**实验一Java程序控制结构**

1. **实验目的：**

1、练习类的构造方法、方法的调用、参数传递、对象的构造与使用;

2、练习循环结构、控制结构;  
3、练习数据的输入与输出;  
4、学习编写结构清晰、逻辑正确、功能完善的 java 代码。 5、学习使用工具对代码进行跟踪调试。

**二，实验环境**

**装有eclipse并配置好Java运行环境的电脑**

1. **实验内容**
2. 题目描述:

(1),某班有 30 名学生，每个学生有学号、姓名和 4 门成绩(学生信息随机产生， 存入一个对象数组)，现新转来一名学生，请输入这位学生的学号、姓名和成绩(四门课的 成绩小于 100 分，从键盘输入，以空格为区分)，然后显示这名学生平均分、最高分，全班 的最高平均分学生信息、新同学在班级中的成绩水平，可能的结果如下所示:  
例1:  
请输入学生姓名:  
jack  
请输入学生学号:  
13201109  
请输入四门课的成绩，以空格分隔:  
89 78 94 0  
jack(ID: 13201109) 平均分 66.50 最高分 94.00  
rose (ID:13201104) 的最高平均分 90.12 最高分 100.00  
jack 的平均成绩低于全班的平均分

例2  
请输入学生姓名:  
jack  
请输入学生学号:  
13201109  
请输入四门课的成绩，以空格分隔:  
100 85 65 85  
jack(ID: 13201109) 平均分 72.38 最高分 100.00  
rose (ID:13201104) 的最高平均分 90.12 最高分 100.00 jack 的平均成绩高于全班的平均分

请输入学生姓名:  
jack  
请输入学生学号:  
13201109 请输入四门课的成绩，以空格分隔: 110 845 100000 100 课程成绩不能高于 100 分!

例4  
请输入学生姓名:  
jack  
请输入学生学号:  
13201109  
请输入四门课的成绩，以空格分隔:  
100 100 100 100  
jack(ID: 13201109) 平均分 100.00 最高分 100.00 jack 现在是班上成绩最好的学生了  
jack 的平均成绩高于全班的平均分

部分参考代码:

**//**学生类  
**class** Student{

String studentID;//学生学号

String studentName;//学生姓名

**int** []scores;//学生成绩

**}**

**源代码：**

**package experiment\_1;**

**import java.util.Random;**

**import java.util.Scanner;**

**//学生类**

**class Student{**

**//属性**

**String studentID; //学生学号**

**String studentName; //学生姓名**

**int []scores; //学生成绩**

**double averageScore = 0.0; //个人平均分**

**double bestScore = 0.0; //个人最高分**

**double classAverage = 0.0;**

**int index = 0; //班上平均成绩最好的学生下标**

**int classNumber = 30;**

**Student(){ //构造方法**

**/\***

**\* 通过随机数原理生成名字，分数等数据**

**\* 名字：随机生成的名字由4个字母组成（首字母大写）**

**\* 分数：生成区间【0，100】**

**\* \*/**

**//填充分数；每个学生四门成绩**

**Random rNumber = new Random(); // Random对象**

**scores = new int[4]; // 每个学生四门成绩**

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

**scores[i] = rNumber.nextInt(100);**

**}**

**//填写名字，由随机字母填充**

**String myChars = "abcdefghijklmnopqrstuvwxyz";**

**int indexNumber;**

**char nameArray[] = new char[4]; //名字含有四个字母**

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

**indexNumber = rNumber.nextInt(26);**

**nameArray[i] = myChars.charAt(indexNumber);**

**}**

**studentName = new String(nameArray);**

**//填充学号:生成最后两位随机数**

**int idNum;**

**String idNumString;**

**idNum = (int)(1 + Math.random() \* 31 ); // 01-30**

**if(idNum < 10) {**

**idNumString = "0" + idNum;**

**}**

**else {**

**idNumString = "" + idNum;**

**}**

**studentID = new String("132011" + idNumString);**

**}**

**//求出个人最高分**

**public void selectBest() {**

**int best = -999999;**

**for(int i = 0; i < this.scores.length; i++) {**

**if(this.scores[i] > best)**

**best = this.scores[i];**

**}**

**this.bestScore = (double)best;**

**}**

**//求出个人平均分**

**public void calAverage() {**

**double sum = 0;**

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

**sum += this.scores[i];**

**}**

**averageScore = (double)sum / 4;**

**}**

**//获取用户输入的信息**

**public boolean getInfo() {**

**boolean isLegal = true;**

**Scanner input = new Scanner(System.in);**

**System.out.println("请输入学生姓名:");**

**studentName = input.next();**

**System.out.println("请输入学生学号:");**

**studentID = input.next();**

**System.out.println("请输入四门课的成绩，以空格分隔:");**

**for(int i = 0; i < this.scores.length; i++) {**

**scores[i] = input.nextInt();**

**}**

**//检测成绩输入成绩是否超过范围【0，100】**

**for(int i = 0; i < this.scores.length; i++) {**

**if(this.scores[i] > 100)**

**isLegal = false;**

**}**

**return isLegal;**

**}**

**//求出班级平均分**

**public void calClassAverage(Student[] wholeClass) {**

**double average = 0.0;**

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

**average += wholeClass[i].averageScore;**

**}**

**average /= classNumber;**

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

**wholeClass[i].classAverage = average;**

**}**

**}**

**/\***

**\* 传入对象数组**

**\* 返回------平均成绩最高---的index**

**\* \*/**

**public void generalBest(Student[] wholeClass) {**

**double best = -99999.0;**

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

**if(wholeClass[i].averageScore > best) {**

**best = wholeClass[i].averageScore;**

**index = i;**

**}**

**}**

**}**

**public void showInfo(Student[] wholeClass, Student s,boolean isLegal) {**

**generalBest(wholeClass); //取得个人平均最高下标**

**calClassAverage(wholeClass); //求得班级平均分**

**if(isLegal){**

**//个人基本情况**

**System.out.printf("%s(ID: %s) 平均分 %.2f 最高分 %.2f\n",s.studentName,s.studentID,s.averageScore,s.bestScore);**

**//个人平均分对比**

**if(wholeClass[index].averageScore > s.averageScore){**

**System.out.printf("%s (ID:%s) 的最高平均分 %.2f 最高分 %.2f\n",wholeClass[index].studentName,wholeClass[index].studentID,**

**wholeClass[index].averageScore,wholeClass[index].bestScore);**

**}else{**

**System.out.printf("%s 现在是班上成绩最好的学生了\n",s.studentName);**

**}**

**//班上平均分对比**

**if(wholeClass[index].classAverage > s.averageScore){**

**System.out.printf("%s 的平均成绩低于全班的平均分\n",s.studentName);**

**}else{**

**System.out.printf("%s 的平均成绩高于全班的平均分\n",s.studentName);**

**}**

**}else{**

**System.out.println("课程成绩不能高于 100 分!");**

**}**

**}**

**//新同学加入**

**public boolean addStudent(Student [] wholeClass,Student s) {**

**boolean isLegal = s.getInfo();**

**if(isLegal) {**

**s.selectBest();**

**s.calAverage();**

**wholeClass[30].averageScore = s.averageScore;**

**wholeClass[30].bestScore = s.bestScore;**

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

**wholeClass[i].classNumber += 1;**

**}**

**}**

**return isLegal;**

**}**

**public static void main(String[] args) {**

**//生成大小为31的对象引用数组**

**Student wholeClass[] = new Student[31];**

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

**wholeClass[i] = new Student(); //生成新的对象**

**wholeClass[i].selectBest();**

**wholeClass[i].calAverage();**

**}**

**//添加新成员**

**Student s = new Student();**

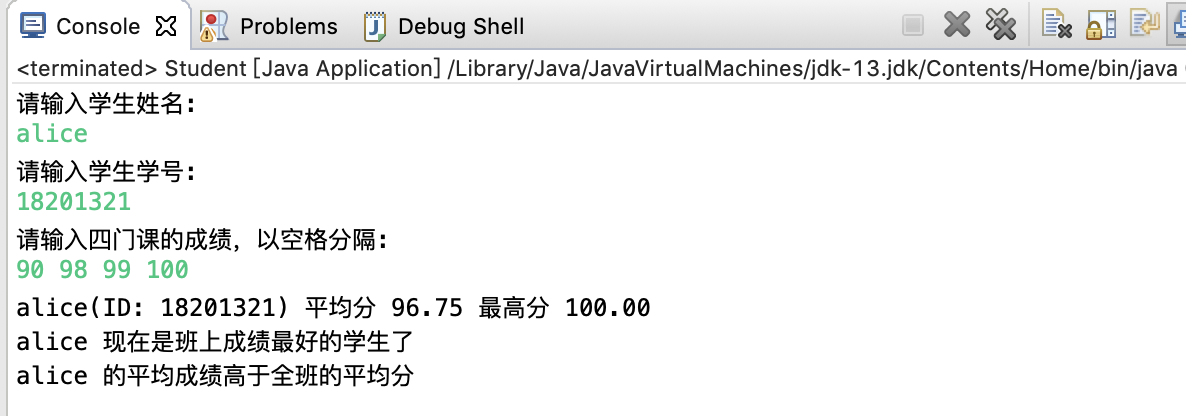
**boolean isLegal = s.addStudent(wholeClass,s);**

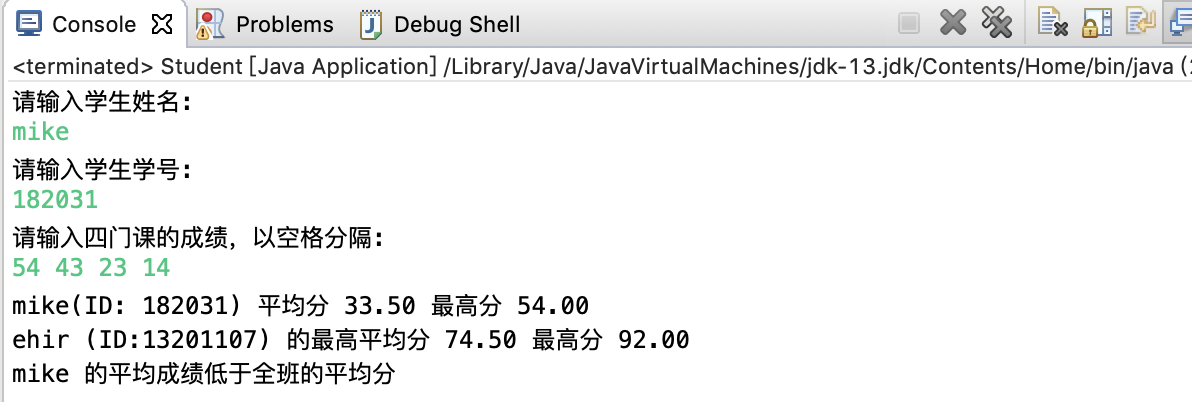
**s.showInfo(wholeClass, s,isLegal);**

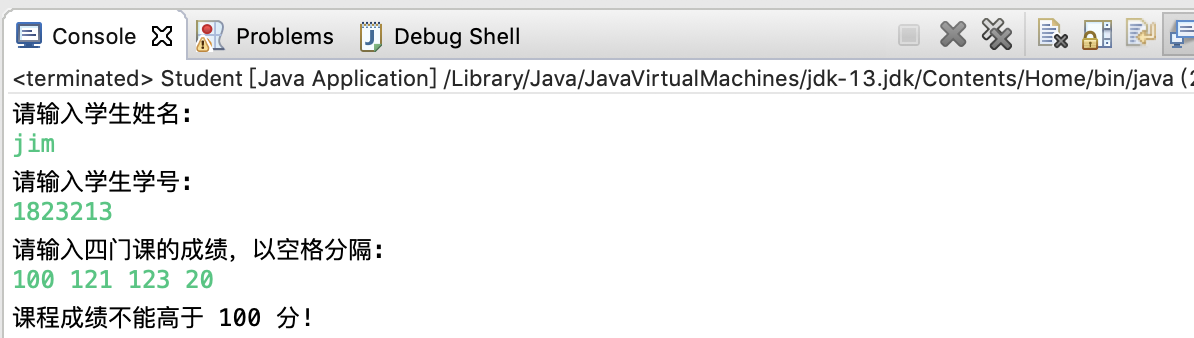
**}**

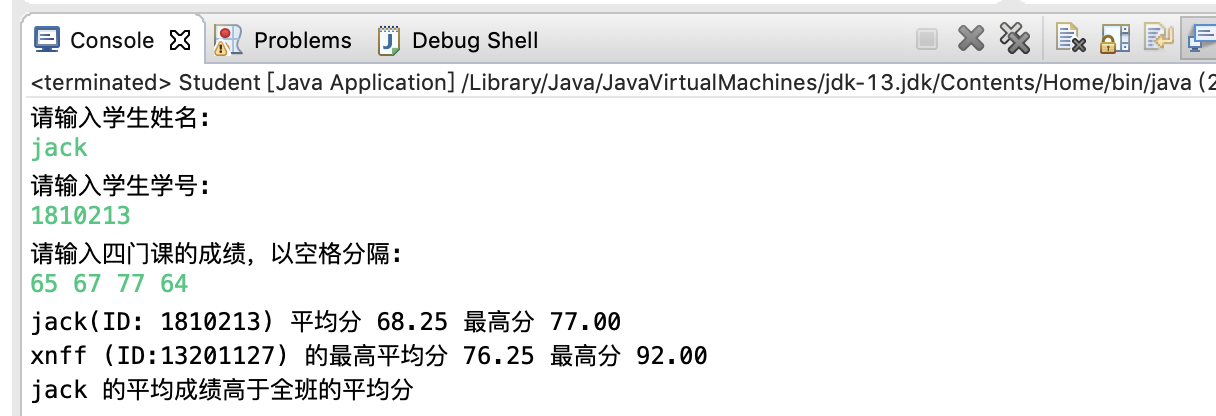
**}**

**运行截图：**

****

****

****

****

**(2),**下列代码中有几处语法错误和逻辑错误，请修正下面代码的错误并输出结果。

**package** experiment\_1;

**import** java.util.Scanner;

/\*\*

\* 分数的四则运算测试类,代码中仅包含了加法运算的输出，调试时仅需输入分数的加 法运算的数据，不要输入减法和乘除法。

\*/

**public class** FractionalCalculator {  
**public static void** main(String[] args) {

// 输入分数内容和运算符  
Scanner input = **new** Scanner(System.***in***); System.***out***.println("请分部输入表达式"); System.***out***.println("请输入参数1的分子:"); String data1 = input.next(); System.***out***.println("请输入参数1的分母:"); String data2 = input.next();

System.***out***.println("请输入运算符(+-\*/) :");

String operation = input.next();

System.***out***.println("请输入参数2的分子:"); S

tring data3 = input.next(); System.***out***.println("请输入参数2的分母:");

String data4 = input.next();

// 运算并输出  
Calculator cal = **new** Calculator(); System.***out***.println("运算结果为:"); cal.*compute*(data1,data2, data3,data4);

}

}

/\*\*  
\* 分数的四则运算类输入:两个以字符串构成的分数，输出其相加的结果，结果仍以

分数行式表示。

\*/

**int** numerator; // 分子 **int** denominator; // 分母 **class** Calculator {

/\*\*设置分子分母  
\* 输入:正整数  
\* 输出:最简分数式，分子分母不包含公约数 \*/

**void** setNumeratorAndDenominator(**int** a, **int** b){ // 设置分子和分母

**int** c = gcd(a,b); numerator = a / c denominator = b / c;

}

**int** getNumerator(){ **return** numerator;

}

**int** getDenominator(){ **return** denominator;

}

// 计算最大公约数

page3image58081024page3image58080640

/\*greatest common divisor 求两整数最大公约数， \* 输入:两正整数a,b  
\* 输出:最大公约数b  
\*/

**int** gcd(**int** a,**int** b){

**if**(a < b){ **int** c = a;

a = b;

b = c; }

**int** r = a % b; **while**(r != 0){

a = b;

b = r;

r = a % b;

} }

Calculator add(Calculator r){ // 加法运算  
**int** a = r.getNumerator();  
**int** b = r.getDenominator();  
**int** newNumerator = numerator \* b + denominator \* a; **int** newDenominator = denominator \* b;

Calculator result = **new** Calculator();

result.setNumeratorAndDenominator(newNumerator, newDenominator);

**return** result; }

Calculator dec(Calculator r){ // 减法运算  
**int** a = r.getNumerator();  
**int** b = r.getDenominator();  
**int** newNumerator = numerator \* b - denominator \* a; **int** newDenominator = denominator \* b;

Calculator result = **new** Calculator();

result.setNumeratorAndDenominator(newNumerator, newDenominator);

**return** result; }

// 转化输入格式，判断并输出运算结果

**public static void** compute(String data1,String data2,String operation,String data3,String data4){

**int** data1\_1 = Math.*abs*(Long.*parse*Long(data1)); **int** data1\_2 = Math.*abs*(Long.*parse*Long(data2)); **int** data2\_1 = Math.*abs*(Long.*parse*Long(data3)); **int** data2\_2 = Math.*abs*(Long.*parse*Long(data4));

Calculator r1 = **new** Calculator();

r1.setNumeratorAndDenominator(data1\_2, data1\_2); Calculator r2 = **new** Calculator(); r2.setNumeratorAndDenominator(data2\_1, data2\_2);

Calculator result;  
**int** a,b; **if**(operation.equals("+")){

result = r1.dec(r2);  
a = result.getNumerator();  
b = result.getDenominator(); System.***out***.println(data1+"/"+data2+" "+operation+" "

+data3+"/"+data4+" = " + a + "/" + b);

}

} }

修改后源代码

**96.public** **static** **void** compute(String data1,String data2,String operation,String data3,String data4)

不可以申明成静态方法

修改：**public** **void** compute(String data1,String data2,String operation,String data3,String data4)

98,**int** data1\_1 = Math.*abs*(Long.*parseLong*(data1));

类型转换错：long 不可以转换成 int(116-119)

修改：**int** data1\_1 = Math.*abs*(Integer.*parseInt*(data1));

cal.compute(data1,data2, operation,data3,data4);

漏参数operation

r1.setNumeratorAndDenominator(data1\_2,data1\_1);

参数位置错误

修改：r1.setNumeratorAndDenominator(data1\_1, data1\_2);