



【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：

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

全部完成后截屏：



C:\ 选择C:\Windows\system32\cmd.exe

```
扁担长  
0 12 16 24 38  
字符串相加的时间: 3186ms 字符串长度: 460000  
StringBuilder的时间: 0ms 字符串长度: 460000  
请按任意键继续. . .
```

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());  
    }  
}
```



中山大学 实验报告

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<>();

        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：

```
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 ShowTags.java
C:\Teach(new)\移动Web应用设计(15)\实验\实验布置\Java\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' + RIPT>: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\Java\test>
```

完成后运行结果截屏：

```
C:\Windows\system32\cmd.exe
<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' + RIPT>: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
请按任意键继续. . .
```



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");
        }
    }
}
```



中山大學
SUN YAT-SEN UNIVERSITY

实验报告

```
static String readFile(String fileName) throws IOException{  
    StringBuilder sb = new StringBuilder("");  
    int c1;  
    FileInputStream fl= new FileInputStream(fileName);  
    InputStreamReader in = new InputStreamReader(fl, "UTF-8");  
  
    while ((c1 = in.read()) != -1) {  
        sb.append((char) c1);  
    }  
    return sb.toString();  
}
```