



Tutorial 5

AST Generation from ANTLR

Question 1.

Download the Exercise code, unzip it, and open file `node.py` in your favorite editor to discover the code. Try to run it by typing `"python node.py"`. Based on its code, write function `sum(root)` that calculates the total of all nodes in the tree.

Question 2.

Download the initial code of Assignment 2, unzip it and test whether it works well. In the folder `src/main/mp/astgen`, make a class named `Count` like class `ASTGeneration` in file `count.py`. Class `Count` performs counting all nodes in a parse tree generated by ANTLR. Modify `run.py`, `TestUtils.py` and add `CountSuite.py` to test your code.

Question 3.

Add one more testcase for `ASTGenSuite.py` with the following input:

```
a. input = """
    function _foo():_integer;
    begin_end
    """
```

Remind that the name of the tests must be meaningful and suitable to the test.

Question 4.

Change the rule in `MP.g4` so that it can accept a procedure/function with many statements in its body. Modify `ASTGeneration.py` so that it can generate AST in this case. Add some more testcases in `ASTGenSuite.py` to test your code.

Question 5.

Change the rule in `MP.g4` so that it can accept the local variable declaration in a procedure/function declaration. Modify `ASTGeneration.py` so that it can generate AST in this case. Add some more testcases in `ASTGenSuite.py` to test your code.