



华南师范大学

*South China Normal University*

## 编译原理课程项目三

子实验二-生成词法分析程序(JFlex)

### 实验报告

设计题目：生成词法分析程序(JFlex)

院    系：计算机学院

指导老师：王欣明

专业班级：15 级数据库 2 班

完成学生：陈天一

一、总结Oberon-0语言的词汇表

操作符	算数运算符	+   -   *   DIV   MOD
	关系运算符	>   <   #   =   >=   <=
	逻辑运算符	&   or   ~
	赋值运算符	:=
	选择运算符	.   []
	括号运算符	(   )
	类型运算符	:
	分割运算符	;   ,
注释符		"(*" {not *})*"
保留字		IF   THEN   ELSIF   WHILE   DO   BEGIN   END   OF   INTEGER   WRITE   WRITELN  READ   RECORD   ARRAY   MODULE   CONST   TYPE   VAR   PROCEDURE
标识符		letter (letter   digit) *
数值常量	INTEGER BOOLEAN	0[1-7]+[ 0-7 ]*   0 [digit]+[digit]* 0 - 10

单词分类的理由：

分类是根据各个单词的属性，例如操作符分为逻辑运算符、赋值运算符、选择运算符等等，其他的分类也同样，例如标识符与数值常量分为两类。具有相似功能的运算符分成一类。

对于关键字与保留字，关键字是保留字的子集，按照对关键字的理解，将类型、操作类型、声明语句列为关键字，保留字主要是程序关键语句。

## 二、抽取Oberon-0语言的词法规则

(1) 根据以上的单词表格，结合 Oberon-0 的语法规则，可得到以下此法规则：

```
operator -> + | - | * | DIV | MOD | := | = | # | > | <
| >= | <= | ( | ) | & | or | ~ | [ | ] | . | :
ReservedWords -> IF | THEN | ELSIF | WHILE | DO | BEGIN
| END | OF | INTEGER | WRITE | READ | WRITELN | RECORD |
ARRAY | MODULE | TYPE | CONST | TYPE | VAR | PROVEDURE
Commment -> "(*" [^*] ~"*)" | "(*" "*" + ")"
Separator -> ; | ,
TypeOperator -> :
Digit -> [1-9]+(0-9)* | 0
Octal -> 0[1-7]?(0-7)*
Letter -> [a-z A-Z]
Identifier -> Letter(Letter | Digit)*
Number -> Digit | Octal
```

## (2) 与其他高级语言的语言词法规则的比较:

- ① 在操作符方面, 与高级语言语言不相同的主要是除法DIV、求余MOD、等于=、不等于#、或OR和非~等;
- ② 在标识符中不可以出现下划线等特殊字符;
- ③ 注释也不相同, Oberon-0中为(\* \*), C++中是//或/\*\*/ ;
- ④ 数字方面, 不可以使用小数点和浮点数。