实验编号：1**四川师大 编译原理 实验报告 2020**年**4**月**12**日

**计算机科学学院** 2017级 4 班 实验名称：词法分析程序

姓名：\_韩勇\_\_ 学号：\_2017110409\_\_ 指导老师：\_\_吴贞东\_\_ 实验成绩:\_\_\_\_\_

**实验 一 词法分析程序\_**

1. 实验目的及要求

通过本实验的编程实践，使学生了解词法分析的任务，掌握词法分析程序设 计的原理和构造方法，使学生对编译的基本概念、原理和方法有完整的和清楚的理 解，并能正确地、熟练地运用。输入的常量说明串，要求最后以分号作结束标志；

1、编程时注意编程风格：空行的使用、注释的使用、缩进的使用等。

2、将标识符填写的相应符号表须提供给编译程序的以后各阶段使用。

3、根据测试数据进行测试。测试实例应包括以下三个部分: 全部合法的输入。 各种组合的非法输入。 由记号组成的句子。

4、词法分析程序设计要求输出形式:

1. 实验内容  
   （1）从源程序文件中读入字符。

（2）统计行数和列数用于错误单词的定位。

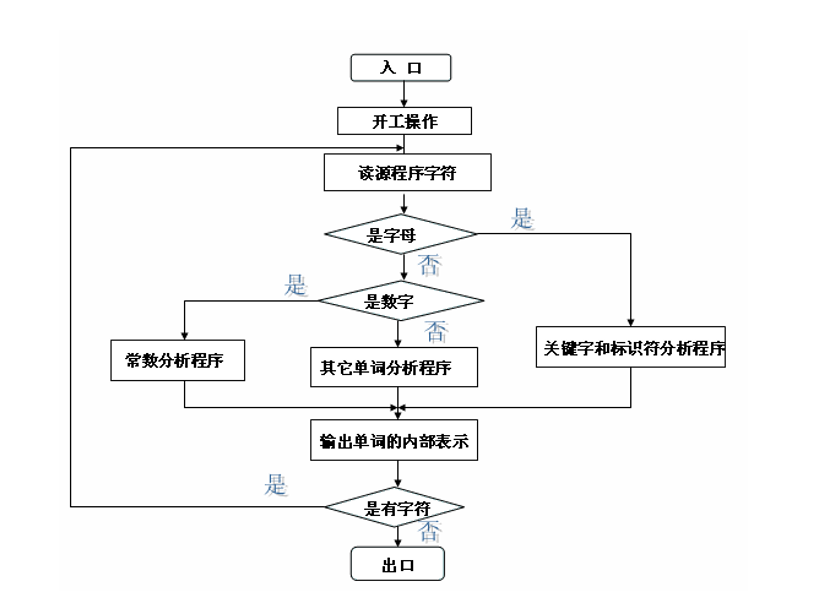
（3）删除空格类字符，包括回车、制表符空格。

（4）按拼写单词，并用（内码，属性）二元式表示。(属性值——token 的机内 表示)

（5）如果发现错误则报告出错 7

（6）根据需要是否填写标识符表供以后各阶段使用实验主要流程、基本操作或核心代码、算法片段（该部分如不够填写，请另加附页）

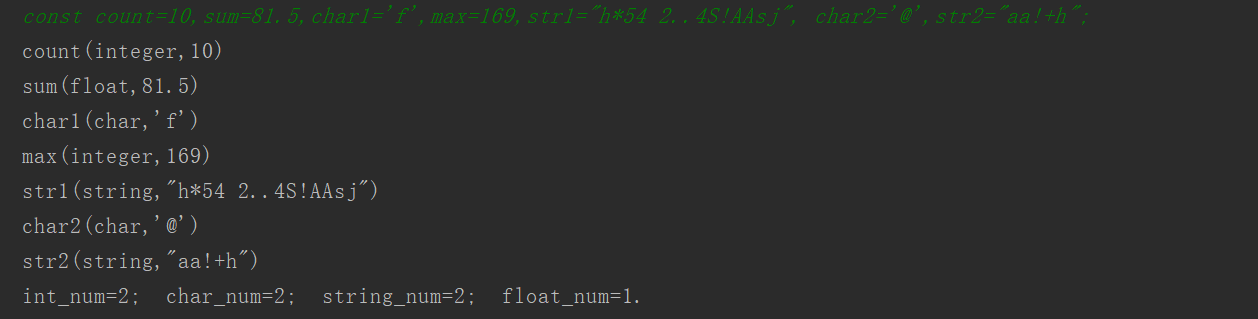
流程图：



代码：

package compiler\_test;  
import java.util.Scanner;  
  
public class Test\_01 {  
 static String *name*; // 存放常量名  
 static String *value*;// 存放常量值  
 static String *type*; // 存放常量值类型  
 static String *errorInfo*;// 存放错误信息  
 static int *correctName*;// 0表示常量名错误，1表示常量名正确  
 static int *correctValue*;// 0表示常量值错误，1表示常量值正确  
 static int *int\_num* = 0;  
 static int *string\_num* = 0;  
 static int *float\_num* = 0;  
 static int *char\_num* = 0;// 用于统计各种类型的常量数量  
  
 public static void main(String[] args) {  
 Scanner in = new Scanner(System.*in*);  
 String s = in.nextLine();  
 s = s.trim();// 去除首尾空格  
 boolean result = s.startsWith("const") && s.endsWith(";");  
 while (!result) {  
 // 如果输入字符串不是以“const"开头并且以”；”结尾，则输出错误信息，并且要求重新输入  
 System.*out*.println("It is not a constant declaration statement! ");  
 System.*out*.println("Please input a string again!");  
 s = in.nextLine();  
 s = s.trim();  
 result = s.startsWith("const") && s.endsWith(";");  
 }  
 *Output*(s);  
 in.close();  
 }  
  
 // 判断常量名是否合法  
 public static int checkName(char[] a, int i) {  
 *name* = "";  
 while (a[i] != '=') {  
 *name* += a[i];  
 i++;  
 }  
 *name* = *name*.trim();  
 String regex = "[a-zA-Z\_][a-zA-Z0-9\_]\*";  
 boolean result = *name*.matches(regex);  
 if (result) {  
 *correctName* = 1;  
 } else {  
 *correctName* = 0;  
 }  
 return i;  
 }  
  
 // 判断常量值的合法性与常量类型  
 public static int checkType(char a[], int i) {  
 *value* = "";  
 *errorInfo* = "";  
 while (a[i] != ',' && a[i] != ';') {  
 *value* += a[i];  
 i++;  
 }  
 *value* = *value*.trim();  
 if (*correctName* == 1) {  
 // 判断该数是否为整数  
 if (*value*.matches("[+|-]?[0-9]\*")) {  
 String s1 = *value*;  
 // 判断符号  
 if (*value*.charAt(0) == '+' || *value*.charAt(0) == '-') {  
 s1 = *value*.substring(1);  
 }  
 if (s1.equals("0") || s1.matches("[1-9][0-9]\*")) {  
 *correctValue* = 1;  
 *type* = "integer";  
 *int\_num*++;  
 } else {  
 *errorInfo* = "Wrong! The integer can’t be started with ‘0’.";  
 *correctValue* = 0;  
 }  
 }  
 // 判断该数是否为浮点数  
 else if (*value*.matches("[+|-]?[0-9]\*[.][0-9]+")) {  
 *correctValue* = 1;  
 *type* = "float";  
 *float\_num*++;  
 }  
  
 // 判断常量值是char型  
 else if (*value*.startsWith("'") && *value*.endsWith("'")) {  
 if (*value*.length() == 3) {  
 *correctValue* = 1;  
 *type* = "char";  
 *char\_num*++;  
 } else {  
 *errorInfo* += "Wrong! There are more than one char in ‘’.";  
 *correctValue* = 0;  
 }  
 }  
  
 // 判断常量名是String型  
 else if (*value*.startsWith("\"") && *value*.endsWith("\"")) {  
 *correctValue* = 1;  
 *type* = "string";  
 *string\_num*++;  
 }  
  
 // 其他错误情况  
 else {  
 *correctValue* = 0;  
 *errorInfo* += "Wrong! The format of the value string is not correct !";  
 }  
 }  
 return i;  
 }  
  
 static void Output(String s) {  
 char[] str = s.toCharArray();  
 int i = 5;  
 while (i < str.length - 1) {  
 i = *checkName*(str, i);  
 i = *checkType*(str, i + 1) + 1;  
 // 常量名定义正确，继续判断常量值  
 if (*correctName* == 1) {  
 // 常量值正确，输出结果，包含常量名，常量类型以及常量值  
 if (*correctValue* == 1) {  
 System.*out*.println(*name* + "(" + *type* + "," + *value* + ")");  
 }  
 // 常量值错误，给出错误类型  
 else {  
 System.*out*.println(*name* + "(" + *errorInfo* + ")");  
 }  
 }  
 // 常量名定义错误  
 else {  
 System.*out*.println(*name* + "("  
 + "Wrong! It is not a identifier!" + ")");  
 }  
 }  
 System.*out*.println("int\_num=" + *int\_num* + "; char\_num=" + *char\_num* + "; string\_num=" + *string\_num* + "; float\_num=" + *float\_num* + ".");  
 }  
}

运行结果：



1. 实验结果的分析与评价（该部分如不够填写，请另加附页）

通过实验更好的了解了语言运行的根本，有利于我们以后更好的进行相关方面的学习。

注：实验成绩等级分为（90－100分）优，（80－89分）良，(70-79分)中，（60－69分）及格，（59分）不及格。