

子实验二：生成词法分析程序

2.1 总结 Oberon-0 语言的词汇表

根据 Oberon-0 语言的 BNF 定义和语言描述，抽取 Oberon-0 语言的词汇表以供词法分析程序设计与实现之用。你需要将 Oberon-0 的所有单词分类，并以表格形式列出各类词汇的预定义单词；譬如在保留字表中列出所有的保留字，在运算符表中列出所有的运算符等等。请在实验报告中说明你的单词分类的理由。

保留字（关键字）	MODULE, PROCEDURE, CONST, TYPE, VAR, RECORD, ARRAY, OF, WHILE, DO, IF, THEN, ELIF, ELSE, BEGIN, END, INTEGER, BOOLEAN
逻辑运算符	OR, &, ~
关系运算符	=, #, <, <=, >, >=
算术运算符	+, -, *, DIV, MOD
类型运算符	:
选择运算符	., []
赋值运算符	:=
分割运算符	,, ;
括号运算符	(,)
注释运算符	(*, *)
标识符	[a-zA-Z] (a-zA-Z, 0-9)*
八进制数值	0[1-7] (0-7)*
十进制数值	[1-9] (0-9)*

单词分类理由：

关键字是保留字的子集，在 Oberon-0 语言中关键字集合和保留字集合相等，所以关键字和保留字相同。

对于其他运算符的分类则是按照运算符的运算意义来划分，具有相似功能和相同优先级的运算符归为一类。还有标识符和数字符号分别归为一类。

上述的划分方法也有利于在使用 JFlex 工具设计出 Oberon-0 的词法分析程序。对于比如算术运算符的划分还可以根据运算符的优先级进一步进行划分。

2.2 抽取 Oberon-0 语言的词法规则

在 Oberon-0 语言的 BNF 定义中，既包括 Oberon-0 语言的语法定义部分，也包括 Oberon-0 语言的词法定义部分。请将词法定义从 Oberon-0 语言的 BNF

中分离出来，并写成正则定义式的形式。

Reserved_word -> "MODULE" | "PROCEDURE" | "CONST" | "TYPE" | "VAR" |
"RECORD" | "ARRAY" | "OF" | "WHILE" | "DO" | "IF" | "THEN" | "ELSIF" | "ELSE"
| "BEGIN" | "END" | "INTEGER" | "BOOLEAN"

Logic_operator -> "OR" | "&" | "~"

Relational_operator -> "=" | "#" | "<" | "<=" | ">" | ">="

Addictive_operator -> "+" | "-"

Multiplicative_operator -> "*" | "DIV" | "MOD"

Type_operator -> ":"

Selected_operator -> "." | "[" | "]"

Assign_operator -> "：="

Seperated_operator -> "," | ";"

Parentheses -> "(" | ")"

identifier -> letter(letter | digit)*

letter -> [a-zA-z]

digit -> [0-9]

number -> decimal | octal

decimal -> [1-9][0-9]*

octal -> 0[1-7]?(0-7)*

Comment -> "(" [^*] ~"*)" | "(" " " + ")"

2.3 下载词法分析程序自动生成工具 JFlex

安装后运行如下：

```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows [版本 10.0.17134.112]
(c) 2018 Microsoft Corporation。保留所有权利。

C:\Users\11788>jflex --version
This is JFlex 1.6.1

C:\Users\11788>w
```

2.4 生成 Oberon-0 语言的词法分析程序

生成的词法分析程序运行如下所示：

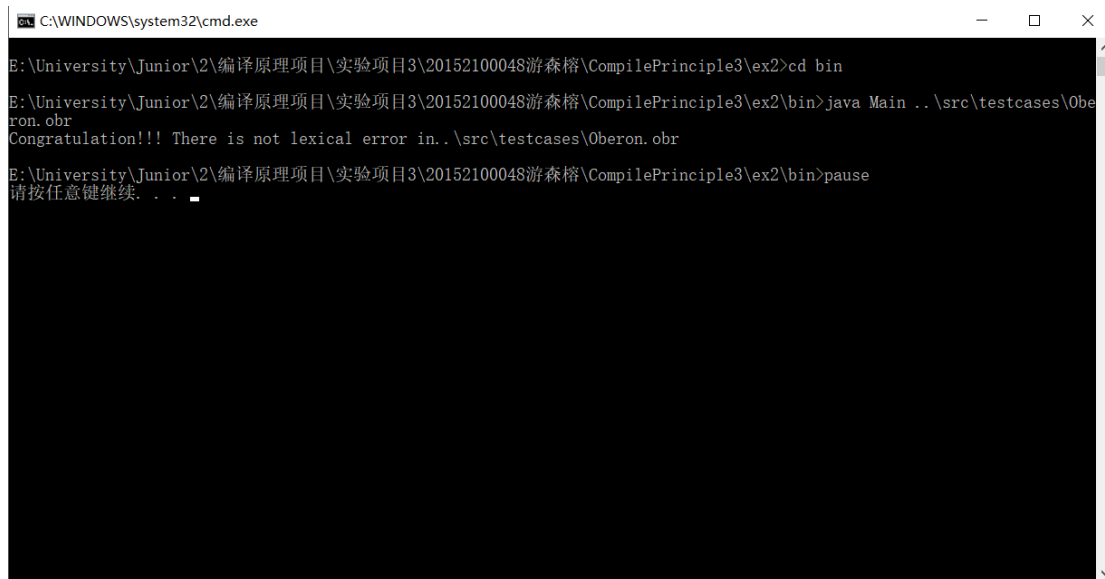
gen.bat 如下：

```
C:\WINDOWS\system32\cmd.exe
E:\University\Junior\2\编译原理项目\实验项目3\20152100048游森榕\ex2>cd src\jflex
E:\University\Junior\2\编译原理项目\实验项目3\20152100048游森榕\ex2\src\jflex>java -jar JFlex.jar ..\oberon.flex
Reading "..\oberon.flex"
Constructing NFA : 306 states in NFA
Converting NFA to DFA :
.....
122 states before minimization, 76 states in minimized DFA
Old file "..\OberonScanner.java" saved as "..\OberonScanner.java~"
Writing code to "..\OberonScanner.java"
E:\University\Junior\2\编译原理项目\实验项目3\20152100048游森榕\ex2\src\jflex>cd ..
E:\University\Junior\2\编译原理项目\实验项目3\20152100048游森榕\ex2\src>pause
请按任意键继续. . .
```

build.bat 如下：

```
C:\WINDOWS\system32\cmd.exe
E:\University\Junior\2\编译原理项目\实验项目3\20152100048游森榕\ex2>cd src
E:\University\Junior\2\编译原理项目\实验项目3\20152100048游森榕\ex2\src>javac -d bin -classpath bin exceptions*.java
E:\University\Junior\2\编译原理项目\实验项目3\20152100048游森榕\ex2\src>javac -d bin -classpath bin OberonLexical.java
E:\University\Junior\2\编译原理项目\实验项目3\20152100048游森榕\ex2\src>cd bin
E:\University\Junior\2\编译原理项目\实验项目3\20152100048游森榕\ex2\src\bin>javac -d .. \OberonScanner.java
E:\University\Junior\2\编译原理项目\实验项目3\20152100048游森榕\ex2\src\bin>javac -d .. \Main.java
E:\University\Junior\2\编译原理项目\实验项目3\20152100048游森榕\ex2\src\bin>cd ..
E:\University\Junior\2\编译原理项目\实验项目3\20152100048游森榕\ex2\src>pause
请按任意键继续. . .
```

run.bat 如下:



```
C:\WINDOWS\system32\cmd.exe
E:\University\Junior\2\编译原理项目\实验项目3\20152100048游森榕\CompilePrinciple3\ex2>cd bin
E:\University\Junior\2\编译原理项目\实验项目3\20152100048游森榕\CompilePrinciple3\ex2\bin>java Main ..\src\testcases\Oberon.obr
Congratulation!!! There is not lexical error in..\src\testcases\Oberon.obr
E:\University\Junior\2\编译原理项目\实验项目3\20152100048游森榕\CompilePrinciple3\ex2\bin>pause
请按任意键继续. . .
```