实验报告

院 系 数据科学与计算机学院 学 号 15352237 姓 名 刘顺宇

【实验题目】**Java 实验(1)**

【实验目的】学习 Java 语言的编程。

【实验准备】

直接从网上或从上传作业的网站上下载并安装 JDK。

http://www.oracle.com/technetwork/cn/java/javase/downloads/jdk8-downloads-2133151-zhs.html

【预备知识】

（1）常量定义

```
final int LEVEL_NUM = 1000;
```

（2）动态数组定义

```
int fibs[]=new int[cnt];
```

（3）foreach 语句

```
double sum = 0;
double scores[] = {100.0, 90.2, 80.0, 78.0,93.5};
for(double score:scores){
   sum = sum + score;
}
```

【注意事项】

（1）按照要求的步骤做，不要进行简化。

（2）运行a.bat可以直接进入目录

【实验内容】

1、(StringFunc.java)已知一个字符串 s 为“扁担长，板凳宽，板凳没有扁担长，扁担没有板凳宽。扁担要绑在板凳上,板凳偏不让扁担绑在板凳上。”，使用以下字符串函数完成任务并显示出来：

（1）用 substring 取出 s 中第一个“板凳宽”并显示出来。

（2）用 indexOf()找出 s 中“扁担”出现的所有位置。

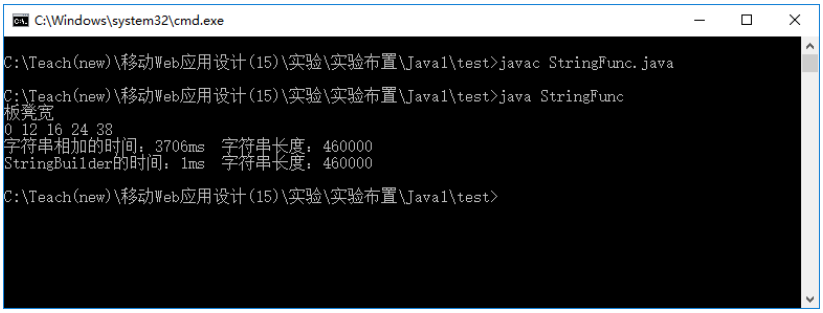
（3）用字符串运算+形成包含重复 10000 次字符串 s 的长字符串，输出计算时间和总长度。

（4）用 StringBuilder 形成上面的长字符串，输出计算时间和总长度。

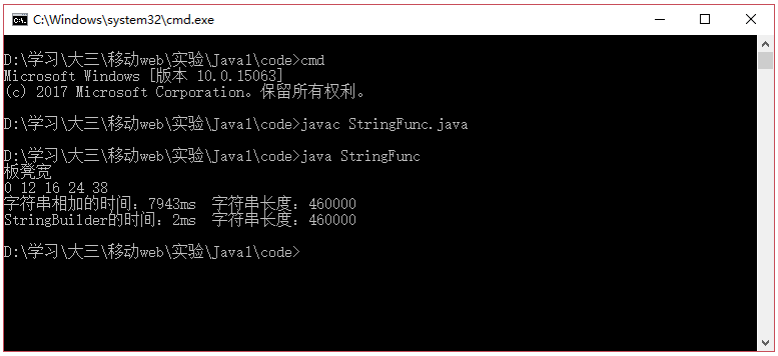
提示：long time= System.currentTimeMillis()

//取得当前时间的毫秒数(距离新纪元时间 1970 年 1 月 1 日 0 时 0 分 0 秒的毫秒数)。

参考结果：



全部完成后截屏：



全部完成后源码(StringFunc.java)：

```
1  import java.util.Arrays;
2  import java.lang.*;
3  public class StringFunc{
4      final static String s="扁担长，板凳宽，板凳没有扁担长，扁担没有板凳宽。扁担要
   绑在板凳上,板凳偏不让扁担绑在板凳上。";
5      public static void main(String args[]){
6          String tem_1 = s.substring(4,7);
7          System.out.println(tem_1);
8
9          int pos = s.indexOf("扁担");
10         while(pos!=-1){
11                 System.out.print(pos + " ");
12                 pos = s.indexOf("扁担",pos+1);
13         }
14         System.out.println();
15
16         String tem_2 ="";
17         long startTime=System.currentTimeMillis();
18         for(int i=0;i<10000;i++){
19                 tem_2 = tem_2+s;
20         }
21         long endTime=System.currentTimeMillis();
22         System.out.print("字符串相加的时间："+(endTime-startTime)+"ms  ");
23         System.out.println("字符串长度："+tem_2.length());
24
25         StringBuilder tem_3=new StringBuilder("");
26         startTime=System.currentTimeMillis();
27         for(int i=0;i<10000;i++){
28                 tem_3.append(s);
29         }
30         endTime=System.currentTimeMillis();
31         System.out.print("StringBuilder的时间："+(endTime-startTime)+"ms
   ");
32         System.out.println("字符串长度："+tem_3.length());
34     }
35 }
```

2、(Fib.java)斐波那契数列（Fibonacci sequence）：第 0 项是 0，第 1 项是 1，从第 2 项开始，每一项都等于前两项之和，结果是 0，1，1，2，3，5，8，13，21，34，55，89，144，233，...。其中，0 为第 0 个斐波那契数。

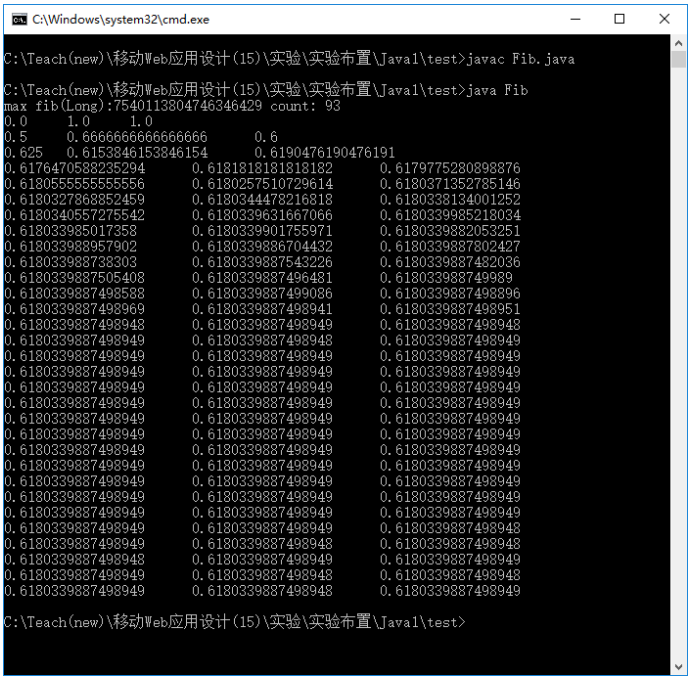
（1）计算斐波那契数列保存在一个 ArrayList 类型的变量 fibs 中，要求一直计算到 Long 类型的最大斐波那契数。显示最后一个数和 fibs 的长度。

*最大长整数为Long.*MAX_VALUE*

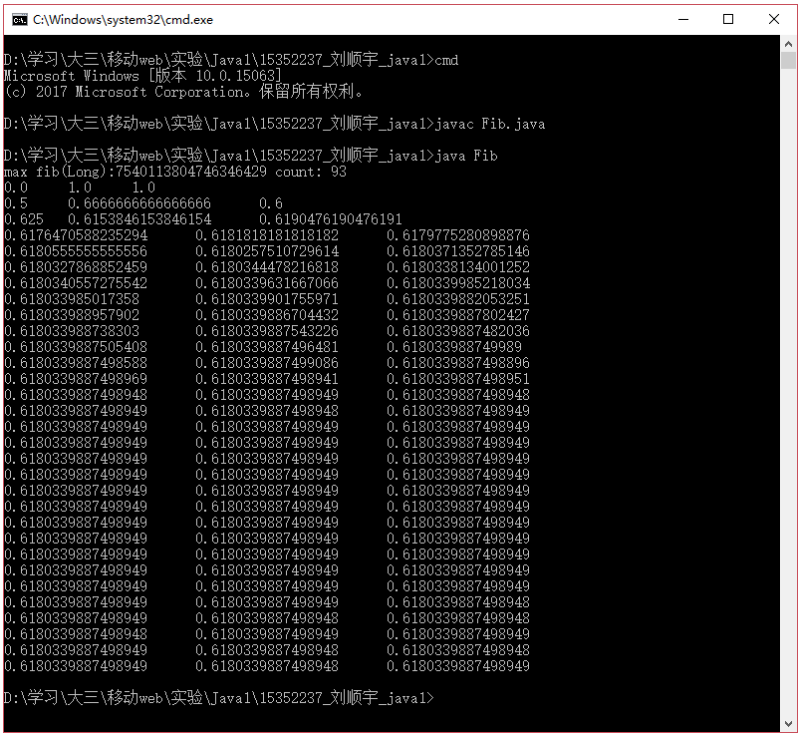
（2）用 Iterator 类依次取出 fibs 中每个元素,计算它与前面一个元素的比值(从第 2 个开始)，保存在 double 类型的动态数组 ratio 中，然后把 ratios 中的所有元素值用 foreach 语句依次显示出来。可以看出这个值逐渐接近黄金分割比例 0.6180339887 4989484820 458683436565。

* ArrayList 的用法见课件，动态数组和 foreach 语句【预备知识】

参考结果：



全部完成后的运行截屏：



全部完成后的源程序(Fib. java)：

```
1  import java.util.*;
2
3  public class Fib{
4      public static void main(String args[]){
5          long fib0=0,fib1=1,fib2=1;
6            ArrayList<Long> fib = new ArrayList<Long>();
7            fib.add(fib0);
8            fib.add(fib1);
9
10           int i = 0;
11           long tem_fib = 0;
12           while(fib.get(i) < Long.MAX_VALUE / 2){
13                    tem_fib = fib.get(i) + fib.get(i + 1);
14                    fib.add(tem_fib);
15                    i++;
16           }
17           System.out.println("max fib(Long):" + fib.get(fib.size()-1)+" coun
   t: " + fib.size());
18
19           double ratios[] = new double[fib.size()];
20           ratios[0] = 0;
21           ratios[1] = 1;
22           i = 2;
23           Iterator<Long> it = fib.iterator();
24           it.next();
25           long tem1 = it.next();
```

```
26          while(it.hasNext()) {
27                  long tem2 = it.next();
28                  ratios[i] = (double)tem1 / (double)tem2;
29                  tem1 = tem2;
30                  i++;
31          }
32
33          i = 0;
34          for(double ratio:ratios){
35                  System.out.print(ratio + "  ");
36                  i++;
37                  if(i == 3){
38                          System.out.println();
39                          i = 0;
40                  }
41          }
42      }
43  }
```

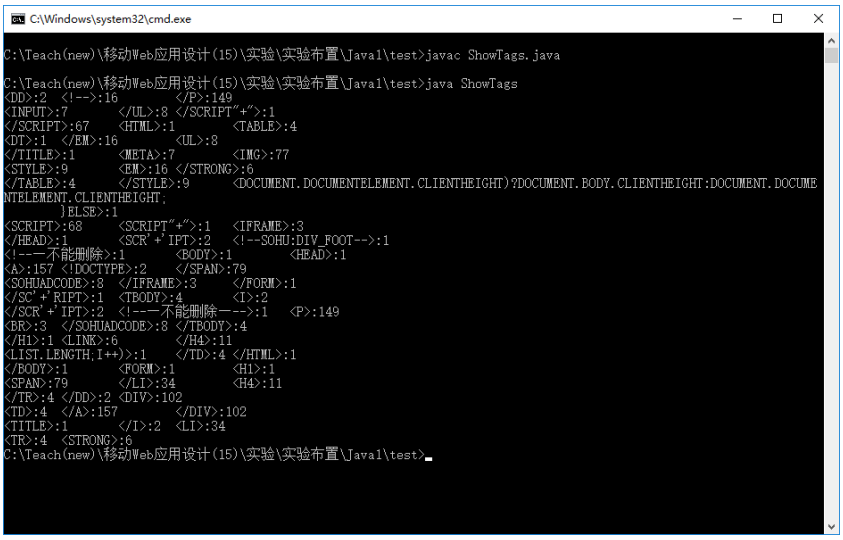
3、(ShowTags.java)找出 html 文件(grassland.htm)中的所有标签名（转换为大写字母），并用 HashMap 保存每个标签出现的次数，最后把所有标签及其出现次数显示出来。

* 要求使用 content.charAt(index)依次取出字符(char 类型)进行判断。char 类型采用“==”进行比较。
* 可能会取到注释和脚本中的标签。
* 不要使用正则表达式

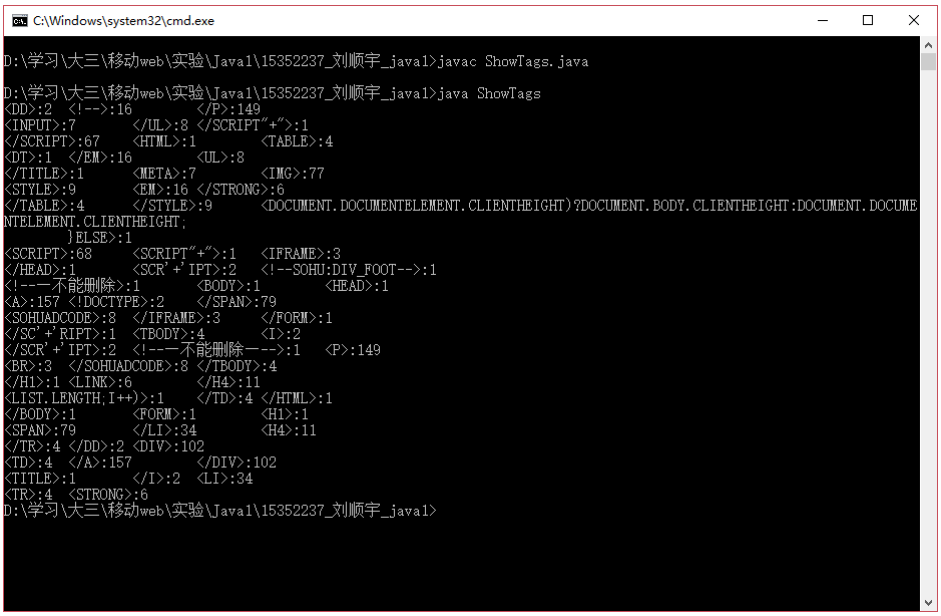
该网页：

http://travel.sohu.com/20161023/n471039505.shtml?pvid=725adae4dbd11180

参考截屏：



完成后运行结果截屏：



问题：DIV 出现多少次？[102]

SPAN 出现多少次？[79]

LI 出现多少次？[34]

源程序(ShowTags.java)：

```
1  import java.io.*;
2  import java.util.*;
3
4  class ShowTags{
5    public static void main(String[] args)throws IOException{
6       String content = readFile(".\\grassland.htm");
7       int len = content.length();
8       HashMap<String, Integer> map = new HashMap<String, Integer>();
9       for(int i = 0; i < len; i++){
10           if(content.charAt(i) == '<' && content.charAt(i+1) != ' '){
11                   for(int j = i+2; j < len; j++){
12                           if(content.charAt(j) == '>' || content.charAt
```

```
(j) == ' '){
13                                      String sub_str = content.substring(i+
1,j).toUpperCase();
14                                              Integer value = map.get(sub_s
tr);
15                                      if (value==null){
16                                              map.put(sub_str, 1);
17                                      }else{
18                                              map.put(sub_str, value+1);
19                                      }
20                                      i = j;
21                                      break;
22                              }else if(content.charAt(j) == '<'){
23                                      i = j - 1;
24                                      break;
25                              }
26                      }
27              }
28      }
29      int i = 0;
30      for(String key:map.keySet()){
31           System.out.print("<"+key+">:"+map.get(key)+"   ");
32           i++;
33           if(i == 3){
34                    System.out.println();
35                    i = 0;
36           }
37      }
38   }
39
40   static String readFile(String fileName) throws IOException{
41           StringBuilder sb = new StringBuilder("");
42           int c1;
43           FileInputStream f1= new FileInputStream(fileName);
44           InputStreamReader in = new InputStreamReader(f1, "gbk");
45
46           while ((c1 = in.read()) != -1) {
47             sb.append((char) c1);
48           }
49         return sb.toString();
50   }
51 }
```

中山大學 SUN YAT-SEN UNIVERSITY 实验报告

【完成情况】

是否完成了这些实验题目？(√完成　×未做或未完成)

1 [√]　2 [√]　3[√]

【实验体会】

写出实验过程中遇到的问题，解决方法和自己的思考；并简述实验体会（如果有的话）。

1. Java 中 print、printf、println 的区别

1.1. System.out.printf 主要是继承了 C 语言的 printf 的一些特性，可以进行格式化输出

1.2. System.out.print 就是一般的标准输出，但是不换行。将它的参数显示在命令窗口，并将输出光标定位在所显示的最后一个字符之后。

1.3. System.out.println 和 print 基本没什么差别,就是最后会换行。将它的参数显示在命令窗口，并在结尾加上换行符，将输出光标定位在下一行的开始。

2. HashMap 更新

2.1. 直接使用 hashmap.put(key, value)函数使用同样的 key 值代替原来的参数即可

3. cmd 窗口输出中文乱码

3.1. 因为 grassland 是 gb2312 编码，readFile 函数用的是 utf-8 编码，将 readFile 函数修改为 gbk 编码即可。

4. 使用 long 的 MAX_VALUE 显示错误需要 class

4.1. 将 long.MAX_VALUE 改为 Long.MAX_VALUE，因为 long 是基本变量类型，Long 是封装过的 long，java 很多数据结构需要使用泛型，不允许使用基本变量类型。

【交实验报告】

(a) 每位同学在宿舍独立完成本实验内容并填写实验报告。

(b) 截止时间：2017 年 11 月 15 日(周三) 23:00

上交作业网站：http://172.18.187.11/netdisk/default.aspx?vm=15web

文件夹: /实验上交/java1

上传文件：学号_姓名_java1.doc　(实验报告)