|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 2018 | 年 | 9 | 月 | 30 | 日 |

实验编号：003 **四川师大Java编程实验报告**

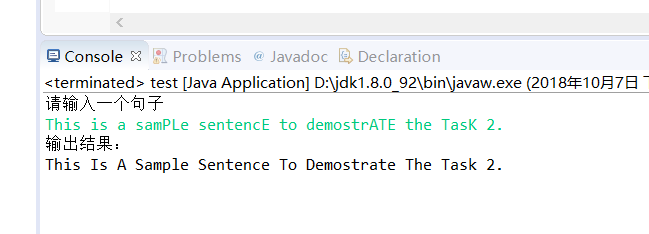
|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **计算机科学学院** | | 2016 | | 级 | | 03 | 班 | 实验名称： | 字符串 | | |
| 姓名： | 何丹阳 | | 学号： | | 2016110308 | | | 指导老师： | 夏羽 | 实验成绩： |  |

**实验三 字符串**

1. **实验目的及要求**
2. 掌握Java字符串的概念和使用。
3. 能够使用字符串的查找、替换、索引等功能。
4. 学会使用字符串的拆分方法。
5. 掌握字符串比较的方法。
6. 培养使用Java字符串解决实际问题的能力。
7. **实验内容**

根据实验说明的要求编写程序并回答实验报告中的问题。

1. 实验主要流程、基本操作或核心代码、算法片段
2. 请将任务一的运行结果截图。



1. String和StringBuffer有什么区别？
   1. String是字符串常量，StringBuffer是字符串变量，
   2. String创建对象是不可更改的，StringBuffer是可以改变的
   3. String对象进行的操作实际上是一个不断创建新的对象并且将旧的对象回收的一个过程，所以执行速度很慢。、
   4. StringBuffer的对象是变量，对变量进行操作就是直接对该对象进行更改，而不进行创建和回收的操作，所以速度要比String快很多
2. 假设s1, s2, s3, s4是四个字符串：

|  |
| --- |
| String s1 = “Welcome to Java”;  String s2 = s1;  String s3 = new String(“Welcome to Java”);  String s4 = “Welcome to Java”; |

那么下面这些表达式的值是什么？

|  |  |  |  |
| --- | --- | --- | --- |
| 表达式 | 值 | 表达式 | 值 |
| s1 == s2 | true | s1.length() | 15 |
| s2 == s3 | false | s1.substring(5) | me to Java |
| s1.equals(s2) | true | s1.substring(5, 11) | me to |
| s2.equals(s3) | true | s1.startsWith(“Wel”) | true |
| s1.compareTo(s2) | 0 | s1.endsWith(“Java”) | true |
| s2.compareTo(s3) | 0 | s1.toLowerCase() | welcome to java |
| S1 == s4 | True | s1.toUpperCase() | WELCOME TO JAVA |
| s1.charAt(0) | W | “ Welcome ”.trim() | Welcome |
| s1.indexOf(‘j’) | -1 | s1.replace(‘o’, ’T’) | WelcTme tT Java |
| s1.indexOf(“to”) | 8 | s1.replaceAll(“o”, ”T”) | WelcTme tT Java |
| s1.lastIndexOf(‘a’) | 14 | s1.replaceFirst(“o”, “T”) | WelcTme to Java |
| s1.lastIndexOf(“o”, 15) | 9 | s1.toCharArray() | Welcome to Java |

1. 以下用于创建字符串对象的两个语句：

|  |
| --- |
| String s = “Hello, world!”; |
| String s= new String(“Hello, world!”); |

哪个更好？为什么？

第一种方法好一点。

第一个创建对象时就创建了一个，而第二个是创建了两个。当使用String时会先在字符串常量池里查看是否有”Hello,world!”字符串对象，有，直接把地址赋值给s.

第二个创建对象就会创建两个，造成空间浪费。在栈中定义一个对象s,去堆中开辟一个内存空间，将内存空间的引用赋值给s，”hello,world!”是常量，然后去字符串常量池 查看是否有hello字符串对象，没有的话分配一个空间存放hello，并且将其空间地址存入堆中new出来的空间中。

1. 请了解StringBuffer类和StringBuilder类

1) 请说明两个类的区别和联系。

区别： 1.StringBuilder速度较快 StringBuffer相对较慢

2.StringBuilder线程不安全 StringBuffer线程安全

联系： 1. 拥有共同的方法名比如append等,都是对字符串进行操作,

2. 大量字符串相加,使用StringBuilder和StringBuffer比直接s1+s2+s3.....性能更好.

2) 请设计一个性能测试程序，测试两个类作为字符串累加过程的性能，并将性能测试结果绘图。

**测试程序：**

**public** **class** bulidsuffer {

**public** **static** **void** main(String[] args) {

**int** loop = 100;

String str = "hdyhdyhdyhdyhdyhdy";

*testStringBuffer*(loop, str);

*testStringBuilder*(loop, str);

}

**public** **static** **void** testStringBuffer(**int** loop,String str) {

StringBuffer strBuffer = **new** StringBuffer();

//测试起始时刻

**long** begin = System.*currentTimeMillis*();

**for**(**int** i = 0 ; i < loop ; i++) {

strBuffer.append(str);

}

//测试截止时刻

**long** end = System.*currentTimeMillis*();

//StringBuffer执行时间

System.***out***.println("StringBuffer: "+ (end - begin));

}

**public** **static** **void** testStringBuilder(**int** loop,String str) {

StringBuilder strBuilder = **new** StringBuilder();

//测试起始时刻

**long** begin = System.*currentTimeMillis*();

**for**(**int** i = 0 ; i < loop ; i++) {

strBuilder.append(str);

}

//测试截止时刻

**long** end = System.*currentTimeMillis*();

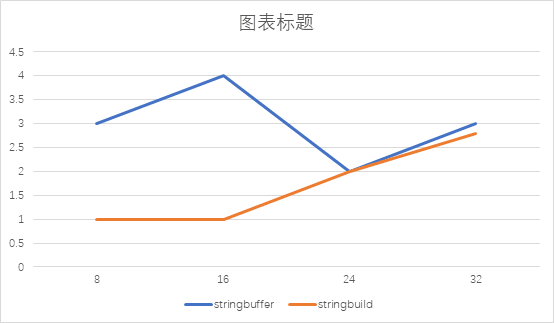
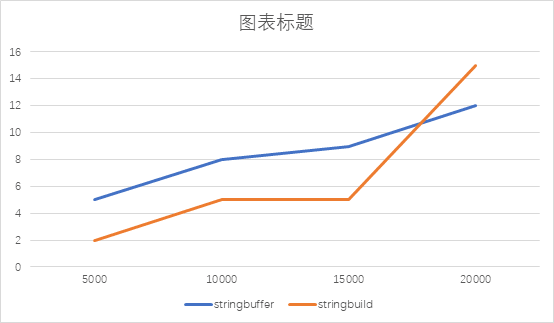
//StringBuilder执行时间

System.***out***.println("StringBuilder: "+ (end - begin));

}

}

**测试结果：**



1. **实验结果的分析与评价**

在本次实验中，老师给了我们一个国庆的时间让我们来完成这次的作业，总体上来说难度有一点点，自己做的健壮性不太好，还有很多要改进的地方。在做用户注册的那些地方时，自己学的正则表达式不太会，去网上找了很多，学习了一些，对正则表达式理解了更多。

注：实验成绩等级分为（90－100分）优，（80－89分）良，(70-79分)中，（60－69分）及格，（59分）不及格。