 

**实 验 报 告**

|  |  |
| --- | --- |
| **课程名称：** | **编译技术** |
| **学生姓名：** | **何博韬** |
| **学生学号：** | **201630676706** |
| **学生专业：** | **软件工程** |
| **开课学期：** | **2017-2018第一学期** |

**软件学院**

**2017年11月**

# XXXX2

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **地 点：** | | | B7 楼 | | 233 房； | | **实验台号：** | | 29 | |
| **实验日期与时间：** | | | 2017年11月1日星期三 | | | | **评 分：** | |  | |
| **预习检查纪录：** | | |  | | | | **实验教师：** | | 徐扬 | |
| **电子文档存放位置：** | | |  | | | | | | | |
| **电子文档文件名：** | | |  | | | | | | | |
| **批改意见：** | | |  | | | | | | | |
| 出勤情况 | 课堂态度 | | 实验报告质量 | | | | | |
| 任务完成情况 | | 源代码注释 | | 报告内容 | |
| □ 出勤  □ 缺勤  □ 早退  □ 请假 | □ 认真  □ 不认真 | | □ 完成  □ 未完成 | | □ 优秀  □ 良好  □ 通过  □ 未通过 | | □ 优秀  □ 良好  □ 通过  □ 未通过 | |

# 报告内容

1、实验目标

实现decaf 语言的词法分析程序

2、实验环境

Window10 JVM Eclipse IDE

3、关键数据结构和核心算法

3.1 数据结构

哈希表

3. 2 核心算法

核心算法是Lexer.java文件中的public Token scan()函数。scan()函数用来扫描输入的文本，并输出Token序列。

scan()函数的伪代码如下

for( ；；readch) 循环读取输入序列

｛

if(是空格或\t)

continue;

else if(是\n)

整形变量line加一；

else

break;

｝

switch(peek)

｛

case 此语言的关键字或者操作符：

return 关键字或操作符对应的Word对象；

｝

if(peek是数字)

｛

do{将连续的多个数字转换为一个十进制数}

while（peek不是数字）;

if（不出现小数点）

return 一个Num类的对象；

else

for（一个无条件的for循环）

｛if（下一个字符不是数字） break;

else 将连续的多个数字转换为十进制小数｝

｝

if（peek是字母）

｛

new一个缓冲区来储存输入的串

do｛将输入的连续的字符储存到缓冲区｝

while（peek是字母或数字）

将缓冲区中的内容赋值给一个String 对象，再将String对象映射的哈希表中的值取出并强制转换为Word 对象。

if（此Word对象值不为null）

return 此Word对象；

else

返回一个tag 为ID的Word对象

｝

3. 3 错误处理

1. **default**:
2. System.out.println("(SYM , "+token.toString()+")");

3.4 运行结果



图 1

4、总结

通过对代码的标注，我对词法分析的理解又加深了。