# MiniJava Compiler 报告

组员:

刘瑞林-15307130083

陈啸天-15307130234

注：课堂报告内容已经为最终版本。此后没有新增内容。此报告与课堂报告一致。

1.工具选择

如果助教不推荐的话, 我们可能会选择JavaCC. 但是在了解了ANTLR4之后, 我们选择了ANTLR4.

优点:

1. 在网络上可以搜索到较多的关于antlr4的博客, antlr4 的语言更接近人能理解的自然语言, 学习曲线低, 可以快速入门.

2. 在github上可以找到大量的样例和guide

3. 输出支持多种语言. 我们选择输出java

4. 使用自顶向下的LL算法, 对于MiniJava来说已经足够了

2. 实现

1. **代码结构**

**/src/g4/**

包含所有词法语法文件. 每一个.g4文件对应一条BNF式. 其中,

Reserved.g4 包含保留字的定义, 以及空白字符的处理等

integer.g4 定义数字, 仅处理由多个数字连接的整型, 不包含"Ee.LlUuFf"等字符

identifier.g4 定义标识符, 仅处理由字母开始的后接字母或数字的标识符

其他词法/语法依照BNF式定义

**/src/java/**

MyMiniJavaVisitor.java 解释执行MiniJava代码.

MiniJavaTest.java 测试所有样例

**.bat文件 : 略**

1. **核心代码工作原理:**

将BNF式翻译成antlr4定义的语法后. antlr4能以一种元编程的方式, 将语法翻译成java逻辑, 其中,

将token转写至.token文件

将分析语法的java代码逻辑写入MiniJavaParser.java

在MiniJavaParser.java中, 实现了一个状态机, 进行语法分析. 定义了每一个语法主题的类, 继承ParserRuleContext, 如果有为语法加label, 则会有更详细的label子类继承父类, 以便在visitor中可以更准确地处理.

MiniJavaBaseVisitor.java定义了访问每一个类的接口

MyMiniJavaVisitor.java中实现了几个顶端的接口, 在额外功能中会详细介绍

1. **碰到的问题和解决方式**

最初在语法分析树中出现错误提示, 于是检查了.g4文件, 发现将某个正则表达式的括号翻译成字符括号, 所以出错, 于是进行改正.

在写解释器时, 碰到的问题较多, 我们通过输出大量的运行时数据, 掌握运行情况, 分析错误点, 并改正.

3. 错误处理机制

词法和语法错误均默认使用antlr4生成的java中的错误处理

删(截图见ppt)如果没放截图说明我忘记了

语义错误在运行时给出, 在MyMiniJavaVisitor.java中大量地使用了类似于

System.err.println("ERROR: Can't get Type "+t.toString()); 的错误提示.

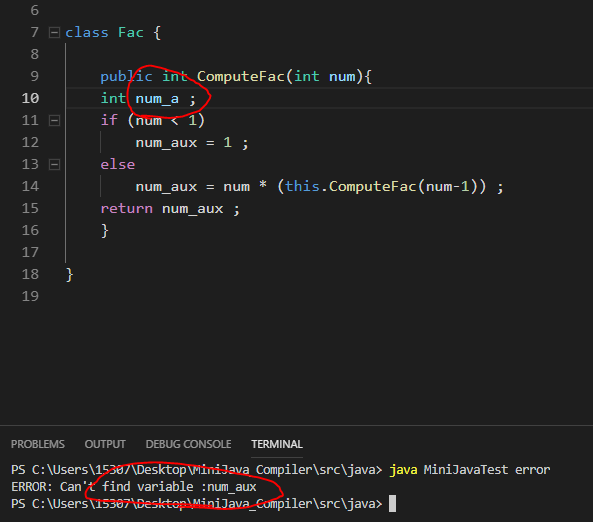
能检测的错误有:

**变量, 类, 类方法找不到.**

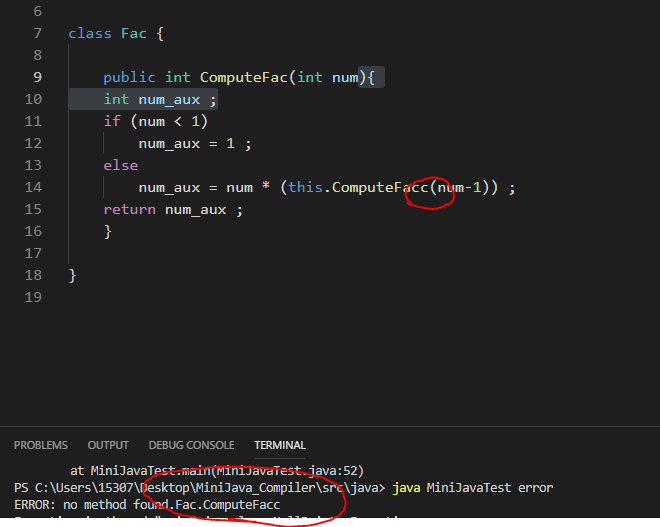
**数组地址错误**

部分错误发生后, 可以继续运行, 但是大多数错误发生后会直接退出.

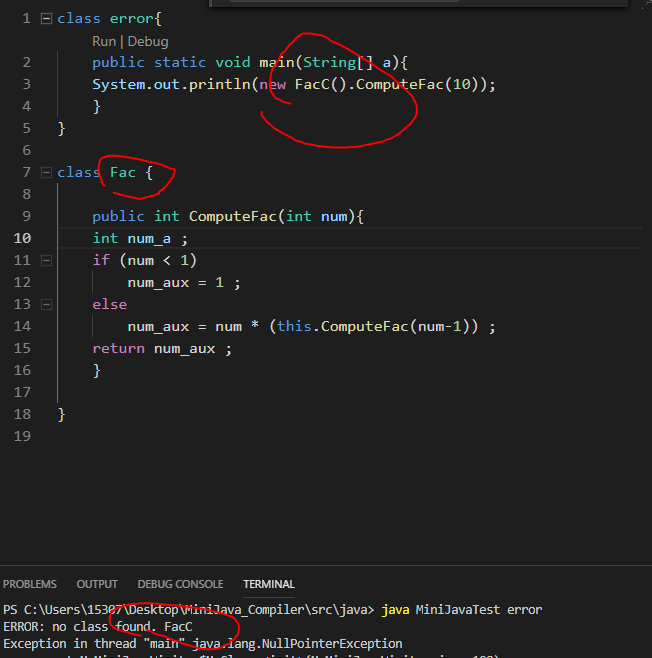
以下是截图



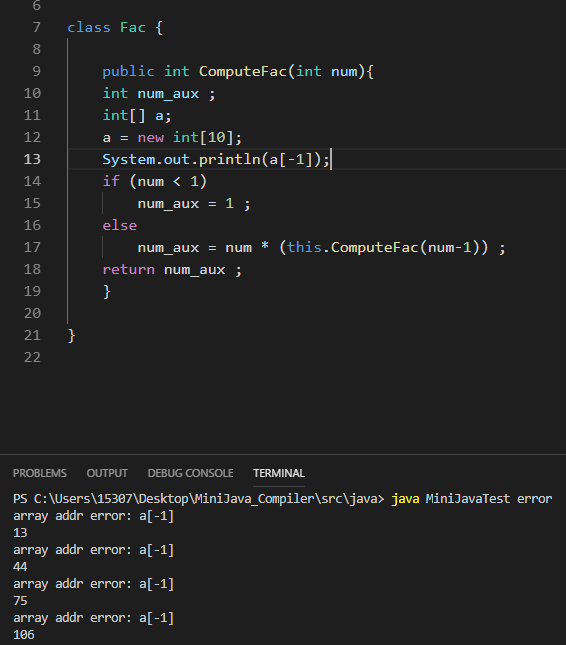
变量名错误



函数名错误



类名错误



数组地址错误

1. 额外功能: MiniJava解释器

基于visitor模式实现

class Myvar # 保存变量

class MyMethod # 保存函数

class MyClass # 保存类

// 全局变量保存类和对象

public List<MyClass> myClasses = new ArrayList<>();

public List<MyClass> myObjects = new ArrayList<>();

```

调用类或函数时, 实例化一个类或者函数, 在实例中保存变量. 查找也在实例中进行.

执行语句时, 因为在语法编写时, 标注了label, 根据每一种语句编写具体执行方式

感想

从头到尾编写一个编译器, 确实很难. 幸好有antlr4工具, 可以生成状态机分析词法和语法. 这样工作就简化到了词法语法编写, 简单许多.

在这个项目中, 受限于时间和精力, 许多很棒的功能没有实现, 比如引入更多的变量类型, 运算符, 但这会给解释器带来更大的困难. 另外也没有实现错误修复的功能, 真的非常遗憾

最后, 这个项目的完成, 要感谢老师在课堂上教授的知识, 感谢助教的文档以及网络上的众多资源. 谢谢!

2018年12月28日