

编译原理课程实验报告

实验 1：词法分析

姓名	张志路	院系	计算机学院	学号	1160300909	
任课教师	辛明影		指导教师	辛明影		
实验地点	格物 208		实验时间	2019 年 4 月 14 日周日 5-6 节		
实验课表现	出勤、表现得分		实验报告 得分		实验总分	
	操作结果得分					
一、需求分析					得分	
要求：阐述词法分析系统所要完成的功能						
1. 设计实现类高级语言的词法分析器，基本功能为识别以下几类单词：						
(1) 标识符						
由大小写字母、数字以及下划线组成，但必须以字母或者下划线开头。						
(2) 关键字						
①类型关键字：整型、浮点型、布尔型、记录型；						
②分支结构中的 if 和 else；						
③循环结构中的 do 和 while。						
(3) 运算符						
①算术运算符；						
②关系运算符；						
③逻辑运算。						
(4) 界符						
①用于赋值语句的界符，如 “=”；						
②用于句子结尾的界符，如 “；”。						
(5) 常数						
无符号整数和浮点数等。						
(6) 注释						
/*.....*/形式。						
2. 具体要求						
(1) 要求基于 DFA 技术设计词法分析器。						
(2) 系统的输入形式：要求能够通过文件导入测试用，测试用例要涵盖“实验内容”中列出的各类单词，并包含各种单词拼写错误。						
(3) 系统的输出分为两部分：一部分是打印输出词法分析器的符号表，另一部分是打印输出源程序对应的 token 序列。						
(4) 在基本要求的基础上，本程序又实现了识别单个字符（char）和字符串（String）的算法。						
二、文法设计					得分	
要求：对如下内容展开描述						
1. 给出各类单词的词法规则描述（正则文法或正则表达式）。						
令 letter_ = [A-Za-z_], digit = [0-9]。						
(1) 标识符						
letter_(letter_ digit)*						

(2) 关键字

char | long | short | float | double | const | Boolean | void | null | false | true | enum | int |
do | while | if | else | for | then | break | continue | class | static | final | extends | new |
return | signed | struct | union | unsigned | goto | switch | case | default | auto | extern |
register | sizeof | typedef | volatile | String

(3) 运算符

< | > | <= | >= | == | != | + | - | * | / | = | ++ | -- | << | >> | || | && | & | || | ! | ^ | % | += | -
= | *= | /=

(4) 界符

() | { | } | ; | [|] | , | =

(5) 无符号常数

digit digit* (\.digit digit*)? ([Ee][+-]? digit digit*)?

(6) 注释

/* */型注释: $\wedge^*(.\backslash r\backslash n)^*\wedge^*/$

//型注释: $//[\backslash s\backslash S]^*\backslash n$

(7) 单引号内的字符 (char)

$\backslash (.)'$

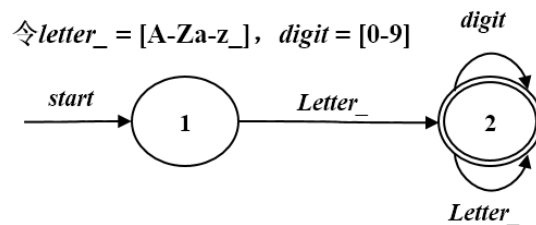
(8) 双引号内的字符串 (String)

$\backslash "(. *?)\backslash "$

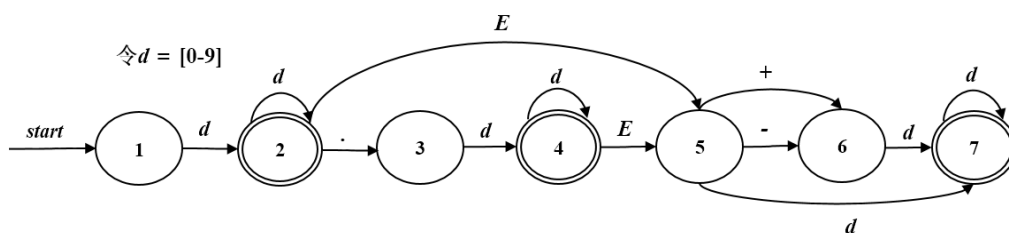
2. 各类单词的转换图

关键字和界符的判断比较简单, DFA 不再给出。

(1) 标识符

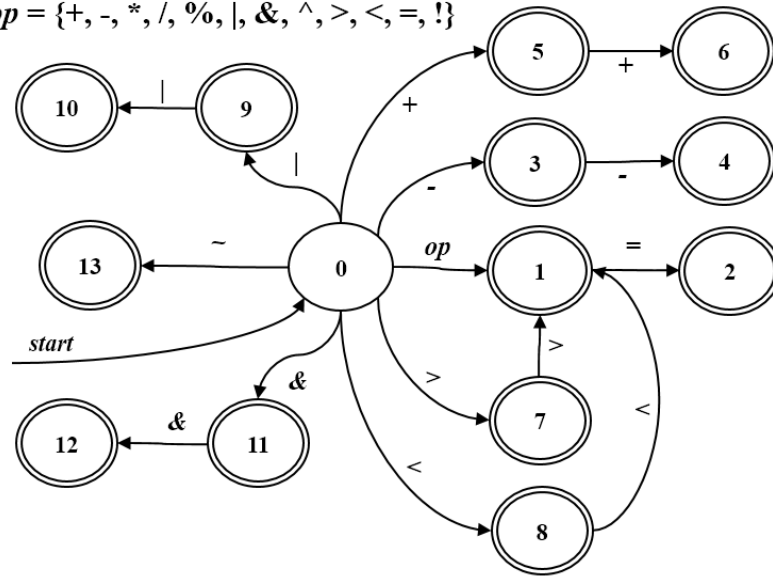


(2) 无符号常数



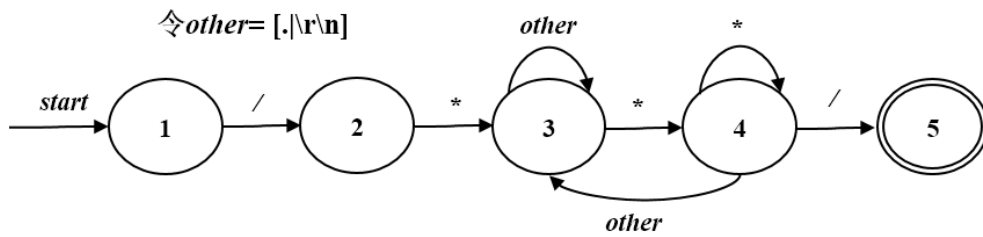
(3) 运算符

令 $op = \{+, -, *, /, \%, |, \&, ^, >, <, =, !\}$

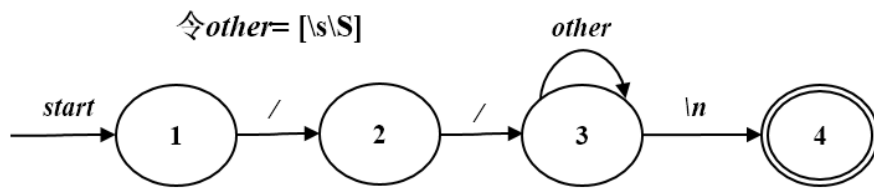


(4) 注释

/* */型注释



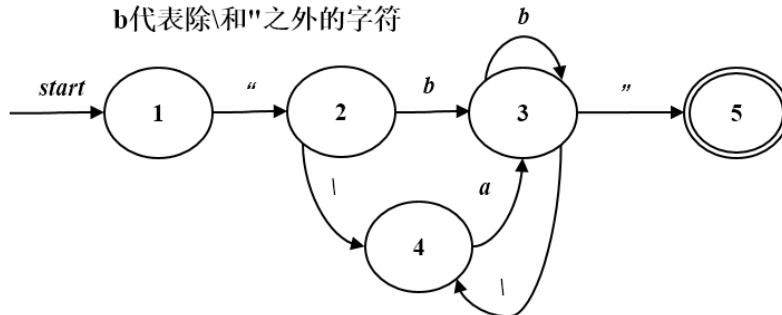
//型注释



(5) 字符串

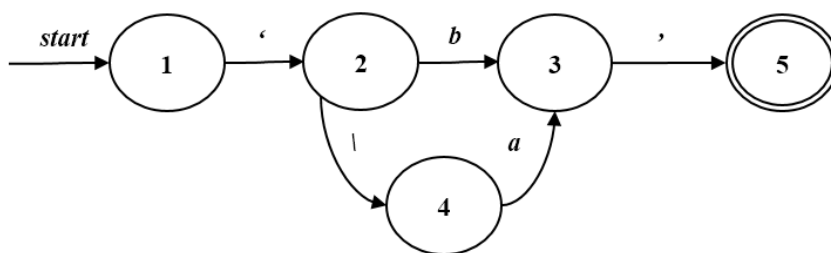
a代表任意字符

b代表除\和"之外的字符



(6) 单个字符

a代表任意字符
b代表除\和'之外的字符



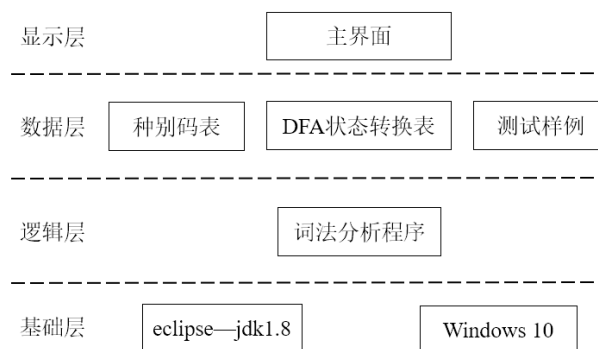
三、系统设计

得分

要求：分为系统概要设计和系统详细设计。

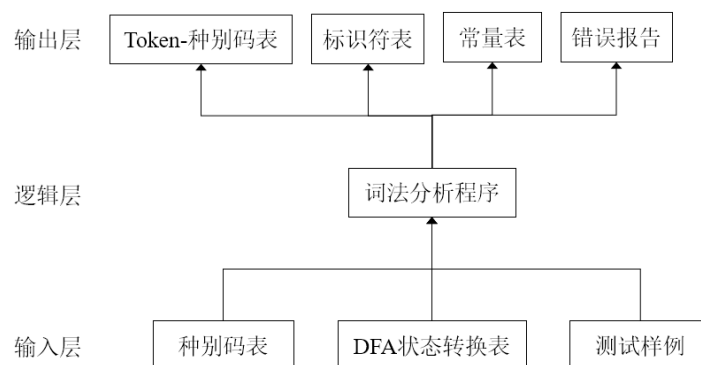
1. 系统概要设计：给出必要的系统宏观层面设计图，如系统框架图、数据流图、功能模块图等以及相应的文字说明。

(1) 系统框架图



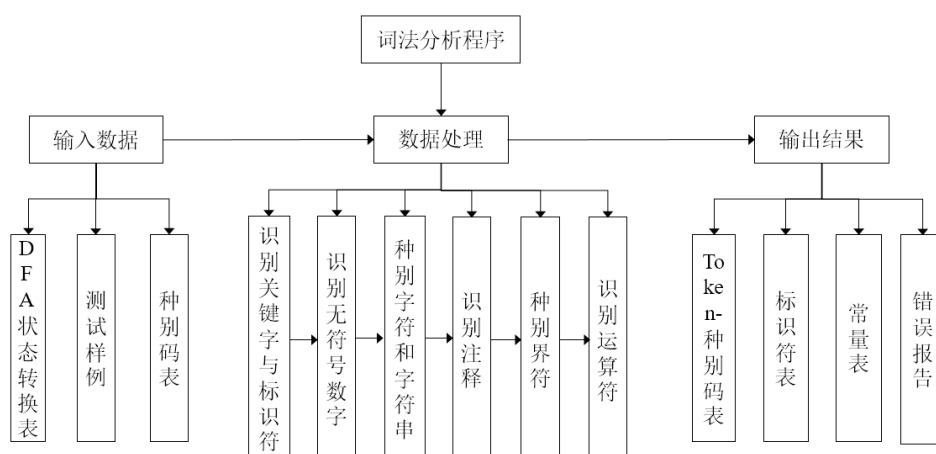
系统分为基础层、逻辑层、数据层和显示层，基础层即在 Windows10、jdk1.8 环境下利用 java 语言进行编程，逻辑层即词法分析程序，数据层包含 DFA 转换表、测试样例和设计的种别码表，显示层显示一个主界面。

(2) 数据流图



词法分析程序通过读入 DFA 转换表和设计的种别码表，对测试样例进行分析，从而对结果进行显示，给出 Token-种别码表、标识符表、常量表和错误报告。

(3) 功能模块图



如上图所示，词法分析程序的功能主要为读入数据、分析数据和给出结果。分析数据的过程即对各类单词进行识别。

2. 系统详细设计：对如下工作进行展开描述

(1) 核心数据结构的设计

```
private String text; // 读入的测试样例文本
private JTable jtable1; // 实为Object[][]数组，存储识别信息，即“行数-Token-种别码-单词类别”
private JTable jtable2; // 实为Object[][]数组，存储错误报告，即“行数-错误内容-错误信息”
private JTable jtable3; // Object[][]数组，存储的是标识符表，即“标识符-标识符位置”
private JTable jtable4; // Object[][]数组，存储的是常量表，即“常量-常量位置”

/**
 * 构造函数
 * @param text String型，读入的测试样例文本
 * @param jtable1 实为Object[][]数组，存储识别信息，即“行数-Token-单词类别-种别码”
 * @param jtable2 实为Object[][]数组，存储错误报告，即“行数-错误内容-错误信息”
 * @param jtable3 实为Object[][]数组，存储的是标识符表，即“标识符-标识符位置”
 * @param jtable4 实为Object[][]数组，存储的是常量表，即“常量-常量位置”
 */
public Lexical(String text, JTable jtable1, JTable jtable2, JTable jtable3, JTable jtable4)
{
    this.text = text;
    this.jtable1 = jtable1;
    this.jtable2 = jtable2;
    this.jtable3 = jtable3;
    this.jtable4 = jtable4;
}
```

(2) 主要功能函数说明

a. lex 函数

```
/**
 * 核心函数
 * 根据已经构成的DFA状态转换表
 * 按行分析数据，识别相应信息
 */
public void lex()
```

该函数是程序的核心函数，其程序流程图如（3）所示，该函数识别各类单词，并输出相应信息和错误报告。

b. is_string_state 函数

```
/**
 * 字符串DFA状态匹配函数
 * @param ch 当前字符
 * @param key 状态表中的字符
 * @return 匹配成功返回true, 否则返回false
 */
public static Boolean is_string_state(char ch, char key)
```

c. is_char_state 函数

```
/**
 * 字符DFA状态匹配函数
 * @param ch 当前字符
 * @param key 状态表中的字符
 * @return 匹配成功返回true, 否则返回false
 */
public static Boolean is_char_state(char ch, char key)
```

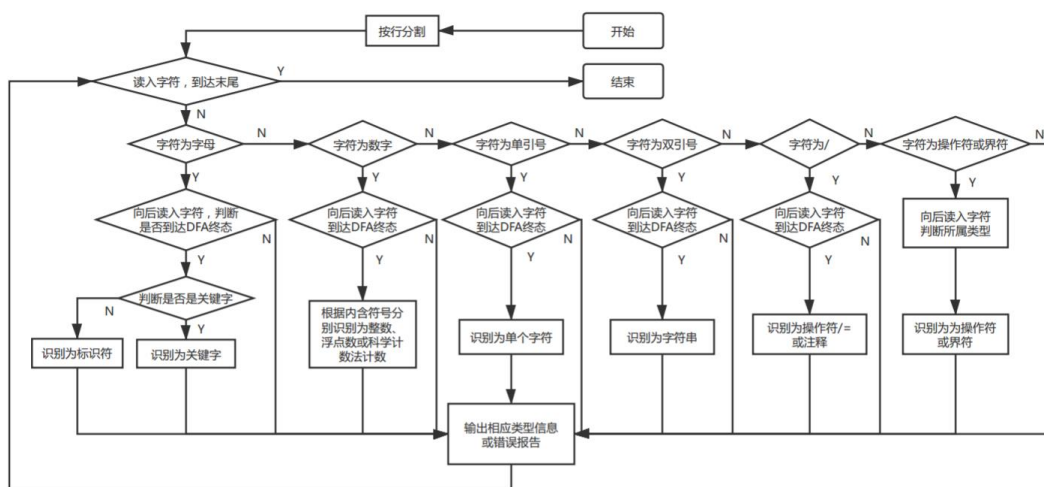
d. is_digit_state 函数

```
/**
 * 数字DFA状态匹配函数
 * @param ch 当前字符
 * @param test 状态表中的字符
 * @return 匹配成功返回true, 否则返回false
 */
public static int is_digit_state(char ch, char test)
```

e. is_note_state 函数

```
/**
 * 注释DFA状态匹配函数
 * @param ch 当前字符
 * @param test 状态表中的字符
 * @return 匹配成功返回true, 否则返回false
 */
public static Boolean is_note_state(char ch, char nD, int s)
```

(3) 程序核心部分的程序流程图



四、系统实现及结果分析	得分	
<p>要求：对如下内容展开描述。</p> <p>1. 系统实现过程中遇到的问题；</p> <p>(1) 种别码编码问题（具体编码见附录）</p> <p>编码的清晰与否将会直接影响后续程序的实现，经过权衡，本系统对标识符关键字、无符号数、注释等大类编码为 1~7；关键字一字一码，以 101 开头；运算符一符一码，以 201 开头；界符一符一码，以 301 开头。之所以编码不连续是为了方便后续对其进行扩展。</p> <p>(2) 常量与标识符的记录问题</p> <p>常量与标识符在编码时都是以大类编码的，不同单词之间未进行区分。为了后续便于语法语义分析，本系统构造了两个 HashMap，分别存储标识符和常量的位置信息，其格式均为“单词—位置”形式。</p> <p>(3) DFA 设计与实现问题</p> <p>关于 DFA 的状态转换表，本系统直接在程序中定义，进而构造相应的转换程序进行状态之间的转移。</p> <p>(4) 无符号数识别问题</p> <p>无符号数对其进行了整形、浮点型和科学计数法的区分。在识别科学计数法的无符号数时，要注意 E 的大小写不影响其表达，例如 1.2E-1=1.2e-1；同时加号的存在与否也不影响表达，例如 1.2E+1=1.2E1。</p> <p>(5) 注释识别问题</p> <p>在进行多行注释识别时，要注意换行符的处理问题，该问题可以通过对 DFA 状态转换表的设计来解决。</p> <p>(6) 错误报告问题</p> <p>在进行错误的识别时，本着“就近”的原则确定其错误类型。例如在进行无符号数的识别时，如果出现“12m”类似的形式，则认为这是一个无符号数错误，而不是标识符错误。</p> <p>2. 针对某测试程序输出其词法分析结果</p> <p>测试样例如下。</p> <pre>1. while(num != 100) 2. { 3. num++; 4. } 5. 6. /*test*/ 7. //test 8. float a = 1.22; 9. float b = 2.32e+1; 10. float c = 22.3E-1; 11. int d = 22E1; 12. s[1] = 'a'; 13. char s = '\t'; 14. char s = '\';</pre>		

```

15. String t = "ZhangZhilu";
16. /*Zhang Zhilu
17. 1160300909
18. */
19.
20. /* Sort function */
21. void bubbleSort(int *arr, int n)
22. {
23.     for (int i = 0; i < n-1; i++)
24.         for (int j = 0; j < n-i-1; j++)
25.             {
26.                 if (arr[j] > arr[j+1])
27.                 {
28.                     int temp = arr[j];
29.                     arr[j] = arr[j+1];
30.                     arr[j+1] = temp;
31.                 }
32.             }
33. }
34.
35. int main()
36. {
37.     int arr[] = { 10,6,5,2,3,8,7,4,9,1 };
38.     bubbleSort(arr, n);
39.     return 0;
40. }
41.
42.
43. /*Error*/
44. int 1_qqq = 3;
45. float e = 2.2.2;
46. float t = 1..1;
47. char s = '\t;
48. String ff = "zhangzhilu;
49. /*zhang
50. zhi
51. lu

```

注：1~43 行为无词法错误程序，44~51 行均有词法错误。

输出的部分“行号—Token—类别—种别码”表如下。

词法分析程序			
行号	Token	类别	种别码
1	while	关键字	131
1	(符号	307
1	num	标识符	1
1	!=	运算符	213
1	100	常量	2
1)	符号	308
2	{	符号	305
3	num	标识符	1
3	++	运算符	206
3	;	符号	302
4	;	符号	306
6	/*test*/	注释	7
7	//test	注释	7
8	float	关键字	118
8	a	标识符	1
8	=	符号	309
8	1.22	浮点型常量	3
8	;	符号	302
9	float	关键字	118
9	b	标识符	1
9	=	符号	309
9	2.32e+1	科学计数法	4
9	;	符号	302
10	float	关键字	118
10	c	标识符	1
10	=	符号	309
10	22.3E-1	科学计数法	4
10	;	符号	302

输出的部分“标识符—位置”和“常量—位置”表如下。

标识符	位置	常量	位置
num	0	100	0
a	1	1.22	1
b	2	2.32e+1	2
c	3	22.3E-1	3
d	4	22E1	4
s	5	1	5
t	6	'a'	6
bubble...	7	'\t'	7
arr	8	'\n'	8
n	9	"Zhang..."	9
i	10	0	10
j	11	10	11
temp	12	6	12
main	13	5	13
e	14	2	14
ff	15	3	15
		0	16

- 输出针对此测试程序对应的词法错误报告
针对以下程序的错误报告如下。

```

43. /*Error*/
44. int 1_qqq = 3;
45. float e = 2.2.2;
46. float t = 1..1;
47. char s = '\t;
48. String ff = "zhangzhilu;
49. /*zhang
50. zhi
51. lu

```

错误行号	Token	详细说明
44	1 qqq	无符号数不合规
45	2. 2. 2	无符号数不合规
46	1. . 1	无符号数不合规
47	'\t	字符常量引号未封闭
48	"zhangzhilu;	字符串常量引号未封闭
52	/*zhangzhilu	注释未封闭

4. 对实验结果进行分析

经过反复地测试与实验，从最后的实验结果来看，该系统对各类单词均进行了细致且正确的识别，例如无符号数“22.3E-1”、字符“‘\t’”、运算符“++”和各类注释。

同时也对一些常见的错误进行了识别，例如错误的无符号数“2.2.2”、错误的字符“‘\t’”和注释未封闭的问题。

结果表明，系统的结果与目标基本一致，系统目标达成。

程序运行的主界面如下图所示。

词法分析程序

```

1 while(num != 100)
2 {
3     num++;
4 }
5
6 /*test*/
7 //test
8 float a = 1.22;
9 float b = 2.32e+1;
10 float c = 22.3E-1;
11 int d = 22E1;
12 s[1] = 'a';
13 char s = '\t';
14 char s = '\t';
15 String t = "ZhangZhilu";
16 /*Zhang Zhilu
17 1160300909
18 */

```

词法分析

打开文件

清空文本

标识符	位置
num	0
a	1
b	2
c	3
d	4
s	5
t	6
bubble...	7
arr	8
n	9
i	10
j	11
temp	12
main	13
e	14
ff	15

常量	位置
100	0
1. 22	1
2. 32e+1	2
22. 3E-1	3
22E1	4
1	5
'a'	6
'\t'	7
'\t'	8
"Zhang...	9
0	10
10	11
6	12
5	13
2	14
3	15

行号	Token	类别	种别码
1	while	关键字	131
1	(界符	307
1	num	标识符	1
1	!=	运算符	213
1	100	整型常量	2
1)	界符	308
2	{	界符	305
3	num	标识符	1
3	++	运算符	206
3	;	界符	302
4	}	界符	306
6	/*test*/	注释	7
7	//test	注释	7
8	float	关键字	118
8	a	标识符	1
8	=	运算符	309
8	1. 22	浮点型常量	3
8	.	界符	302
9	float	关键字	118
9	b	标识符	1
9	=	运算符	309
9	2. 32e+1	科学计数法	4
9	.	界符	302
10	float	关键字	118
10	c	标识符	1
10	=	运算符	309
10	22. 3E-1	科学计数法	4
10	-	界符	302

错误行号	Token	详细说明
44	1 qqq	无符号数不...
45	2. 2. 2	无符号数不...
46	1. . 1	无符号数不...
47	'\t	字符常量引...
48	"zhangzhilu;	字符串常量...
56	/*zhangzhilu	注释未封闭

指导教师评语：

日期：2019 年 4 月 13 日

附录

单词类型	具体类型	种别码
标识符	—	1
无符号数	整型常量	2
	浮点型常量	3
	科学计数法	4
字符常量	—	5
字符串常量	—	6
注释	—	7
关键字	auto	101
	double	102
	int	103
	struct	104
	break	105
	else	106
	long	107
	switch	108
	case	109
	enum	110
	register	111
	typedef	112
	char	113
	extern	114
	return	115
	union	116
	const	117
	float	118
	short	119
	unsigned	120
	continue	121
	for	122

	signed	123
	void	124
	default	125
	goto	126
	sizeof	127
	volatile	128
	do	129
	if	130
	while	131
	static	132
	String	133
运算符	+	201
	-	202
	*	203
	/	204
	%	205
	++	206
	--	207
	<	208
	<=	209
	>	210
	>=	211
	==	212
	!=	213
	&&	214
		215
	!	216
	~	217
	&	218
		219
	^	220
	>>	221
	<<	222
	+=	223
	-=	224
	*=	225
	/=	226
	%=	227
	&=	228
	^=	229
	=	230
	<<=	231
	>>=	232

界符	,	301
	;	302
	[303
]	304
	{	305
	}	306
	(307
)	308
	=	309