



院系 数据科学与计算机学院 学号 15352408 姓名 张稼伟

【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 c “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：

```
C:\Windows\system32\cmd.exe
C:\Teach(new)\移动Web应用设计(15)\实验\实验布置\Java\test>javac StringFunc.java
C:\Teach(new)\移动Web应用设计(15)\实验\实验布置\Java\test>java StringFunc
板凳宽
0 12 16 24 38
字符串相加的时间: 3706ms 字符串长度: 460000
StringBuilder的时间: 1ms 字符串长度: 460000
C:\Teach(new)\移动Web应用设计(15)\实验\实验布置\Java\test>
```

全部完成后截屏：

```
字符串相加的时间: 2534ms 字符串总长度: 460000
zhangjiwei@Air:code$ freedom$ javac StringFunc.java
zhangjiwei@Air:code$ freedom$ java StringFunc
扁担长
0 12 16 24 38
字符串相加的时间: 3057ms 字符串总长度: 460000
StringBuilder的时间: 1ms 字符串总长度: 460000
zhangjiwei@Air:code$ freedom$
```

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[]){
        System.out.println(s.substring(0,3));
        int w = s.indexOf("扁担");
        while(w != -1){
            System.out.printf("%d ", w);
            w = s.indexOf("扁担", w+1);
        }
        System.out.println();

        String ss="";
        long st = System.currentTimeMillis();
        for(int i=0; i<10000; i++)ss = ss + s;
        long ed = System.currentTimeMillis();
        System.out.printf("字符串相加的时间: %dms   字符串总长度: %d\n", ed-st, ss.length());

        st = System.currentTimeMillis();
        StringBuilder sb = new StringBuilder("");
        for(int i=0; i<10000; i++)sb.append(s);
        ed = System.currentTimeMillis();
        System.out.printf("StringBuilder的时间: %dms   字符串总长度: %d\n", ed-st, sb.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:

[illegible]

全部完成后的运行截屏：

[illegible]

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<Long>();
        fibs.add(fib0);
        fibs.add(fib1);
        fibs.add(fib2);
        while(true){
            long x = fibs.get(fibs.size()-1);
            long y = fibs.get(fibs.size()-2);
            if (Long.MAX_VALUE - x < y)
                break;
            fibs.add(x+y);
        }
        System.out.printf("max fib(Long): %d count: %d\n", fibs.get(fibs.size()-1), fibs.size());
        Iterator iter = fibs.iterator();
        Double oldFib = 0.0;
        Double nextFib;
        ArrayList<Double> ratio = new ArrayList<Double>();
        int cnt = 0;
        while(iter.hasNext()){
            Long tmp = (Long) iter.next();
            nextFib = tmp.doubleValue();
            if (++cnt > 1)
                ratio.add(oldFib/nextFib);
            oldFib = nextFib;
        }
        cnt = 0;
        for (Double x : ratio) {
            System.out.print(x);
            System.out.print("  ");
            if (++cnt == 4){
                System.out.println();
                cnt = 0;
            }
        }
    }
}
```

- 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：

```
C:\Windows\system32\cmd.exe

C:\Teach(new)\移动Web应用设计(15)\实验\实验布置\Java1\test>javac ShowTags.java

C:\Teach(new)\移动Web应用设计(15)\实验\实验布置\Java1\test>java ShowTags

<DD>:2 <!-->:16 <P>:149
<INPUT>:7 </UL>:8 </SCRIPT"&>:1
</SCRIPT>:67 <HTML>:1 <TABLE>:4
<DT>:1 </EM>:16 <UL>:8
</TITLE>:1 <META>:7 <IMG>:77
<STYLE>:9 <EM>:16 </STRONG>:6
</TABLE>:4 </STYLE>:9 <DOCUMENT, DOCUMENTELEMENT, CLIENTHEIGHT>?DOCUMENT, BODY, CLIENTHEIGHT:DOCUMENT, DOCUMENTELEMENT, CLIENTHEIGHT;
)ELSE>:1
<SCRIPT>:68 <SCRIPT"&>:1 <IFRAME>:3
</HEAD>:1 <SCR'+'IPT>:2 <!--SOHU:DIV_FOOT-->:1
<!--不能删除>:1 <BODY>:1 <HEAD>:1
<A>:157 <!DOCTYPE>:2 </SPAN>:79
<SOHUADCODE>:8 </IFRAME>:3 </FORM>:1
</SC'+'IPT>:1 <TBODY>:4 <I>:2
</SCR'+'IPT>:2 <!--不能删除-->:1 <P>:149
<BR>:3 </SOHUADCODE>:8 </TBODY>:4
</H1>:1 <LINK>:6 </H4>:11
<LIST.LENGTH,I++>:1 </TD>:4 </HTML>:1
</BODY>:1 <FORM>:1 <H1>:1
<SPAN>:79 </LI>:34 <H4>:11
</TR>:4 </DD>:2 <DIV>:102
<TD>:4 </A>:157 </DIV>:102
<TITLE>:1 </I>:2 <LI>:34
<TR>:4 <STRONG>:6
C:\Teach(new)\移动Web应用设计(15)\实验\实验布置\Java1\test>
```

完成后运行结果截屏：

```
zhangjiweideAir:code freedom$ java ShowTags
</SCRIPT>: 67 </DD>: 2 <BODY>: 1
<H4>: 11 </HEAD>: 1 <TABLE>: 4
<TD>: 4 <!DOCTYPE>: 2 </TBODY>: 4
</TABLE>: 4 </H4>: 11 <DD>: 2
</TD>: 4 </EM>: 16 <SOHUADCODE>: 8
<STRONG>: 6 <SCR'+'IPT>: 2 <SCRIPT"&>: 1
<H1>: 1 </HTML>: 1 <P>: 149
</SOHUADCODE>: 8 <BR>: 3 <LIST.LENGTH;I++>: 1
</SCRIPT"&>: 1 </SCR'+'IPT>: 2 </STRONG>: 6
```



```
<SCRIPT>: 1 <HTML>: 1 </TITLE>: 1
<LI>: 34 </BODY>: 1 <TITLE>: 1
</P>: 149 </SPAN>: 79 </H1>: 1
<!--不能删除-->: 1 <!-->: 16 <DIV>: 102
<EM>: 16 </SC+'RIPT>: 1 </TR>: 4
</A>: 157 </FORM>: 1 <!--不能删除>: 1
</UL>: 8 <I>: 2 </I>: 2
<UL>: 8 <TBODY>: 4 <TR>: 4
<IMG>: 77 <A>: 157 </IFRAME>: 3
<HEAD>: 1 <IFRAME>: 3 <FORM>: 1
<META>: 7 <!--SOHU:DIV_FOOT-->: 1 <SPAN>: 79
<STYLE>: 9 <INPUT>: 7 </LI>: 34
<DT>: 1 <DOCUMENT.DOCUMENTELEMENT.CLIENTHEIGHT)?DOCUMENT.BODY.CLIENTHEIGHT:DOCUMENT.DOCUMENTELEMENT.CLIENTHEIGHT;
}ELSE>: 1 <SCRIPT>: 67
</STYLE>: 9 </DIV>: 102 <LINK>: 6
```

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{
        String content = readFile("grassland.htm");
        int len = content.length();
        HashMap<String, Integer> map = new HashMap<String, Integer>();
        int st, ed;
        for (int i=0; i<len; i++){
            if (content.charAt(i) == '<' && content.charAt(i+1) != ' '){
                st = i;
                for (int j=st; j<len; j++){
                    if (content.charAt(j)==' ' || content.charAt(j)=='>'){
                        ed = j;
                        String sub = new String(content.substring(i, j));
                        sub = sub + '>';
                        sub = sub.toUpperCase();
                        if (map.containsKey(sub) == false)
                            map.put(sub, 0);
                        int nowValue = map.get(sub);
                        map.put(sub, nowValue + 1);
                        i = j;
                        break;
                    }
                }
            }
        }

        int cnt = 0;
        Iterator iter = map.entrySet().iterator();
        while (iter.hasNext()) {
            Map.Entry entry = (Map.Entry) iter.next();
            Object key = entry.getKey();
```



```
Object val = entry.getValue();
System.out.print(key.toString() + ": " + val.toString() + " ");
if (++cnt == 3){
    cnt = 0;
    entry = (Map.Entry) iter.next();
    System.out.println();
}
}
}

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】

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）。

第一次 java 实验算比较简单。稍微遇到的一点困难是 java 在类型转换时有时不能直接强制转换，需要使用特定的函数，比如 toString、doubleValue。

【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）