

---

# 廈門大學



## 信息学院软件工程系

### 《JAVA 程序设计》实验报告

#### 实验 8

姓名：黄勛

学号：22920212204392

学院：信息学院

专业：软件工程

完成时间：2023.4.18

## 一、实验目的及要求

- 熟悉集合类

## 二、实验题目及实现过程

实验环境: Windows 10 21H2、jdk17、utf-8 编码

### 题目一

#### (一) 实验题目

- ◆ 编写一个程序，读取一个名字序列，将他们保存在一个 set 中，然后删除重复的名字。应允许用户输入名字。

#### (二) 实现过程

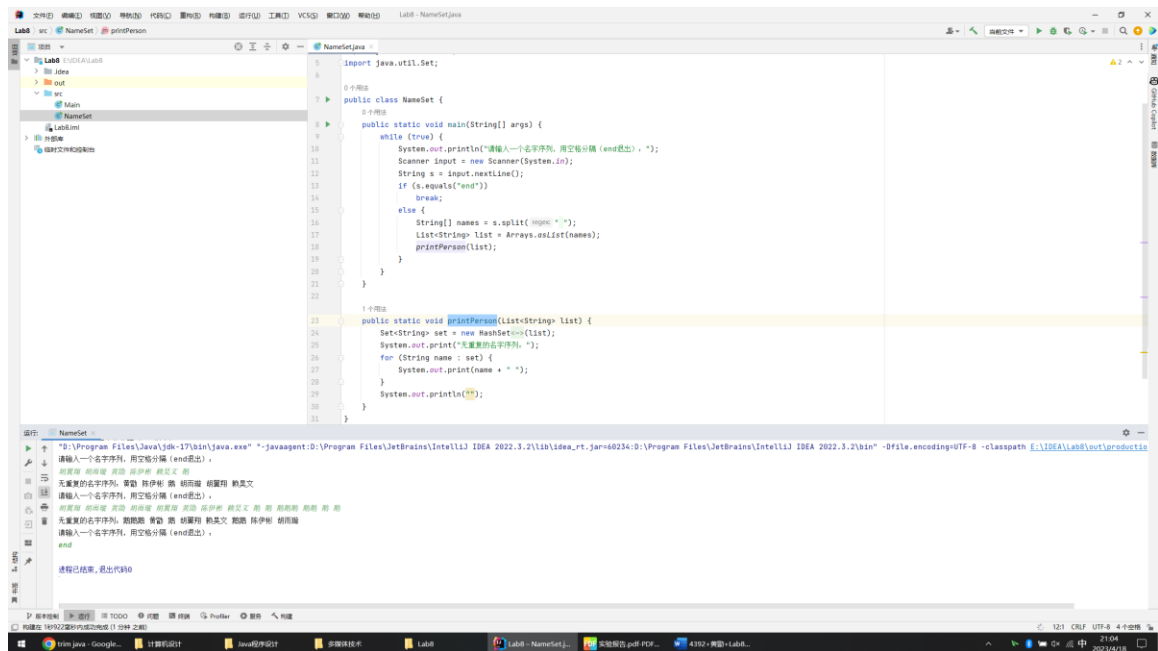
设计类: NameSet, main 方法获取用户的输入

具体思路:

- 1) 将输入的字符串用 split 函数分隔开存入 names 数组
- 2) 用 Array.asList 方法将其转换为 list
- 3) 调用 printPerson 方法，将 list 转换为 set，利用 set 中对象不能重复的特点将其中重复的名字删除然后输出

#### (三) 过程截图

最终结果 (全屏截图)



代码展现：

```

8  public static void main(String[] args) {
9      while (true) {
10         System.out.println("请输入一个名字序列，用空格分隔（end退出）：");
11         Scanner input = new Scanner(System.in);
12         String s = input.nextLine();
13         if (s.equals("end"))
14             break;
15         else {
16             String[] names = s.split(regex: " ");
17             List<String> list = Arrays.asList(names);
18             printPerson(list);
19         }
20     }
21 }
22
23 1 个用法
24 public static void printPerson(List<String> list) {
25     Set<String> set = new HashSet<>(list);
26     System.out.print("无重复的名字序列：");
27     for (String name : set) {
28         System.out.print(name + " ");
29     }
30     System.out.println("");
31 }

```

## 题目二

### (一) 实验题目

- ◆ 编写一个程序，它确定并输出用户输入的某个句子中重复的单词数。不区分字母的大小写。忽略标点符号。

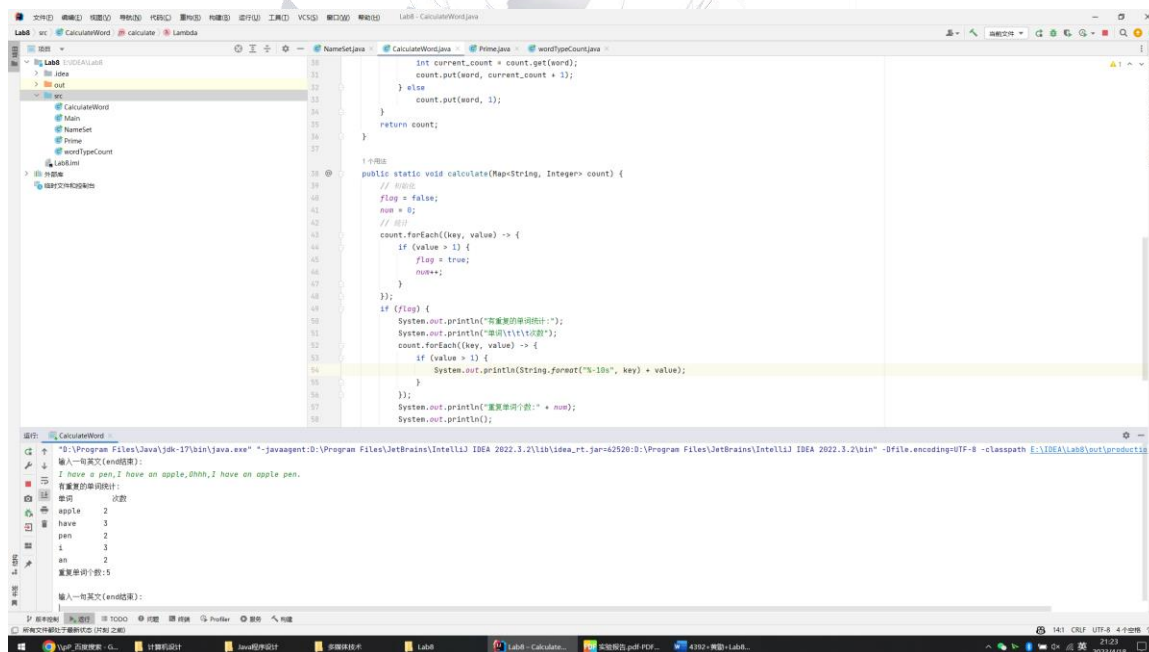
### (二) 实现过程

思路：设计了一个类 CalculateWord，main 方法获取用户输入

- 1) 调用 createMap 方法，用 replaceAll 方法将标点符号替换为空格，用 split 函数将输入的字符串分隔开存入数组
- 2) 遍历单词数组，将单词转换为小写，用 containsKey 方法判断是否存过该单词，若没存过则放入一个（单词，1）的元素，若存过则将其 value 值加一。
- 3) 调用 calculate 方法，遍历 map 若其中某个单词的 value 值，如果 value 值大于 1 则说明是重复的单词，需要将其输出，最后输出 value 值大于 1 的单词数。

### (三) 过程截图

最终结果（全屏截图）



代码展现：

```

5  ▶ public class CalculateWord {
    1 个用法
6      static Scanner scanner = new Scanner(System.in);
    3 个用法
7      static boolean flag = false;
    3 个用法
8      static int num = 0;
9
    0 个用法
10 ▶ public static void main(String[] args) {
11     while (true) {
12         System.out.println("输入一句英文(end结束):");
13         String s = scanner.nextLine().trim();
14         if (s.equals("end"))
15             break;
16         else {
17             Map<String, Integer> map = createMap(s);
18             calculate(map);
19         }
20     }
21 }

```

创建 map:

```

23 @ public static Map<String, Integer> createMap(String s) {
24     Map<String, Integer> count = new HashMap<>();
25     s = s.replaceAll( regex: "\\pP", replacement: " "); // 去除标点符号
26     String[] sentence = s.split( regex: " ");
27     for (String w : sentence) {
28         String word = w.toLowerCase();
29         if (count.containsKey(word)) {
30             int current_count = count.get(word);
31             count.put(word, current_count + 1);
32         } else
33             count.put(word, 1);
34     }
35     return count;
36 }

```

统计单词书并输出:

```

38 @ public static void calculate(Map<String, Integer> count) {
39     // 初始化
40     flag = false;
41     num = 0;
42     // 统计
43     count.forEach((key, value) -> {
44         if (value > 1) {
45             flag = true;
46             num++;
47         }
48     });
49     if (flag) {
50         System.out.println("有重复的单词统计:");
51         System.out.println("单词\t\t\t\t\t次数");
52         count.forEach((key, value) -> {
53             if (value > 1) {
54                 System.out.println(String.format("%-10s", key) + value);
55             }
56         });
57         System.out.println("重复单词个数:" + num);
58         System.out.println();
59     } else
60         System.out.println("无重复单词");
61     }
62 }

```

### 题目三

#### (一) 实验题目

- ◆ 修改 wordTypeCount.java，使其计算每个字母的出现次数而不是每个单词的出现次数。

#### (二) 实现过程

修改思路：

1) 修改的 map 的数据类型

```
Map<Character, Integer> myMap = new HashMap<>();
```

2) createMap 方法，将输入的字符串转化为小写，并转换成 char 类型数组

```

// tokenize the input
tokens = tokens.toLowerCase().replaceAll(regex: " ", replacement: ""); // 转换为小写并去除空格
tokens = tokens.replaceAll(regex: "\\pP", replacement: ""); // 去除标点符号

```

3) 用 `containsKey` 方法判断 `map` 是否存过该字母，若没存过则放入一个（字母，1）的元素，若存过则将其 `value` 值加一。

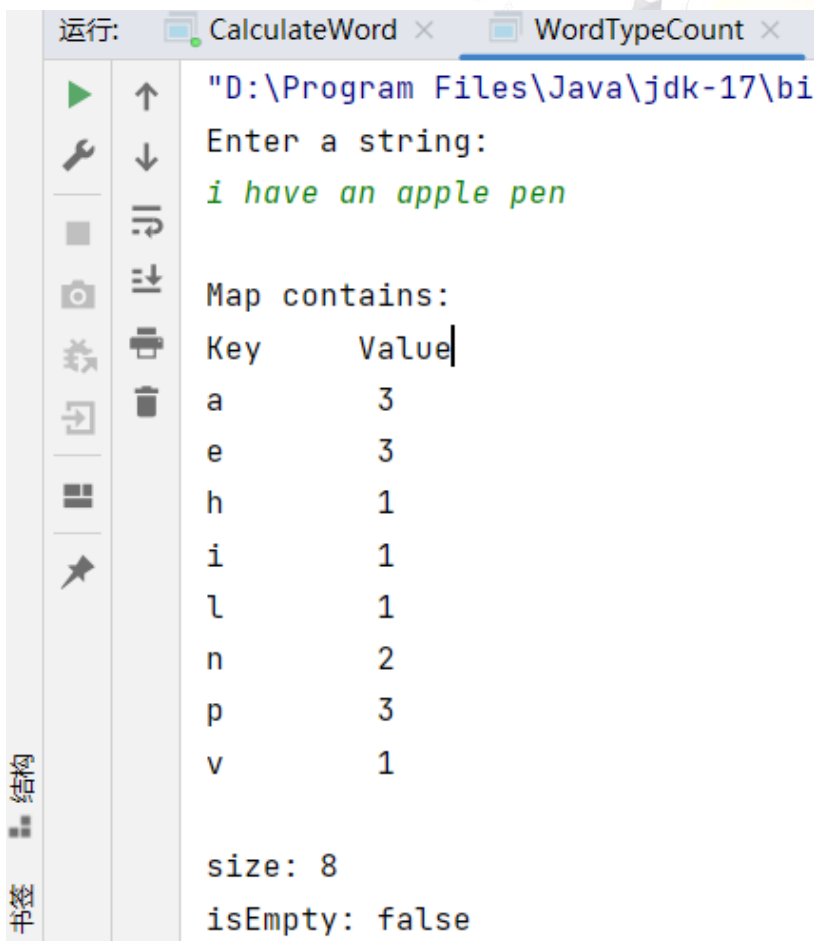
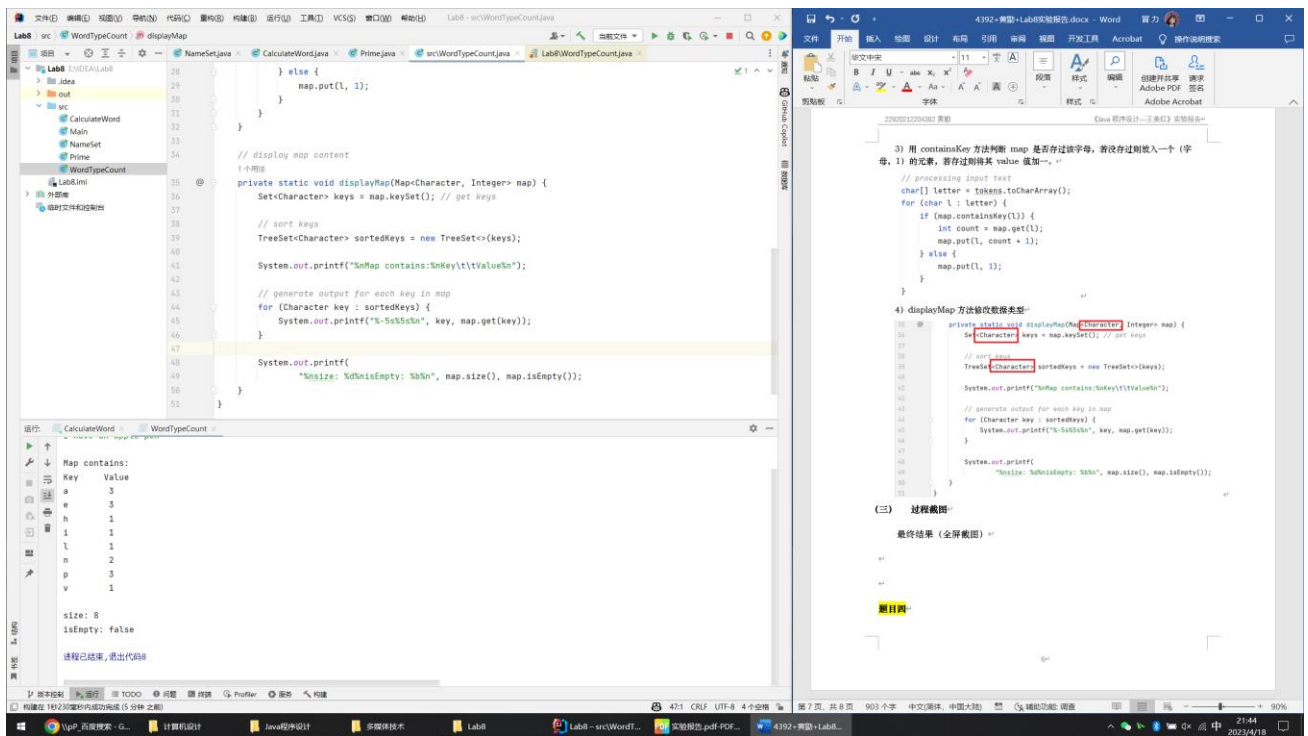
```
// processing input text
char[] letter = tokens.toCharArray();
for (char l : letter) {
    if (map.containsKey(l)) {
        int count = map.get(l);
        map.put(l, count + 1);
    } else {
        map.put(l, 1);
    }
}
```

4) `displayMap` 方法修改数据类型

```
35  @ private static void displayMap(Map<Character, Integer> map) {
36      Set<Character> keys = map.keySet(); // get keys
37
38      // sort keys
39      TreeSet<Character> sortedKeys = new TreeSet<>(keys);
40
41      System.out.printf("\nMap contains:\nKey\t\tValue\n");
42
43      // generate output for each key in map
44      for (Character key : sortedKeys) {
45          System.out.printf("%-5s%-5s\n", key, map.get(key));
46      }
47
48      System.out.printf(
49          "\nsize: %d\nisEmpty: %b\n", map.size(), map.isEmpty());
50  }
51  }
```

### (三) 过程截图

最终结果（全屏截图）





## 题目四

### (一) 实验题目

- 编写一个程序，它从用户处获得一个整数输入，然后判断它是否为质数。如果不显示该数的全部质因子。质数的因子是 1 和自身。不为质数的数，都具有唯一的因子分解形式。例如 54 的因子为 2、3、3、3，这些值相乘的结果是 54。因此对于 54，输出因子应为 2 和 3。需在程序中使用 Set。

### (二) 实现过程

思路：设计了一个类 Prime，main 方法获取用户的输入

1) 检测输入，若输入异常则 catch 要求用户重新输入；

```
7 public static void main(String[] args) {  
8     while (true) {  
9         System.out.println("请输入一个大于1的整数（-1退出）：");  
10        try {  
11            int num = scanner.nextInt();  
12            if (num == -1)  
13                return;  
14            else if (num < 0)  
15                throw new Exception();  
16            else  
17                calcDivisor(num);  
18        } catch (Exception e) {  
19            System.err.println("输入错误，请重新输入");  
20        }  
21    }  
22 }
```

2) 调用 calcDivisor 方法，利用 set 要求对象不重复的特点，初始化了一个 HashSet 用来存放因子

```
23 public static void calcDivisor(int n) {  
24     Set<Integer> set = new HashSet<>();
```

3) 判断 n 是否为质数时用 for 循环从 2 开始遍历，如果遍历到 n 的平方根次时，仍没有找到可以除尽的数则 n 为质数，因子为 1 和它本身；若找到除尽的数，则先判断 set 是否为空，再判断该数是否为 set 中因子的倍数，若不是则存入 set 中

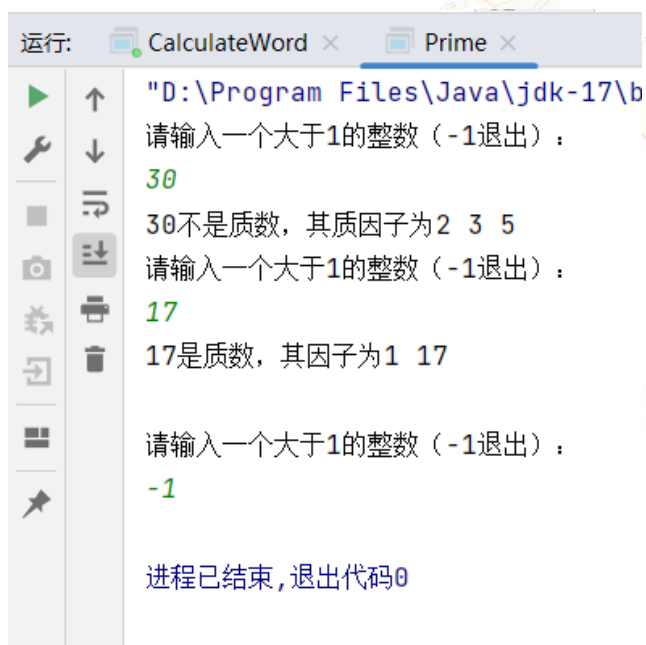
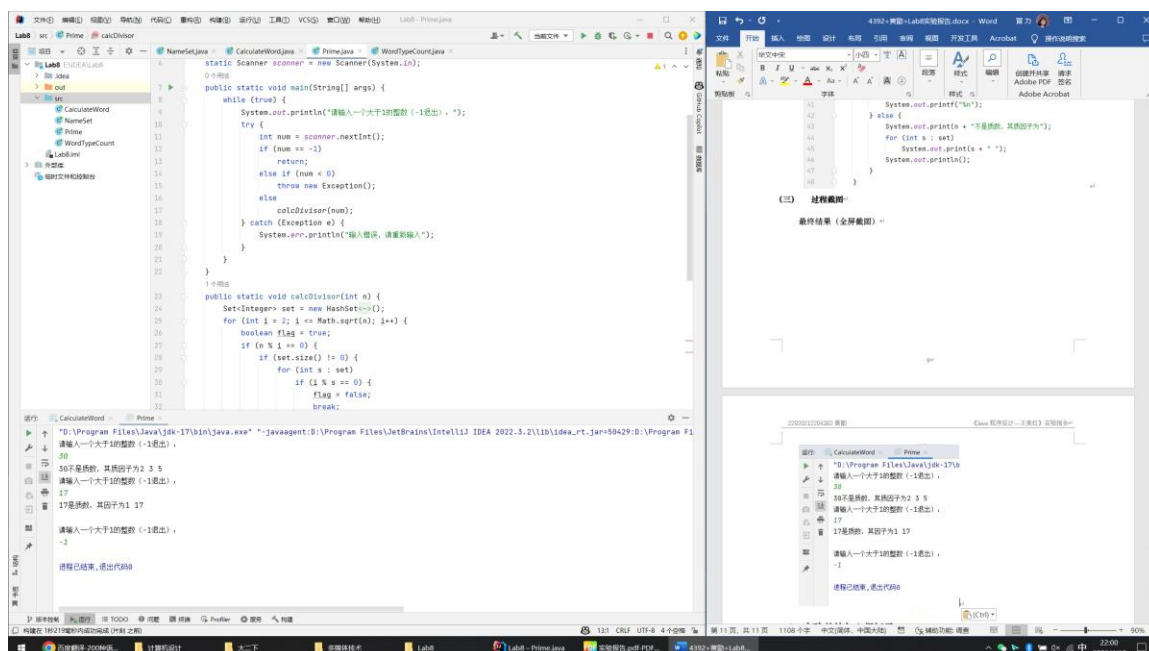
```
25      for (int i = 2; i <= Math.sqrt(n); i++) {
26          boolean flag = true;
27          if (n % i == 0) {
28              if (set.size() != 0) {
29                  for (int s : set)
30                      if (i % s == 0) {
31                          flag = false;
32                          break;
33                      }
34              }
35              if (flag)
36                  set.add(i);
37          }
38      }
```

4) 如果最后 set 不为空, 遍历 set 输出该数的因子, 反之输出质数

```
39      if (set.size() == 0) {
40          System.out.println(n + "是质数, " + "其因子为1 " + n);
41          System.out.printf("%n");
42      } else {
43          System.out.print(n + "不是质数, 其质因子为");
44          for (int s : set)
45              System.out.print(s + " ");
46          System.out.println();
47      }
48  }
```

### (三) 过程截图

最终结果 (全屏截图)



### 三、实验总结与心得记录

通过本次实验,我对 Java 集合类的使用有了更深入的了解。我学会了如何创建集合对象、向集合中添加元素、删除元素、修改元素以及遍历集合中的元素等操作。此外,我还

了解了 Java 集合类的特点和常用方法，如 ArrayList、LinkedList、HashSet、TreeSet 等集合类的区别和使用方法。

