Lexical_analyzer.h

```
  //前置运算符
  set<string> PRE_OPERATORS = {"+", "-"};

  //界符
  set<char> BORDERS = {'(', ')', '{', '}', ',', ';'};

  ifstream code_reader_;
  ofstream lexical_analyser_printer_;

  unsigned int line_counter_;                          //用于词法分析发生错误时的行数定位
  bool print_detail_;                                  //选择是否将词法分析结果输出到txt中
  unsigned int step_counter_;                          //词法分析计步器
  bool IsDot(const unsigned char ch);                  //是否为小数点
  bool IsLetter(const unsigned char ch);               //是否字母
  bool IsDigital(const unsigned char ch);              //是否数字
  bool IsSingleCharOperator(const unsigned char ch);   //是否单符号运算符
  bool IsDoubleCharOperatorPre(const unsigned char ch); //是否双符号运算符
  bool IsBlank(const unsigned char ch);                //是否是空白符
  unsigned char GetNextChar();                         //获取字符流中的下一个字符，同时计算行数。
  void PrintDetail(WordInfo word);                     //打印词法分析信息
  WordInfo GetBasicWord();                             //获取文法符号,但是无法进行打印
```

Lexical_analyzer.cpp

```
bool IsDot(const unsigned char ch)
{
    if (ch == '.')
        return true;
    else
        return false;
}
```

```
WordInfo LexicalAnalyzer::GetBasicWord()
```

在这个函数中添加的部分：

```
WordInfo LexicalAnalyzer::GetBasicWord()
{
    unsigned char ch;
    string str_token;
    WordInfo word;
    while (!code_reader_.eof())
    {
        int flag = 0; //区分int 和 float的标志位
        ch = GetNextChar();
        if ('/' == ch) //处理注释
        {…
        }
        else if (IsDigital(ch)) //可能是小数float或者整数int
        {
            str_token += ch;
            while (!code_reader_.eof())
            {
```

```
else if (IsDigital(ch)) //可能是小数float或者整数int
{
    str_token += ch;
    while (!code_reader_.eof())
    {
        unsigned char next = GetNextChar();
        if (IsDigital(next)) //下一个是数字
        {
            str_token += ch;
        }
        else //下一个不是数字
        {
            if (ch == '.') //是小数，继续循环判断后面的数字
            {
                str_token += ch;
                flag = 1; //是小数
                continue;
            }
            else //不是小数,是int型数
            {
                code_reader_.seekg(-1, ios::cur); //输入文件指针回退,因为多读了next
                word.value = str_token;
                if (flag == 0)
                    word.type = LCINT; //int数
                else if (flag == 1)
                    word.type = LCFLOAT; //float数
                return word;
            }
        }
    }
}
```

```
else if (ch == '\'') //开头是字符型标志\'
{
    str_token += ch;
    while (!code_reader_.eof())
    {
        unsigned char next = GetNextChar();
        while (!code_reader_.eof())
        {
            if (IsDigital(ch) || IsLetter(ch) || IsBlank(ch)) //数字，字母，' ',\n,\t,255
                str_token += ch;
            else if (ch == '\'') //\'(字符结束标志)
                str_token += ch;
            else
            {
                code_reader_.seekg(-1, ios::cur); //输入文件指针回退,因为多读了next
                word.value = str_token;
                word.type = LCCHAR; //char型常量
                return word;
            }
        }
    }
}
```