

词法分析程序设计

一、针对本次实验的设计

为了支持更多的数据类型和调用库函数，本次作业采用 C++ 语言编写。

因为不考虑字符串以及字符常量中出现转义字符的情况，所以可以判定源程序的任意两行之间相互独立，因此

我采用的基本思路是按行读取源码文件，每读取一行就处理一行，不必处理行于行之间存在耦合的情况，其次，每提取到一个词法成分就立即将其输出。

对于每一行源码，维护一个索引变量 ind 来定位当前处理的位置，并且在遍历每一行的时候采用 switch-case 分支结构。

step1: 提取字符串常量。当检测到 " 双引号字符时，就清空 string 类型的变量 strcon，然后用 strcon 存储字符串常量，直到遇到下一个 "

，没有考虑字符串内部出现 \ 的情况。

step2: 提取字符常量。当检测到 ' 单引号字符时，就将紧随其后的字符作为字符常量并输出，并且让 ind++ 以跳过接下来的 '，也就是说，默认"中有且仅有一个字符，并不考虑其中出现 \ 的情况。

step3: 提取符号TK。一堆类似的 case 语句，特判处理所有的符号。

step4: 提取整型常量。此处的数字都是以 string 的形式提取并存储在变量 intcon 中的。

step5: 提取标识符。标识符以 `_ | a | . . . | z | A | . . . | Z` 开头，并以 `_ | a | . . . | z | A | . . . | Z | 1 | . . . | 9` 作为组成。提取出标识符后，需要判断是否是保留字，我事先采用 map 数据结构用来存储了所有 <保留字, 类别码>键值对，只需要将提取的标识符转换成小写后查找 map，根据是否是保留字来对应输出相关的信息。

空白字符的处理：此处我考虑的空白字符只有 '\t' 和 ' ' 两种，专门写了这样一个函数来跳过多余的空白字符：

```
1  int getNewIndex(int start, std::string line)
2  {
3      int i = start;
4      while (line[i] == '\t' || line[i] == ' ') i++;
5      return i;
6  }
```

二、针对以后实验的预留

本次作业的程序仅采用了简单的循环结构和分支结构，没有较为复杂的写法，也没有面向对象，并且代码长度相对较短（<200行），因此可修改的灵活性较高。再者，由于课程进度暂时还没讲到符号表管理和错误处理，因此，还不能准确预测今后具体要做哪些修改，断然预留一些接口显得很牵强，而且容易出错，于是在此只做一些简单的分析预测：

- **符号表管理**：需要根据之后的具体需求来建立合适的数据结构来保存标识符以及其对应的数据类型和值，这个数据需要保证便于添加和查找，可以考虑面向对象建类或者建立结构体，并且需要能够进行高效的查重，以进行错误处理。
- **错误处理**：
 1. 标识符命名是否符合规范
 2. 是否在单词之间有空白符分隔
 3. 是否有对不同数据类型的变量进行相同的命名
 4. ...还有很多细枝末节的小规范，需要根据之后实验的具体要求进行的添加处理，在此无法具体预测。