

电子科技大学 信息与软件工程 学院

标准实验报告

(实验) 课程名称编译技术

电子科技大学教务处制表

电 子 科 技 大 学

实 验 报 告

学生姓名： 邓萌达 学 号： 2018091620008

指导教师： 周尔强

实验时间： 2019.10.06 实验地点： 信软 303/304

一、实验室名称：信软 303/304

二、实验项目名称：词法分析器

三、实验学时：4 学时

四、实验内容及步骤：

1. 运用 c++ & c 编写词法分析程序

（1）运用 advance 函数一个个的读取文本中的字符

（2） analysis 函数将所得到的字符按照各个词法的特性拼接后检查属于哪一类词法。

（3） output 函数将其打印出来。

（4）同时，在 analysis 函数中，分析了几种常见的词法错误，例如小数出现了两点，例如数字开头的标识符等。

2. 运用 flex 编写词法分析程序

运用正则表达式匹配各个词法，更加简单的编写，分析，输出，纠错的一个程序的

词法结构

五、实验运行结果及测试：

1.1 运用 c++编写词法分析程序

```
→ handin make main
→ handin c++ -std=c++11 main.cpp -o main
→ handin ./main test.java > ans1.out
→ handin cat ans1.out
<keyword, package, (1, 1)>
<identifier, com, (1, 9)>
<delimiter, ., (1, 12)>
<identifier, gitblit, (1, 13)>
<delimiter, ., (1, 20)>
<identifier, client, (1, 21)>
<delimiter, ;, (1, 27)>
<keyword, import, (3, 1)>
<identifier, java, (3, 8)>
<delimiter, ., (3, 12)>
<identifier, awt, (3, 13)>
<delimiter, ., (3, 16)>
<identifier, Color, (3, 17)>
<delimiter, ;, (3, 22)>
<keyword, import, (4, 1)>
<identifier, java, (4, 8)>
<delimiter, ., (4, 12)>
<identifier, awt, (4, 13)>
<delimiter, ., (4, 16)>
<identifier, Component, (4, 17)>
<delimiter, ;, (4, 26)>
<keyword, import, (5, 1)>
<identifier, java, (5, 8)>
<delimiter, ., (5, 12)>
<identifier, text, (5, 13)>
<delimiter, ., (5, 17)>
<identifier, SimpleDateFormat, (5, 18)>
<delimiter, ;, (5, 34)>
```

1.2 运用 c 编写词法分析器

```
→ compilers cc main_lex.c -o main
→ compilers ./main test.java > ans1.out
→ compilers cat ans1.out
<package, package, (1, 1)>
<identifier, com, (1, 9)>
<delimiter, ., (1, 12)>
<identifier, gitblit, (1, 13)>
<delimiter, ., (1, 20)>
<identifier, client, (1, 21)>
<delimiter, ;, (1, 27)>
<import, import, (3, 1)>
<identifier, java, (3, 8)>
<delimiter, ., (3, 12)>
<identifier, awt, (3, 13)>
<delimiter, ., (3, 16)>
<identifier, Color, (3, 17)>
<delimiter, ;, (3, 22)>
<import, import, (4, 1)>
<identifier, java, (4, 8)>
<delimiter, ., (4, 12)>
<identifier, awt, (4, 13)>
<delimiter, ., (4, 16)>
<identifier, Component, (4, 17)>
```

2.运用 flex 编写词法分析器。

```
→ handin flex main.l
→ handin gcc-8 lex.yy.c -o main
→ handin ./main < test.java > ans2.out
→ handin cat ans2.out
```

```
TOKEN-TYPE          TOKEN-VALUE
```

```
-----
<keyword, package, lines:1 >
<identifier, com, lines:1 >
<delimiter, ., lines:1 >
<identifier, gitblit, lines:1 >
<delimiter, ., lines:1 >
<identifier, client, lines:1 >
<delimiter, ;, lines:1 >
<keyword, import, lines:3 >
<identifier, java, lines:3 >
<delimiter, ., lines:3 >
<identifier, awt, lines:3 >
<delimiter, ., lines:3 >
<identifier, Color, lines:3 >
<delimiter, ;, lines:3 >
<keyword, import, lines:4 >
<identifier, java, lines:4 >
<delimiter, ., lines:4 >
<identifier, awt, lines:4 >
<delimiter, ., lines:4 >
<identifier, Component, lines:4 >
<delimiter, ;, lines:4 >
<keyword, import, lines:5 >
<identifier, java, lines:5 >
```

六、实验结论与总结：

在这个实验中，两种方法编写的词法分析器均能按照要求完成对 test.java 程序的分析，也符合了词法分析器的基本的要求。但是更加隐晦的词法错误仍然不能指出，flex 所编写的词法分析器只能显示出每一个词的行数不能显示出列数，这是不足的地方。

这次词法分析器的实验，让我对编译器中的词法分析过程有了更加深入的了解，同时也让我更加深入的学习了 c++ 以及 stdc++11 的各种特性。更加让我感到满足的是，以这次实验为契机，我学习了强大的 flex 以及正则表达式的运用。

由于这次词法分析的任务要求是 c 语言，我最后又用 c 语言写了一遍词法分析器，巩固了我 c 的基础。

最后，感谢周尔强老师的授课以及细心指导。

另外，我已经将这次实验的代码传至
github: <https://github.com/kelolemon/lexical-analysis>

报告评分：

指导教师签字：