

词法分析设计文档

1.主函数内的设定

```
int main() {
    open();

    while (!getSym());
    printSym();

    close();
}
```

*open*和*close*分别负责输入输出文件流的打开和关闭，因为在整个程序当中需要从*testfile.txt*中读取文件并写入*output.txt*，因此在程序中需要打开以上两个文件，在之后的迭代当中可以修改函数当中打开的文件以及加入文件流来方便地更改目标文件，并且在结束时进行*close*，保证文件流有开启有关闭。

*getSym*是词法分析的主要程序，一次*getSym*读入一个符合当前已定义类的字符串，因此需要通过返回来判断是否已经读入完成，采用了如上的*while*内调用*getSym*的形式，而*printSym*只是输出本次的信息记录，在之后的程序中可能并不需要，因此在读入结束之后调用此函数输出内部记录的词。

2.词法分析头文件

内部定义了由类别码构成的枚举类，方便赋予读入词对应的属性，定义了记录词属性所用的数据结构，以及主函数中需要调用的词法分析内的函数。

3.词法分析函数

采用编译技术教材上所采用的设计模式。

内部首先定义了需要记录的变量，包括输入输出流，记录当前未完成词的缓冲区，当前读入字符，当前完成词的属性，词的总记录。

然后定义了两个*map*用于快速的查找保留字，以及特殊字符（空间扩大，但是代码更加整洁并且方便修改）。

读入函数首先读入一个字符然后根据此字符判断接下来可能读入的形式。首先判断当前是空白符则跳过，如果是字符则持续读入至非字符、数字和下划线，此时判断是否为保留字否则为标识符。判断得到数字后持续读入数字即为整型常量，然后与特殊字符*map*进行比对，此*map*放入了读入单个字符即可判断类别的字符，*map*中含有即结束当前词读入。然后是涉及到大于小于等需要读入下个字符才能判断是否为单字符还是双字符的词。然后判断单引号双引号，然后持续读入直到读到下一个单引号双引号结束，中间内容为字符或字符串，读入*EOF*则返回*true*用于结束主函数的读入。在结束时新建记录结构赋值并存储。

其中我认为最重要的函数为*retract*，因为很多时候需要继续读入才能判断是否结束当前次读入，但当前已经读入又会破坏下一次的起始读入，因此在读入到不属于当前词的字符时需要使用此函数回退然后结束读入。

本次额外写入了出错情况，保留以后错误分析的可能性并且有助于输出错误信息方便debug。