**词法分析器实验报告**

151250209 朱晨乾

南京大学软件学院

**目录**

1. 实验目的
2. 内容描述
3. 思路方法
4. 假设
5. 相关FA描述
6. 重要数据结构
7. 核心算法
8. 运行截图
9. **实验目的**

完成一个词法分析程序，对语句进行词法分析，从而更好地理解词法分析的原理

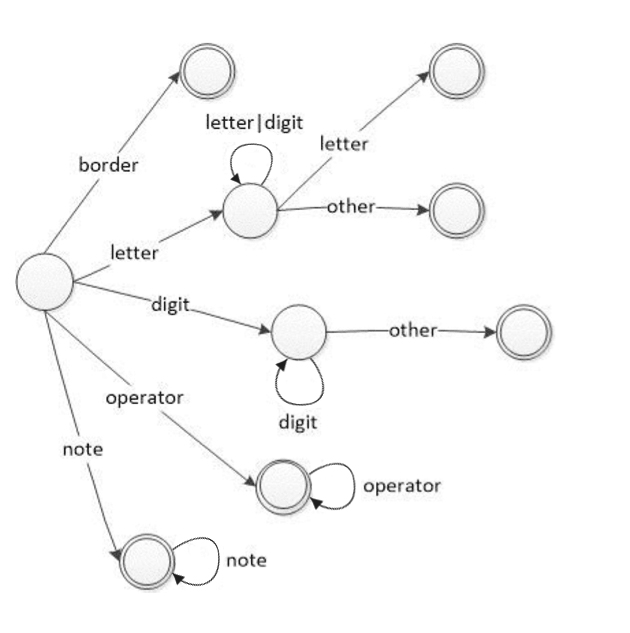
1. **内容描述**

此程序基于Java语言，程序读取一个文本文件并对其中内容进行词法分析。此程序基本实现了对Java程序的词法识别。可识别保留字、操作符、过界符、注释符、变量名（保留字优先级高于变量名），并输出格式为<种别码，识别的单词符号>的Token序列，同时对未识别字符、整形变量溢出、文件过大进行异常处理。

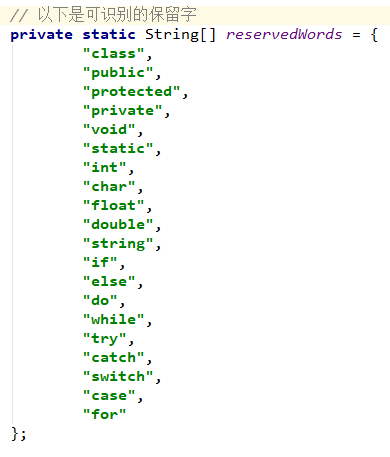
1. **思路方法**
   1. 针对要识别的单词符号写出正则表达式
   2. 构造出正则表达式对应的NFA
   3. 合并所有NFA并化简为DFA
   4. 基于DFA编写代码
   5. 代码中的具体实现：先读取一个输入字符，判断其可能的类别，再读取下一个继续判断。若已经识别出则添加到输出连，多读的字符需要退回，否则继续读取下一个字符。
2. **假设**

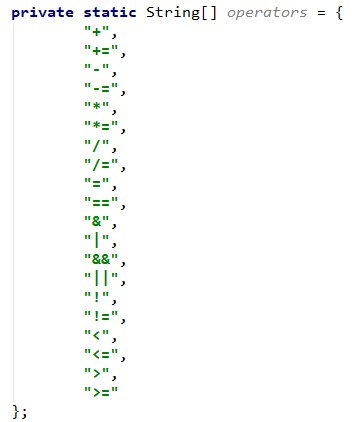
假设输入的是正常的Java程序，即包含合法的保留字和运算符。

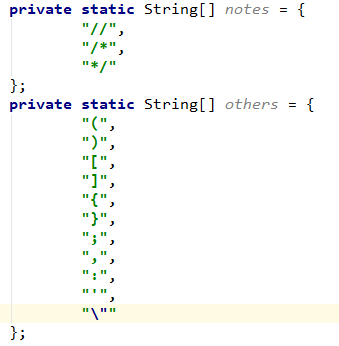
1. **相关FA描述**

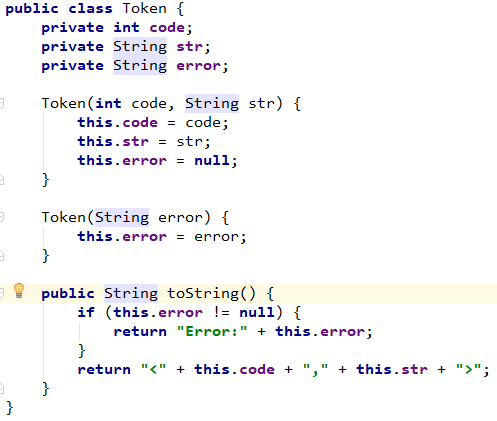


1. **重要数据结构**





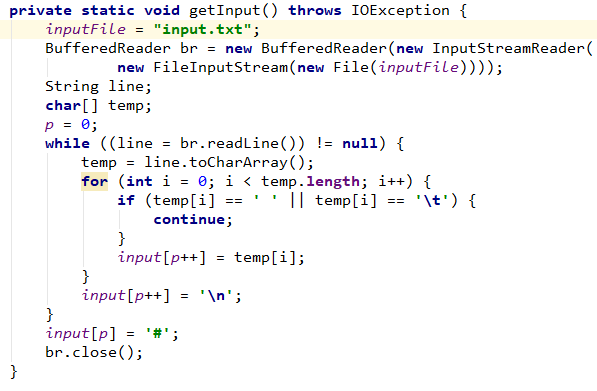


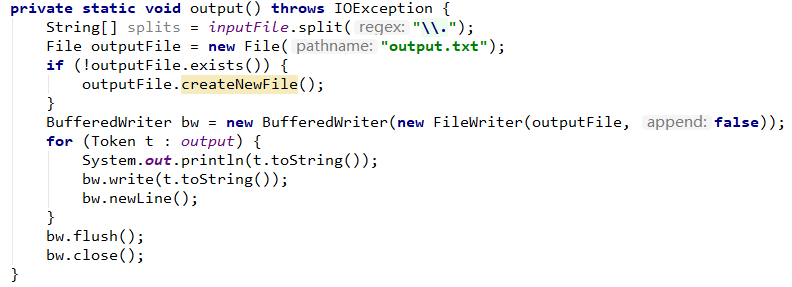


1. **核心算法**

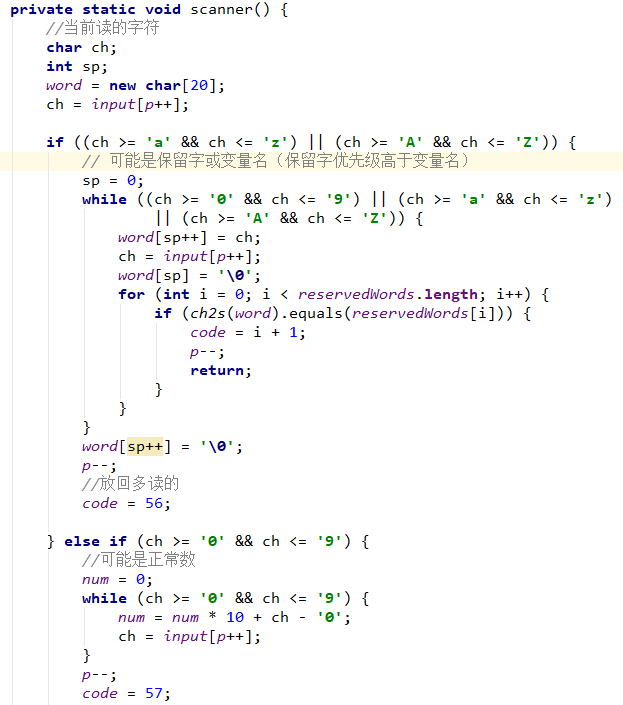
程序的主要方法：

getInput（）——读取txt文件得到输入

  
output（）——将结果输出到控制台和输出文件中



scanner（）——扫描输入进行分析，此函数为核心算法



Scanner（）方法包含了程序进行词法分析的核心算法，每次只能识别一个单词符号，所以程序通过反复调用scanner（）方法分析整个输入。

通过读取的第一个字符类型，预测接下来的单词符号可能的类型。读到英文字符，可能为保留字或变量名（类型一）；读到数字，就是常书（正数）（类型二）；读到其他字符（类型三），则可能是操作符或边界符或注释符；如果是‘-’字符，后面是数字就组成负数，当然可能是换行符或是未定义的字符。

类型一：继续读取，没读一位，都判断是否属于保留字。若是，则直接输出（因为保留字优先级高于变量名），否则一直读到不是英文字符为止，此时需要退回前一位，并输出变量名。

类型二：一直读到不是数字为止，同样需要退回一位，并输出正常数。

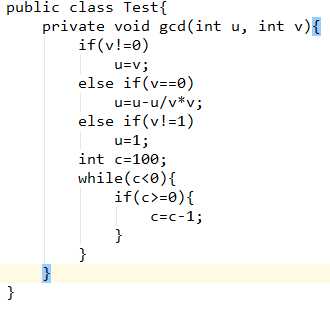
类型三：如果单字符就可以确定种别，则直接输出；否则继续读取下一位，直到可以确定种别为止；若负号（减号）后面是数字，按类型二进行读取，最后需要输出负数。同样地，多读的字符需要退回。

下面为单词符号和种别码对应表：

|  |  |  |  |
| --- | --- | --- | --- |
| 单词符号 | 种别码 | 单词符号 | 种别码 |
| Class | 1 | = | 30 |
| Public | 2 | == | 31 |
| Protected | 3 | & | 32 |
| Private | 4 | && | 33 |
| Void | 5 | | | 34 |
| Static | 6 | || | 35 |
| Int | 7 | ! | 36 |
| Char | 8 | != | 37 |
| Float | 9 | < | 38 |
| Double | 10 | <= | 39 |
| String | 11 | > | 40 |
| If | 12 | >= | 41 |
| Else | 13 | // | 42 |
| Do | 14 | /\* | 43 |
| While | 15 | \*/ | 44 |
| Try | 16 | ( | 45 |
| Catch | 17 | ) | 46 |
| Switch | 18 | [ | 47 |
| Case | 19 | ] | 48 |
| For | 20 | { | 49 |
|  | 21 | } | 50 |
| + | 22 | , | 51 |
| += | 23 | : | 52 |
| - | 24 | ; | 53 |
| -+ | 25 | ‘ | 54 |
| \* | 26 | “ | 55 |
| \*= | 27 | Letter(letter|digit)\* | 56 |
| / | 28 | Digitdigit\* | 57 |
| /= | 29 |  |  |

1. **运行截图**

Input.txt



Output.txt

