**《Java程序设计》上机作业4**

|  |  |
| --- | --- |
| **学号：** | **202103151422** |
| **姓名：** | **温家伟** |
| **班级：** | **大数据分析2101** |
| **日期：** | **2023.11.9** |

**1.** 设计一个程序，使其可以对一个英文单词构成的字符串数组，实现以下功能：

(1) 统计以字母h开头的单词数；

(2) 统计单词中含“or”字符串的单词数；

(3) 统计长度为3的单词数；

(4) 对该字符串数组进行排序（忽略大小写）。

如String s[]={"hello", "world", "welcome", "hi", "hey"}，按单词表顺序排序输出{"hello", "hey", "hi", "welcome", "world"}。

提示：String类中不考虑大小写按字典顺序比较两个字符串的方法为public int compareToIgnoreCase(String str)。也可选用Arrays.sort(s)方法。

|  |
| --- |
| * **问题分析与解答**   这道题可以分为以下几个步骤来解决  统计以字母h开头的单词数：遍历字符串数组，对每个单词进行检查，如果以字母"h"开头，则对计数器加一。  统计单词中含“or”字符串的单词数：同样遍历字符串数组，对每个单词进行检查，如果包含子字符串"or"，则对计数器加一。  统计长度为3的单词数：遍历字符串数组，对每个单词进行检查，如果长度为3，则对计数器加一。  对该字符串数组进行排序（忽略大小写）：使用Arrays.sort(s)方法进行排序。 |
| * **代码实现** |
| * **运行结果** |
| * **思考及总结**   可以将每个功能封装成一个独立的函数，使得程序结构清晰，同时便于维护和扩展。此外，要注意处理字符串大小写不敏感的情况，使用相应的方法进行比较和排序。 |

**2.** 请设计一个equal方法，用来判断两个整型数组（list1和list2）是否严格相同（*strictly identical*）的。如果两个数组的大小，以及其中相同下标i的两个元素list1[i] 和list2[i]都相同，那么返回true，否则返回false。该方法的方法头如下：

public boolean equal(int[] list1, int[] list2)

编写一个程序，提示用户输入两个整型数组的大小和数组中每个元素的值，通过调用程序内equal方法判断，两个数组是否严格相同。输入输出样例如下：

注： 第一个输入值是数组的大小。

|  |
| --- |
| Enter list1: 5 2 5 6 1 6  Enter list2: 5 2 5 6 1 6  Two lists are strictly identical |
|  |
| Enter list1: 5 2 5 6 6 1  Enter list2: 5 2 5 6 1 6  Two lists are not strictly identical |

|  |
| --- |
| * **问题分析**   设计equal方法：编写一个equal方法，接收两个整型数组list1和list2作为参数，通过比较数组的大小以及每个对应位置上的元素是否相同，来判断两个数组是否严格相同。可以使用for循环遍历两个数组，并逐个比较对应位置上的元素值。  编写主程序：在主程序中，首先提示用户输入两个整型数组的大小，然后分别输入每个数组的元素值。接着调用equal方法判断两个数组是否严格相同，根据返回结果输出相应的信息。 |
| * **代码实现** |
| * **运行结果** |
| * **思考及总结**   编写equal方法时，需要注意判断两个数组的大小是否相同，以及对应位置上的元素值是否相同。  在主程序中，要合理处理用户输入的数组大小和元素值，可以使用Scanner类来实现用户输入的获取。  调用equal方法进行比较时，根据返回的布尔值输出相应的判断结果。 |

**3.** 在上一题的基础上，修改equals方法，使其能判断两个整型数组（list1和list2）是否相同（*identical*）的。如果两个数组具有相同的元素内容（即使下标不一致），那么返回true，否则返回false。该方法的方法头如下：

public boolean equal(int[] list1, int[] list2)

编写一个程序，提示用户输入两个整型数组的大小和数组中每个元素的值，通过调用程序内equal方法判断，两个数组是否相同。输入输出样例如下：

注： 第一个输入值是数组的大小。

|  |
| --- |
| Enter list1: 5 2 5 6 6 1  Enter list2: 5 5 2 6 1 6  Two lists are identical |
|  |
| Enter list1: 5 5 5 6 6 1  Enter list2: 5 2 5 6 1 6  Two lists are not identical |

|  |
| --- |
| * **问题分析**   这道题在前一题的基础上只要加上对数组排序的代码即可，即先对数组排序，再依次比较每一个元素。 |
| * **代码实现** |
| * **运行结果** |
| * **思考及总结**   数组的排序方法是Arrays.sort(arr); |

**4.** (*Binary to decimal*) 请设计一个方法，可以将一串二进制数字组成的字符串转换为十进制的整数。方法头如下:

**public static int** binaryToDecimal(**String** binaryString)

例如：二进制字符串10001是十进制的17，因此 **binaryToDecimal("10001")** 方法调用后返回整数17。

注意：（**请不要使用！**）现成的方法 **Integer.parseInt("10001", 2)** 能够实现二进制字符串到十进制的转换，这里需要自己写语句实现。不要调用这个现成的方法。

编写一个测试程序，提示用户输入一个二进制字符串，调用binaryToDecimal方法并输出对应的十进制数值。

|  |
| --- |
| * **问题分析**   二进制转十进制，只需要每一位乘上对应的基数，然后加起来即可。 |
| * **代码实现** |
| * **运行结果** |
| * **思考及总结**   要注意下标不要越界。 |

**5.** 编写一个程序，从键盘输入10个数，最终显示互不相同的数：即同一个数出现多次，则仅显示一次。要求至少使用数组、List、Set 三种方式中的两种进行实现（可以在同一个程序中定义两个方法）。

提示：读入一个数，如果它不是一个新数，则将它存储在数组或集合中，如果该数已经在数组中，则忽略它。输入之后，数组包含的都是不同的数。

程序的运行示例：

|  |
| --- |
| Enter ten numbers: 1 2 3 2 1 6 3 4 5 2  The number of distinct number is 6  The distinct numbers are: 1 2 3 6 4 5 |

|  |
| --- |
| * **问题分析**   List方式：借助提示，读入一个数，在List中，就不继续插入，不在就插入。  Set方式：Set有去重的功能，所以直接输入即可。 |
| * **代码实现** |
| * **运行结果** |
| * **思考及总结**   通过使用数组和List，我们可以按照输入的顺序来存储数字，但需要手动进行重复性检查。  通过使用Set，我们可以自动确保其中不包含重复的元素，在插入时自动进行重复性检查，但无法保证元素的顺序。 |

**6.** 编写程序:

(1) 包含学生类Student，及其子类研究生类GraduateStudent。

* 学生类包含公有成员变量name，credit；研究生类有自己的私有变量postcredit；
* 学生类包括方法getName(), getCredit(), printInfo()，研究生类除继承上述方法外，还有自己的setPostcredit方法，并需重写printInfo()；

(2) 写一个测试类TestStudent，首先创建一个研究生并设置其postcredit，输出对象信息。然后建立一个Student数组，包含三个研究生（GraduateStudent）元素，要求打印输出该学生数组中所有学生的姓名和学分信息。

|  |
| --- |
| * **问题分析**   首先，我们需要创建一个Student类和一个GraduateStudent类。其中，Student类包含公有成员变量name和credit，以及方法getName()、getCredit()和printInfo()；而GraduateStudent类继承自Student类，拥有自己的私有变量postCredit，并且需要添加setPostCredit()方法，并且重写printInfo()方法。  接下来，我们需要创建一个TestStudent测试类。在该类中，我们首先创建一个研究生对象，并设置其postCredit，然后输出该对象信息。接着，我们建立一个Student类型的数组，其中包含三个研究生（GraduateStudent）元素，并要求打印输出该学生数组中所有学生的姓名和学分信息。 |
| * **代码实现**   **Student.java:**    GraduateStudent.java:    TestStudent.java: |
| * **运行结果** |
| * **思考及总结**   重写父类的方法是面向对象编程中的一个重要概念，它允许子类修改父类的行为以适应自己的需求。 |