Java课程实验报告

专业：软件工程 姓名：王佳颉 学号：1412641 年级：大二

完成日期：2015/12/5 授课教师：刘明铭

1.题目

（1）将5个学生姓名写入一个Set<Student>集合中，学生具有学号，姓名，成绩列表等属性，学生以学号区分，按照平均分、最高分、最低分对学生进行打印；

（2）编写泛型传递参数的程序求集合a中包含元素个数以及a、b元素的对应关系。

2.要求

通过对学生的排序以及泛型传递程序的编写掌握java中Set、List、Map集合的各种方法操作。

3.程序实现

（1）Student.java：

a.创建一个Student类，成员变量包括姓名name、学号number、成绩数组score[]、最高成绩max、最低成绩min、平均成绩average；构造器中随机产生10个1-100分的成绩放在score[]数组中，实现了求最高分、最低分、平均分；重写了toString方法输出各种成员变量；重写了equals和hashCode方法以学号区分对象。

**public** **class** Student {

**public** String name;

**public** **int** number;

**public** **int**[] score=**new** **int**[10];

**public** **double** average;

**public** **int** max;

**public** **int** min;

**public** Student(String name,**int** number){

**this**.name=name;

**this**.number=number;

**double** sum=0;

**for**(**int** i=0;i<10;i++){

score[i]=(**int**)(Math.*random*()\*100+1);

}

max=min=score[0];

**for**(**int** i=1;i<10;i++){

sum=sum+score[i];

**if**(max<score[i]){

max=score[i];

}

**if**(min>score[i]){

min=score[i];

}

}

average=sum/10;

}

**public** String toString(){

**return** "学号:"+number+" 姓名:"+name+" 平均分:"+average+" 最高分:"+max+" 最低分:"+min;

}

**public** **boolean** equals(Object obj){

**if**(**this**==obj){

**return** **true**;

}

**if**(obj!=**null**&&obj.getClass()==Student.**class**){

Student s=(Student)obj;

**return** s.number==**this**.number;

}

**return** **false**;

}

**public** **int** hashCode(){

**return** **this**.number;

}

b.在main方法中创建了Student类的5个实例并通过构造器进行初始化，由于Lambda表达式只能在java8中实现，因此把书中TreeSet定制排序的comparator转换为一般表达式来按照不同的标准（最高分、最低分、平均分）对学生进行排名并且打印出来。下面是按照平均分进行排名：

Student student1=**new** Student("张三",1412641);

Student student2=**new** Student("李四",1412642);

Student student3=**new** Student("王五",1412643);

Student student4=**new** Student("钱六",1412644);

Student student5=**new** Student("孙七",1412645);

TreeSet<Student> ts1=**new** TreeSet<Student>(**new** Comparator<Student>(){

**public** **int** compare(Student m1,Student m2){

**return** m1.average>m2.average?-1

:m1.average<m2.average?1:0;

}

});

ts1.add(student1);

ts1.add(student2);

ts1.add(student3);

ts1.add(student4);

ts1.add(student5);

System.*out*.println("平均分排名：");

System.*out*.println(ts1);

（2）CollectionTest.java：

a.将List集合转换为HashSet集合便可以将重复的元素去掉，然后输出HashSet集合便输出了List集合的元素种类。

**static** <T> **void** Amount(List<T> a){

Set<T> aset=**new** HashSet<T>(a);

System.*out*.println(a+"中包含"+aset+aset.size()+"种不同的数");

}

b.先将List集合a转换为HashSet：aset得到List集合a的种类，然后通过Iterator对aset进行遍历，定义迭代器为it，如果有值while(it.hasNext())：定义一个HashSet为bset，将it.next()赋值给X，然后对List集合a进行循环，如果X对应的索引存在（java中List集合中不存在元素X，则indexOf的返回值为-1，所以a.indexOf(X)!=-1），则将List集合b在X在a中对应的索引处的值加入HashSet集合bst中，并将X在a中的位置的值重新复制为空null用于X在下一次循环中获得X在a中的后面的索引值。循环结束后输出a中X对应b中的值就是bset的元素。然后对a中下一个元素进行遍历。

**static** <T> **void** Corresponding(List<T> a,List<T> b){

Set<T> aset=**new** HashSet<T>(a);

Iterator<T> it=aset.iterator();

**while**(it.hasNext()){

Set<T> bset=**new** HashSet<T>();

T X=(T)it.next();

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

**if**(a.indexOf(X)!=-1){

T Y=b.get(a.indexOf(X));

a.set(a.indexOf(X), **null**);

bset.add(Y);

}

}

System.*out*.println("a中的"+X+"对应于b中的"+bset);

}

}

c.在main方法中通过数组构建List集合a、b然后进行上述操作即可。

List<Integer> a=**new** ArrayList<Integer>();

List<Integer> b=**new** ArrayList<Integer>();

**int**[]A={2,1,2,2};

**int**[]B={1,0,0,1};

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

a.add(A[i]);

}

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

b.add(B[i]);

}

*Amount*(a);

*Corresponding*(a,b);