**编译原理课程设计**

**1、问题背景**

编译原理课程设计 与 实用编译器的比较

要点：实用编译器比编译原理课程设计要完成的复杂的多

**2、目的**

通过实践（设计与开发），对编译程序的设计原理及技术进一步的掌握和了解，培养学生计算思维能力及系统能力的提升。

**3、问题分析**

编译原理课程设计做什么？

设计并实现一个简单Pascal语言的编译器，主要包括几个模块：

词法分析模块

语法分析模块

语义分析模块

**要点：**

先练习处理文本文件，

然后练习处理文法结构，

最后练习根据结构完成处理变换。

遵循软件的开发过程：分析、设计、编码、集成和测试

**课程设计指导**

**Simple-PASCAL语言语法**

**1、源程序设计语言 G[<语句表>]**

参考编译原理教材179 上机实习题

**<语句表>→<语句> | <语句>;<语句表>**

**<语句>→<赋值语句>|<条件语句>|<WHILE语句>|<复合语句>**

**<赋值语句>→<变量>:=<算术表达式>**

**<条件语句>→IF<关系表达式>THEN<语句>ELSE<语句>**

**<WHILE语句>→WHILE<关系表达式>DO<语句>**

**<复合语句>→BEGIN<语句表>END**

**<算术表达式>→<项>|<算术表达式>+<项>|<算术表达式>-<项>**

**<项>→<因式>|<项>\*<因式>|<项>/<因式>**

**<因式>→<变量>|<常数>|(<算术表达式>)**

**<关系表达式>→<算术表达式><关系符><算术表达式>**

**<变量>→<标识符>**

**<标识符>→<标识符><字母>|<标识符><数字>|<字母>**

**<常数>→<整数>**

**<整数>→0|<非零数字><泛整数>**

**<泛整数>→<数字>|<数字><泛整数>|ε**

**<关系符>→<|<=|==|>|>=|<>**

**<字母>**

**→A|B|C|D|E|F|G|H|I|J|K|L|M|N|O|P|Q|R|S|T|U|V|W|X|Y|Z**

**<非零数字>→1|2|3|4|5|6|7|8|9**

**<数字>→<非零数字>|0**

**注释：**从接连出现的/\*到下一次接连出现的\*/之间的任何文字都是注释。

从某行接连出现的//到该行的结尾的任何文字都是注释。

**白空格：**两个单词之间的任何空格，制表符，回车，换行都是白空格，除了用来分隔单词以外，没有意义。

**限制条件：**标识符的长度不大于8，整数不大于65535。

**任务1：**写出符合上面文法的两个不同源程序, 要求基本覆盖所有产生式, 包含两种注释, 包含若干白空格, 满足限制条件。

**词法分析** （源程序文本文件处理）

1）什么是源程序？

需要被处理的程序叫做源程序。

转换程序的输入时源程序。

在这里，源程序就是使用语句表文法编写的程序

2）什么是文本文件？

文件每个字节或者若干个字节表示一个字符的文件。

众所周知，一般情况下，源文件编写完毕以后，保存为特定后缀名的文本文件。

本次课程设计，所有的源文件后缀名约定为：YJB

这里强调 本次课程设计要求处理保存在文本文件中的源程序。

**练习2：**将练习1写好的源程序输入计算机，用1.YJB作为文件名保存

**3、词法分析的“处理”做什么事情？**

1）处理文件

读取源文件的内容

标准套路，（参考C语言的文件操作函数，fopen函数等）

2）把源文件的内容分解为词法单元

词法单元就是单词。

单词有类别（本次课程设计，请设计好单词的分类）：

标识符为1类：如ABC，XYZ，C

整数设为2类：如123，900

IF为3类

THEN为4类

ELSE, WHILE, DO, BEGIN, END…….

运算符+， -， \*， /,>, <, =, >= , <=, <>, :=为101/102……

分隔符类： ；，（，）为201/201…….

白空格不是单词

注释以及注释标记不是单词

**练习3** 手工把练习1写好的源程序分解为单词类别和单词词文组成的二元式流

**4、怎样描述词法单元：正规文法，正规表达式**

上面给出的语句表文法实际上是该语言的BNF，其中包了单词描述和文法描述。需要从中提取出描述单词的正规文法或者正规表达式。

采用正规文法:

<保留字>→IF|THEN|……

<整数>→ 0|1<泛整数>|2<泛整数>

采用正规表达式:

(A|…|Z)(A|…|Z|0|…|9)\*|(0|…|9)|(1|…|9)(0|…|9)\*|IF|THEN|………….

**练习4** 写出描述语句表语言的词法单元的正规文法或者正规表达式

**5、怎样分解词法单元：有穷自动机，递归分析程序**

**怎样编写词法分析程序：**

1）预处理：把源文件一个字符一个字符的读入词法分析程序设置的输入字符结构体数组中（输入缓冲区），读入过程要删除注释，删除多余的白空格.

2）输入缓冲区要记录每个符号的ASCII码，以及该符号在源程序中的坐标值（行号和列号）

输入符号结构体数组：

struct inchar

{

char input；

int x，y；

}INCHAR[]；

**练习5** 编写预处理函数，从源文件得到输入符号流，要求按照上面的形式保留每个符号的坐标

3）从源程序字符数组中获得单词, 编码为二元式.

二元式采用结构体数组存储, 把单词类型和词元记录下来

例如:

struct dual

{ int dual\_type;

union {

char lexeme\_text[]；

int lexeme\_num[];

} lexeme

} DUAL[]

分解单词的方法：

利用case多路转换语句根据单词的特点直接编写。

通过描述单词的正规文法得到相应的有穷自动机，通过case多路

转换语句完成有穷自动机的处理流程。参考课本P78的例子

3）将有穷自动机转换为状态矩阵，构造状态矩阵驱动程序完成有穷自动机的流程。参考课本p55例子。

通过递归分析（正规文法是上下文无关文法）

**6、编写词法分析程序要注意的问题**：

1. 检查词法是否有错误

检查是否有非法字符：如a, aBC, @, &, !

检查是否有非法单词：如12A, BC\_1,

检查标志符和数字是否满足限制条件

检查注释符号是否配对

1. 界符分隔单词

能够区分两个单词的符号为界符

有些界符不是单词：如白空格

有些界符仅仅用来分隔：如；

有些界符本身还是源程序不可缺少的单词，如(, ), +, /, 等等

有些界符包含两个字符：如<>, >=等等

3）输出词法错误

如果没有错误，返回0

如果有错误，需要报告词法错误在源程序中的位置。并且，要能够越过错误，分解下一个单词，直到源程序结束。

1. 输出的二元式流保存在二元式结构体数组中，供语法分析使用。

为了方便接下来的语法分析能够定位语法错误在源程序中的位置，需要在二元式结构体数组中保留每个单词在源程序中的坐标（单词的首字母所在的行号列号）。

实际的二元式结构体数组应该是：

struct dual

{ int dual\_type;

union {

char lexeme\_text[]；

int lexeme\_num[];

} lexeme

int x, y;

} DUAL[]

我们还是把它叫做二元式

**练习6**  编写词法分析程序，从输入符号流得到二元式流

**词法分析程序的总结**

1）预处理函打开源文件，读取源程序内容到设置好的字符数组（输入缓冲区），这个过程要删除注释，删除多余的白空格，所有的白空格最好都转为单个空格。预处理得到了字符流。

2）预处理后关闭源文件，对输入缓冲区进行词法处理：分解输入字符流，得到的二元式存入二元式结构体。词法处理以后得到二元式流

3）错误处理要完善，特别是坐标准确

**语法分析**

1、语法分析做什么事情？

语法分析检查源程序是否符合语句表语言的文法。语法分析程序处理词法分析程序得到的二元式流，检查是否存在语法错误。

**练习7** 根据语法写源程序，并画出语法树

2、怎样完成语法分析

两种策略：自顶向下或者自底向上

自顶向下：采用递归下降分析或者预测分析

（1）递归分析简单，容易完成，参考课本p113程序。

（2）预测分析需要首先构造预测分析表，然后构造表驱动程序。

自底向上分析：

（3）SLR分析比较合适，首先根据文法构造SLR分析表，然后编写分析表驱动程序完成语法分析

语法分析设计过程需要考虑的问题：

（1）数据结构设计

（2）分析器原理的实现流程

（3）产生式及分析的存储结构设计

（4）语法树节点的表示

**练习8** 选择一种分析方法，完成语法分析程序

3、语法分析要注意的问题

检查语法错误，类型很多，不一一列举

报告语法错误，要给出语法在源程序中的位置

语法错误恢复，通过语法界符同步，语法界符就是 ；或者 END

**语义分析**

工作：通过语法制导翻译给出中间代码

**练习9** 按照课本上通用的四元式格式，设计中间语言，手工把练习1编写的源程序翻译为四元式流

**练习10**根据语法分析所用的策略编写语法制导翻译文法，

自顶向下分析，L属性文法

自底向上分析，S属性文法

**练习11** 根据语法分析采用的策略，编写相应的属性翻译文法