|  |  |
| --- | --- |
| **数据结构与算法 实验报告** | |
| 第二次 | |
| 图片包含 标牌  已生成极高可信度的说明 | |
|  | |
|  | |
|  | |
| **姓名** | 代珉玥 |
| **班级** | 软件001班 |
| **学号** | 2205223077 |
| **电话** | 18585038226 |
| **Email** | 2040257842@qq.com |
| **日期** | 2020-10-30 |

目录

[实验1 2](#_Toc19482)

[1. 题目 2](#_Toc29561)

[2. 数据结构设计 2](#_Toc5689)

[3. 算法设计 3](#_Toc10700)

[4. 主干代码说明 3](#_Toc26697)

[5. 运行结果展示 5](#_Toc15138)

[6. 总结和收获 6](#_Toc27100)

[7. 参考文献 6](#_Toc12626)

[实验2 7](#_Toc17329)

[8. 题目 7](#_Toc26435)

[9. 数据结构设计 7](#_Toc28081)

[10. 算法设计 7](#_Toc17256)

[11. 主干代码说明 8](#_Toc4587)

[12. 运行结果展示 8](#_Toc22365)

[13. 总结和收获 8](#_Toc21574)

[14. 参考文献 9](#_Toc5433)

[实验3 9](#_Toc3340)

[15. 题目 9](#_Toc4228)

[16. 数据结构设计 9](#_Toc13428)

[17. 算法设计 9](#_Toc12295)

[18. 主干代码说明 9](#_Toc28630)

[19. 运行结果展示 9](#_Toc7871)

[20. 总结和收获 9](#_Toc30004)

[21. 参考文献 9](#_Toc32184)

# 实验1

## 题目

回文串就是指不论正读还是反读都是一样的字符串。比如以下都是回串：”radar”、”able was ere saw elba”，另外，如果我们做一个可以忽略文本串中的空格的话，那么下面串也是回文串：“a man a plan a canal panama”。

题目任务：   
 1.要求使用递归思想，编写一个函数 testPalindromeByRecursion，该函数用来判断给定的字符串参数是否是回文串，如果是则返回true，否则返回false。   
 2.要求使用栈数据结构，将 1 中编写的程序转换成非递归函数，函数名为：testPalindromeByStack，该函数用来判断给定的字符串是否是回文串，如果是则返回true，否则返回false。   
 注：本实验中使用的栈和队列都只能使用自定义的数据结构，以下同。

## 数据结构设计

设计一个栈，储存字符串。

public class Node {  
 private char element;  
 private Node below;

}

public class MyStack {  
 private Node top;  
 private int length;

}

## wps算法设计

testPalindromeByStack方法：

将字符串中的除了空格的字符输入两个栈中，将第二个栈中的元素弹入第三个栈中，比较第一个栈和第三个栈中弹出的元素是否相同，若均相同则返回true。

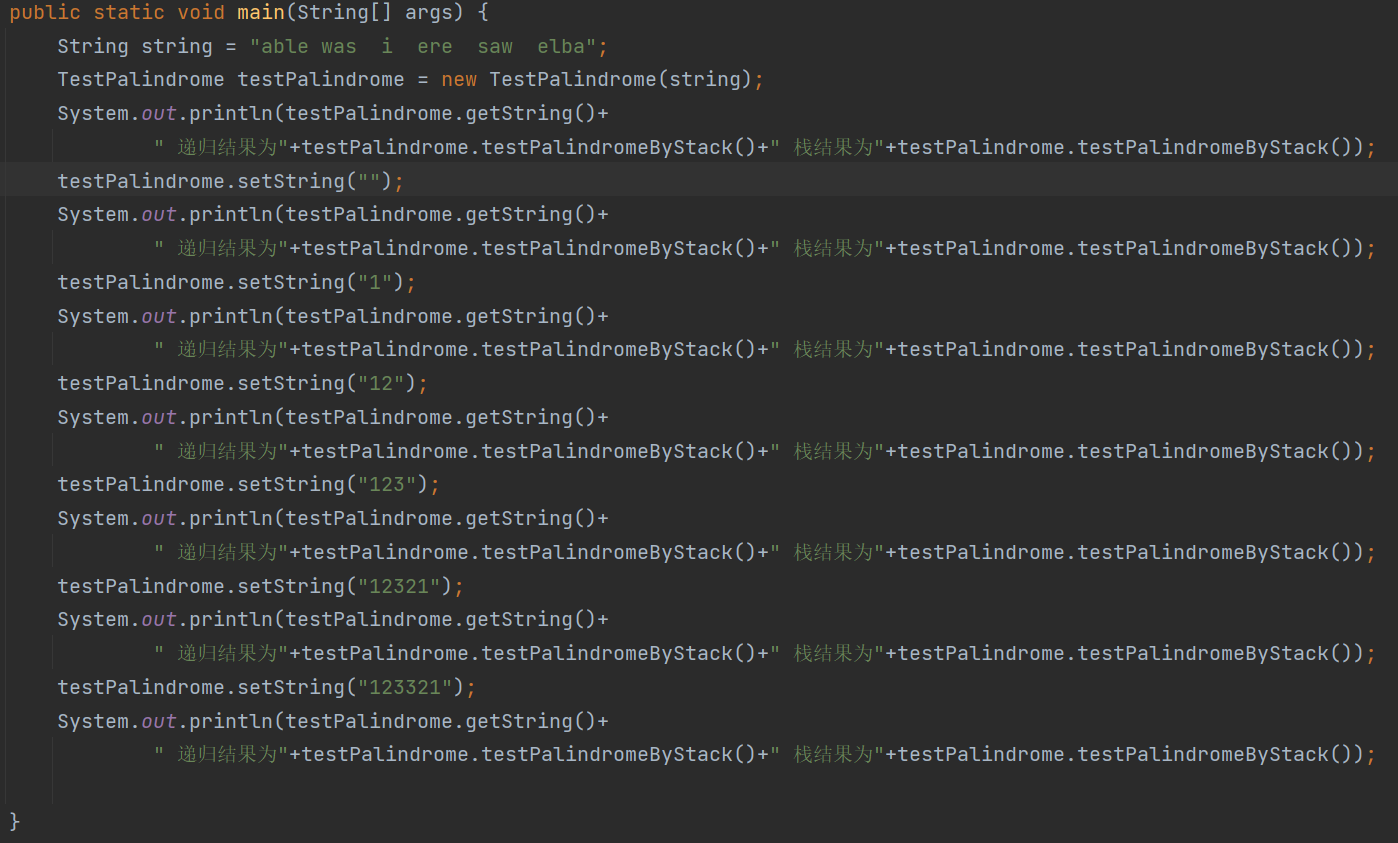
## 主干代码说明

boolean testPalindromeByRecursion(int start,int end){  
 if(end-start<1){  
 return true;  
 }  
 if (string.charAt(start)==' '){  
 return testPalindromeByRecursion(start+1,end); //后移起始下标  
 }  
 if (string.charAt(end)==' '){  
 return testPalindromeByRecursion(start,end-1); //前移终止下标  
 }  
 if (string.charAt(start)!=string.charAt(end)){  
 return false;  
 }  
 return testPalindromeByRecursion(start+1,end-1); //后移起始下标，前移终止下标  
}

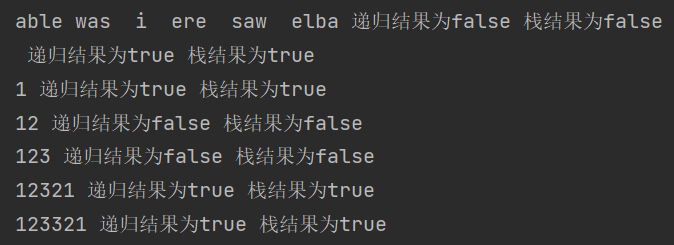
boolean testPalindromeByStack(){  
 MyStack myStack1 = new MyStack(); //原始顺序  
 MyStack myStack2 = new MyStack(); //暂存栈  
 MyStack myStack3 = new MyStack(); //倒序  
 for (int i = 0; i < string.length(); i++) {  
 if (string.charAt(i) != ' ') {  
 myStack1.push(string.charAt(i));  
 myStack2.push(string.charAt(i));  
 }  
 }  
 for (int i = 0; i < myStack1.getLength(); i++) {  
 myStack3.push(myStack2.pop()); //倒序存入  
 }  
 for (int i = 0; i < myStack1.getLength(); i++) {  
 if(myStack1.pop()!=myStack3.pop()){  
 return false;  
 }  
 }  
 return true;  
}

## 运行结果展示

测试代码：



运行结果：



## 总结和收获

在写第二个方法时，刚开始节点中的两个成员分别是element和precursor，后来感觉这个名字不太好，应该改为element和below（这个节点下面的元素）。但是在构造方法中的传入变量名忘记改了，导致pop的时候top = new Node(element, top)没有给below赋值，出现了空指针引用。

之后一是应该先做好全局的构思再写代码，二是改名的时候应该注意是否全都改了，尤其在传参的时候。

## 参考文献

无。

# 实验2

## 题目

## 数据结构设计

## 算法设计

## 主干代码说明

## 运行结果展示

## 总结和收获

## 参考文献

# 实验3

## 题目

回文串就是指不论正读还是反读都是一样的字符串。比如以下都是回文串：”radar”、”able was i ere saw elba”，另外，如果我们做一个可以忽略文本串中的空格的话，那么下面串也是回文串：“a man a plan a canal panama“。   
 题目任务：   
 1. 要求使用递归思想，编写一个函数 testPalindromeByRecursion，该函数用来判断给定的字符串参数是否是回文串，如果是则返回true，否则返回false。   
 2. 要求使用栈数据结构，将 1 中编写的程序转换成非递归函数，函数名为：testPalindromeByStack，该函数用来判断给定的字符串是否是回文串，如果是则返回true，否则返回false。   
 注：本实验中使用的栈和队列都只能使用自定义的数据结构，以下同。

## 数据结构设计

## 算法设计

## 主干代码说明

## 运行结果展示

## 总结和收获

## 参考文献