PA 1-A 实验报告

计62 徐晟 2016011253

1. 实验要求

在本实验PA1-A中，实验要求编码实现Decaf语言编辑器的词法分析和语法分析部分，同时生成抽象语法树。

1. 实验完成

分为以下几个部分：

1. 支持scopy：

在lexer.l中加入scopy关键字

在parse.y中添加相应token

在tree.java中添加对应节点

1. 添加sealed关键字

在lexer.l中加入sealed关键字

在parse.y中添加相应token

在tree.java中添加对应节点

1. 支持串行条件卫士语句：

在lexer.l中加入关键字if，|||

在parser.y中添加IfStmt, GuardedStmt, IfSubStmtWrapper, IfSubStmtList

在tree.java中添加对应节点

1. 支持var自动类型推导

在lexer.l中添加关键字var

在parser.y中添加VarBind

在tree.java中添加对应节点

1. 支持一维数组操作

在parser.y中添加ArrayConstant, ArrayList, ConstantList

在tree.java中添加对应节点

1. 数据设计

我个人感觉给定的testcase覆盖了几乎所有可能，一开始我的程序在array-1, array-2, array-access这三个点和guarded-1, guarded-2这两个点是错误的，在修改了parser的字串顺序的逻辑之后就正确了.

1. 总结

这次试验虽然是最开始的一次，但是由于实验的体量复杂性，我还是花了大量的时间配置环境以及调试，一下是我对环境配置和调试的总结：

1. 不能使用最新版的eclipse photon，因为它能够运行的最低版本jdk是1.8，我修改为了eclipse 4.2 juno +jdk 1.7
2. 在使用ant调用byacc的时候有时候不能调用(amd64版本)，我把相对路径直接改成了绝对路径
3. 因为第一次作业的规模还是相对比较大的，功能要一项一项加方便检查错误
4. 当符号多次被使用的时候（例如if条件卫士语句中的”:”），很容易出现unknown identifier的情况，这时候可以把冒号换成其他特殊符号以验证是哪里写错了
5. 还没有找到一个比较好的检查和调试的方案…
6. 参考：https://github.com/acst1223/decaf\_PA1（部分参考tree.java的结构）