编译原理 Lab1

1. 实验目的
   编写调试一个词法分析程序，对语句进行词法分析。

2. 内容描述

3. 实验方法
   1) 针对要识别的单词符号写出正则表达式
   2) 构造出正则表达式对应的 NFA
   3) 合并所有 NFA
   4) 将 NFA 化简为 DFA
   5) 基于 DFA 编写代码
   6) 代码中具体的实现：

4. 假设
   假设输入的文件内容是正常的 java 程序，即包含合法的保留字和运算符。

5. 相关 FA 描述

   词法单元分类：

   关键字：
   static String[] *keywords*=
   {"public","protected","private","class","static","void","main","String","int","double","float","
   char","if","else","else
   if","do","while","try","catch","finally","case","switch","case","break","for"};

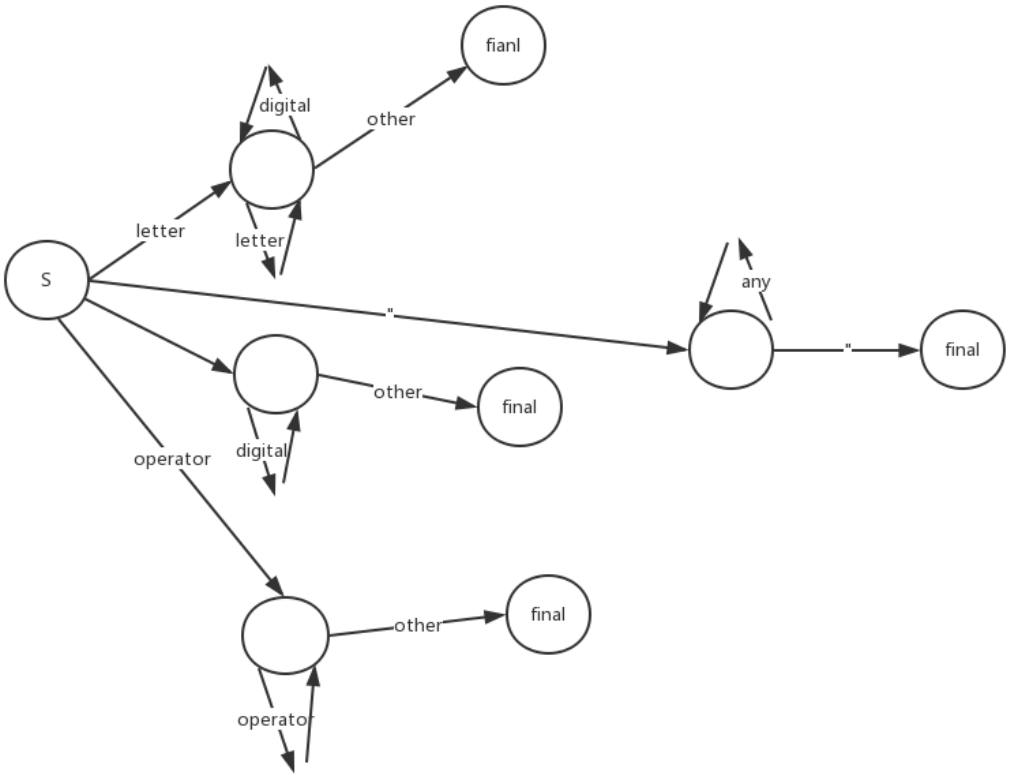
   操作符
   static String[] *operation*=
   {"[","]","(",")","++","--","!","~","*","/","%","+","-
   ","<<",">>","<>","<","=",">","==","!=","|","&","^","&&","||","+=","-=","<=",">="};

   标点符号
   static String[] *punctuation*={",","{","}",";","."};

   标识符：
   letter(letter|digit)*

   int:
   digitdigit*

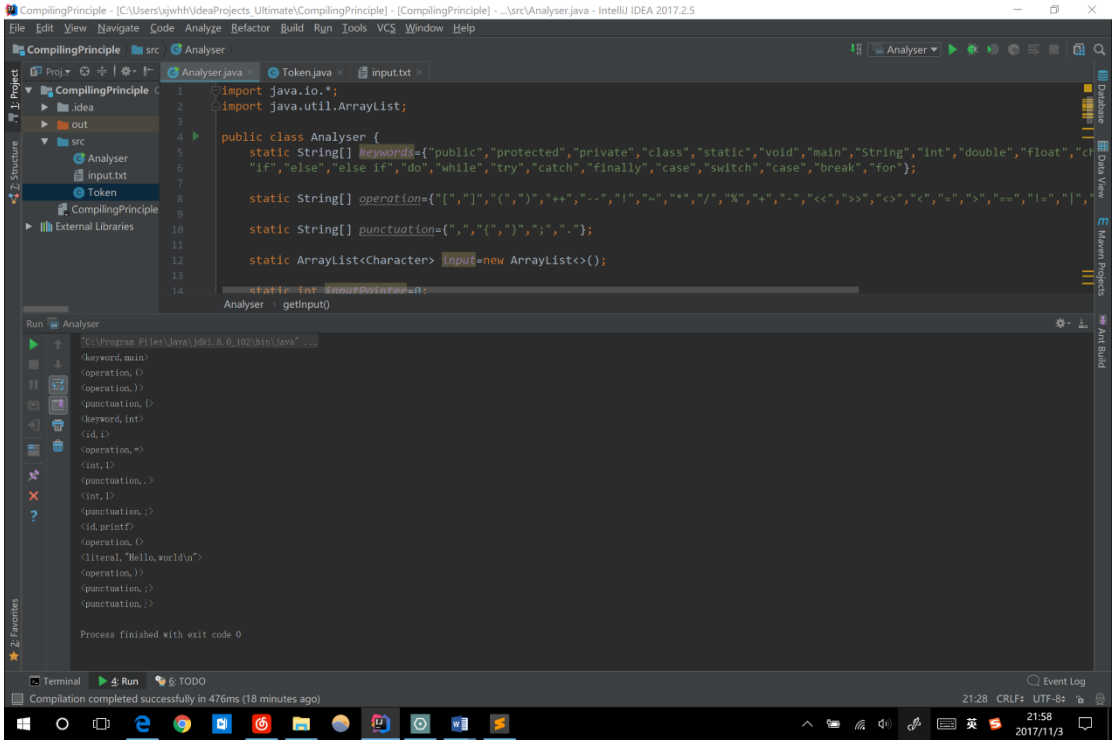
字符串：
“(letter|digit)*”



6. 重要数据结构描述
   input:ArrayList<Character>，用来存储一个字符一个字符读取的输入文件内容
   word: :ArrayList<Character>,用来存储每次 scanner 过程中需要暂存的子字符串
   inputPointer：全局变量，input 指针
   Token:token 类，用来存储将要输出的 token，重载了 toString()方法，便于输出

7. 代码算法描述
   getInput()：获取文件输入
   scanner()：
   一个字符一个字符的扫描输入
   英文字母开头，可能是保留字或者标识符，每读一位，都判断是否属于保留字，若是就直接输出，否则一直读到不是英文字符为止，指针回退一位
   数字开头，就是数字类型，读到不是数字为止，指针回退一位
   “开头，就是字面值字符串，读到不是”为止
   其余的就是标点符号或者操作符，需要一直读到能够确定是哪种为止，要注意“-”之后可能接数字表示负数

8. 示例运行



9. 产生的问题和解决方法
   1) int 类型可能会溢出，要进行判断
   2) -后面可能接数字表示负数
   3) 对于保留字，操作符，标点符号的兼容难以完全包含

10. 感受
    对于词法分析的状态和变化有了更好的理解