

实验一 词法分析器

1、实验名称

词法分析器编程实验

2、动机/目标

本实验旨在设计并实现一个简单的词法分析器,用于将源代码分解为关键字、标识符、数字、运算符、分隔符等词法单元。

3、内容描述

实验内容包括实现基本的词法分析功能,能够识别并分类不同类型的词法单元,并输出相应的结果。

4、思路/方法

1. 使用 C++ 编程语言编写词法分析器。
2. 定义关键字、分隔符、运算符和过滤符的集合。
3. 实现判断函数用于识别字符或字符串的类别。
4. 逐字符读取源代码,依据判断函数分类处理。

5、假设

1. 输入文件包含了简单的源代码,格式合法。
2. 不考虑复杂的多字符运算符和注释。

6、相关有限自动机描述

实验中的词法分析器相当于实现了一个简单的有限自动机,通过状态转换来识别和分类不同的词法单元。

7、重要数据结构描述

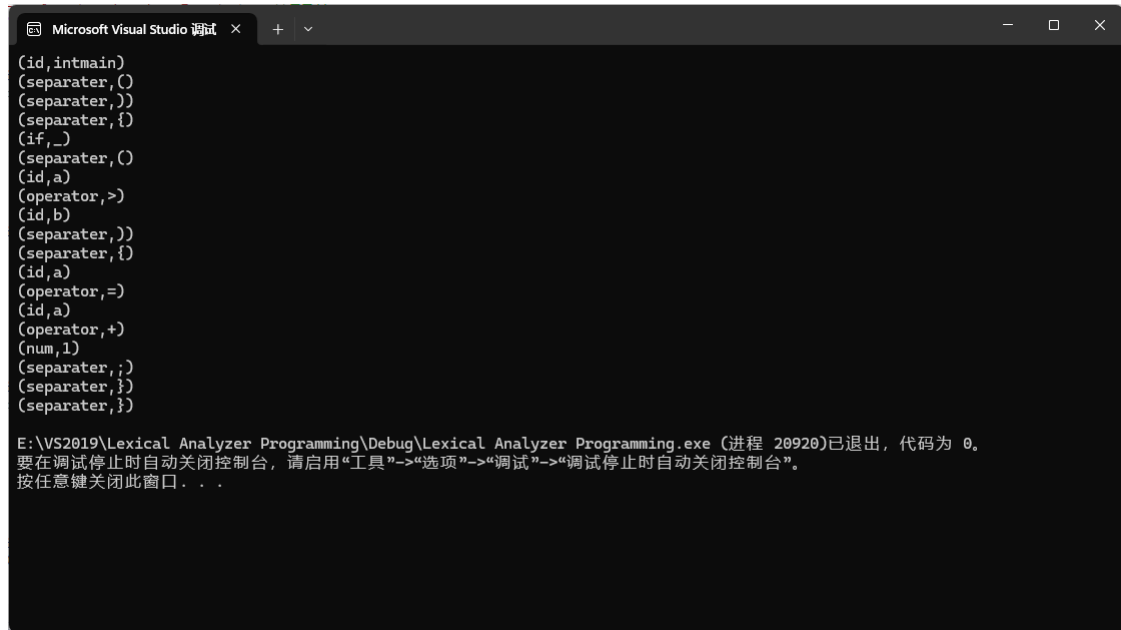
- string KEYWORD[5]: 存储关键字。
- char SEPARATER[5]: 存储分隔符。
- char OPERATOR[8]: 存储运算符。
- char FILTER[4]: 存储过滤符。

8、核心算法描述

1. 逐字符读取输入。
2. 根据当前字符类型调用相应的判断函数。
3. 生成并输出对应的词法单元。
4. 处理过程中跳过过滤符,识别错误字符时输出提示。

9、运行案例

```
int main() {  
    if (a > b) {  
        a = a + 1;  
    }  
}
```



```
(id,intmain)
(separator,())
(separator,))
(separator,{)
(if,_)
(separator,())
(id,a)
(operator,>)
(id,b)
(separator,))
(separator,{)
(id,a)
(operator,=)
(id,a)
(operator,+)
(num,1)
(separator,;)
(separator,{)
(separator,{)

E:\VS2019\Lexical Analyzer Programming\Debug\Lexical Analyzer Programming.exe (进程 20920)已退出, 代码为 0。
要在调试停止时自动关闭控制台, 请启用“工具”->“选项”->“调试”->“调试停止时自动关闭控制台”。
按任意键关闭此窗口...
```

10、出现的问题及相关解决方案

1. 问题：无法识别复杂运算符和注释。
解决方案：未来可以扩展支持多字符运算符和注释。
2. 问题：代码逻辑中字符读取存在冗余。
解决方案：优化了读取逻辑，避免重复读取。

11、个人感想

通过本次实验，我深刻理解了词法分析器的基本原理和实现过程。编写代码过程中遇到了一些困难，但通过不断调试和改进，最终实现了目标。