**集美大学计算机工程学院实验报告**

|  |  |  |
| --- | --- | --- |
| **课程名称**：Java程序设计 | **班级**：软件1912 | **实验成绩**： |
| **指导教师**：李　旺 | **姓名**：游嘉升 |  |
| **实验项目名称**：实验7：集合 | **学号**：201921122050 | **上机实践日期**：11月27日 |
| **实验项目编号**： | **组号**： | **上机实践时间**： 2 学时 |

**一、目的**

掌握常用异常的处理方法；掌握常用集合的基本用法；理解泛型及其应用。

**二、实验内容与设计思想**

1. 使用异常机制处理程序错误

为下面的代码加入适当的异常处理, 使得该程序可以处理输入的字符串不为数字字符串的情况（要求重新输入，最后得到3个从键盘输入的浮点数）。

|  |
| --- |
| public static void main(String[] args) {  double[] x = new double[3];double sum = 0.0;  Scanner inputScan = new Scanner(System.in);  for(int i = 0; i < x.length;i++){  System.out.println("Please input the "+(i+1)+" number:");  String input = inputScan.nextLine();  x[i] = Double.parseDouble(input);  sum += x[i];  }  System.out.println(Arrays.toString(x));  System.out.println("Sum:" + sum); } |

2. List与Iterator接口

(1)编写NameDao接口，包含如下方法：

//将数组中的字符串放入列表，该方法需要使用ListIterator接口实现添加

**public List getNameFromArray(String[] names);**

//在列表中搜索到与name相同的项目，就返回该name所在的序号，找不到返回-1。

**public int searchName(List nameList, String name);**

//根据指定id，从列表中移除指定的项目，如果id指定错误(负数或者超出列表最大值)抛出IllegalArgumentException异常，移除成功返回true

**public boolean removeFromList(List nameList, int id);**

(2)编写NameDaoImpl实现NameDao接口。

(3)编写一个测试类，测试上述代码的功能。

3. Map与HashMap

(1)编写Student类，只有String name与int age两个属性和他们的setter/getter方法，覆盖其hashCode与equals方法。

(2)在Util类中编写方法：List<Student> getRandomStudents(int n)随机生成n个学生的列表。

说明：调用“基础代码”中的Util.getRandomName (int min, int max)获取随机名字，调用public static int getRandomAge(int min, int max)生成随机年龄。

(3)在Util类中编写方法：public static Map<String, Student> makeStudentMap(List<Student>);，将List中每个Student的name作为key, Student对象本身作为value放入Map中，并返回。

(4)在Util类中编写方法：public static String searchName(List<String> nameList, String name);，根据指定的name返回相应的字符串，如果没有找到返回null。

(5)在Util类中编写方法：public static Student searchStudent(Map<String,Student> stuMap, String name);，根据指定的name返回对应的学生，如果没有找到返回null。

(6)在Util类中编写方法：public static List<String> generateNameList(int n);，生成n个字符串的NameList并返回。

(7)在Util类中编写方法：Map<Student> generateStudentMap(int n);，生成包含n个键的的StudentMap并返回。

说明：调用getRandomStudents方法；调用makeStudentMap方法；返回StudentMap。

(8)对比测试。在初始化阶段调用generateNameList方法生成nameList列表，里边包含1000万个字符串。在初始化阶段调用getRandomStudents生成1000万个Student对象，然后调用makeStudentMap方法生成stuMap映射表。

对nameList调用searchName方法搜索指定字符串并统计执行时间。

对stuMap调用searchStudent方法搜索指定字符串对应的学生对象并统计执行时间。

思考：步骤(8)中，哪个方法搜索速度快？为什么？

(9)遍历输出StudentMap前10个的key，前10个value，前10个entry。

**备注：可以设置虚拟机选项，让JVM能够拥有大内存。Eclipse中的设置方法：打开Window->Preferences菜单，左边列选择Java->Installed JREs，在需要设置的JRE上，点击Edit，设置Default VM arguments的值为“-Xms1024m -Xmx4096m”，其中-Xms设置最小内存，-Xmx设置最大内存。**

4. Collections

(1)改写Student类，增加int grade属性。

(2)随机生成100个Student对象放入students数组。

(3)将students数组转换成studentList(List类型)。

(4)使用Collections.sort对student的成绩进行排序。

(5)使用Collections的min和max方法找到成绩最高的和成绩最低学生。

(6)使用Collections的binarySearch查找成绩等于60的学生。

(7)编写方法从studentList找出grade小于60的学生，放入otherList。

(8)使用List的removeAll方法，从studentList中移除otherList。

(9)使用Collections的shuffle方法，打乱studentList中学生排列的顺序。

(10)将studentList转换为数组。

**三、实验使用环境**

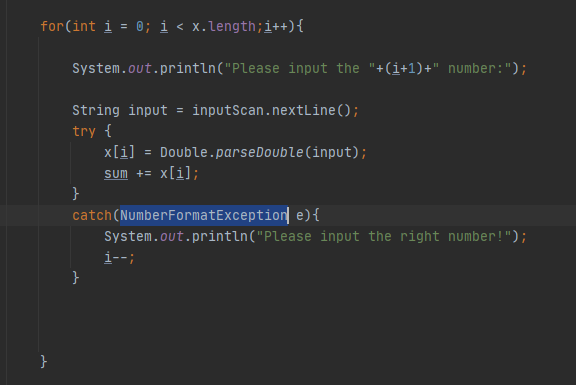
Java version "13.0.2"

IntelliJ IDEA 2020.2.2 x64

**四、实验步骤和调试过程**

1. 使用异常机制处理程序错误

1. 设计思想：使用捕获异常机制，当输入的字符串被转换为double类型时，若无法转换，Double.parseDouble()方法会抛出NumberFormatException的异常，捕获异常并要求重新输入。
2. 关键代码：



1. 测试数据：

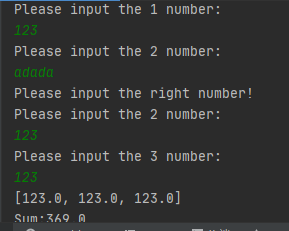
123

adada

123

123

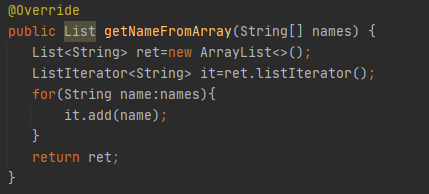
1. 测试结果：



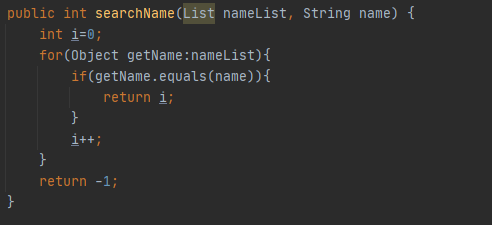
2. List与Iterator接口

1. 设计思想：创建一个NameDaolmpl类有NameDao接口，实现getNameFromArray，searchName和removeFromList方法。getNameFromArray遍历Array迭代器添加入List，searchName遍历List寻找，removeFromList使用迭代器，找到remove，未找到抛出IllegalArgumentException异常。
2. 关键代码：

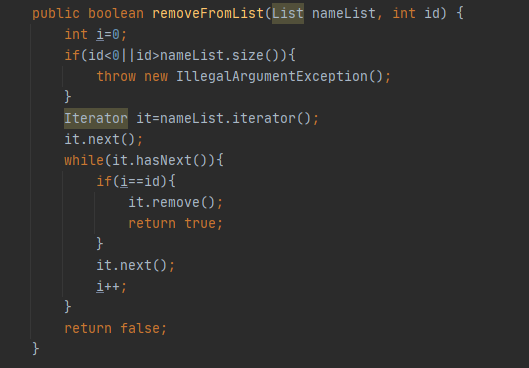
getNameFromArray实现



searchName实现



removeFromList实现



1. 测试数据：

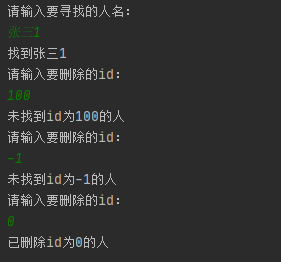
张三1

100

-1

0

1. 测试结果：



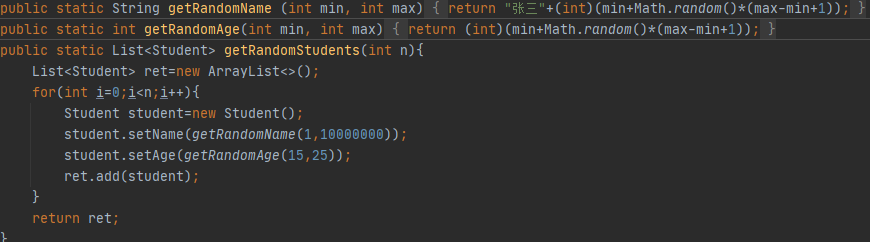
3. Map与HashMap

1. 设计思想：创建Student类，具有name和age属性，创建Util类具有多种方法。getRandomName和getRandomAge方法通过Math.random方法生成随机姓名与年龄,

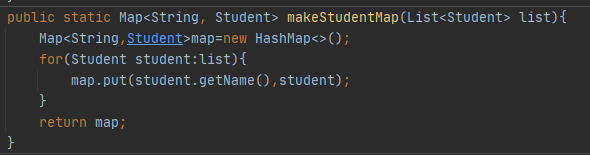
getRandomStudents通过调用getRandomName和getRandomAge方法实现生成学生List，makeStudentMap方法通过调用getRandomStudents和getRandomAge生成学生Map，searchName和searchStudent通过遍历List和Map的映射实现对名字和学生的查找，generateNameList和generateStudentMap通过循环调用getRandomStudents生成List和makeStudentMap生成Map。

1. 关键代码：

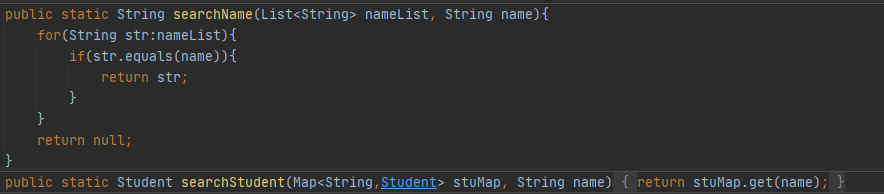
getRandomStudents，getRandomAge，getRandomStudents实现



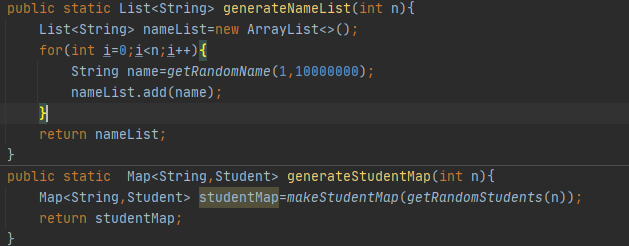
makeStudentMap实现



searchName和searchStudent实现



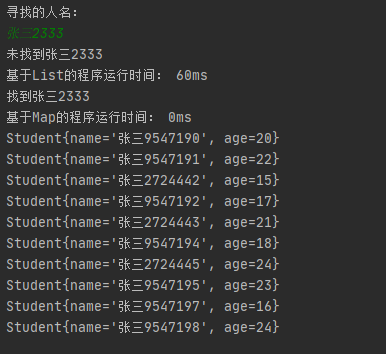
generateNameList和generateStudentMap



1. 测试数据：

随机生成的1000万学生数据

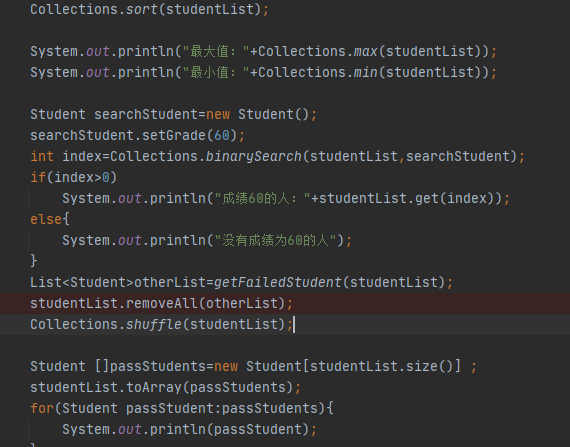
1. 测试结果：



4. Collections

1. 设计思想：创建Student类，具有name，age，grade属性，并重写根据grade属性比较大小的Comparable，在Test类中调用Collection的各种方法
2. 关键代码：

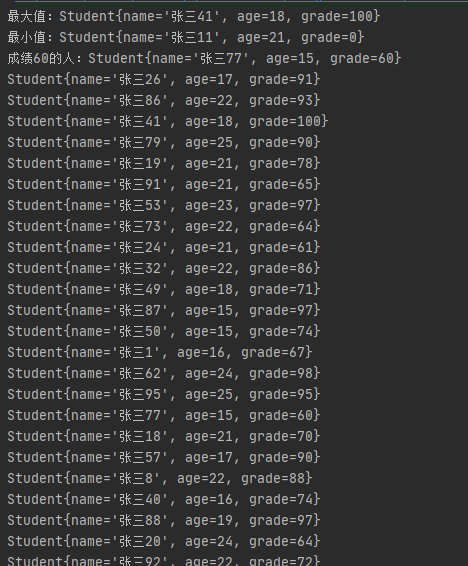
Collection的各种方法调用



1. 测试数据：

随机生成的100个学生数据

1. 测试结果：



**五、实验小结**

1．实验中遇到的问题及解决过程

2．实验中产生的错误及原因分析

①把Arrays.toArray()得到的结果Object[]强制类型装换成别的类数组，引发报错。

解决过程：百度了一下发现，Object[]与其他类数组并不存在方法可以转换，如果需要得到其他类的数组需要遍历Object[]，或者直接调用带有参数的Arrays.toArray()，传入接受的类数组，该方法可以自动转化为对应的类数组。

②调用Colletion.binarySearch()时，传入的key不是搜索的类数组的那一类引发报错。

解决过程：Colletion.binarySearch()的默认情况下是使用对象中的Comparable来作比较，所以只需要传入带有那个想要搜索的属性的空对象就可以根据空对象在类数组里找到那个对象。

3．实验体会和收获。

Java的库的强大，默认集成了大部分的数据结构与方法，在开发下，可以快速的调用适合当前情况下的数据结构与方法，合理利用Java库中的类可以事半功倍。

**六、附录**

[1][美] 凯.S.霍斯特曼（Cay S. Horstmann）.Java核心技术·卷 I（原书第10版）[M].机械工业出版社