**暨南大学本科实验报告专用纸**

课程名称 java程序设计实验 成绩评定

实验项目名称 替换错别字 指导教师 张庆丰

实验项目编号 0806016904 实验项目类型 实验地点 N502

学生姓名 张印祺 学号 2018051948

学院 信息科学技术学院 系 计算机科学系 专业 网络工程

实验时间 2020 年 11 月 11 日 下 午～ 11 月 11 日 下 午

**一、实验目的**

本实验的目的是掌握怎样使用Pattern类和Match类检索字符串。

**二、实验要求**

下列字符串中将“登录网站”错写为“登陆网站”，将“惊慌失措”错写为“惊慌失错”：

“忘记密码，不要惊慌失错，请登陆我们的网站，我们有办法”

编写一个java应用程序，输出把错别字替换为正确用字的字符串。

**三、程序代码**

**import** java.util.regex.Matcher;

**import** java.util.regex.Pattern;

**public** **class** PatternTest {

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

String str,arr1[]= {"失错","登陆"},arr2[]= {"失措","登录"};

Matcher m=**null**;

str="忘记密码，不要惊慌失错，请登陆我们的网站，我们有办法";

System.***out***.println("原句:"+str);

**for**(**int** i=0;i<arr1.length;i++) {

m=Pattern.*compile*(arr1[i]).matcher(str);

**while**(m.find()) {

str=str.replace(m.group(), arr2[i]);

}

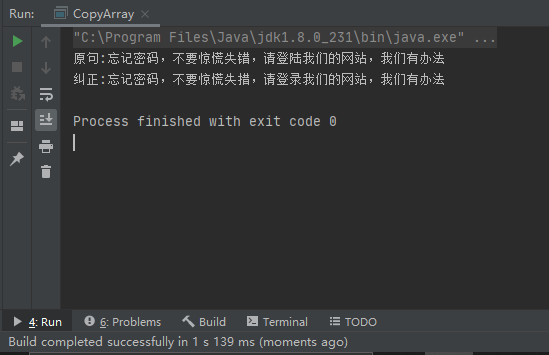
}

System.***out***.println("纠正:"+str);

}

}

**四、实验结果**



**五、感悟与提升**

1. Pattern类的作用在于编译正则表达式后创建一个匹配模式.

public final class Pattern extends Object implements Serializable正则表达式的编译表示形式。用于编译正则表达式后创建一个匹配模式。

指定为字符串的正则表达式必须首先被编译为此类的实例。然后，可将得到的模式用于创建Matcher对象，依照正则表达式，该对象可以与任意字符序列匹配。执行匹配所涉及的所有状态都驻留在匹配器中，所以多个匹配器可以共享同一模式。

因此，典型的调用顺序是：

Pattern p =Pattern.compile("a\*b");

Matcher m =p.matcher("aaaaab");

boolean b = m.matches();

在仅使用一次正则表达式时，可以方便地通过此类定义matches方法。此方法编译表达式并在单个调用中将输入序列与其匹配。语句：

boolean b =Pattern.matches("a\*b", "aaaaab");

等效于上面的三个语句，尽管对于重复的匹配而言它效率不高，因为它不允许重用已编译的模式。

此类的实例是不可变的，可供多个并发线程安全使用。Matcher类的实例用于此目的则不安全。

1. Matcher类使用Pattern实例提供的模式信息对正则表达式进行匹配

public final class Matcher extends Object implements MatchResult通过调用模式（Pattern）的matcher方法从模式创建匹配器。创建匹配器后，可以使用它执行三种不同的匹配操作：

1、matches方法尝试将整个输入序列与该模式匹配。

2、lookingAt尝试将输入序列从头开始与该模式匹配。

3、find方法扫描输入序列以查找与该模式匹配的下一个子序列。

每个方法都返回一个表示成功或失败的布尔值。通过查询匹配器的状态可以获取关于成功匹配的更多信息。

匹配器在其输入的子集（称为区域）中查找匹配项。默认情况下，此区域包含全部的匹配器输入。可通过region方法修改区域，通过regionStart和regionEnd方法查询区域。区域边界与某些模式构造交互的方式是可以更改的。

此类还定义使用新字符串替换匹配子序列的方法，需要时，可以从匹配结果计算出新字符串的内容。可以先后使用appendReplacement和appendTail方法将结果收集到现有的字符串缓冲区，或者使用更加便捷的replaceAll方法创建一个可以在其中替换输入序列中每个匹配子序列的字符串。

匹配器的显式状态包括最近成功匹配的开始和结束索引。它还包括模式中每个捕获组捕获的输入子序列的开始和结束索引以及该子序列的总数。出于方便的考虑，还提供了以字符串的形式返回这些已捕获子序列的方法。

匹配器的显式状态最初是未定义的；在成功匹配导致IllegalStateException抛出之前尝试查询其中的任何部分。每个匹配操作都将重新计算匹配器的显式状态。匹配器的隐式状态包括输入字符序列和添加位置，添加位置最初是零，然后由appendReplacement方法更新。

可以通过调用匹配器的reset()方法来显式重置匹配器，如果需要新输入序列，则调用其reset(CharSequence)方法。重置匹配器将放弃其显式状态信息并将添加位置设置为零。

此类的实例用于多个并发线程是不安全的。