编译原理与设计实验报告

姓名：卜梦煜 学号：1120192419 班级：07111905

# 1. 实验名称

语法分析实验

# 2. 实验目的

（1）熟悉 C 语言的语法规则，了解编译器语法分析器的主要功能；

（2）熟练掌握典型语法分析器构造的相关技术和方法，设计并实现具有一定复杂度和分析能力的 C 语言语法分析器；

（3）了解 ANTLR 的工作原理和基本思想，学习使用工具自动生成语法分析器；

（4）掌握编译器从前端到后端各个模块的工作原理，语法分析模块与其他模块之间的交互过程。

# 3. 实验内容

该实验选择 C 语言的一个子集，基于 BIT-MiniCC 构建 C 语法子集的语法分析器，该语法分析器能够读入词法分析器输出的存储在文件中的属性字符流，进行语法分析并进行错误处理，如果输入正确时输出 JSON 格式的语法树，输入不正确时报告语法错误。

# 4. 实验环境

Intellij IDEA Community 2021.3.2 + ANTLR 4.8

# 5. 实验过程与步骤

本实验首先阅读和理解BITMiniCC中AST结点定义，根据C11标准中对C语言语法、AST结点定义、四个测试用例，定义了一个C语言语法子集，写出相应语法产生式，并通过文法等价变换消除左递归；之后使用递归下降分析方法，基于BITMiniCC手动编码实现语法分析器，并将语法树输出为JSON文件；最后运行四个测试用例，验证语法分析器的正确性。

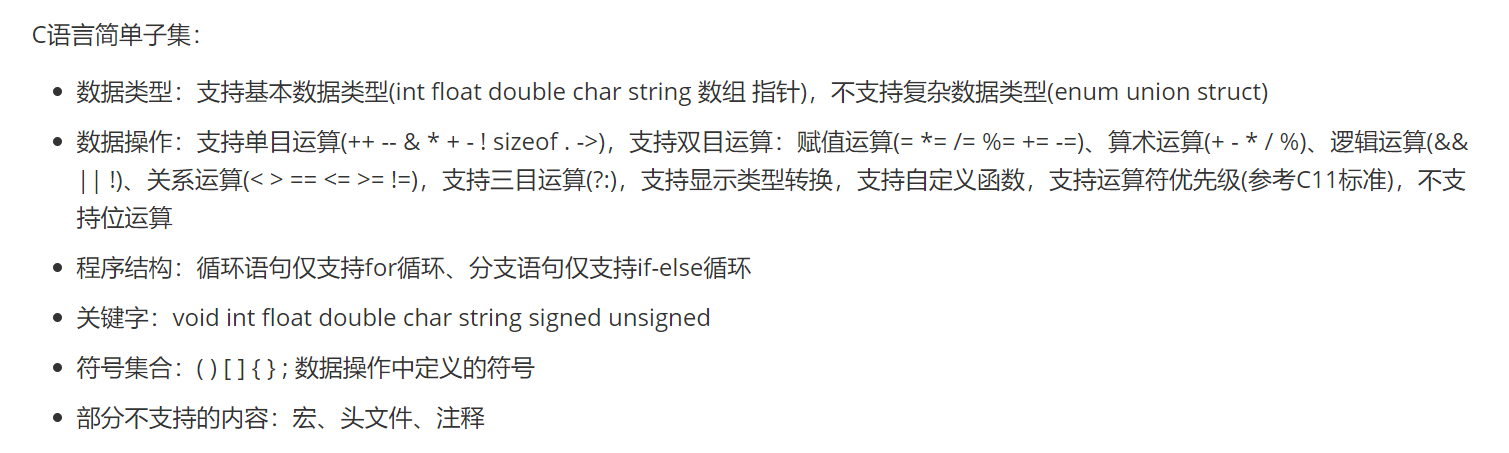
## 5.1 定义C语言语法子集

首先根据实验说明书中AST结点定义，结合BITMiniCC内部语法分析器运行测试用例输出的JSON文件，较为深入地理解了BITMiniCC中各AST结点含义、组成，以及AST结点之间继承、实现、聚合、组合关系。根据理解，将BITMiniCC中AST结点大致分为五大类，理解如下：

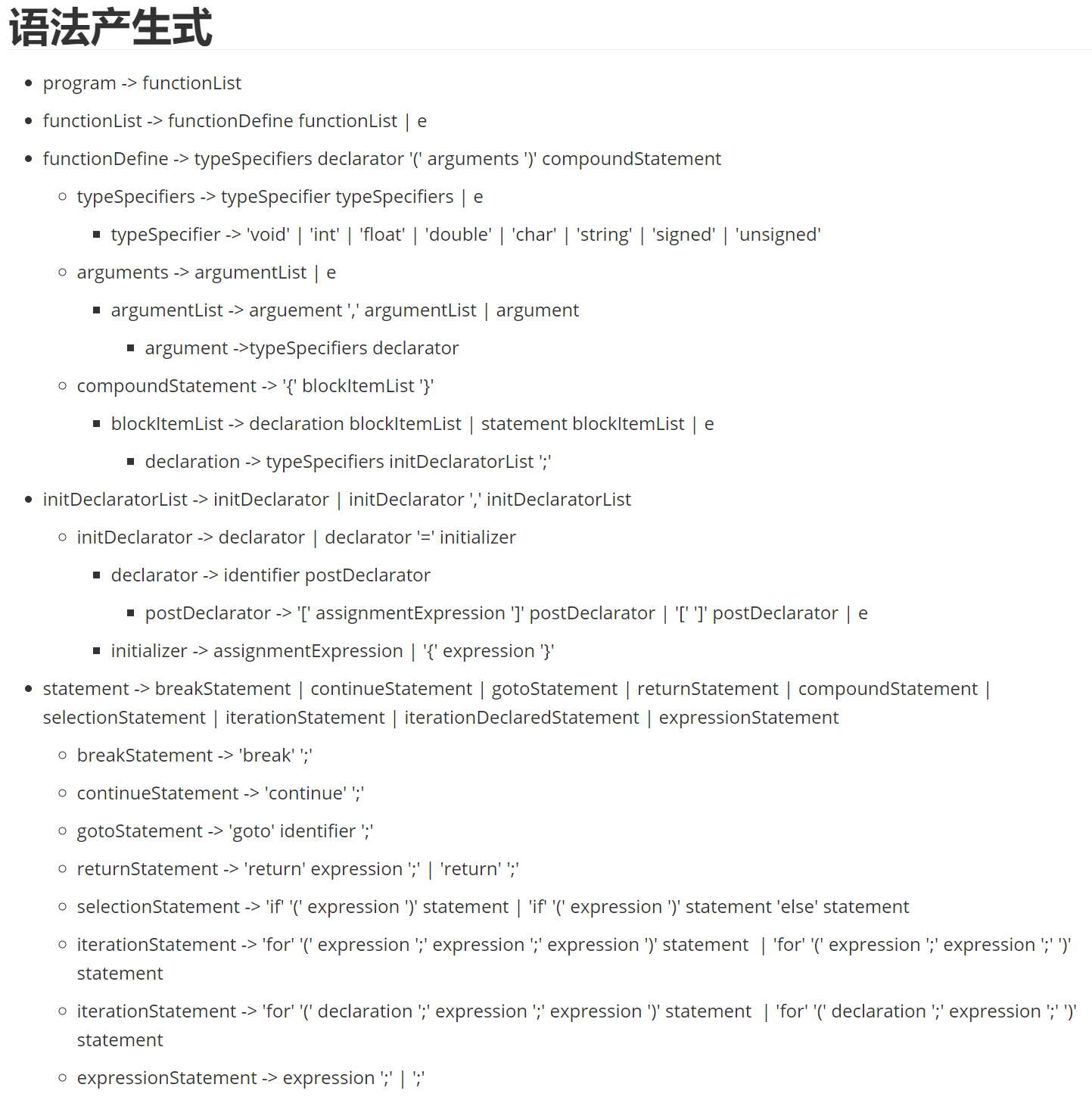




根据C11标准中对C语言语法、AST结点定义，以四个测试用例中涉及的C语言语法要求为基础，修改实验四中定义的C语言语法子集，得到新的自定义的C语言语法子集如下：



根据自定义的C语言语法子集，参考C11标准，得到该子集的语法产生式如下：

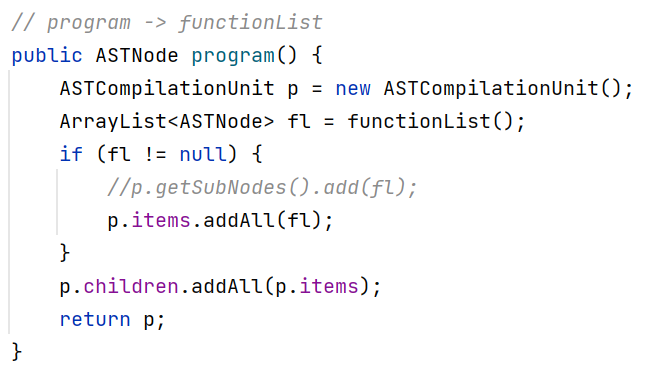




## 5.2 编码实现递归下降分析器

使用递归下降分析方法，基于BITMiniCC中的ExampleParser.java，手动编码实现语法分析器，并将语法树输出为JSON文件。部分关键代码如下：

语法分析入口的定义：



函数定义结点的定义：



语句结点的定义：

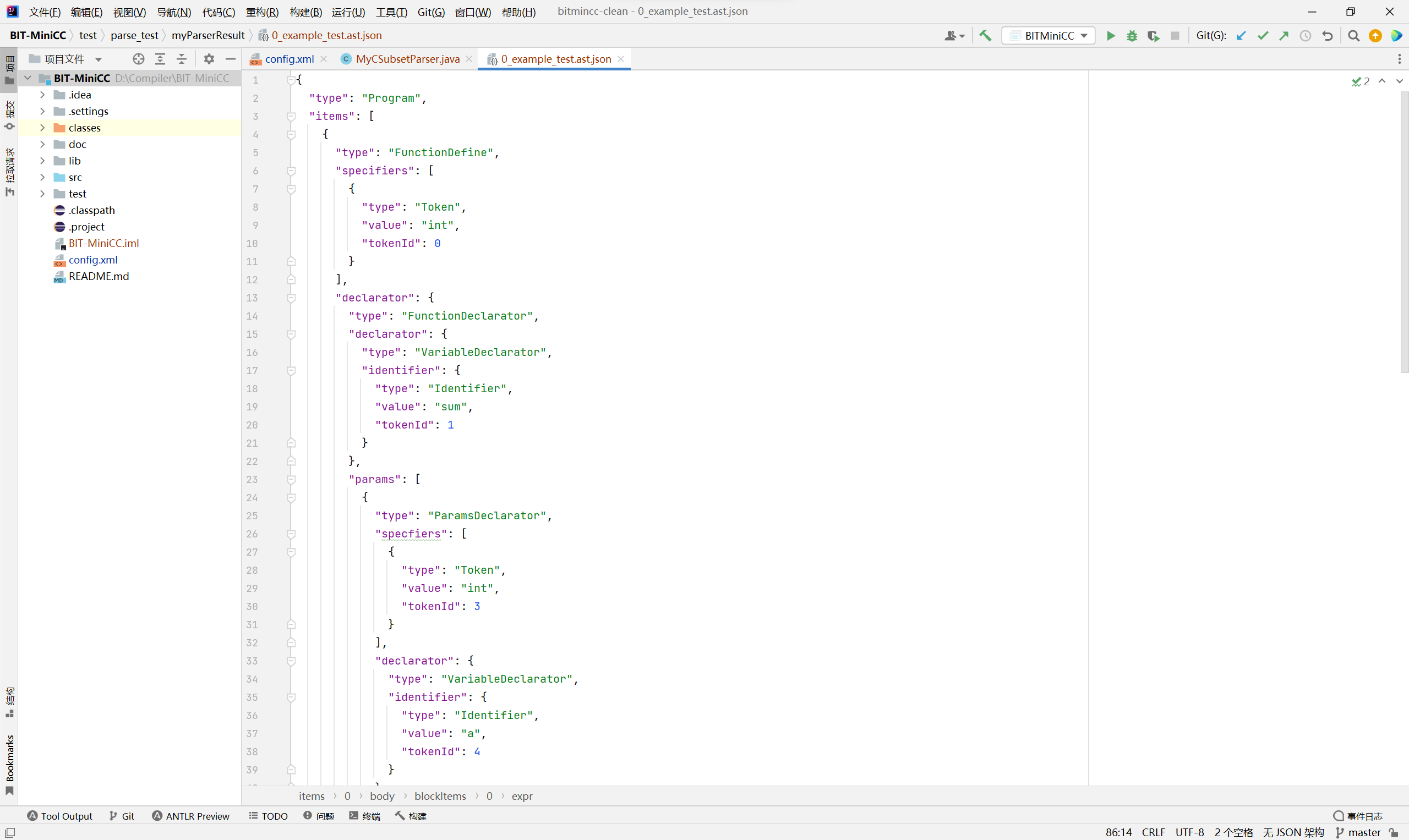


## 5.3 实验测试

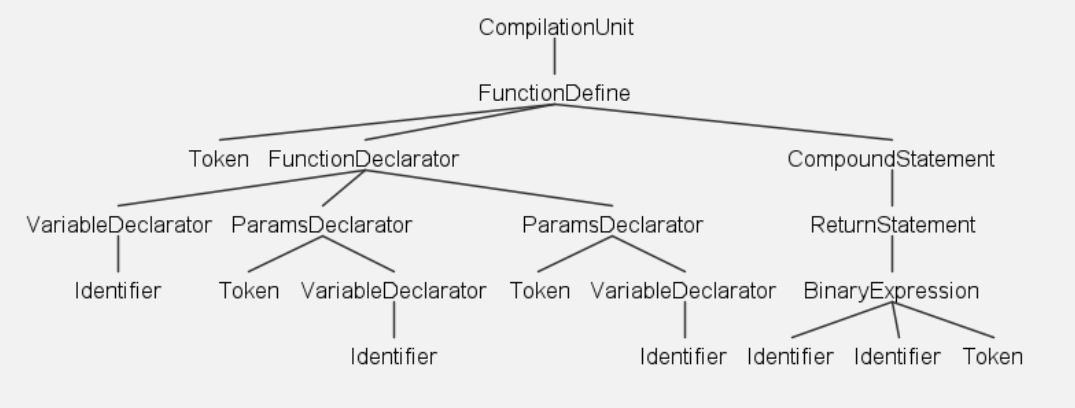
用BITMiniCC中的四个语法分析器测试用例对自定义的语法分析器进行测试，四个测试用例上测试结果分别如下：

（1）测试用例一：

生成JSON文件如下，与BITMiniCC内部语法分析器分析该测试用例生成的JSON文件内容相同：

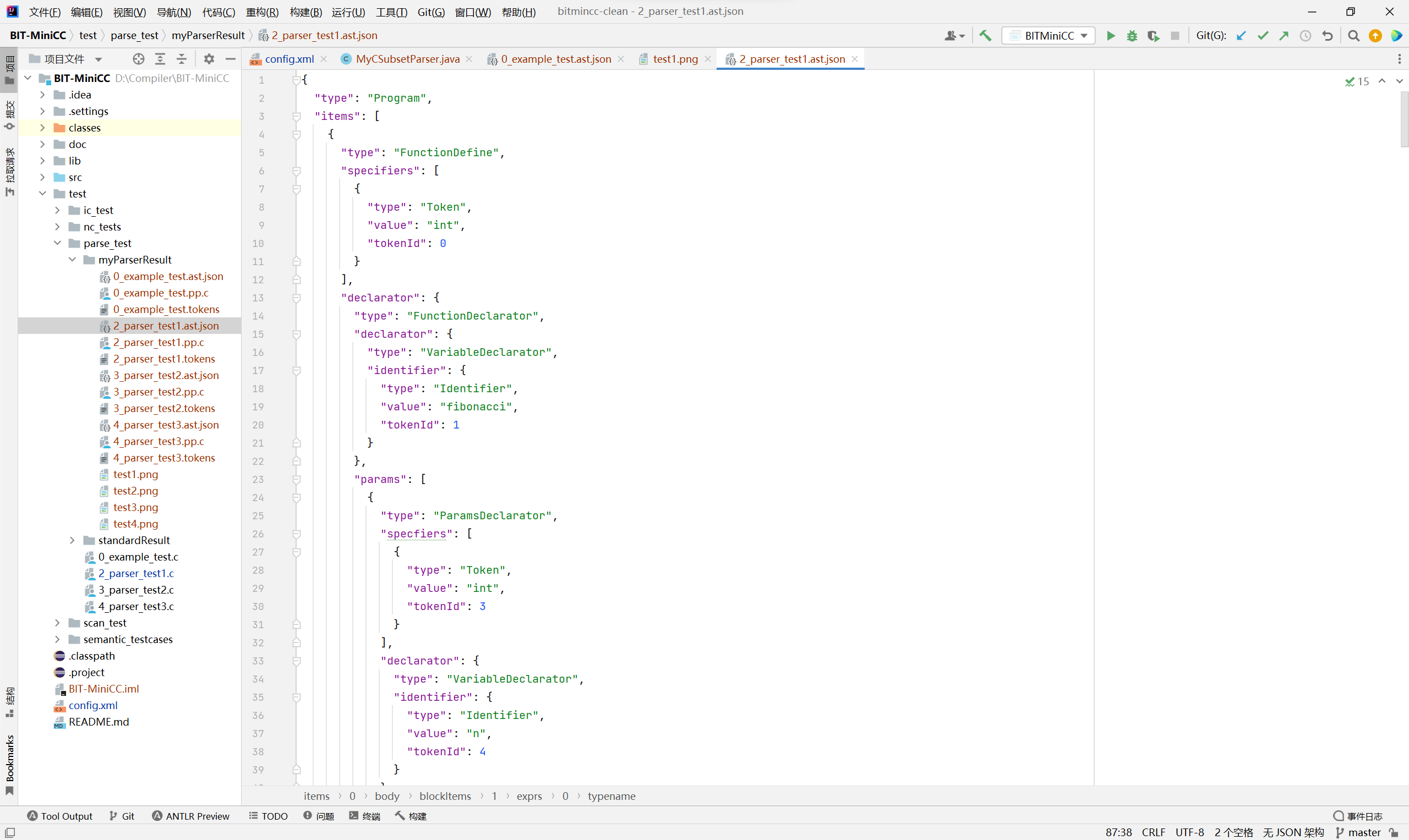


生成的语法树结构如下：

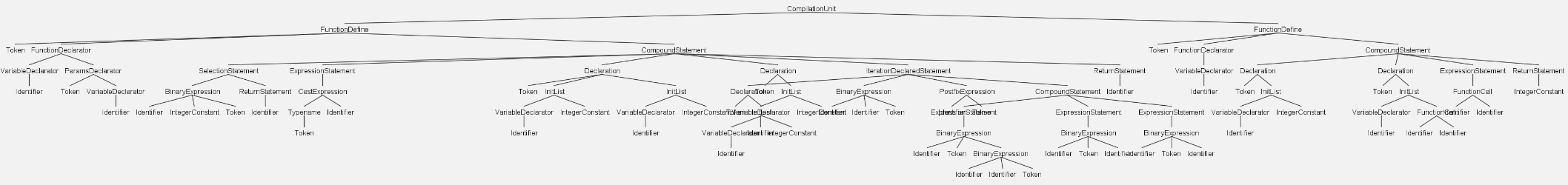


（2）测试用例二：

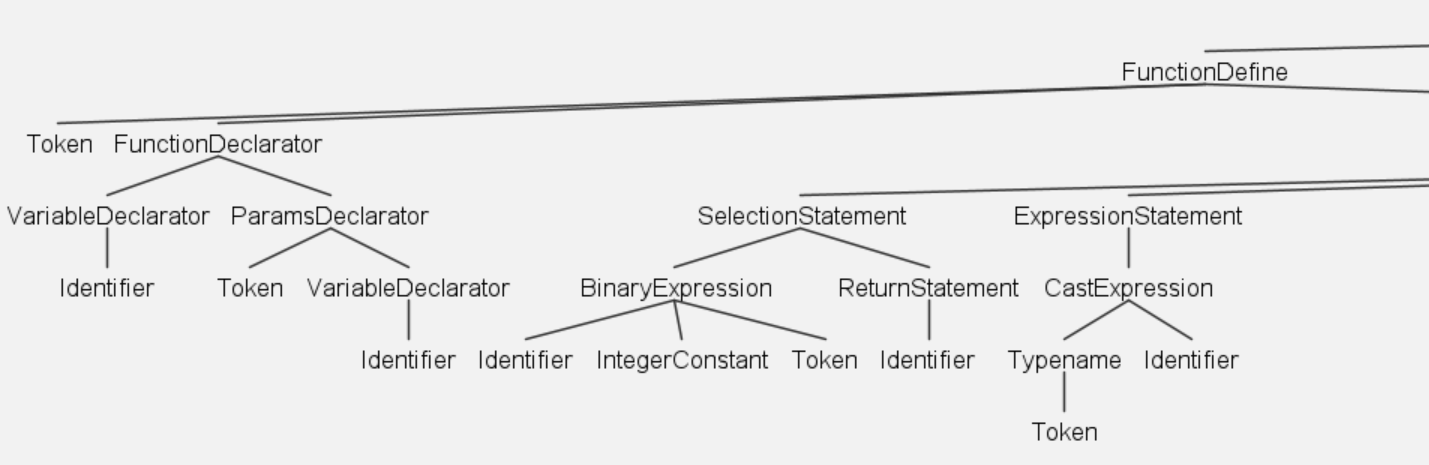
生成JSON文件如下，与BITMiniCC内部语法分析器分析该测试用例生成的JSON文件内容相同：



生成的语法树结构如下：

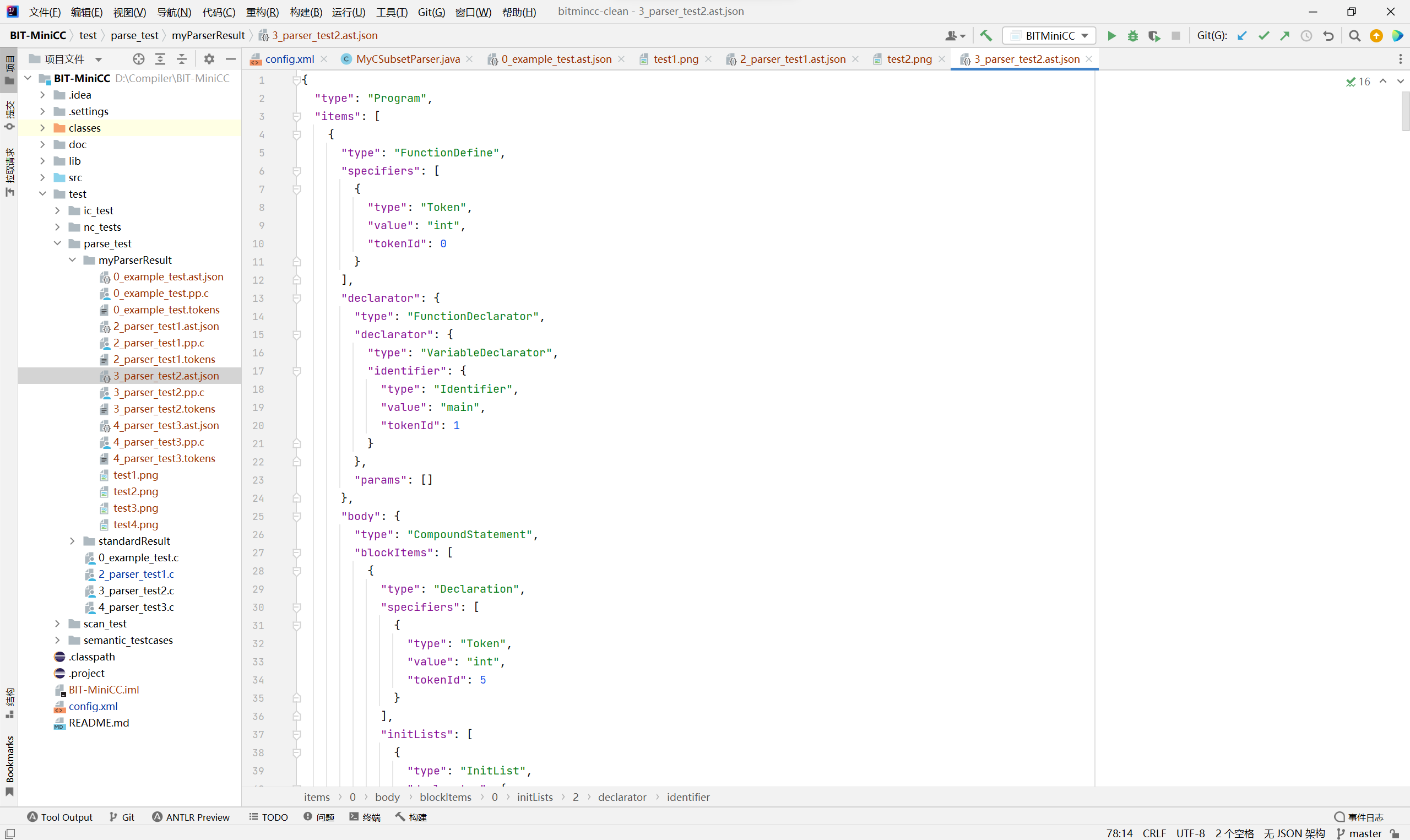


部分结构如下：

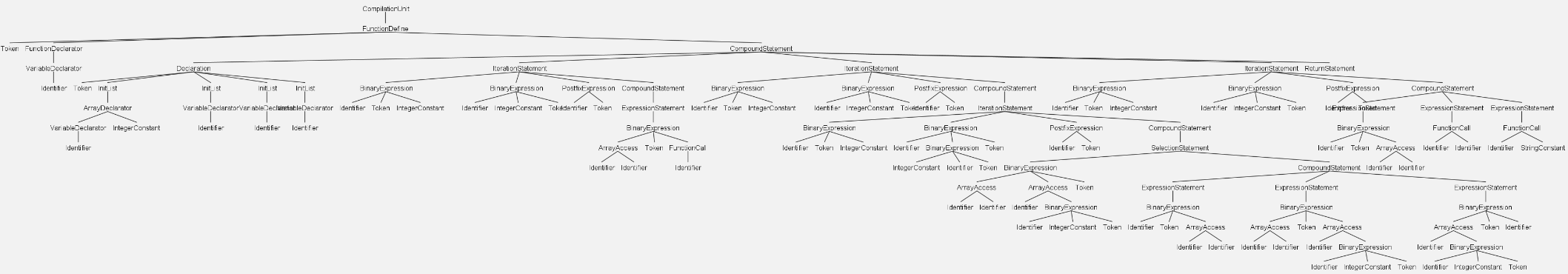


（3）测试用例三：

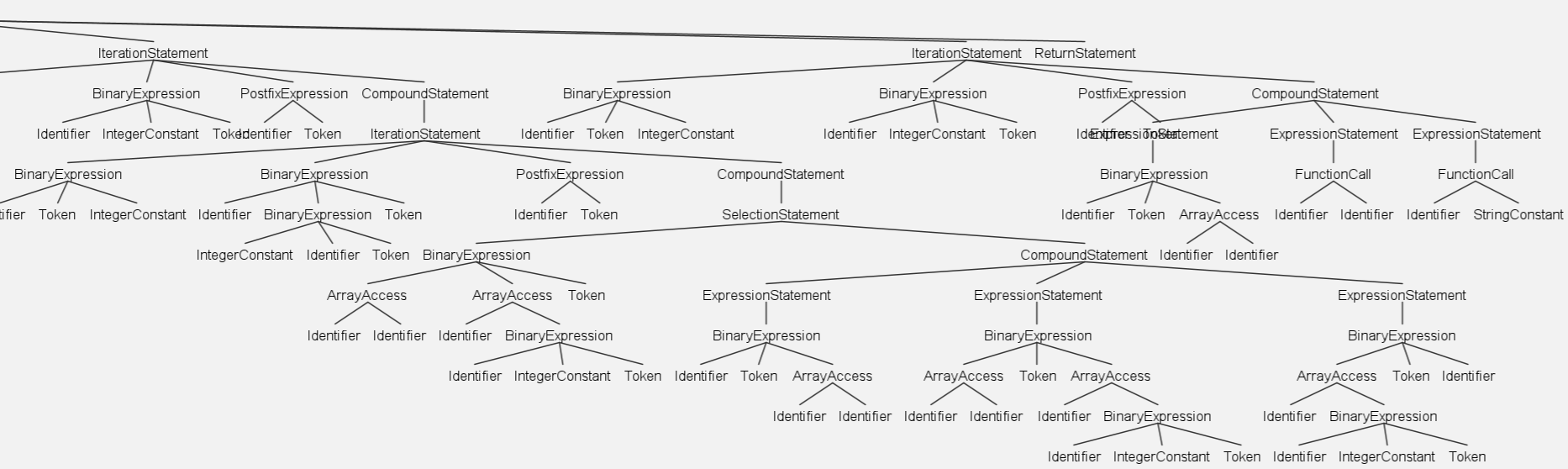
生成JSON文件如下，与BITMiniCC内部语法分析器分析该测试用例生成的JSON文件内容相同：



生成的语法树结构如下：

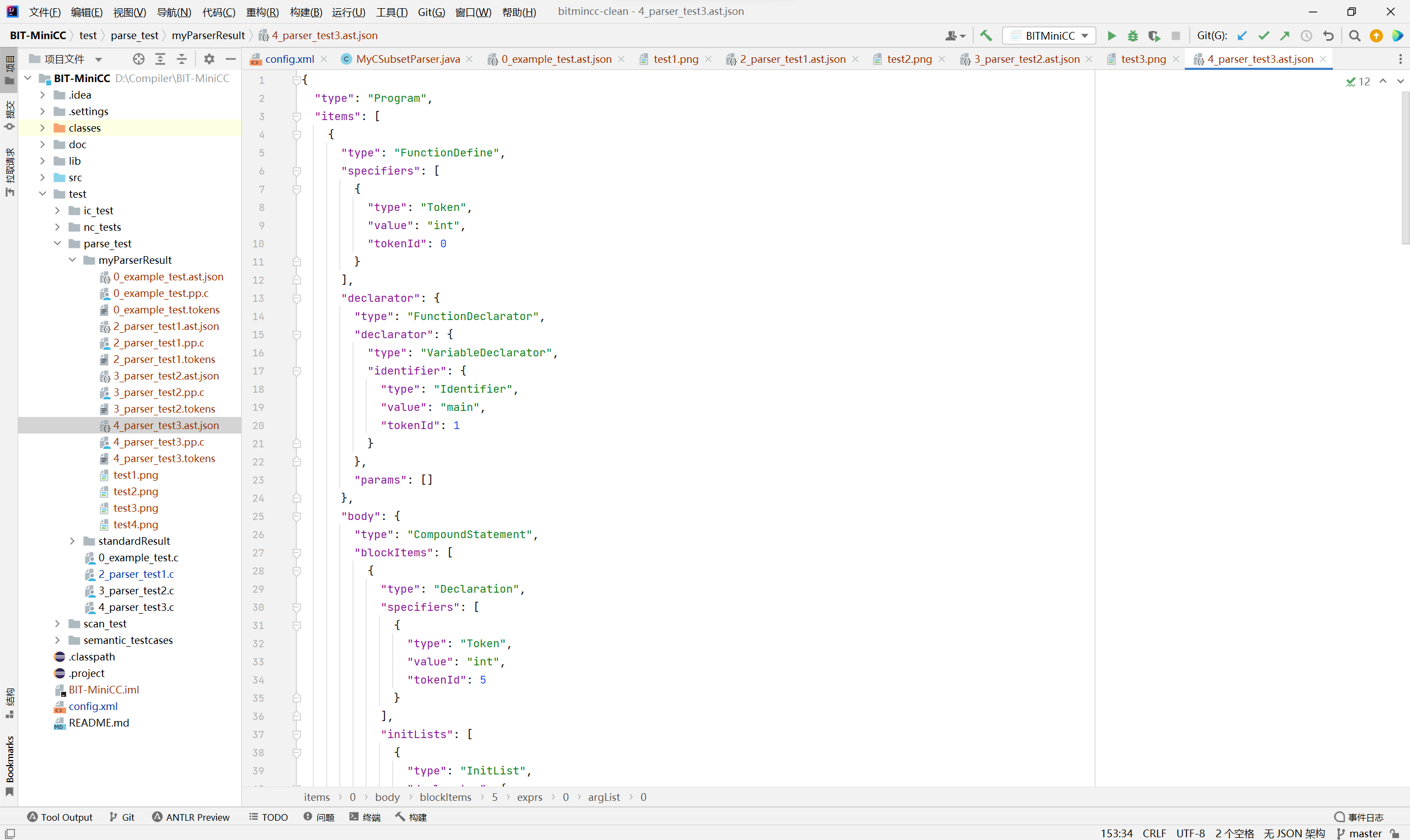


部分结构如下：

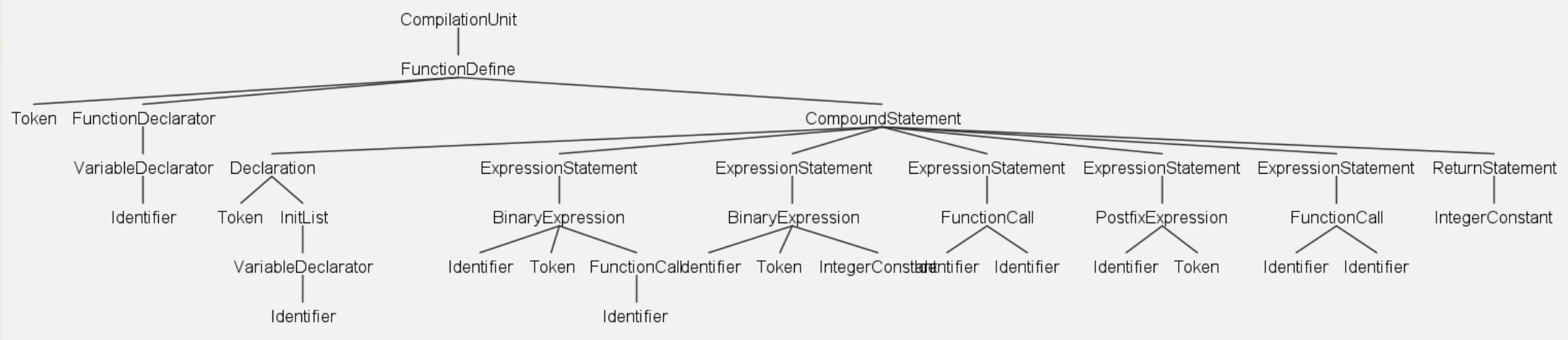


（4）测试用例四：

生成JSON文件如下，与BITMiniCC内部语法分析器分析该测试用例生成的JSON文件内容相同：



生成的语法树结构如下：



# 6. 实验心得体会

本次实验是目前遇到的难度最大的一次编译原理实验。实验中我首先采用了ANTLR自动生成工具，但在如何嵌入BITMiniCC的过程中遇到了一些困难，包括token文件读入方式，以及如何将AST结点写入语法分析器中，进而生成符合BITMiniCC定义的JSON文件。第一个问题可以通过手动编写符合结构的输入函数解决。第二个问题是主要问题，因为需要完全改写自动生成的语法分析器，并改写访问者模式的visit()函数，工作量较大。因此换用了手动编写递归下降分析器的方式编写能够嵌入BITMiniCC的语法分析器。

通过本次实验，我有如下收获：

（1）在实验四自定义的C语言语法子集的基础上，进一步加深了对C11标准中定义的C语言语法结构的理解。

（2）对BITMiniCC中语法分析器的AST结点定义有了更深入的理解，基本理解了BITMiniCC中语法分析器的结构，包括语法分析器的输入输出、语法产生式与AST结点的关系、JSON文件的理解、JAVA访问者模式的理解等等。