



HARBIN INSTITUTE OF TECHNOLOGY

哈尔滨工业大学

Java 程序设计实验报告

学号:

姓名:

专业: 软件工程类

班级: 1637102

实验三：面向对象程序设计

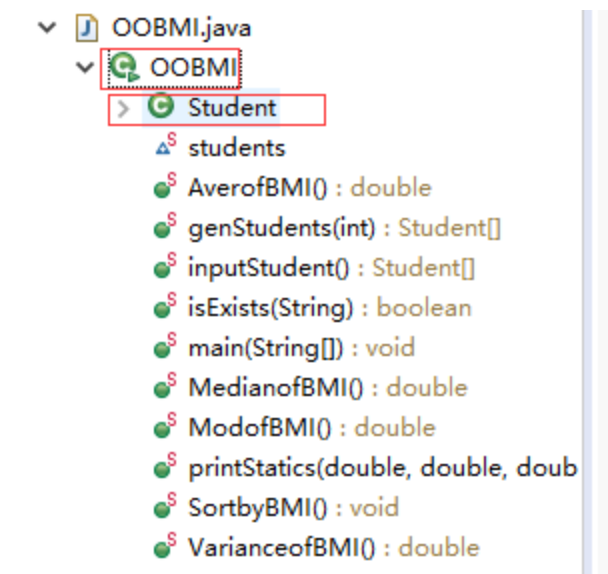
一、实验目的

- 1) 掌握面向对象的基本概念（成员变量、成员函数等）
- 2) 掌握类的定义、内部类的定义
- 3) 掌握对象的声明
- 4) 掌握对象数组的使用
- 5) 了解集合的概念和基本接口
- 6) 掌握增强 for 循环语句
- 7) 掌握范型的应用
- 8) 掌握基本集合对象 ArrayList 的应用
- 9) 了解流输入输出的基本概念。
- 10) 了解异常的概念。
- 11) 掌握 File、FileReader、FileWriter 等文件读写类库的使用。
- 12) 掌握 IOException、FileNotFoundException 等异常的使用。

二、实验内容

1) 编写 OOBMI 类；并在其中定义内部类 Student，该类包含学号、姓名、身高、体重和 bmi 等属性；在 OOBMI 中增加成员属性 ArrayList<Student> students；

Step1.定义该类在 OOBMI 类下



Step2.编写代码如下：

```

public class Student{
    String number="",name="";
    double height=0.0,weight=0.0,bmi=0.0;

    public Student(){
    public Student(String number,String name,double height,double weight){
        this.number=number;
        this.name=name;
        this.height=(int)(height*100)/100.0;
        this.weight=(int)(weight*100)/100.0;
        this.bmi=(int)(weight/(height*height))*100/100.0;
    }
}

```

Step3. 增加成员属性 `ArrayList<Student> students` :

```

public class OOBIM {
    static ArrayList<Student> students=new ArrayList<Student>();
    public static void main(){

```

2) 在 OOBMI 类中增加成员函数 `inputStudent`，从键盘输入学生信息，函数返回值为 `Student` 类型。

Step1.定义函数 `inputStudent` () 如下

```

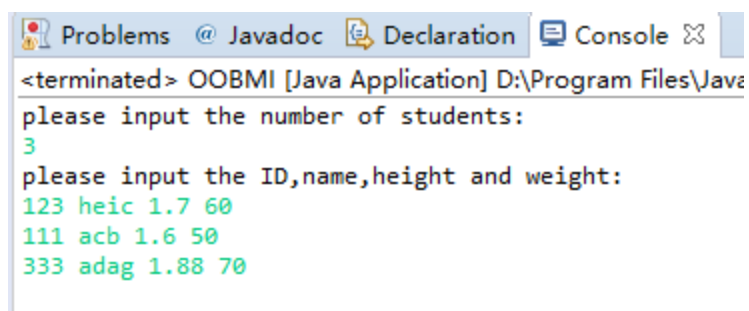
public static Student inputStudent(Scanner in){

    String number=in.next();
    String name=in.next();
    double height=in.nextDouble();
    double weight=in.nextDouble();
    Student student=new OOBIM().new Student(number,name,height,weight);

    return student;
}

```

Step2.运行结果如下



```

Problems @ Javadoc Declaration Console
<terminated> OOBIM [Java Application] D:\Program Files\Java
please input the number of students:
3
please input the ID,name,height and weight:
123 heic 1.7 60
111 acb 1.6 50
333 adag 1.88 70

```

3) 在 OOBMI 类中增加 isExists(String id) 函数, 判断该学生是否已经输入 (是否在 students 中), 函数返回值为 boolean 类型, 如果已经存在, 返回 false ; 否则, 返回 true。

Step1.增加 isExists 函数如下 :

```
public static boolean isExists(String id){
    for(Iterator<Student> iter = students.iterator();iter.hasNext();){
        Student temp=iter.next();
        if(temp.number==null)continue;
        if(id.compareTo(temp.number)==0)return true;
    }
    return false;
}
```

4) 编写 inputStudents 函数, 不要求用户输入学生人数, 通过询问用户是否继续输入来决定是否继续输入学生, 并判断输入的学生是否已经存在(调用 isExists 函数), 如果不存在则将输入的学生对象保存到 students 中。

Step1.编写 inputStudents 函数如下 :

```
public static void inputStudents(Scanner in){
    String str=null;
    do{
        System.out.println("please input the ID,name,height and weight:");
        String number=in.next();
        String name=in.next();
        double height=in.nextDouble();
        double weight=in.nextDouble();
        if(isExists(number)){
            System.out.println("error!");
            continue;
        }
        students.add(new Student(number,name,height,weight));
        System.out.println("Do you want to creat next random student?");
        str=in.next();
    }while(str.compareTo("Y")==0||str.compareTo("y")==0);
}
```

Step2.实验运行结果如下 :

```

<terminated> OOBIM [Java Application] D:\Program Files\J
please input the ID,name,height and weight:
111 ada 177 60
Do you want to creat next random student?
y
please input the ID,name,height and weight:
111 ad 177 60
error!
please input the ID,name,height and weight:
112 444 166 58
Do you want to creat next random student?
n
|

```

5) 在 OOBMI 中增加 genStudents 函数, 参数为整数, 能够随机生成指定数量的名学生对象, 并保存到 students 中。

Step1. 增加 genStudents 函数如下

```

public static void genStudents(int n){
    Random ron=new Random();

    char Char[]={
        'A','B','C','D','E','F','G',
        'H','I','J','K','L','M','N',
        'O','P','Q','R','S','T',
        'U','V','W','X','Y','Z',
        'a','b','c','d','e','f','g',
        'h','i','j','k','l','m','n',
        'o','p','q','r','s','t',
        'u','v','w','x','y','z'};

    for(int i=0;i<n;i++){
        String number=String.valueOf(100+ron.nextInt()%100);
        if(isExists(number)){
            i--;
        }
        else {
            String name="";
            for(int j=0;j<5;j++){name+=Char[ron.nextInt(46)];}
            double height=ron.nextDouble()+1;
            double weight=ron.nextDouble()*60+40;
            Student student=new Student(number,name,height,weight);
            students.add(student);
        }
    }
    System.out.println("Success!");
}

```

6) 修改 OOBMI 中的 genStudents 和 inputStudent 函数，调用 isExists 函数避免输入或生成重复的学号的学生。

Step1.修改 genStudents 函数如下

```
public static void genStudents(int n){
    Random ron=new Random();

    char Char[]={
        'A','B','C','D','E','F','G',
        'H','I','J','K','L','M','N',
        'O','P','Q','R','S','T',
        'U','V','W','X','Y','Z',
        'a','b','c','d','e','f','g',
        'h','i','j','k','l','m','n',
        'o','p','q','r','s','t',
        'u','v','w','x','y','z'};

    for(int i=0;i<n;i++){
        String number=String.valueOf(100+ron.nextInt()%100);
        if(isExists(number)){
            i--;
        }
        else {
            String name="";
            for(int j=0;j<5;j++){name+=Char[ron.nextInt(46)];}
            double height=ron.nextDouble()+1;
            double weight=ron.nextDouble()*60+40;
            Student student=new Student(number,name,height,weight);
            students.add(student);
        }
    }
    System.out.println("Success!");
}
```

Step2.修改 inputStudents 函数如下

```
public static Student inputStudent(Scanner in){
    String number=null,name=null;
    double height=0.0,weight=0.0;
    do{
        System.out.println("please input the ID,name,height and weight:");
        number=in.next();
        name=in.next();
        height=in.nextDouble();
        weight=in.nextDouble();
    }while(!isExists(number));

    Student student=new Student(number,name,height,weight);

    return student;
}
```

7) 在 OOBMI 中增加 4 个成员函数，分别统计 bmi 的均值、中值、众数、方差等统计信息(bmi 值保留两位小数存储)。

Step1. 增加各个函数如下（另加了排序函数用于查中位数）

```
public static double AverofBMI(){
    double sum=0;
    for(int i=0;i<students.length;i++)sum+=students[i].bmi;

    return sum/students.length;
}

public static double ModofBMI(){
    int hash[]=new int[10000];
    for(int i=0;i<students.length;i++)hash[(int)students[i].bmi*100-1]++;
    int max=0;
    for(int i=1;i<hash.length;i++){
        if(hash[i]>hash[max])max=i;
    }
    return max/100.0;
}

public static double MedianofBMI(){
    SortbyBMI();
    if(students.length%2==0)
        return students[students.length/2].bmi+students[students.length/2-1].bmi;

    return students[students.length/2].bmi;
}

public static double VarianceofBMI(){
    double sum=0;
    double aver=AverofBMI();
    for(int i=0;i<students.length;i++)
        sum+=(students[i].bmi-aver)*(students[i].bmi-aver);

    return sum/students.length;
}

public static void SortbyBMI(){
    for(int i=0;i<students.length;i++){
        int min=i;
        for(int j=i+1;j<students.length;j++){
            if(students[i].bmi<students[min].bmi)min=j;
        }
        if(min!=i){
            Student temp=students[i];
            students[i]=students[min];
            students[min]=temp;
        }
    }
}
```

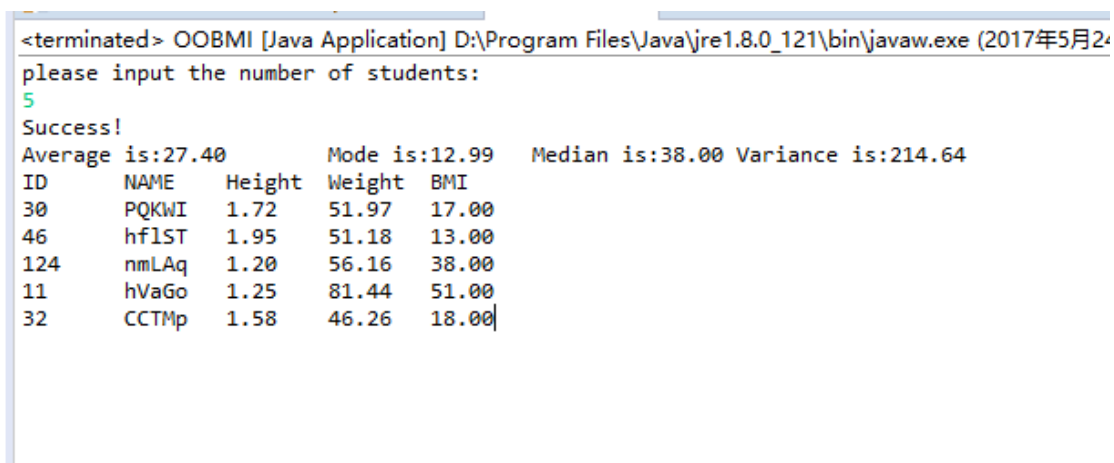
8) 在 OOBMI 中增加 printStatics 函数，该函数可以打印所有学生基本信息（利用增强的 for 语句），以及统计结果信息。打印时，每个学生的信息打印为

一行，为了清晰，学号、姓名、身高、体重和计算后的 bmi 值之间用制表符(\t) 隔开；打印完学生信息后，打印 BMI 统计信息。思考：如何将统计信息传递给 printStatics 函数？

Step1. 增加 printStatics 函数如下

```
public static void printStatics(double aver,double mod,double median,double variance){
    System.out.printf("Average is:%.2f\tMode is:%.2f\tMedian is:%.2f\tVariance is:%.2f\n",
        ,aver,mod,median,variance);
    System.out.printf("ID\tNAME\tHeight\tWeight\tBMI\n");
    for(int i=0;i<students.length;i++)
        System.out.printf("%s\t%s\t%.2f\t%.2f\t%.2f\n",
            students[i].number,students[i].name,
            students[i].height,students[i].weight,students[i].bmi);
}
```

Step2. 运行结果如下



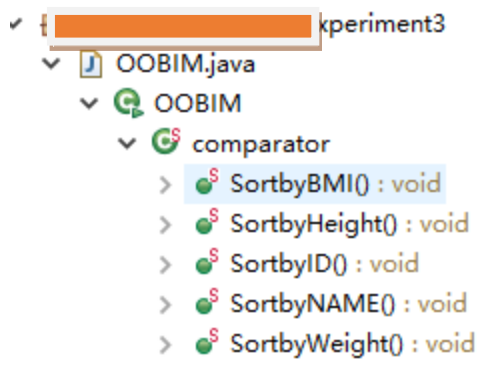
```
<terminated> OOBMI [Java Application] D:\Program Files\Java\jre1.8.0_121\bin\javaw.exe (2017年5月24
please input the number of students:
5
Success!
Average is:27.40      Mode is:12.99   Median is:38.00  Variance is:214.64
ID      NAME      Height  Weight  BMI
30      PQKWI    1.72    51.97   17.00
46      hf1ST    1.95    51.18   13.00
124     nmLAq    1.20    56.16   38.00
11      hVaGo    1.25    81.44   51.00
32      CCTMp    1.58    46.26   18.00
```

Step3. 思考：传参给 printStatics 函数即可

9) 在 OOBMI 类中增加五个 comparator 子类(内部类)，能够利用

Collections.sort 函数对学生分别按照学号、姓名、身高、体重、BMI 进行排序。

Step1. 分析题意，增加五个 comparator 子函数即可实现功能，故无需添加子类，冗增无用代码



Step2. 对应代码如下：

```
public static void SortbyBMI(){
    Collections.sort(students, new Comparator<Student>() {
        @Override
        public int compare(Student o1, Student o2) {
            //排序
            return (int)(100*o1.bmi-100*o2.bmi);
        }
    });
}

public static void SortbyID(){
    Collections.sort(students, new Comparator<Student>() {
        @Override
        public int compare(Student o1, Student o2) {
            //排序
            return o1.number.compareTo(o2.number);
        }
    });
}

public static void SortbyNAME(){
    Collections.sort(students, new Comparator<Student>() {
        @Override
        public int compare(Student o1, Student o2) {
            //排序
            return o1.name.compareTo(o2.name);
        }
    });
}

public static void SortbyHeight(){
    Collections.sort(students, new Comparator<Student>() {
        @Override
        public int compare(Student o1, Student o2) {
            //排序
            return (int)(100*o1.height-100*o2.height);
        }
    });
}
```

```

public static void SortbyWeight(){
    Collections.sort(students, new Comparator<Student>() {
        @Override
        public int compare(Student o1, Student o2) {
            //升序
            return (int)(100*o1.weight-100*o2.weight);
        }
    });
}

```

10)在 OOBMI 类中增加方法 saveFile(ArrayList<Student> students, String filename), 该方法可以将学生信息 students 写入到指定的文本文件中, 每一行写入一个学生。

Step1.增加 saveFile 方法如下：

```

public static void saveFile(ArrayList<Student> students2, String filename){
    File file = new File("."+File.separator+filename);
    if(!file.exists()) {
        try {
            file.createNewFile();
            System.out.println(file.getName() + " 创建成功");
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
    FileWriter writer;
    try {
        writer = new FileWriter(filename, false);
        writer.write(String.format("ID\tNAME\tHEIGHT\tWEIGHT\tBMI\r\n"));
        for(Student st: students2){
            writer.write(String.format("%s\t%s\t%.2f\t%.2f\t%.2f\r\n",
                st.number,st.name,st.height,st.weight,st.bmi));
        }
        writer.close();
    } catch (IOException e) {
        e.printStackTrace();
    }
    System.out.println("SAVE SUCCESS!");
}

```

Step2.运行测试结果如下：

```

<terminated> FileBMI [Java Application] D:\Program
do you want:
1生成学生信息生成及文件保存:
2学生查询、增加、删除及修改
3计算并显示统计结果1
please input the ID
111
Success!
Do you want to creat next random student?
Y
please input the ID
123
Success!
Do you want to creat next random student?
Y
please input the ID
133
Success!
Do you want to creat next random student?
N
student.txt 创建成功
SAVE SUCCESS!

```

Step3.输出文件如下：

student.txt - 记事本

文件(F)	编辑(E)	格式(O)	查看(V)	帮助(H)
ID	NAME	HEIGHT	WEIGHT	BMI
111	DkVrD	1.32	45.88	26.00
123	GSsGb	1.34	77.37	42.00
133	IkWEq	1.70	86.10	29.00

11) 在 OOBMI 类中增加方法 `ArrayList<Student> readFile(String filename)` 读文件中的数据到学生 `ArrayList` 中，并返回该 `ArrayList`。

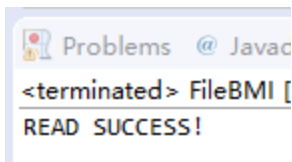
Step1.增加造 `readFile` 方法如下：

```

public static ArrayList<Student> readFile(String filename){
    ArrayList<Student> buff=new ArrayList<Student>();
    File file = new File("."+File.separator+filename);
    if(file.exists()) {
        try {
            BufferedReader reader =new BufferedReader(new FileReader(file));
            String tempString = null;
            while ((tempString = reader.readLine()) != null) {
                String[] a= tempString.split(" ");
                Student st= new FileBMI().new Student(a[0],a[1],
                    Double.parseDouble(a[2]),Double.parseDouble(a[3]));
                buff.add(st);
            }
            reader.close();
            System.out.println("READ SUCCESS!");
        } catch (FileNotFoundException e) {
            e.printStackTrace();
        } catch (NumberFormatException e) {
            e.printStackTrace();
        } catch (IOException e) {
            e.printStackTrace();
        }
    } else {
        System.out.println("the file have noe been found!!");
    }
    return buff;
}

```

Step2.运行测试结果如下：



12) 在 OOBMI 类中增加 findStudent、modifyStudent、deleteStudent 三个函数。findStudent 可根据学号查询学生信息并显示，然后提示用户是否修改或删除学生。如果用户选择在增加，调用 inputStudent 函数增加学生；如果用户选择修改，调用 modifyStudent 函数修改学生信息，注意学号不能修改；如果用户选择删除学生，则调用 deleteStudent 函数删除该学生。注意，所有增删改操作完成后，都应重新保存到文件。

Step1.增加造 findStudent 方法如下（本人认为 modifyStudent、deleteStudent 函数可以用 ArrayList 类中的方法更轻易实现，故无需赘添函数，只需对 inputStudent 稍作修改）：

```
public static void findStudent(ArrayList<Student> stu,Scanner in){
    System.out.println("please input the id:");
    String id=in.next();

    for(int i=0;i<students.size();i++){

        if(id.compareTo(students.get(i).number)==0){
            System.out.printf("The ID has been found,do you want:\n"
                + "1 modify\n"
                + "2 delete\n");
            switch(in.nextInt()){
                case 1:students.set(i,inputStudent(in));break;
                case 2:students.remove(i);break;
                default:System.out.println("ERROR CHIOSE");break;
            }
            return;
        }
    }
    System.out.printf("The ID has not been found,do you want creat?input 1 to do it\n");
    if(in.nextInt()==1)stu.add(inputStudent(in));
    saveFile(students, "student.txt");
}
```

13) 在 OOBMI 的 main 函数中，调用上述函数，完成输入、统计结果输出的完整过程。

Step1.在主函数中调用各函数：

```
public static void main(String[] agrv){
    genStudents(in.next());

    double aver=AverofBMI(),mod=ModofBMI(),med=MedianofBMI(),var=VarianceofBMI();
    System.out.printf("\nsort by bmi\n");
    printStatics(aver,mod,med,var);

    System.out.printf("\nsort by ID\n");
    comparator.SortbyID();
    printStatics(aver,mod,med,var);

    System.out.printf("\nsort by name\n");
    comparator.SortbyNAME();
    printStatics(aver,mod,med,var);

    System.out.printf("\nsort by height\n");
    comparator.SortbyHeight();
    printStatics(aver,mod,med,var);
}
```

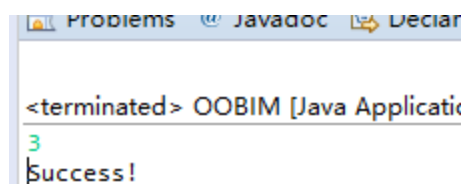
```

        System.out.printf("\nsort by weight\n");
        comparator.SortbyWeight();
        printStatics(aver,mod,med,var);

        in.close();
    }
}

```

Step2.运行并按照引导生成 3 组随机学生信息：



Step3.所有的排序方式的运行结果如下：

```

sort by bmi
Average is:52.33      Mode is:35.99      Median is:53.00 Variance is:170.89
ID   NAME   Height Weight  BMI
113  LUodU   1.33   65.73   36.00
112  bqWLd   1.13   68.98   53.00
111  iFJHI   1.11   85.35   68.00

```

```

sort by ID
Average is:52.33      Mode is:35.99      Median is:53.00 Variance is:170.89
ID   NAME   Height Weight  BMI
111  iFJHI   1.11   85.35   68.00
112  bqWLd   1.13   68.98   53.00
113  LUodU   1.33   65.73   36.00

```

```

sort by name
Average is:52.33      Mode is:35.99      Median is:53.00 Variance is:170.89
ID   NAME   Height Weight  BMI
113  LUodU   1.33   65.73   36.00
112  bqWLd   1.13   68.98   53.00
111  iFJHI   1.11   85.35   68.00

```

```

sort by height
Average is:52.33      Mode is:35.99      Median is:53.00 Variance is:170.89
ID   NAME   Height Weight  BMI
111  iFJHI   1.11   85.35   68.00
112  bqWLd   1.13   68.98   53.00
113  LUodU   1.33   65.73   36.00

```

```

sort by weight
Average is:52.33      Mode is:35.99      Median is:53.00 Variance is:170.89
ID   NAME   Height Weight  BMI
113  LUodU   1.33   65.73   36.00
112  bqWLd   1.13   68.98   53.00
111  iFJHI   1.11   85.35   68.00

```