

# 编译原理PJ报告

14307130176 钱文渊

## 0、工具与配置

本次试验我采用了工具ANTLR来进行语法分析。ANTLR的安装也非常简单，下载安装antlr4后，更改环境变量，即可编译一个语法文件\*.g4。

## 1、绘制语法树：

运行流程：

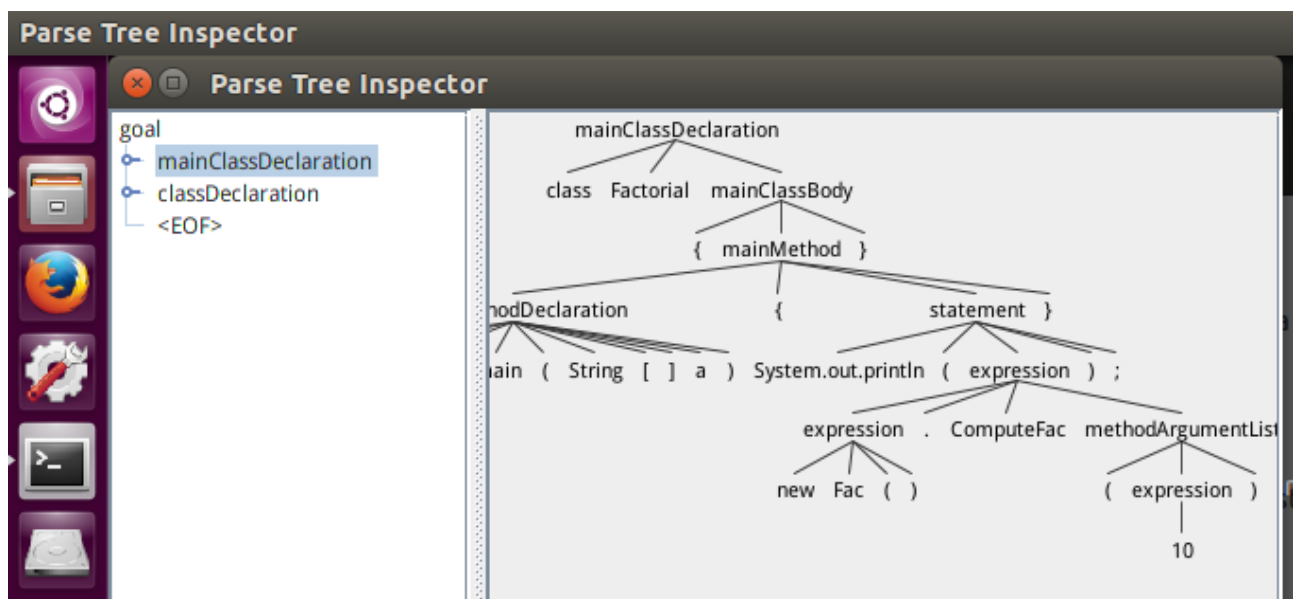
- (1) 编写文法文件MiniJava.g4；
- (2) 编译Minijava.g4；
- (3) 由编译产生的MiniJavaLexer、MiniJavaParser等文件，对待测试文件factorial.java进行语法分析，并绘制出语法树。

(1) 首先根据cambridge.org官网上，MiniJava Project中给出的BNF编写相应的适用于ANTLR的语法文件MiniJava.g4。

(2) 编写编译、运行MiniJava.g4的脚本文件drawtree.sh如下：

```
qwy@qwy: ~/qwy/Tree
#!/usr/bin/env bash
export CLASSPATH=".;./antlr-4.5-complete.jar:$CLASSPATH"
java -jar ./antlr-4.5-complete.jar MiniJava.g4
javac MiniJava*.java
java org.antlr.v4.runtime.misc.TestRig MiniJava goal factorial.java -gui
```

(3) 运行drawtree.sh脚本，即可得到对应测试MiniJava文件factorial.java的语法树如图：



(4) 此外，ANTLR自带丰富功能，运行时参数-tree可以将语法树输出至文件（字符串形式）

## 2、错误处理：

利用ANTLR编译MiniJava产生的MiniJavaLexer.java、MiniJavaParser.java等文件，调用部分参数，可以遍历语法树上的每一个节点，并且检查其是否有词法、语法错误。具体代码见项目Tree.class。

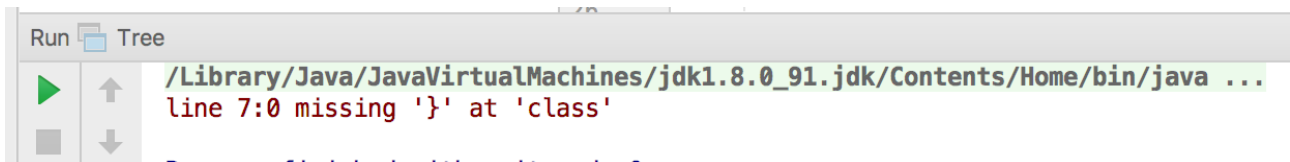
### (1) 词法错误：

对factorial.java文件第10行变量类型int前加上“+”号，显然没有这样的类型“+int”，报出一个词法错误：



### (2) 语法错误：

将factorial.java文件第5行的括号'}'去掉，显然函数后括号丢失，形成一个语法错误：



## 3、部分说明：

### (1) 为什么选ANTLR：

你TA推荐了啊。。。而且用了之后确实相当方便，自带了-gui参数，可以绘制语法树，使得我的工作集中在了编写文法文件MiniJava.g4上，对于可视化、排版的工作就省略了。

### (2) 代码结构、核心工作原理：

代码核心其实是文法文件。有了文法文件，就能用ANTLR编译产生Lexer和Parser，就能去遍历语法树，做很多的事情。项目最大困难其实是配文件吧。。。

### (3) 错误处理：

前面提到了报错的功能。因为菜，所以没修复。。。