院 系 数据科学与计算机学院 学 号 　　 姓 名

【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。

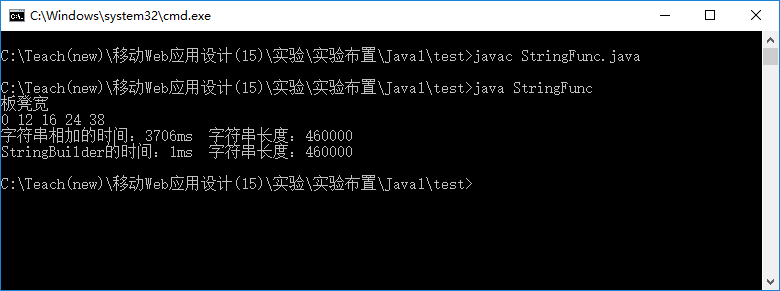
（2）用indexOf()找出s中“扁担”出现的所有位置。

（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。

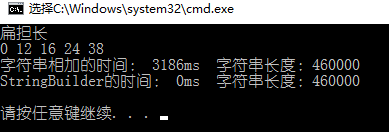
（4）用StringBuilder形成上面的长字符串，输出计算时间和总长度。

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)：

public class StringFunc{

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。";  
 public static void main(String args[]){  
  
 String s1=*s*.substring(0,3);  
 System.*out*.println(s1);  
  
 for(int i=0;i<*s*.length();i++){  
 i = *s*.indexOf("扁担",i);  
 if(i<0)break;  
 System.*out*.print(i+" ");  
 }  
 System.*out*.println();  
 long time= System.*currentTimeMillis*();  
 String s2= new String();  
 s2="";  
 for(int i=0;i<10000;i++){  
 s2+=*s*;  
 }  
 System.*out*.println("字符串相加的时间： "+(System.*currentTimeMillis*()-time)+"ms 字符串长度："+s2.length());  
  
 time= System.*currentTimeMillis*();  
 StringBuilder stringBuilder = new StringBuilder();  
 for(int i=0;i<10000;i++){  
 stringBuilder.append(*s*);  
 }  
 System.*out*.println("StringBuilder的时间： "+(System.*currentTimeMillis*()-time)+"ms 字符串长度："+stringBuilder.toString().length());  
 }  
}

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。

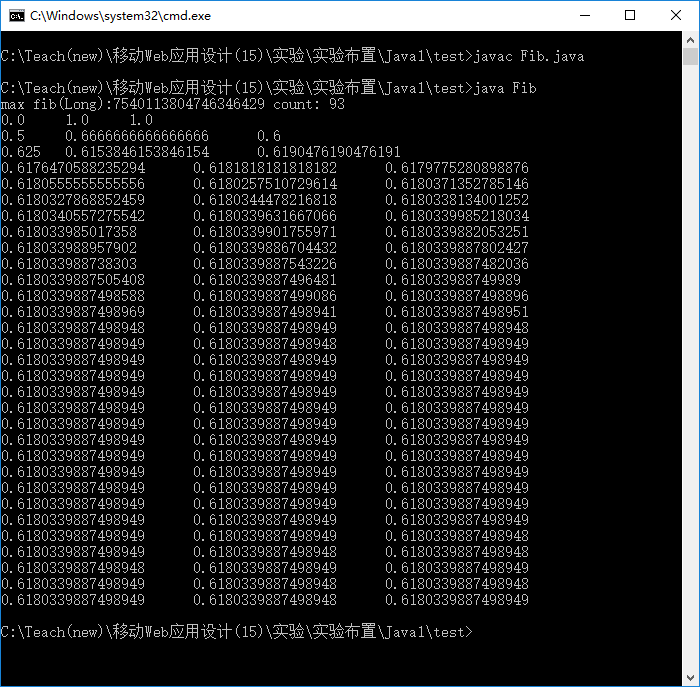
（1）计算斐波那契数列保存在一个ArrayList类型的变量fibs中，要求一直计算到Long类型的最大斐波那契数。显示最后一个数和fibs的长度。

\**最大长整数为*Long.*MAX\_VALUE*

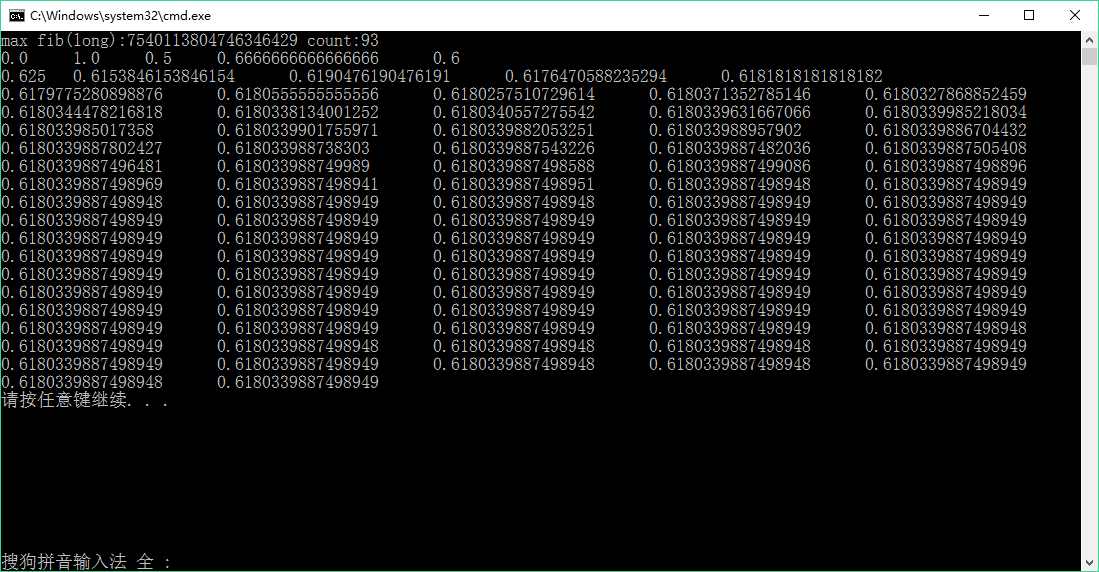
（2） 用Iterator类依次取出fibs中每个元素,计算它与前面一个元素的比值（从第2个开始），保存在double类型的动态数组ratio中，然后把ratios中的所有元素值用foreach语句依次显示出来。可以看出这个值逐渐接近黄金分割比例0.6180339887 4989484820 458683436565。

\* 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)：

import java.util.\*;

public class Fib{  
 public static void main(String args[]){  
 long fib0=0,fib1=1,fib2=1;  
 ArrayList<Long>fibs=new ArrayList<>();  
 fibs.add(fib0);  
 fibs.add(fib1);  
 long tmp=fib2;  
 while(tmp<Long.*MAX\_VALUE*&&tmp >= 0){ //计算Fib  
 fibs.add(tmp);  
 tmp=fibs.get(fibs.size()-1)+fibs.get(fibs.size()-2);  
 }  
 System.*out*.println("max fib(long):"+fibs.get(fibs.size()-1)+ " count:"+fibs.size());  
  
 double ratios[]=new double[fibs.size()-1];  
 Iterator iterator = fibs.iterator();  
 long pre=(long)iterator.next();  
 int i=0;  
 while(iterator.hasNext()){ //求出比值  
 long now=(long)iterator.next();  
 ratios[i++]=(double) pre/now;  
 pre=now;  
 }  
 int count=0;  
 for (double r:ratios) {  
 if((count++)%5==4)  
 System.*out*.println(r);  
 else System.*out*.print(r+"\t");  
 }  
 }  
}

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。

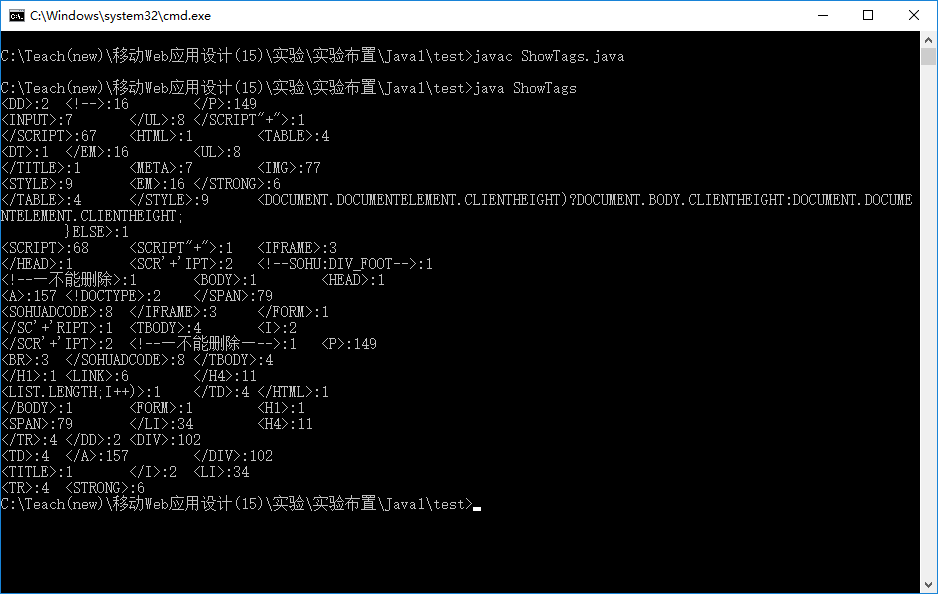
\* 不要使用正则表达式

该网页：

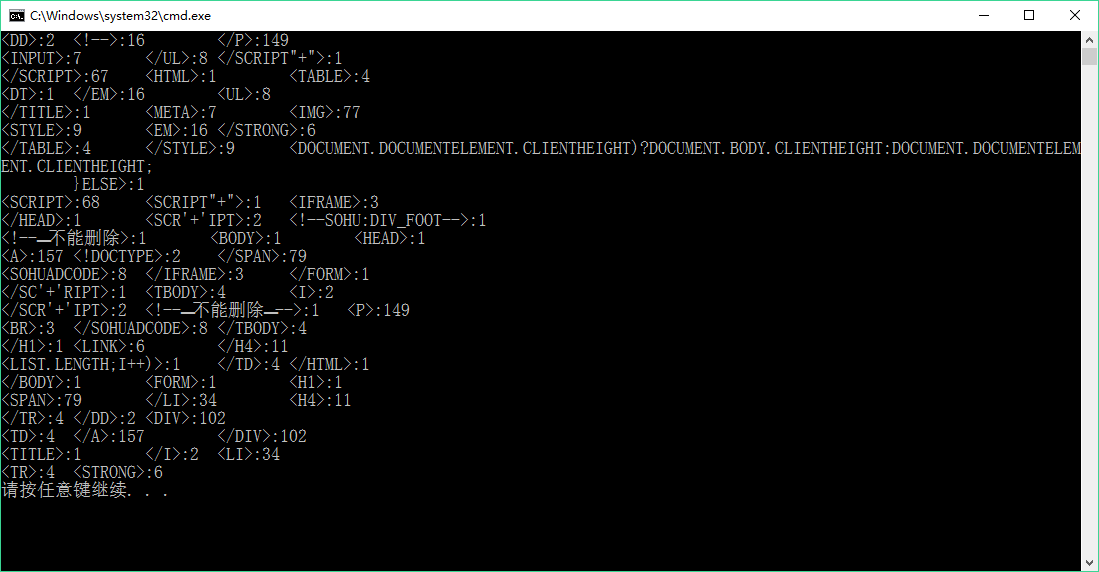


<http://travel.sohu.com/20161023/n471039505.shtml?pvid=725adae4dbd11180>

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)：

import java.io.\*;

import java.util.\*;

class ShowTags{  
 public static void main(String[] args)throws IOException{  
 HashMap<String,Integer>map=new HashMap<>();  
 String content = *readFile*(".\\grassland.htm");  
 content=content.toUpperCase();  
 boolean flag = false;  
 ArrayList<String>out=new ArrayList<>();  
 String label="";  
 for(int i=0;i<content.length();i++){ //分离标签  
 char temp=content.charAt(i);  
 if(flag&& (temp == '>' || temp == ' ')) {  
 flag = false;  
 out.add(label);  
 }  
 if(flag)  
 label+=temp;  
 if(temp == '<') {  
 label = "";  
 flag=true;  
 }  
 }  
 for(String s:out){//map映射   
 if(!s.equals("")){  
 if(map.containsKey(s))  
 map.put(s,map.get(s)+1);  
 else map.put(s,1);  
 }  
 }  
 int count=0; //遍历map输出  
 Iterator iter = map.entrySet().iterator();  
 while (iter.hasNext()) {  
 Map.Entry entry = (Map.Entry) iter.next();  
 String key = (String) entry.getKey();  
 int val = (Integer) entry.getValue();  
 if((count++)%3==2)System.*out*.println("<"+key+">:"+val);  
 else System.*out*.print("<"+key+">:"+val+"\t");  
 }  
 }  
  
 static String readFile(String fileName) throws IOException{  
 StringBuilder sb = new StringBuilder("");  
 int c1;  
 FileInputStream f1= new FileInputStream(fileName);  
 InputStreamReader in = new InputStreamReader(f1, "UTF-8");  
  
 while ((c1 = in.read()) != -1) {  
 sb.append((char) c1);  
 }  
 return sb.toString();  
 }  
}

【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】

这次实验只是初步的java编程，虽然说之前Android课上已经接触过一段时间的java了，但是有一些还是新东西，不过编程部分因为java其实和C++差不到那里去，所以编程实现倒是挺简单的。

第二题斐波那契那边给出的参考效果错了，第三题那个GB2312编码的Htm文件也是很毒性。

【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)