**实 验 报 告**

|  |  |
| --- | --- |
| **课程名称：** | **编译技术** |
| **学生姓名：** | **陈卓文** |
| **学生学号：** | **201936380215** |
| **学生专业：** | **软件工程** |
| **开课学期：** | **2021-2022第一学期** |

**软件学院**

**2021年11月**

# 语义分析

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **地 点：** | | | B7 楼 | | 233 房； | | **实验台号：** | | 无 | |
| **实验日期与时间：** | | | 2021/12/17 8:50 ~ 12:15 | | | | **评 分：** | |  | |
| **预习检查纪录：** | | |  | | | | **实验教师：** | | 应伟勤 | |
| **电子文档存放位置：** | | |  | | | | | | | |
| **电子文档文件名：** | | |  | | | | | | | |
| **批改意见：** | | |  | | | | | | | |
| 出勤情况（20%） | 课堂态度（20%） | | 实验报告质量（60%） | | | | | |
| 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

1. 参考第6讲中Semantic Analysis of a Program知识理解读懂提供的Decaf语言语义分析源码，重点理解对变量声明与使用、表达式和语句进行语义分析的代码，在其基础上添加相关的注释。

2. 提供的源码中没有对for语句的条件表达式进行类型检查，参考源码中inter包中的while语句的类型检查代码，实现对for语句的类型检查，并能提示类型检查的相关语义错误。

2、实验环境

Ubuntu 20.04；openjdk 11.0.8；

3、关键数据结构和核心算法

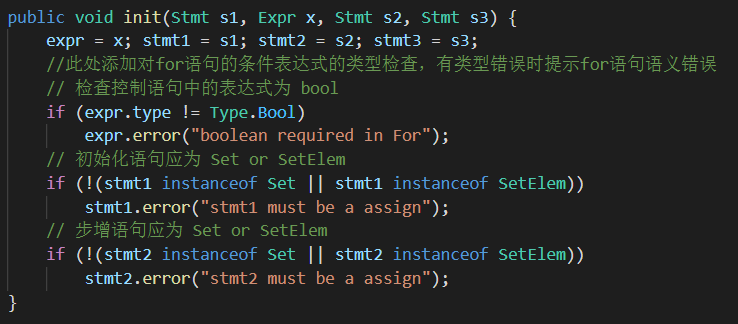
本次实验代码基于第二次实验，变动较少，语义检查方面仅仅在inter中的类添加了若干个check函数及相关逻辑，以供代码检查。

* 1. 数据结构

Inter中储存了ast中的节点定义，parser为语法分析器，负责构建ast。

3. 2 核心算法

Ast中的节点会储存自己的类型，父节点会对自己的子节点进行类型检查。

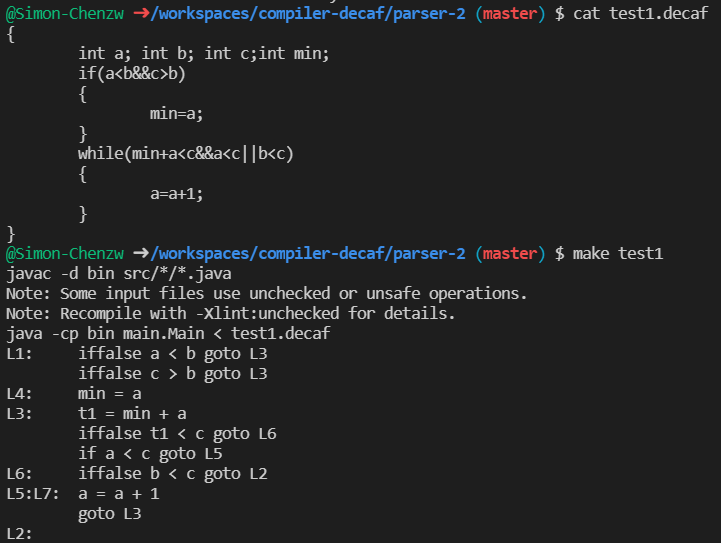


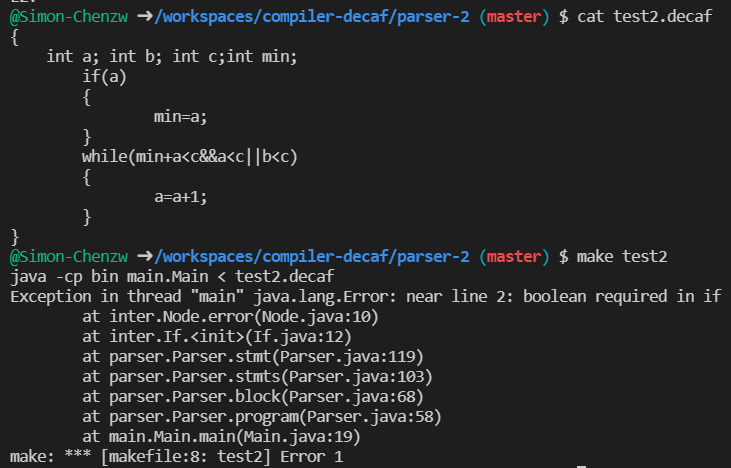
增加代码的部分如上，易见，代码对各个表达式的类型进行了限制。

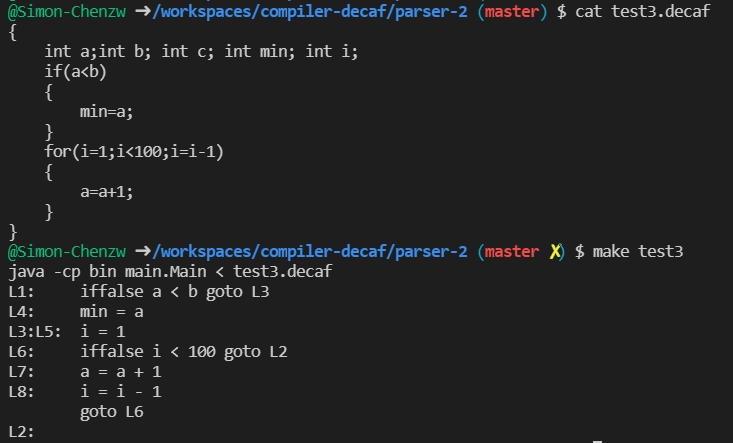
3. 3 错误处理

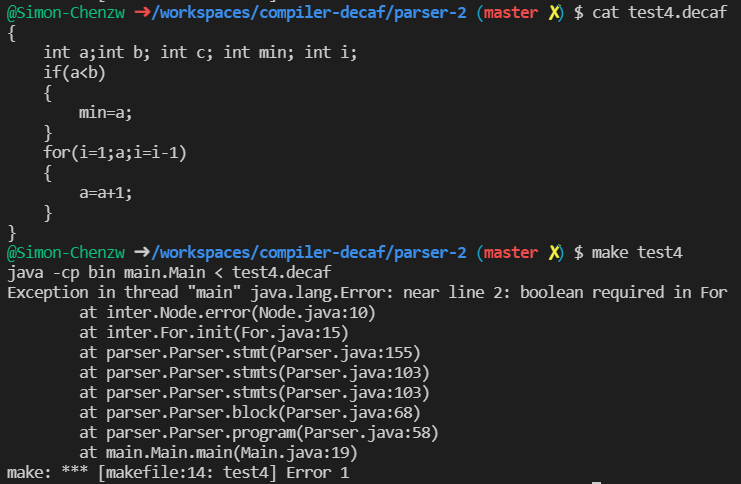
遇到语法错误时，通过调用 Node.error 扔出错误并打印信息。

3.4 运行结果









4、总结

　可以看出，语法分析器正确的解析decaf代码，并且正确的进行了语法检查，对固定位置的表达式进行了要求，并且正常的生成了中间代码。