词法分析器

10011603 2016302731 冯宇

**一.主要功能**

实现对程序的阅读，区分出关键字和一般变量以及数字。程序能将文件中的变量识别并输出。

**二.实现代码**

**重要变量：**

在main.h中

typedef enum {

identi, // 0

constsy, // 1

intsy, charsy, // 2, 3

voidsy, // 4

intcon, charcon, stringcon, // 5, 6, 7

plus, minus, times, idiv, // 8, 9, 10, 11

eql, neq, gtr, geq, lss, leq, // 12, 13, 14, 15, 16, 17

becomes, // 18

lparent, rparent, lbrack, rbrack, lbrace, rbrace, // 19, 20, 21, 22, 23, 24

comma, semicolon, colon, // 25, 26, 27

ifsy, elsesy, // 28, 29

switchsy, casesy, defaultsy, // 30, 31, 32

forsy, // 33

scanfsy, printfsy, // 34, 35

mainsy, // 36

returnsy, // 37

} symbol; // 利用枚举类举出所有关键字、运算符和标点符号的类型

typedef char alpha[maxIdentLength + 1]; //alpha类型定义为char类型的数组，且有大小限制

在symbol.cpp中

alpha keywords[keywordCount]; //存储关键字的char形式

symbol keySymbols[keywordCount];//存储关键字的枚举类

map<char, symbol> specialSymbols;//存储运算符和标点符号的map<char,symbol>

**关键函数**

/\*

输入字符判断是否为空字符

输入：char ch 字符

输出：1为空字符 0非空字符

\*/

int isspace(char ch)

/\*

如果读取到一行的某位读取一行代码进line[]，并且保存最新的char进ch

否则，读取最新的char进ch

输入：(文本数据流)

输出：（ch）

\*/

void nextch()

/\*

读取程序中的量,ch永远为现读取量的下一个字符，因为在每块最后都有nextch();

如果为变量，变量名保存到token[]，sy=identi,

如果为关键字，token[]不更新，sy更新为对应的枚举类

如果为数字，数字保存到token[]中并计算值保存到inum中

如果为运算符或者逻辑符号或者标点符号，token[]不更新，sy更新为对应的枚举类

\*/

void insymbol(){

while (isspace(ch))读取下一个字符

if (isLetter(ch))读取整个变量名到token并判断是变量名还是关键字，更新sy

else if (isdigit(ch))}{读取为数字，数字保存到token[]中并计算值保存到inum中}

else //读取为逻辑符号，sy=枚举类

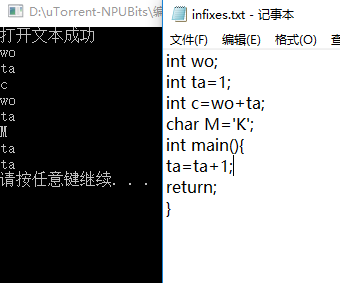
else {读取到“”循环读取}

else{读取到运算符号或者标点符号，sy=specialSymbols[ch]; }

else error(UnknownWord);读取下一个词

}

**三.测试结果**



**四.Debug以及C++知识点**

**1.枚举类enum**

一般枚举类声明就从0开始

enum colors{blue=5,black=3,red,green};

这样我们就修改了枚举成员值，之后的red，green都是按blue的值依次加一。

枚举变量只能使用同类型的枚举成员和变量赋值，不能使用整型常量。

**2.map类型**

#include <map>

当中map 的元素以键－值（key-value）对的形式组织：键用作元素在 map 中的索引，而值则表示所存储和读取的数据。set 仅包括一个键。并有效地支持关于某个键是否存在的查询

插入 ：改变map中的条目非常简单，因为map类已经对[]操作符进行了重载

enumMap[1] = "One";

enumMap.insert(map<int, CString> :: value\_type(2, "Two"))

获得一个值

CString tmp = enumMap[2];

但是,只有当map中有这个键的实例时才对，否则会自动插入一个实例，值为初始化值。我们可以使用Find()和Count()方法来发现一个键是否存在。 查找map中是否包含某个关键字条目用find()方法，传入的参数是要查找的key，在这里需要提到的是begin()和end()两个成员，分别代表map对象中第一个条目和最后一个条目，这两个数据的类型是iterator. 这个结构体在后续代码中出现次数很多。

1) map最基本的构造函数；

map<string , int >mapstring;

map<int ,string >mapint;

map<sring, char>mapstring;

map< char ,string>mapchar;

map<char ,int>mapchar;

map<int ,char >mapint;

2) map添加数据；

map<int ,string> maplive;

1.maplive.insert(pair<int,string>(102,"aclive"));

2.maplive.insert(map<int,string>::value\_type(321,"hai"));

3, maplive[112]="April";//map中最简单最常用的插入添加！

3)map中元素的查找：

find()函数返回一个迭代器指向键值为key的元素，如果没找到就返回指向map尾部的迭代器。

map<int ,string >::iterator l\_it;;

l\_it=maplive.find(112);

if(l\_it==maplive.end())

cout<<"we do not find 112"<<endl;

else cout<<"wo find 112"<<endl;

4)map中元素的删除：

如果删除112；

map<int ,string >::iterator l\_it;

l\_it=maplive.find(112);

if(l\_it==maplive.end())

    cout<<"we do not find 112"<<endl;

else

    maplive.erase(l\_it);

    //delete 112;