

**《数据结构》**

**实验报告**

**实验（ 二 ）**

**年 级： 2023级**

**专 业： 计算机科学与技术**

**姓 名： 袁田**

**学 号： 2314022**

# 一．实验内容

1.实现以下内容：实现一个栈，包括出栈、入栈、获取栈顶元素、判断是否为空和统计栈中元素个数5个基本方法。

2.使用内容1中的栈进行表达式计算，表达式是一个可能包括“+”，“-”，“\*”，“/”，“(”，“)”，数字以及未知数的字符串（未知数定义为A~Z，a~z）。

2.1 判断表达式是否合法。

2.2 表达式求值：计算表达式的运算结果，如果包含未知数，返回一个包含未知数的简化表达式。

2.3 表达式优化：减少不必要的计算步骤，例如将 x+0 简化为x，将 y\*0 简化为0

# 二．设计思想

T1.设计一个栈，利用链表的形式构建

T2.1 判断表达式的合法性：

1. 通过遍历输入的表达式，检验表达式中是否均为合法字符，包括数字，大小写字母，左右括号，和规定的+-\*/四个运算符
2. 检验括号的匹配性：设计一个栈遍历字符串，将左括号直接插入，右括号插入前确定栈顶是否为左括号，若不是直接返回非法；若确为左括号直接删除栈顶元素，遍历结束后判断栈是否为空，若不为空，则为非法
3. 检验字符前内容的合法性：

遍历字符串，确定类型后对于数字，字母，左括号，右括号，运算符均有进行判定的标准，并且注意在字符串结尾不得为运算符

T2.2.表达式第一次化简，构建函数Simplify

T2.3表达式第二次化简，将其中与0和1进行加减乘除的特殊情况进行考虑

# 三．程序效果

# 程序的运行效果，输入及输出的相关要求和具体执行结果如下所示：

Part1. 设计一个栈，利用链表的形式构建。

Part2.

案例一：

输入： ((3+4\*(2-1))/5+x+0\*y)

输出：

True

1.4+x+0\*y

1.4+x

先是判断表达式的合法性，然后进行两次化简。

案例二：

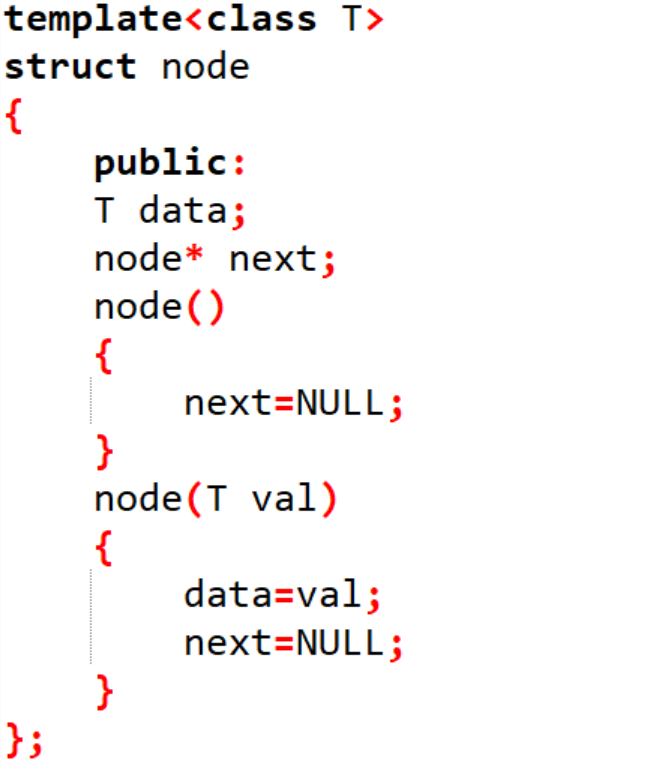
输入： 3+4\*(2-1)+

输出： False

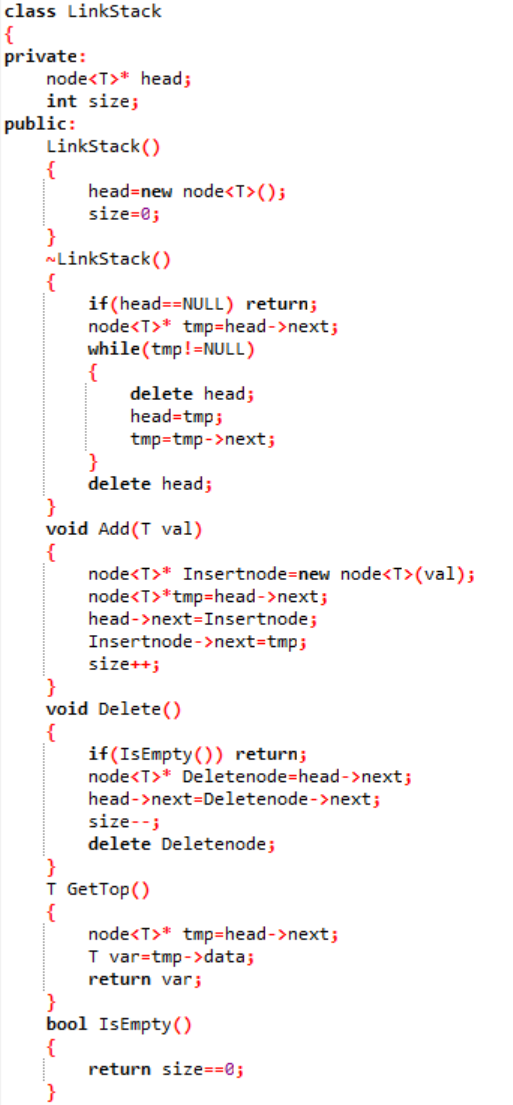
# 四．核心代码

T1.

构建node结构体和LinkStack类(并实现Add,Delete,GetTop,IsEmpty)

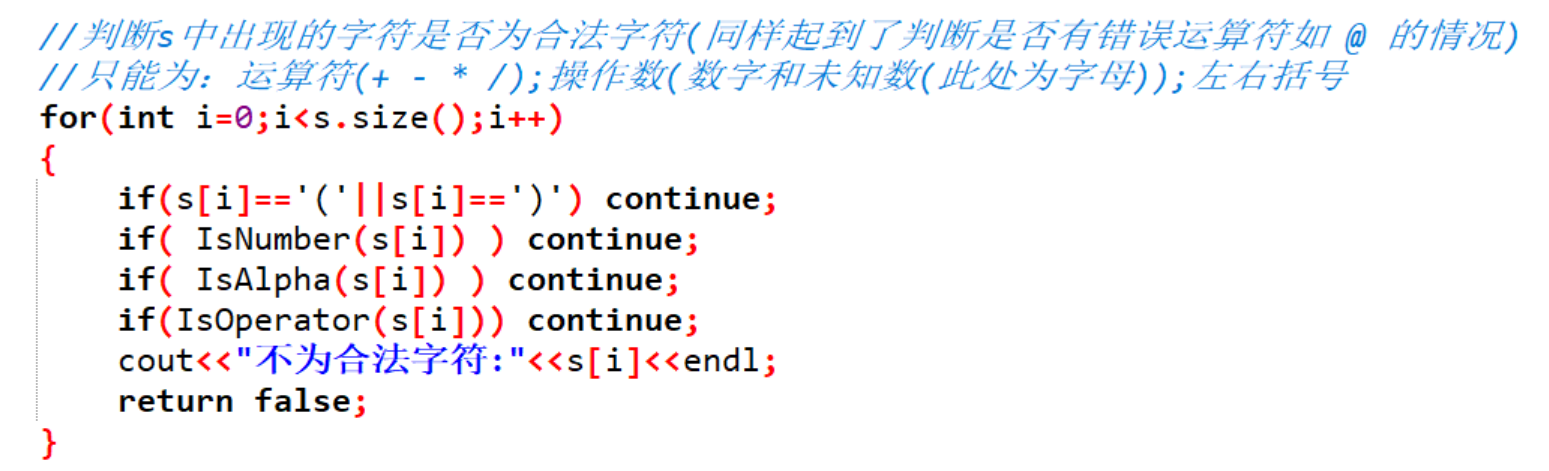


Node结构体

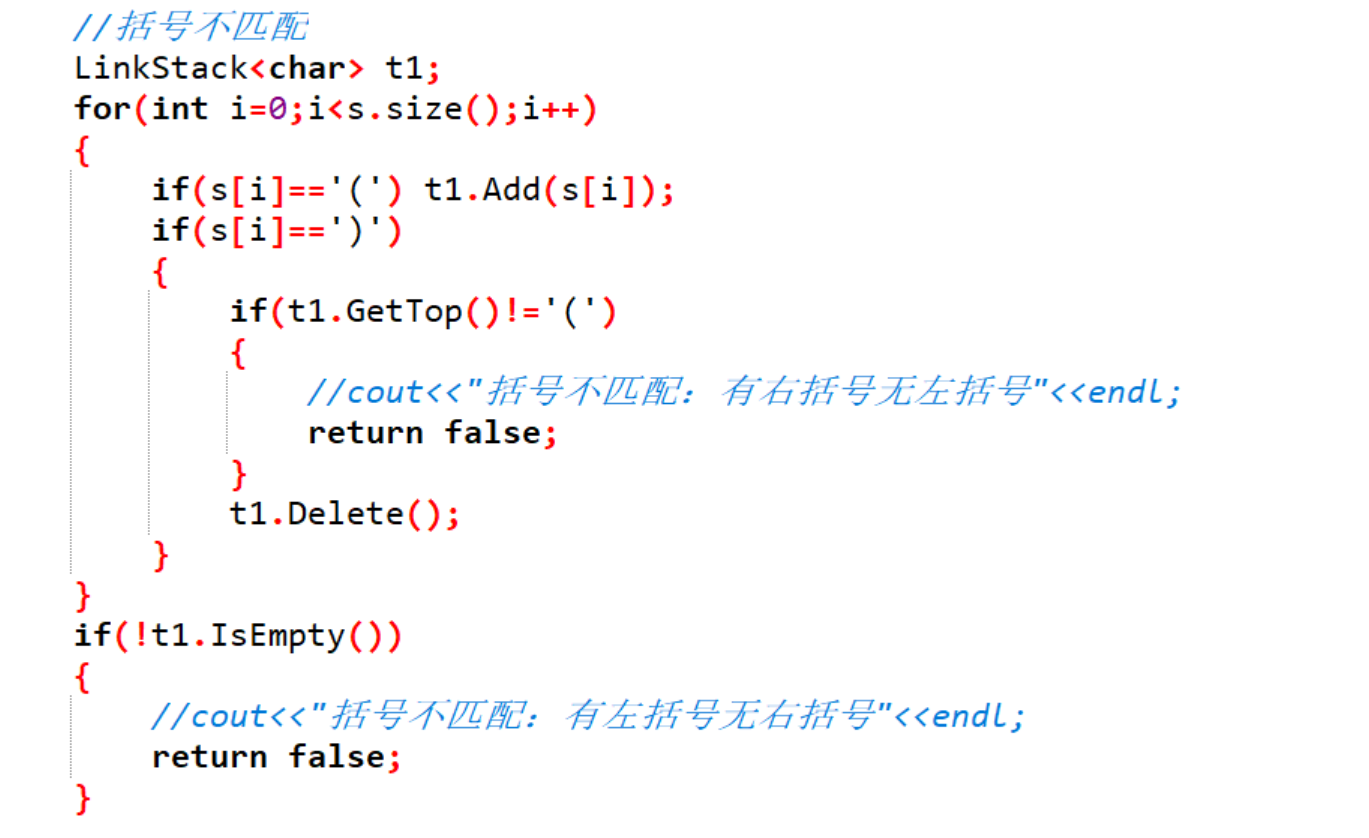


Link Stack类

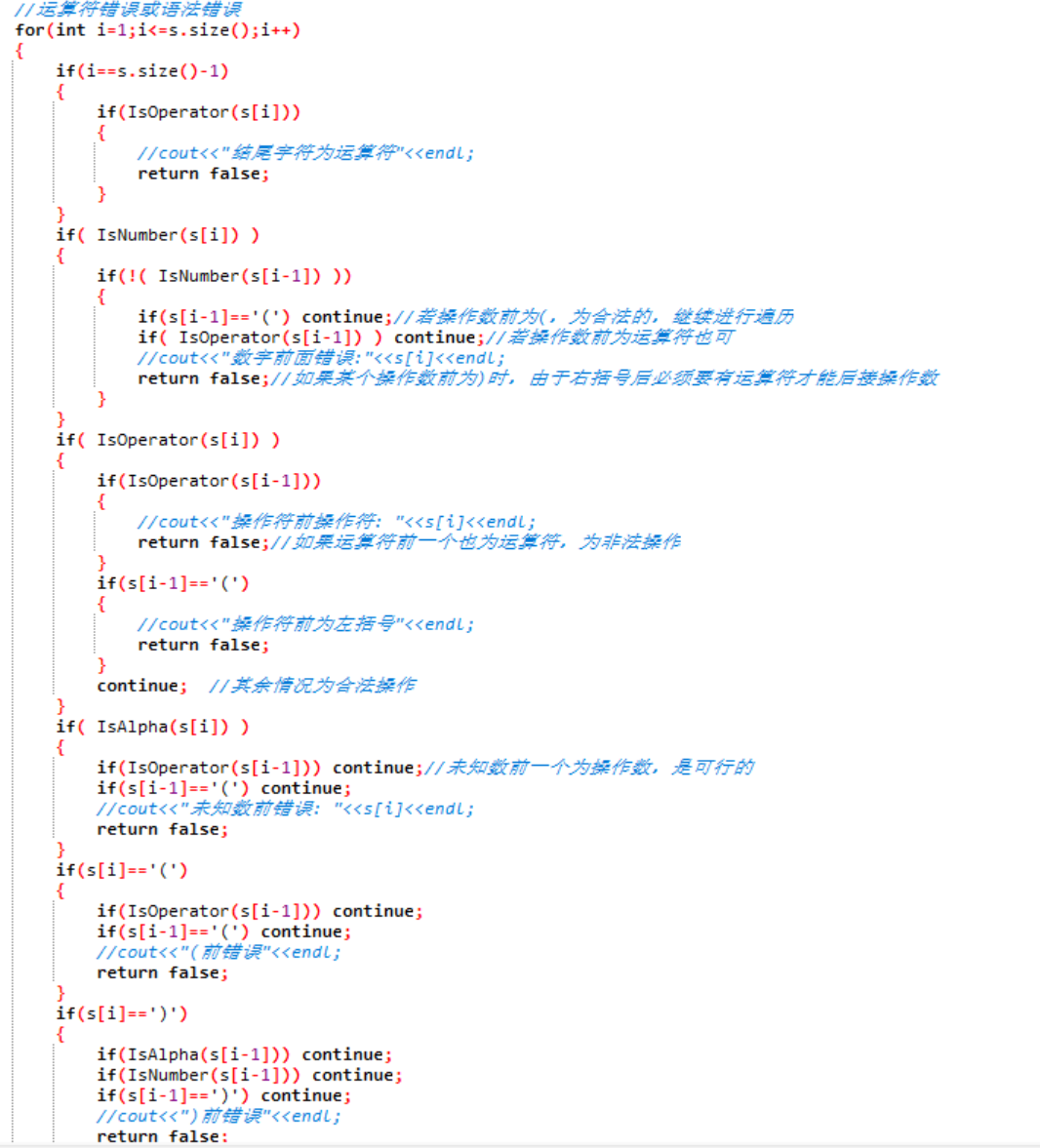
T2.Part1判断表达式的合法性，若合法返回True



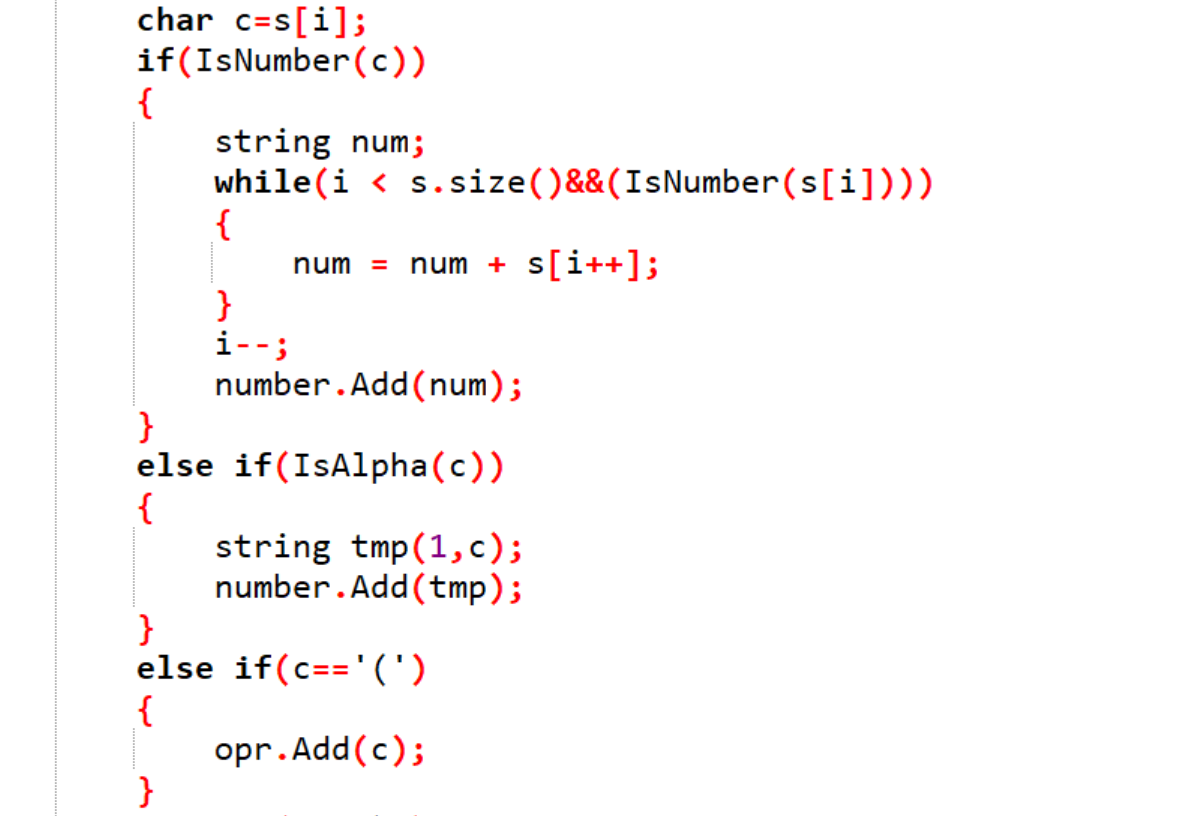
判断字符的合法性，避免有￥@等非法字符的出现

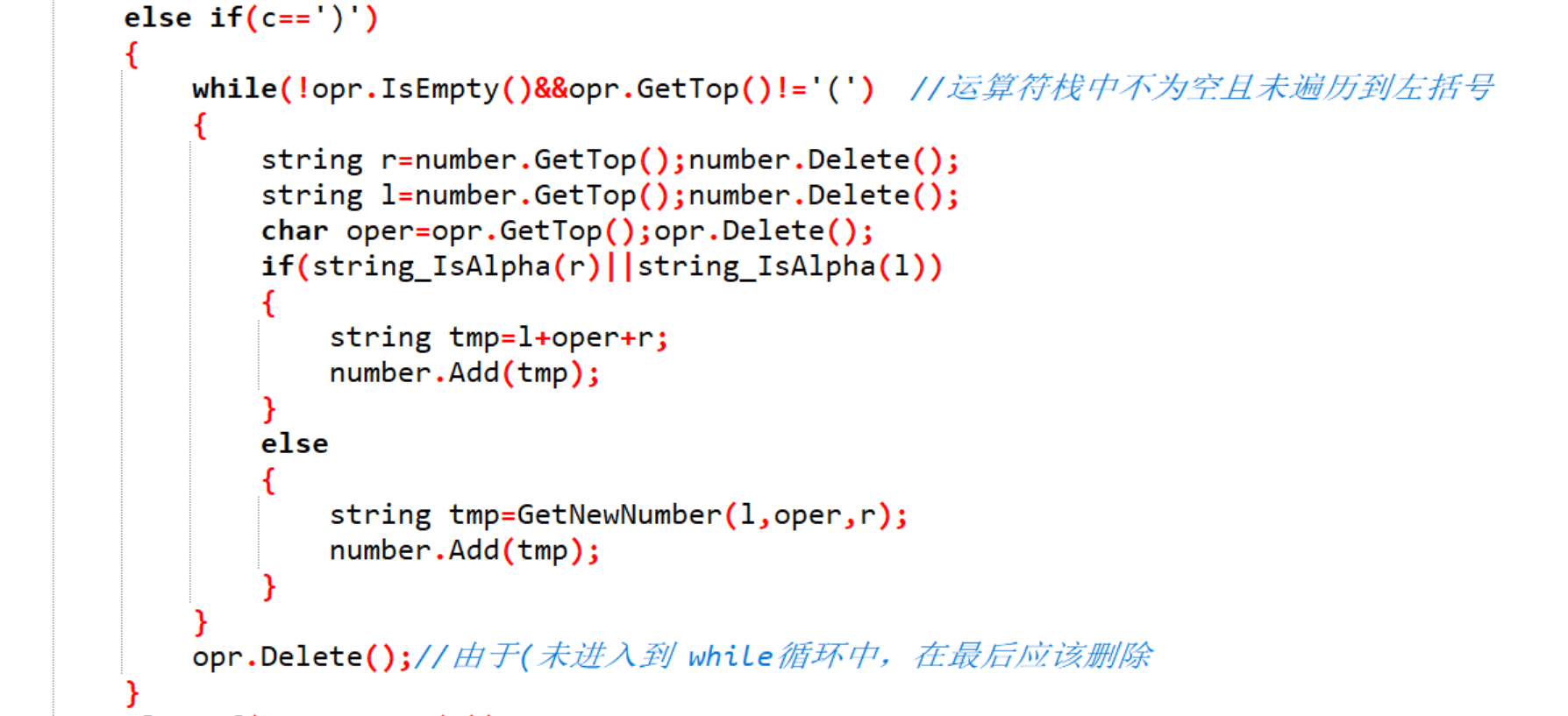


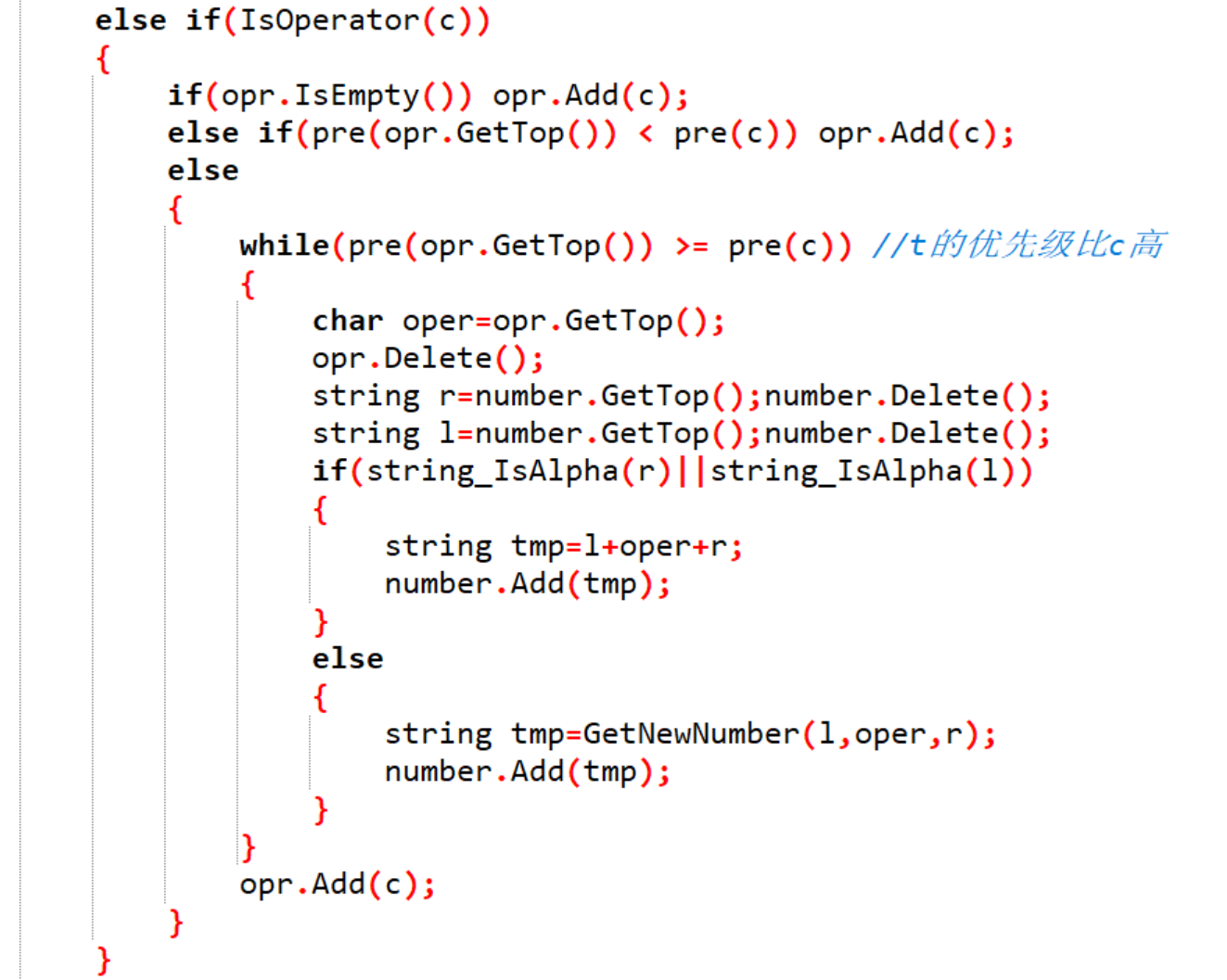
判断括号的匹配性

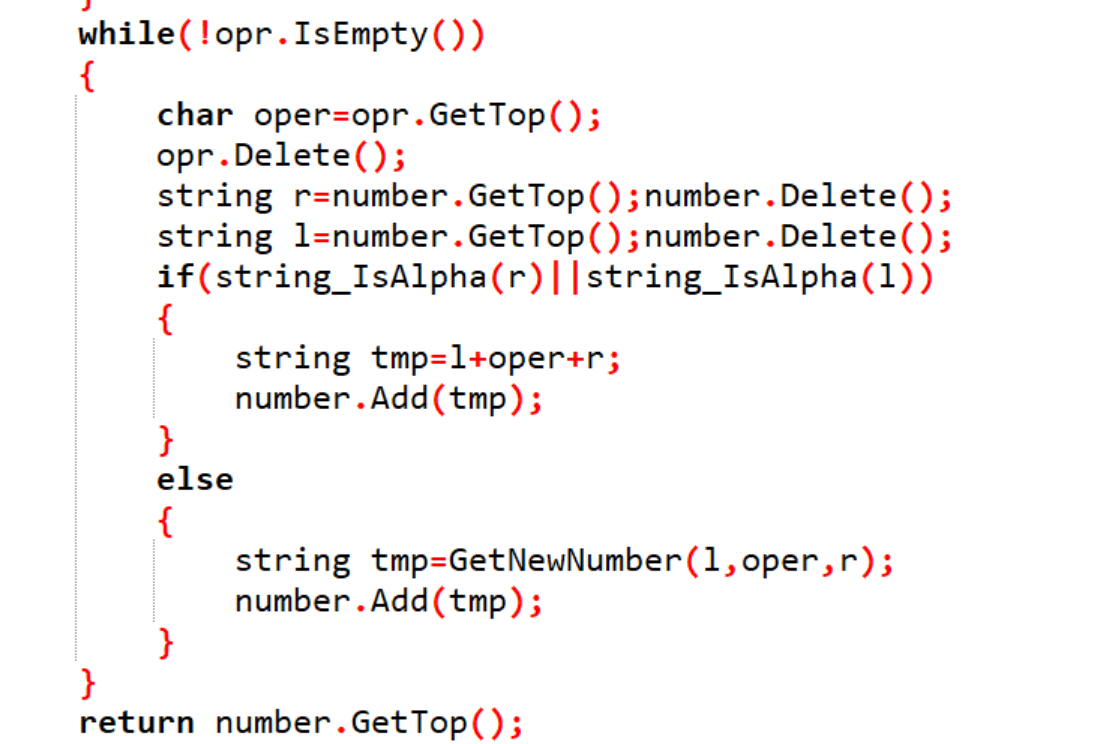


Part2.简化表达式（若为数值则计算得到值）

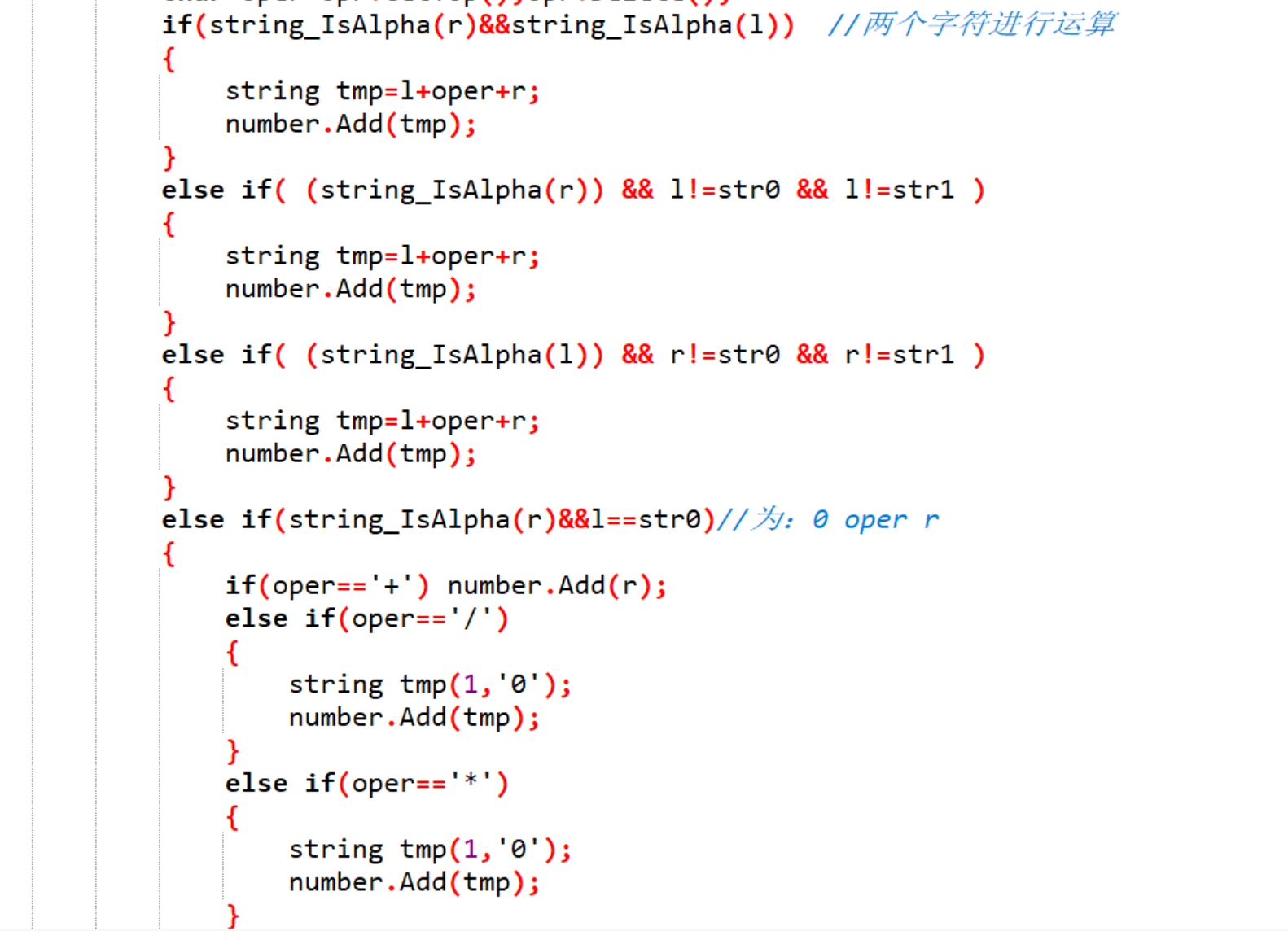


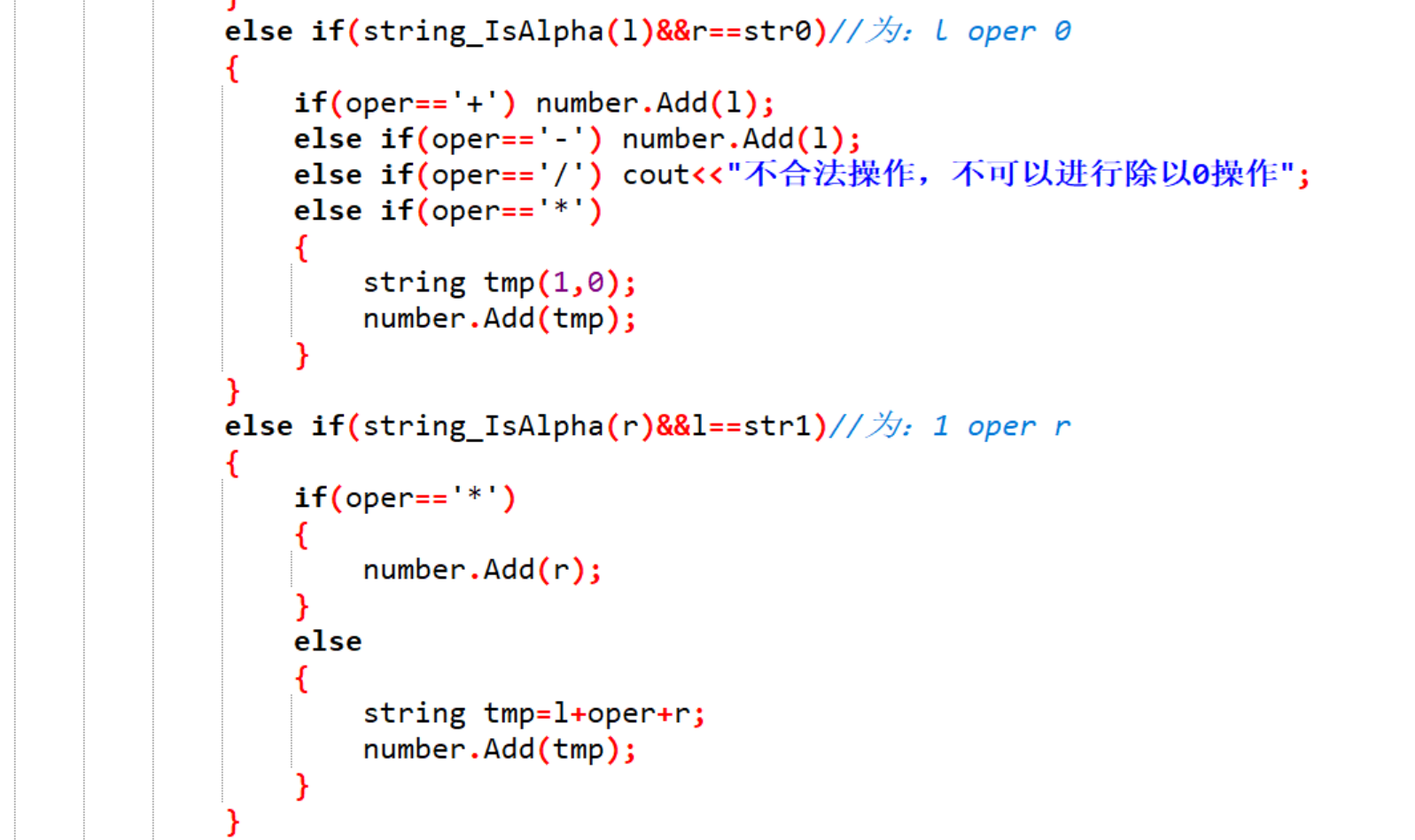


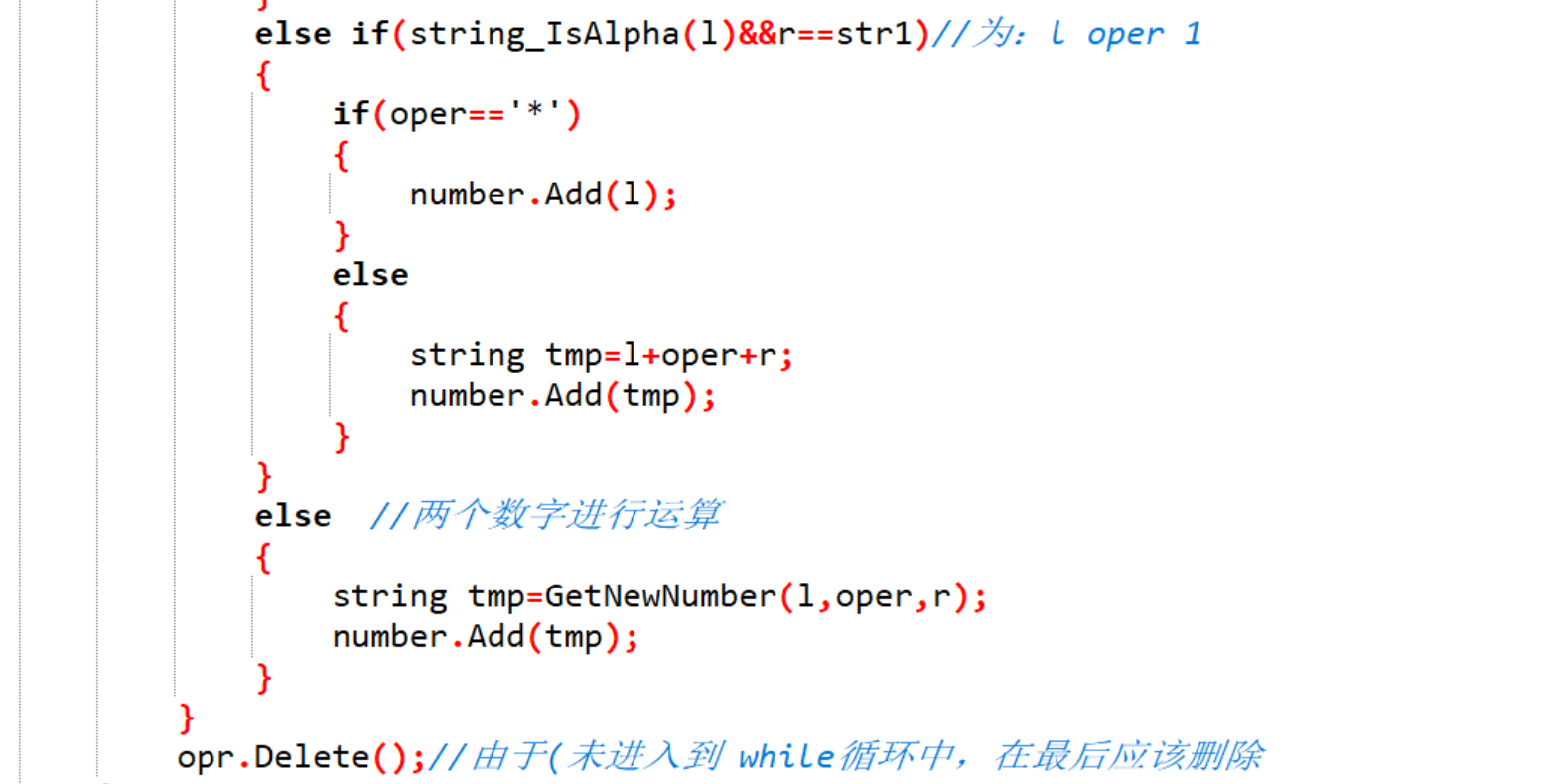




Part3.类似操作，增加判断







# 五．总结

通过本次实验，我对于栈的各项功能有了更加深入的理解：

利用栈实现了表达式的计算，让我理解了栈先进后出的性质