



HARBIN INSTITUTE OF TECHNOLOGY

哈尔滨工业大学

Java 程序设计实验报告

学号: XXXXXXXXXX

姓名: XXXXXX

专业: XXXXXXXXXX

班级: XXXXXXXXXX

实验五：集合对象程序设计

一、实验目的

- 1) 了解集合的概念和基本接口
- 2) 掌握增强 for 循环语句
- 3) 掌握范型的应用
- 4) 掌握基本集合对象 ArrayList 的应用

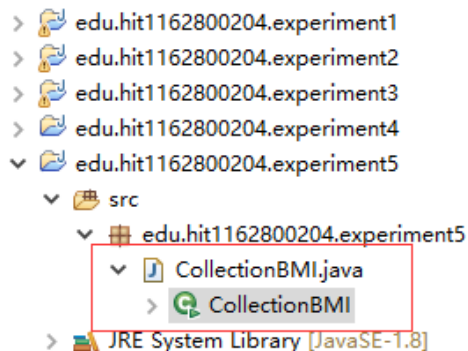
二、实验内容

- 1) 将 OOBMI 改造为 CollectionBMI 类；将 CollectionBMI 中的 Student[] students 改造为 **ArrayList<Student> students**;
- 2) 改造 genStudents 函数，将随机生成的学生对象保存到 **ArrayList<Student> students** 中。
- 3) 改造 inputStudents 函数和相关的 isExists 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，如果不存在则将输入的学生对象保存到 **ArrayList<Student> students** 中。
- 4) 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
- 5) 改造 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。
- 6) 在 CollectionBMI 的 main 函数中，调用上述函数，完成学生信息生成、按不同属性排序、结果输出的完整过程。

三、实验步骤

- 1) 将 OOBMI 改造为 CollectionBMI 类；将 CollectionBMI 中的 Student[] students 改造为 **ArrayList<Student> students**；

Step1.改造该类如下



Step2.更改 students 的数据类型如下：

```
public class CollectionBMI {  
  
    static List<Student> students=new ArrayList<Student>();  
    public CollectionBMI(){}
```

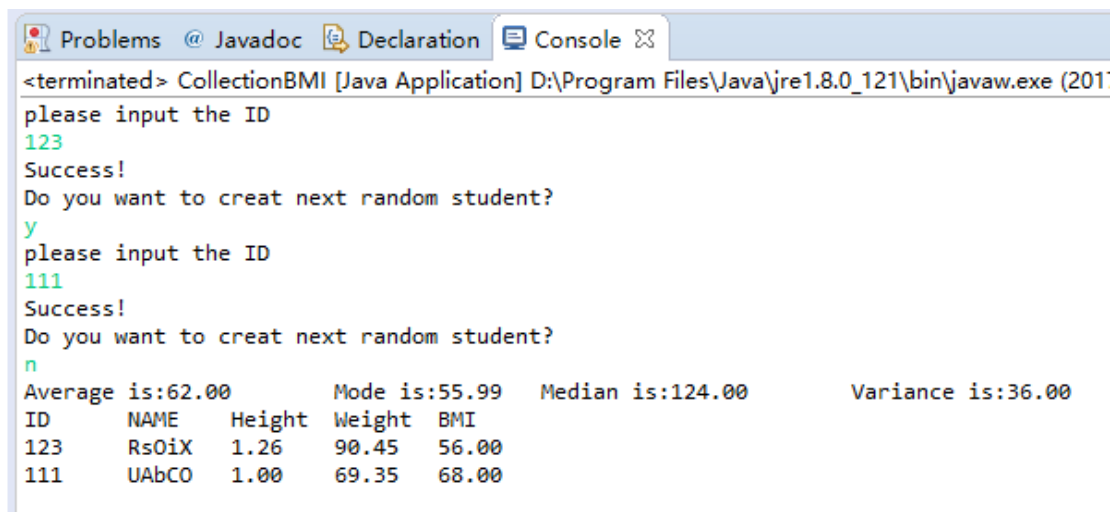
2) 改造 genStudents 函数，将随机生成的学生对象保存到

ArrayList<Student> students 中。

Step1.改造 genStudents 函数如下

```
public static void genStudents(String id){  
  
    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'};  
  
    if(isExists(id))return;  
    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;  
    students.add(new CollectionBMI().new Student(id,name,height,weight));  
  
    System.out.println("Success!");  
}
```

Step2.运行 genStudents 函数添加数据到 students 中结果如下：



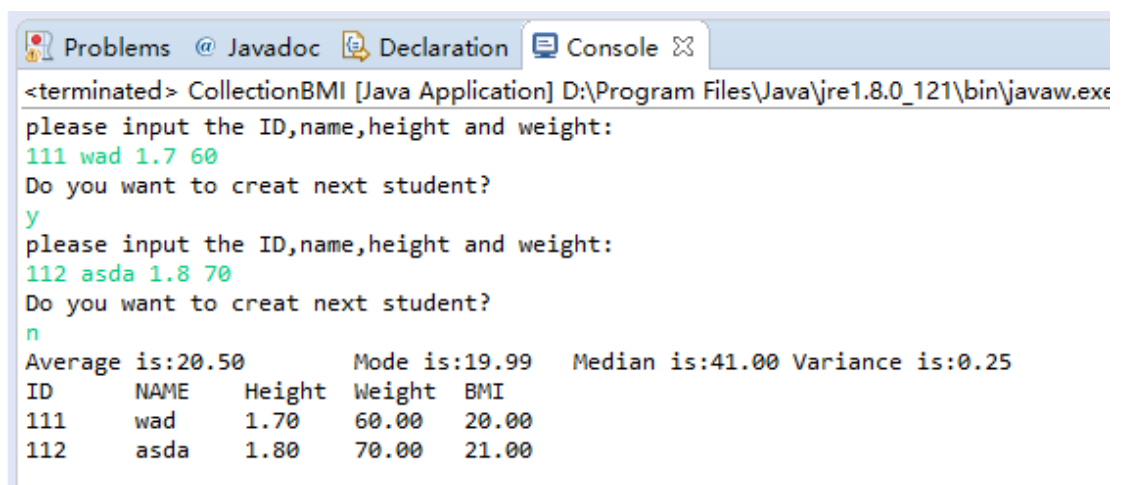
```
<terminated> CollectionBMI [Java Application] D:\Program Files\Java\jre1.8.0_121\bin\javaw.exe (201  
please input the ID  
123  
Success!  
Do you want to creat next random student?  
y  
please input the ID  
111  
Success!  
Do you want to creat next random student?  
n  
Average is:62.00      Mode is:55.99      Median is:124.00      Variance is:36.00  
ID      NAME      Height      Weight      BMI  
123      RsOiX      1.26      90.45      56.00  
111      UAbCO      1.00      69.35      68.00
```

3) 改造 inputStudents 函数和相关的 isExists 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，如果不存在则将输入的学生对象保存到 ArrayList<Student> students 中。

Step1.改造 inputStudents 函数和相关的 isExists 函数如下：

```
public static Student inputStudent(Scanner in){  
    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();  
    Student student=new CollectionBMI().new Student(number,name,height,weight);  
  
    return student;  
}  
  
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;  
}
```

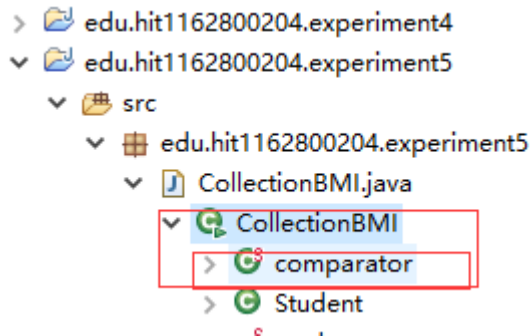
Step2.实验运行结果如下：



```
<terminated> CollectionBMI [Java Application] D:\Program Files\Java\jre1.8.0_121\bin\javaw.exe  
please input the ID,name,height and weight:  
111 wad 1.7 60  
Do you want to creat next student?  
y  
please input the ID,name,height and weight:  
112 asda 1.8 70  
Do you want to creat next student?  
n  
Average is:20.50      Mode is:19.99      Median is:41.00 Variance is:0.25  
ID      NAME      Height  Weight  BMI  
111     wad      1.70   60.00   20.00  
112     asda     1.80   70.00   21.00
```

4) 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.增加内部类 comparator 如下：



Step2.增加各函数如下：

SortbyBMI()

```
public static void SortbyBMI(){
    for(int i=0;i<students.size();i++){
        int min=i;
        for(int j=i+1;j<students.size();j++){
            if(students.get(j).bmi<students.get(min).bmi)min=j;
        }
        if(min!=i){
            Student temp=students.get(i);
            students.set(i, students.get(min));
            students.set(min,temp);
        }
    }
}
```

SortbyID ()

```
public static void SortbyID(){
    for(int i=0;i<students.size();i++){
        int min=i;
        for(int j=i+1;j<students.size();j++){
            if(students.get(j).number.compareTo(students.get(min).number)<0)min=j;
        }
        if(min!=i){
            Student temp=students.get(i);
            students.set(i, students.get(min));
            students.set(min,temp);
        }
    }
}
```

SortbyNAME()

```
public static void SortbyNAME(){
    for(int i=0;i<students.size();i++){
        int min=i;
        for(int j=i+1;j<students.size();j++){
            if(students.get(j).name.compareTo(students.get(min).name)<0)min=j;
        }
        if(min!=i){
            Student temp=students.get(i);
            students.set(i, students.get(min));
            students.set(min,temp);
        }
    }
}
```

SortbyHeight()

```
public static void SortbyHeight(){
    for(int i=0;i<students.size();i++){
        int min=i;
        for(int j=i+1;j<students.size();j++){
            if(students.get(j).height<students.get(min).height)min=j;
        }
        if(min!=i){
            Student temp=students.get(i);
            students.set(i, students.get(min));
            students.set(min,temp);
        }
    }
}
```

SortbyWeight()

```
public static void SortbyWeight(){
    for(int i=0;i<students.size();i++){
        int min=i;
        for(int j=i+1;j<students.size();j++){
            if(students.get(j).weight<students.get(min).weight)min=j;
        }
        if(min!=i){
            Student temp=students.get(i);
            students.set(i, students.get(min));
            students.set(min,temp);
        }
    }
}
```

5) 改造 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。

Step1.改造 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(Iterator<Student> iter = students.iterator();iter.hasNext();){
        Student temp=iter.next();
        System.out.printf("%s\t%s\t%.2f\t%.2f\t%.2f\n",
            temp.number,temp.name,
            temp.height,temp.weight,temp.bmi);
    }
}
```

6) 在 CollectionBMI 的 main 函数中，调用上述函数，完成学生信息生成、按不同属性排序、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){
    new CollectionBMI();
    Scanner in=new Scanner(System.in);
    String str;
    |
    do{
        System.out.println("please input the ID");
        genStudents(in.next());
        System.out.println("Do you want to creat next random student?");
        //students.add(inputStudent(in));
        //System.out.println("Do you want to creat next student?");
        str=in.next();
    }while(str.compareTo("Y")==0||str.compareTo("y")==0);

    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：

```

Problems @ Javadoc Declaration Console
<terminated> CollectionBMI [Java Application] D:\Program Files\Java\jre1.8.0_1
please input the ID
111
Success!
Do you want to creat next random student?
y
please input the ID
112
Success!
Do you want to creat next random student?
y
please input the ID
113
Success!
Do you want to creat next random student?
n

```

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
```

至此，实验圆满结束！！