**编译原理课程实验报告**

**实验1：词法分析**

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 姓名 | 王晨懿 | | 院系 | 计算机科学与技术学院 | | | 学号 | | | 1162100102 | | |
| 任课教师 | |  | | | 指导教师 |  | | | | | | |
| 实验地点 | | 格物208 | | | 实验时间 | 2019.4.14 | | | | | | |
| 实验课表现 | | 出勤、表现得分 | |  | 实验报告  得分 |  | | 实验总分 | | |  | |
| 操作结果得分 | |  |
| 1 需求分析 | | | | | | | | | 得分 | | |  |
| 要求：阐述词法分析系统所要完成的功能  **识别以下几类单词：**   1. 标识符（由大小写字母、数字以及下划线组成，但必须以字母或者下划线开头） 2. 关键字（①类型关键字：整型、浮点型、布尔型、记录型；②分支结构中的if和else；③循环结构中的do和while； 3. 运算符（①算术运算符；②关系运算符**；③**逻辑运算） 4. 界符（①用于赋值语句的界符，如“=”；②用于句子结尾的界符，如“；”； 5. 常数（无符号整数和浮点数等） 6. 注释（/\*……\*/形式）   **用户界面包括如下输出：**   1. token序列   打印token序列，每个token打印信息包括：行号、token、种别码、属性值。   1. 符号表   打印符号表，符号表包括信息：入口地址、符号。   1. 错误信息   打印错误信息，错误说明会指出错误行数，以及错误原因。  **用户使用方法：**  用户首先需要在文本编辑区输入待分析的文本；或者可以点击“打开文件”按钮，从文件导入文本；也可以点击“测试用例”按钮，导入程序默认的测试用例。  在输入好文本后，用户需要点击“词法分析”按钮，程序将进行词法分析，分析结果将在token序列表、符号表、错误信息表中展示出来。 | | | | | | | | | | | | |
| 2 文法设计 | | | | | | | | | 得分 | | |  |
| 要求：对如下内容展开描述   1. 给出各类单词的词法规则描述（正则文法或正则表达式） 2. 各类单词的转换图  2.1各类单词的词法规则描述 **关键字**  keyWords -> int | float | double | if | else | switch | do | while | for | void | return  **标识符**  digit -> 0 | 1 | 2 | … | 9  letter\_ -> A | B | … | Z | a | b | … | z | \_  id -> letter\_(letter\_ | digit)\*  **整型常数**  int -> (+|-)?digit+  **运算符**  operator -> + | - | \* | / | & | | | ! | && | || | == | != | < | > | <= | >=  **分界符**  assign -> =  separator -> ; | ( | ) | [ | ] | { | }  **注释**  comment -> /\\*.\*\\*/  （’\’ 为转义字符） 2.2各类单词的转换图 **关键字**    **标识符**    **整型常数**    **运算符**    **分界符**    **注释** | | | | | | | | | | | | |
| 3 系统设计 | | | | | | | | | 得分 | | |  |
| 要求：分为系统概要设计和系统详细设计。  （1）系统概要设计：给出必要的系统宏观层面设计图，如系统框架图、数据流图、功能模块图等以及相应的文字说明。  （2）系统详细设计：对如下工作进行展开描述   * 核心数据结构的设计 * 主要功能函数说明 * 程序核心部分的程序流程图  3.1系统概要设计3.1.1 系统框架图  3.1.2 数据流图  3.1.3 功能模块图  3.2 系统详细设计3.2.1 核心数据结构设计 **输入缓冲区**  用于词法分析时的数据结构。构造函数需要输入文本，即为待词法分析的文本，然后在其最后加上字符’\0’，便于判断文本是否读到末尾。  其主要功能为缓存token以及模拟读入字符指针的移动。  **public** **class** MyBuffer  //组成token的StringBuffer  **private** StringBuffer tokenBuffer = **new** StringBuffer();  //输入文本  **private** String text = "";  //当前读入字符在文本中的行数  **private** **int** rowNum = 1;  //当前读入的字符索引  **private** **int** idx = 0;  **public** MyBuffer(String text) {  **this**.text = text + '\0'; //在文本最后添加’\0’，判断输入是否结束  }  //将字符ch存入tokenBuffer  **public** **void** push(**char** ch)  //清空tokenBuffer  **public** **void** clear()  //获取tokenBuffer存入的字符串  **public** String getToken()  //读入换行符’\n’时调用，行数++  **public** **void** nextLine()  //获取当前行数  **public** **int** getRowNum()  //读入下一个字符  **public** **char** getChar()  //回退一个字符，即idx--  **public** **void** retract() 3.2.2主要功能函数说明3.2.2.1 词法分析程序private boolean scanToken() 主要功能：分析出输入的下一个token。  返回值：如果读入文本末尾手动添加的’\0’，返回false；否则返回true。  程序中用到的全局变量和方法如下：   1. **buffer**   MyBuffer类（见3.1.2节），控制输入以及存储token。会用到MyBuffer的一系列方法来控制输入缓冲区。   1. **private** **void** acceptToken(**int** type, String token, **int** row)   接受token，具体功能见3.2.2.2节   1. **private** **void** outputError(String description, **int** row)   输出错误信息  程序会从输入缓冲区中依次读入字符。  如果满足自动机状态转换条件则一直读入知道到达终态；  一旦状态转移失败则输入缓冲区回退 tokenBffer.retract() ；  当到达终态时，调用accrptToken() 方法接受token；  如果出现错误，则调用outputError()方法，输出错误信息。 3.2.2.2 接受token程序private void acceptToken(int type, String token, int row) 程序根据输入的参数type 和token本身为其分配种别码和属性值，并调用outputToken()方法和outputSymbol()方法，输出token序列和符号表。 3.2.3程序核心部分的程序流程图 该流程图展示了词法分析核心部分的流程。 | | | | | | | | | | | | |
| **4 系统实现及结果分析** | | | | | | | | | 得分 | | |  |
| 要求：对如下内容展开描述。   1. 系统实现过程中遇到的问题； 2. 针对某测试程序输出其词法分析结果； 3. 输出针对此测试程序对应的词法错误报告； 4. 对实验结果进行分析。   注：其中的测试样例自行产生。 4.1 系统实现过程中遇到的问题  1. 实现dfa时候没有依靠状态转移矩阵，而是单纯地使用if-else和switch语句进行判别，导致逻辑很容易出现错误。 2. 对于gui的构建有些生疏 3. 一些细节的处理不知道应该如何实现才更接近一个高级语言编译器的正确做法。比如对于一个未封闭的注释符号“/\*”，应该判定其一直到文本末尾皆为注释，还是应该报错（本次实验选择报错）。  4.2 针对某测试程序输出其词法分析结果 测试用例：  分析结果： 4.3 输出针对此测试程序对应的词法错误报告4.4 对实验结果进行分析 | | | | | | | | | | | | |
| 指导教师评语：  日期： | | | | | | | | | | | | |