

# 实验3报告

19335019 陈浣

## 1. 比较静态成员与非静态成员

### (1) 修改后不会对正确性产生影响：

在这个程序中lookahead是否为静态的对于程序正确性没有影响，删除static运行结果如下所示：

```
Input an infix expression and output its postfix notation:  
1+2-5  
12+5-  
End of program.
```

### (2) 对此问题的看法：

在Java中，类的静态数据成员在全局是唯一的，该类的任何一个对象都可以直接访问该类的静态数据成员且获得相同的数据值。

在编程过程中，如果Paser类中的lookahead需要被多个所有对象同时访问，那么需要声明为static，这样可以保证数据的一致性。如果不需要被同时访问，甚至不同的Paser对象的lookahead是不同的时，则不能声明为static。

## 2. 比较消除尾递归前后程序的性能

在此程序中，将递归程序转换为等价的循环程序没有提高程序的效率。

实验验证（略）

## 3. 扩展错误处理功能

### (1) 如何准确地划分错误类型？

词法错误：

term()和rest()会对输入字符进行识别，若为不可识别的字符（数字与+、-之外的字符），则为此法错误，都会分别进行报错，并进行忽略。空格会报错说明表达式中不能出现空格，空格之外的字符会报错为非法字符。空格有专门的处理函数stripSpace()。

语法错误：

term()预期接受一个数字字符，若接受的不是数字字符，则说明前一个运算符缺少运算量。若此时的result列表还没有存储内容，则说明缺少左运算量，若有存储内容，则缺少有运算量。

rest()预期接受一个运算字符，若接受的是数字字符，则说明该运算量与前一个运算量之间缺少运算符。若接受的是空格，则说明是词法错误，调用相应的处理函数stripSpace()进行处理。若接受的是其他的字符，则说明是词法错误，用户输入了非法字符。

## (2) 如何实现错误的定位?

先将所有的输入进行存储，然后按照索引从头开始逐个处理输入的字符，当当前处理字符出现错误时，索引所指位置即为错误出现的位置。另外还设置了指向上一个尚未接受到右运算量的运算符的索引 `indexOfPre`，若到表达式结束时还是没有获得右运算量，则另外根据这一索引额外进行报错，说明其缺少右运算量。

## (3) 如何实现出错后的恢复?

- 错误类型为 表达式中含有空格 时，则忽略该输入字符，直到读取到一个非空格的字符或者是输入表达式结束。这一过程在 `stripSpace` 函数中实现。
- 错误类型为 表达式中含有非法字符 时，则忽略该输入字符，直到读取到一个合法字符或者是输入表达式结束。
- 错误类型为 两个（或以上）运算量之间缺少运算符 时，由于不能预测缺少的运算符是什么，所以对第一个运算量之后的其他运算量进行忽略，继续向后读取字符，直到读取到一个合法运算符或者是输入表达式结束。
- 错误类型为 运算符缺少左（或右）运算量 时，由于不能预测缺少的运算量是什么，所以对第一个运算符之后的非运算量输入字符进行忽略，继续向后读取字符，直到读取到一个合法运算量或者是输入表达式结束。若该运算符左边缺少运算量，则直接忽略该运算符。

## (4) 各类测试用例的屏幕截图

说明：由于 `cmp` 窗口中 `^` 显示不明显，后来出错位置的显示符修改为 `!`，截图保留原来截图，不再修改。

### 测试用例1

```
1 | ++ *[]
```

测试内容：

- 空格
- 非法字符
- 输入串到最后还是缺少左运算量

完整处理过程：

Input an infix expression and output its postfix notation:

```
++ *[]
=====
++ *[]
^
词法错误，表达式中不应有空格，已忽略该空格
=====
++ *[]
^
语法错误，缺少左运算量，已忽略此输入字符
=====
++ *[]
^
语法错误，缺少左运算量，已忽略此输入字符
=====
++ *[]
^
词法错误，表达式中不应有空格，已忽略该空格
=====
++ *[]
^
词法错误，非法运算符，已忽略该字符
=====
++ *[]
^
词法错误，非法运算符，已忽略该字符
=====
++ *[]
^
词法错误，非法运算符，已忽略该字符
=====
++ *[]
^
词法错误，表达式中不应有空格，已忽略该空格
=====
The output should be:
End of program.
```

## 测试用例2

1 | 5+ -+

测试内容：

- 空格
- 非法字符
- 运算符到最后还是缺少右运算量

完整处理过程：

Input an infix expression and output its postfix notation:

5+ -+

=====

5+ -+

^

词法错误，表达式中不应有空格，已忽略该空格

=====

5+ -+

^

词法错误，表达式中不应有空格，已忽略该空格

出现空格

=====

5+ -+

^

语法错误，前一运算符缺少右运算量，已忽略此输入字符

=====

5+ -+

^

语法错误，前一运算符缺少右运算量，已忽略此输入字符

=====

5+ -+

^

语法错误，该运算符缺少右运算量，已忽略该运算符字符

运算符到最后还是缺少右运算量

The output should be: 5

End of program.

### 测试用例3

1 | ++4+555- 6 7 8-++1++

测试内容：

- 空格
- 输入串到最后还是缺少右运算量
- 多个运算量间缺少运算符
- 缺少左运算量

完整处理过程：

Input an infix expression and output its postfix notation:

++4+555- 6 7 8-++1++

++4+555- 6 7 8-++1++

^

语法错误，缺少左运算量，已忽略此输入字符

++4+555- 6 7 8-++1++ |

^

语法错误，缺少左运算量，已忽略此输入字符

++4+555- 6 7 8-++1++

^

语法错误，该运算量与前一个运算量之间缺少运算符，已忽略该运算量

++4+555- 6 7 8-++1++

^

语法错误，该运算量与前一个运算量之间缺少运算符，已忽略该运算量

++4+555- 6 7 8-++1++

^

词法错误，表达式中不应有空格，已忽略该空格

++4+555- 6 7 8-++1++

^

词法错误，表达式中不应有空格，已忽略该空格

++4+555- 6 7 8-++1++

^

词法错误，表达式中不应有空格，已忽略该空格

++4+555- 6 7 8-++1++

^

语法错误，该运算量与前一个运算量之间缺少运算符，已忽略该运算量

++4+555- 6 7 8-++1++

^

词法错误，表达式中不应有空格，已忽略该空格

++4+555- 6 7 8-++1++

^

语法错误，该运算量与前一个运算量之间缺少运算符，已忽略该运算量

++4+555- 6 7 8-++1++

^

语法错误，前一运算符缺少右运算量，已忽略此输入字符

++4+555- 6 7 8-++1++

^

语法错误，前一运算符缺少右运算量，已忽略此输入字符

++4+555- 6 7 8-++1++

^

语法错误，前一运算符缺少右运算量，已忽略此输入字符

++4+555- 6 7 8-++1++

^

语法错误，前一运算符缺少右运算量，已忽略此输入字符

++4+555- 6 7 8-++1++

^

词法错误，表达式中不应有空格，已忽略该空格

++4+555- 6 7 8-++1++

^

语法错误，该运算符缺少右运算量，已忽略该运算符字符

The output should be: 45+6-1-

End of program.

#### 测试用例4

1 | 5+6-9+4+5-0-5+2+6+1-3

测试内容：合法输入

完整处理过程：

Input an infix expression and output its postfix notation:

5+6-9+4+5-0-5+2+6+1-3

|=====

The output should be: 56+9-4+5+0-5-2+6+1+3-

End of program.