向对象（一）**

1. **实验目的**

在集成开发环境下，运用Java语言的面向对象的特性来实现类的封装、构造方法的定义以及对象的实例化和对象方法的调用。

1. **实验目标**

(1)能够根据需要解决的实际问题，利用Java语言实现类的定义；

(2)能够熟练运用构造方法来实例化对象以及调用对象的方法。

1. **实验内容**

**3.1 实验环境**

Eclipse IDE for Java Developers

**3.2 具体实验内容**

1．定义一个计算机类，要求该类中至少包含3个私有的属性（为每个属性定义设置和获取方法）,该类中至少包含两个构造方法（一个无参，一个有参），以及包含2-4个方法来描述计算机对象的行为（其中一个方法要能打印输出计算机对象的所有属性值）

2．定义一个类，该类中包含以下几个方法（静态）：

实现两个字符串数组的逆序排序，输出结果为字符串数组；

求两个整形数组的交集；

求两个浮点型数组的并集；

3．定义一个表示学生信息的类Student，要求如下：

（1）类Student的成员变量：

sNO　表示学号；sName表示姓名；sSex表示性别；sAge表示年龄；sJava：表示Java课程成绩。

（2）类Student带参数的构造方法：

在构造方法中通过形参完成对成员变量的赋值操作。

（3）类Student的方法成员：

getNo（）：获得学号；

getName（）：获得姓名；

getSex（）：获得性别；

getAge（）获得年龄；

getJava（）：获得Java 课程成绩

setNo（）：设置学号；

setName（）：设置姓名；

setSex（）：设置性别；

setAge（）；设置年龄

setJava（）：设置Java 课程成绩

根据类Student的定义，创建五个该类的对象，输出每个学生的信息，计算并输出这五个学生Java语言成绩的平均值，以及计算并输出他们Java语言成绩的最大值和最小值。

**四、实验过程分析**

**4.1 实验步骤**

1．定义一个计算机类，要求该类中至少包含3个私有的属性（为每个属性定义设置和获取方法）,该类中至少包含两个构造方法（一个无参，一个有参），以及包含2-4个方法来描述计算机对象的行为（其中一个方法要能打印输出计算机对象的所有属性值）。

实验代码：

//定义计算机类

public class Computer {

private String computerName;

private double computerPrice;

private String computerType;

//无参的构造方法

public Computer(){}

//有参的构造方法

public Computer(String computerName, double computerPrice, String computerType) {

this.computerName = computerName;

this.computerPrice = computerPrice;

this.computerType = computerType;

}

//看书

public void read() {

System.out.println("使用" + this.computerName + "电脑阅读");

}

//打印电脑的属性

public void printComputer() {

System.out.println("============打印电脑属性============");

System.out.println("电脑品牌为: " + this.computerName);

System.out.println("电脑类型为: " + this.computerType);

System.out.println("电脑价格为: " + this.computerPrice);

System.out.println("=================================");

}

//追剧

public void watch() {

System.out.println("使用" + this.computerName + "电脑追剧");

}

}

//测试类

class Test {

public static void main(String[] args) {

Computer computer = new Computer("小米",4000,"笔记本");

computer.printComputer();

computer.read();

computer.watch();

}

}

结果输出：

2．定义一个类，该类中包含以下几个方法（静态）：

实现两个字符串数组的逆序排序，输出结果为字符串数组；

求两个整形数组的交集；

求两个浮点型数组的并集；

实验代码：

package experiment2

public class Myclass {

//实现两个字符串数组的逆序排序，输出结果为字符串数组

public static String[] reverse(String[] str1, String[] str2) {

String[] ret = new String[str1.length+str2.length];

int index = 0;

for(int i = str1.length-1; i >= 0; i--) {

ret[index++] = str1[i];

}

for (int i = str2.length-1; i >= 0 ; i--) {

ret[index++] = str2[i];

}

return ret;

}

//求两个整形数组的交集

public static void intersect(int[] arr1, int[] arr2) {

int flag = 0;

for(int i = 0; i < arr1.length; i++) {

for (int j = 0; j < arr2.length; j++) {

if(arr1[i] == arr2[j]) {

flag = 1;

System.out.print(arr1[i] + " ");

}

}

}

if(flag == 0) {

System.out.println("两个整形数组的交集为空");

}

}

//求两个浮点型数组的并集

public static void union(double[] arr1, double[] arr2) {

double[] ret = new double[arr1.length+arr2.length];

for (int i = 0; i < ret.length; i++) {

ret[i] = Integer.MAX\_VALUE;

}

int index = 0;

for (int i = 0; i < arr1.length; i++) {

ret[index++] = arr1[i];

}

for (int i = 0; i < arr2.length; i++) {

int flag = 0;

for (int j = 0; ret[j] < Integer.MAX\_VALUE; j++) {

if(arr2[i] == ret[j]) {

flag = 1;

}

}

if(flag == 0) {

ret[index++] = arr2[i];

}

}

for (int i = 0; ret[i] < Integer.MAX\_VALUE; i++) {

System.out.print(ret[i] + " ");

}

}

}

public class Test3 {

public static void main(String[] args) {

String[] str1 = {"abc","bcd","abd"};

String[] str2 = {"cda","aaa","bbb"};

String[] ret = Myclass.reverse(str1,str2);

System.out.println("字符串1逆序后的结果为: ");

for (int i = 0; i < str1.length; i++) {

System.out.print(ret[i] + " ");

}

System.out.println();

System.out.println("字符串2逆序后的结果为: ");

for (int i = 0; i < str2.length; i++) {

System.out.print(ret[i] + " ");

}

System.out.println();

System.out.println("=======================");

int[] arr1 = {1,3,5,7,9};

int[] arr2 = {2,4,9,8};

System.out.println("两个整型数组的交集为: ");

Myclass.intersect(arr1,arr2);

System.out.println();

System.out.println("=======================");

double[] arr3 = {3.2,4.1,5.8,6.9};

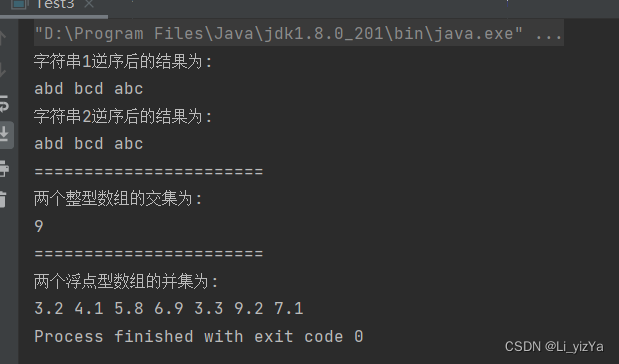
double[] arr4 = {3.3,6.9,9.2,7.1};

System.out.println("两个浮点型数组的并集为: ");

Myclass.union(arr3,arr4);

}

}

结果输出：

3．定义一个表示学生信息的类Student，要求如下：

（1）类Student的成员变量：

sNO　表示学号；sName表示姓名；sSex表示性别；sAge表示年龄；sJava：表示Java课程成绩。

（2）类Student带参数的构造方法：

在构造方法中通过形参完成对成员变量的赋值操作。

（3）类Student的方法成员：

getNo（）：获得学号；

getName（）：获得姓名；

getSex（）：获得性别；

getAge（）获得年龄；

getJava（）：获得Java 课程成绩

setNo（）：设置学号；

setName（）：设置姓名；

setSex（）：设置性别；

setAge（）；设置年龄

setJava（）：设置Java 课程成绩

根据类Student的定义，创建五个该类的对象，输出每个学生的信息，计算并输出这五个学生Java语言成绩的平均值，以及计算并输出他们Java语言成绩的最大值和最小值。

实验代码;

public class Student {

private int sNo;

private String sName;

private String sSex;

private int sAge;

private int sJava;

public Student(int sNo, String sName, String sSex, int sAge, int sJava) {

this.sNo = sNo;

this.sName = sName;

this.sSex = sSex;

this.sAge = sAge;

this.sJava = sJava;

}

public int getsNo() {

return sNo;

}

public void setsNo(int sNo) {

this.sNo = sNo;

}

public String getsName() {

return sName;

}

public void setsName(String sName) {

this.sName = sName;

}

public String getsSex() {

return sSex;

}

public void setsSex(String sSex) {

this.sSex = sSex;

}

public int getsAge() {

return sAge;

}

public void setsAge(int sAge) {

this.sAge = sAge;

}

public int getsJava() {

return sJava;

}

public void setsJava(int sJava) {

this.sJava = sJava;

}

@Override

public String toString() {

return "学号: " + sNo +

", 姓名: " + sName + ' ' +

", 性别: " + sSex + ' ' +

", 年龄: " + sAge +

", Java成绩: " + sJava;

}

}

public class Test2 {

public static void main(String[] args) {

Student[] students = new Student[5];

students[0] = new Student(1,"张三","男",18,92);

students[1] = new Student(2,"李四","男",19,81);

students[2] = new Student(3,"王五","男",18,76);

students[3] = new Student(4,"张三金","男",20,99);

students[4] = new Student(5,"赵东","男",18,87);

int count = 0;

int maxScore = 0;

int minScore = 100;

System.out.println("=====================学生信息表========================");

for (Student student : students) {

//计算五名学生总分

count += student.getsJava();

//获取最高成绩

if(student.getsJava() > maxScore) {

maxScore = student.getsJava();

}

//获取最低成绩

if(student.getsJava() < minScore) {

minScore = student.getsJava();

}

System.out.println(student.toString());

}

System.out.println("=====================================================");

System.out.println();

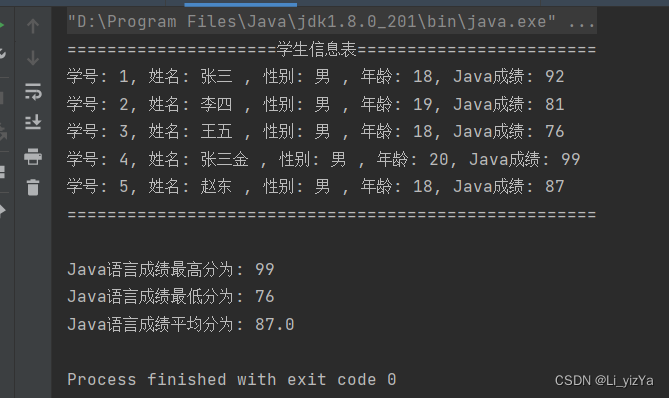
System.out.println("Java语言成绩最高分为: " + maxScore);

System.out.println("Java语言成绩最低分为: " + minScore);

System.out.println("Java语言成绩平均分为: " + (double)count/ students.length);

}

}

结果输出：

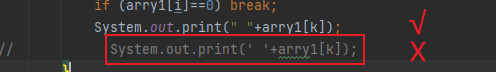
* 1. **错误分析**

错误情况：

在第二题输出时,输出结果与预期不符且相差较大。

解决方案：

经过反复查找与调试，发现罪魁祸首是因为在输出语句中的' ' 导致的输出失常； 因此调整代码的输出语句，将所有的' '改换为" "



1. **实验总结**

这次实验内容与上学期的数据结构知识相似，例如求数组（集合）的交并补集，逆序输出等等，都与数据结构里面的排序和查找算法有关，两者结合，相辅相成。在本次实验中我顺利的完成了实验任务，并在实验的过程中经过调试找出了问题所在。通过本次实验，经过问题思考与问题解决的规划，对实验的完成有较大的帮助，同时在实验的过程中也加深了对于已有知识点的理解与运用，虽不能融会贯通但也不至于纸上谈兵，这也算本次实验中较大的收获之一。在本次实验的过程中当然也离不开同学的相互交流，当然，对实验题目的理解出现偏差时也需要及时向老师寻求帮助，以免因为错误的理解方向，导致产生的错误结果但是自己不知道的情况。在部分题目中通过借鉴网上同类题目的解题思路对理解题目与解决题目也同样具有极大的帮助，积极的汲取知识将会对学习产生很大影响，不同的正确解题思路也会碰撞出不一样的火花。