实验一 TINY语言的词法分析

实验目的

构造tiny语言的词法分析器（扫描器），要求利用第三方的lex工具进行构造。实验结果：构造出的扫描器，能够读入教材样例中给出的tiny语言的示例代码，分解成token输出。

Experimental Requirement

Build the lexical parser (a scanner) of TINY language, using the Lex tool of third party. The experimental result Scanner can receive the sample program of Tiny language, and output is a sequence of tokes that are defined by regular expression.

预习要求：

Preview requirements:

1.查阅各种专业文献，针对任意二种现代程序设计语言编译器的词法分析技术，找出各自的标志性的特征点，并加以比较分析和比较，指出其优缺点，并试图给出自已的结论。

Referring to professional literatures, aiming at the lexical analysis technology of any two modern programming language compilers, finds out their respective features, compares them, points out their advantages and disadvantages, and tries to give your Evaluation and conclusions.

1. ***Java编译器***

java语言是基于java虚拟机(JVM)而实现的一种高级语言，它需要通过java编译器编译成JVM识别的语言，最后由JVM实现到目标语言的转换。

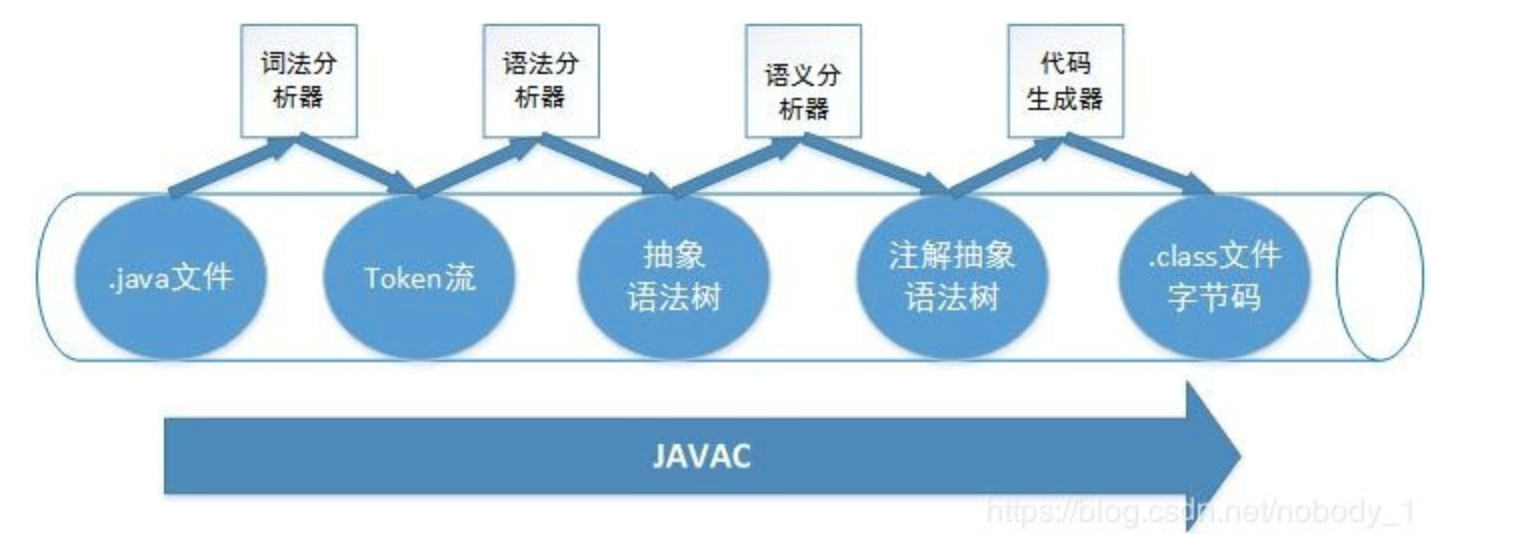


Javac的作用

javac是把java高级语言转变成JVM识别的一种二进制代码；具体体现就是.java文件到.class文件的转变；JVM识别的.class文件存储的是字节码；而转变的正确性则是由JVM的语言规范来保证，所以java编译器的作用可以理解为把java语言规范转变成JVM语言规范。

Javac的主要过程

从java语言到字节码的转变要经过四个过程：①java语言到Token流的过程，称为词法分析；②Token流到抽象语法树的过程，称为语法分析；③解析复杂的树节点，如语法糖的解析等，称为语义分析；④抽象语法树到字节码的过程，称为代码生成。



**特点**：通过java编译器编译成JVM识别的语言，最后由JVM实现到目标语言的转换。

**优点**：javac的任务就是将java源代码语言转换成jvm能够识别的语言，然后jvm将jvm语言再转化成当前机器能够识别的语言(这样使得对开发者屏蔽与机器相关的细节，并且使得语言的执行与平台无关)

JIT除了具有缓存的功能外，还会对代码做各种优化

**结论**：java编译器主要是通过javac转化成jvm能够识别的语言，而使语言可以与执行的平台无关

***（2）Python编译器***

1. Tokenizer进行词法分析，把源程序分解为Token
2. Parser根据Token创建CST
3. CST被转换为AST
4. AST被编译为字节码
5. 执行字节码

**特点**：Python的做法，只需分析CST，无需考虑词法分析的内容，当然增加的工作量是构造一个生成器从Grammar生成对应的分析代码。

**优点**：维护的代码会简单一些。

**结论：**python只需分析CST，无需考虑词法分析的内容

2.请描述自已拟定的实验计划和步骤，以及对输入和输出的设计。

Please design your experimental plan and procedures, as well as the design of input and output of the SCANNER

**计划：**

掌握词法分析器的构造和使用，学会使用 fex 工具来构造词法分析程序

**步骤：**

1. 学会使用flex工具
2. 构造tiny语言的词法分析器（扫描器），要求
3. 构造出的扫描器，能够读入教材样例中给出的tiny语言的示例代码，分解成token输出。

**输入：**

{ Sample program

in TINY language -

computes factorial

}

read x; { input an integer }

if 0 < x then { don't compute if x <= 0 }

fact := 1;

repeat

fact := fact \* x;

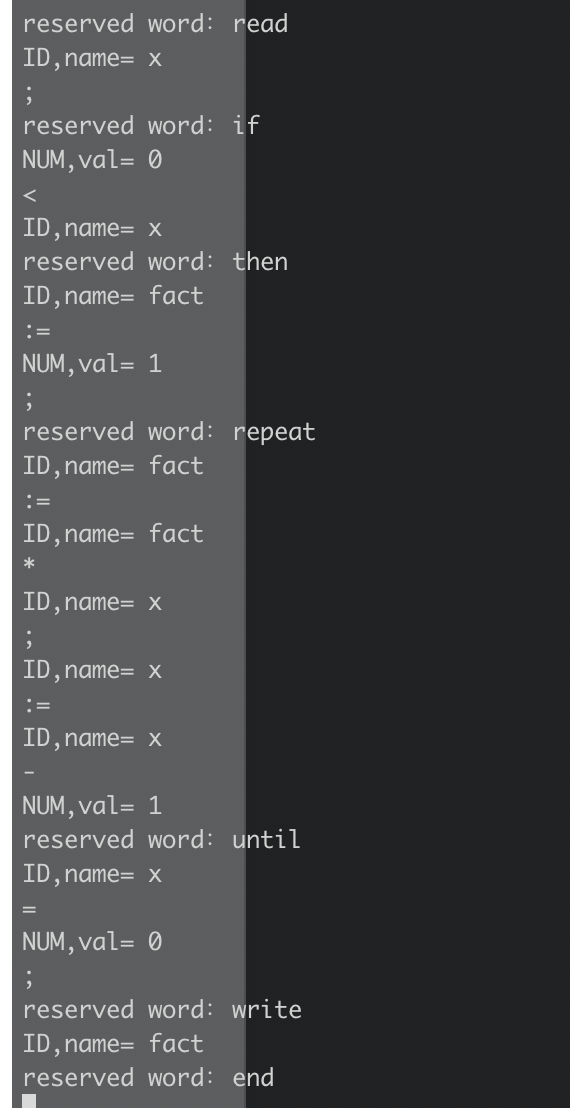
x := x - 1

until x = 0;

write fact { output factorial of x }

end

**输出：**

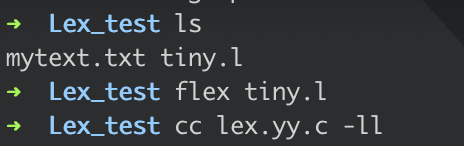
****

3.选择三方工具，阅读其使用说明，并尝试建立起工作环境。

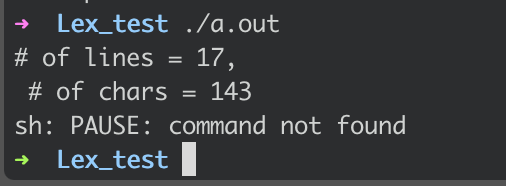
Choose the LEX tools, read their instructions, and try to establish a working environment.

我使用的是mac系统，在终端中是可以直接编译lex文件的，以下为编译实例。

（1）使用老师给的tiny.l和mytest.txt进行测试



（2）



完成工作环境搭建

资料来源：

<https://blog.csdn.net/ATField/article/details/1439068>

<https://blog.csdn.net/ATField/article/details/1475840>

<https://blog.csdn.net/ATField/article/details/1488934>

Appel A . 现代编译器的JAVA实现[M]. 电子工业出版社, 2004.