

Parse Tree Walking Activity

For this activity, you will practice using ANTLR-generated listener classes to process a parse tree.

Step 0. Checkout the treewalker project and generate the Java parser, etc.

In Eclipse, checkout this Maven Project (ignore line wrapping in the URL):

<https://utopia.cs.memphis.edu/course/comp4040-2013fall/examples/treewalker/trunk/>

In the `src/main/java/edu/memphis/cs/treewalker/` folder, you will find three separate sub-projects: **calculator**, **csv**, and **json**. Each of these sub-projects will be associated with a task below. Each one contains an ANTLR grammar (.g4) file.

The project uses Maven, so you must do **Maven -> Update Project** on the project. This step is necessary to make the project eventually compile; however, at this point the project will still have errors. We will fix those next.

Warning: You must use ANTLRWorks 2.1 for the following. Otherwise, your code will not run properly.

In ANTLRWorks, for each grammar file, do the following:

1. Open the grammar file.
2. **Run -> Generate Recognizer...** This should launch a **Generate Recognizer** wizard.
3. In the wizard, make sure that **Output directory** is set to the directory that the .g4 is in, and make sure that **Target language** is set to **Java**. Leave **Library directory** blank. Click **Next**.
4. On the next screen of the wizard:
 - a. Check **Generate listener**.
 - b. Uncheck **Generate visitor**.
 - c. Check **Package**, and set it to **edu.memphis.cs.treewalker.X** where **X** is the current sub-project's package (e.g., **csv**, **json**, etc.).
 - d. Click **Next**.
5. On the next wizard screen, uncheck all boxes, and click **Finish**.

These steps should generate four Java classes. If **Y.g4** is the current grammar file, then the classes will be **YParser.java**, **YLexer.java**, **YListener.java**, and **YBaseListener.java**. There will also be a couple **.tokens** files, which you can ignore. By generating these files all compiler errors should be fixed. You may need to **Refresh** the project to make it rebuild.

You can test-run each sub-project by right-clicking on the **MyXApp.java** file and doing **Run As... -> Java Application**. The applications should do nothing at this point (i.e., show an empty **Console** view, with no thrown exceptions, etc.).

Now you are all set to begin the tasks. For each, the only file you need to modify is **MyYListener.java**. You may print all output to **System.out**. There are test input files for each task in the top-level folder of the project.

Task 1. CSV translator

For this task you must rewrite a CSV file, giving it a new syntax. For example, if this is the input:

```
Name, Sex, Species, Weight  
Ralph, Male, Canine, 88
```

Then your output should look like this:

```
{Name;Sex;Species;Weight}{Ralph;Male;Canine;88}
```

Each row is surrounded by {}'s, the fields are separated by ;'s, and there are no newlines.

Task 2. JSON indentation

For this task, you must indent (i.e., "pretty print") a JSON file. For example, if this is your input (with no newlines and long lines wrapping):

```
{ "firstName": "John", "lastName": "Smith", "address": {  
"streetAddress": "21 2nd Street", "city": "New York",  
"state": "NY", "postalCode": 10021 } }
```

Then your output should look like this:

```
{  
    "firstName": "John",  
    "lastName": "Smith",  
    "address": {  
        "streetAddress": "21 2nd Street",  
        "city": "New York",  
        "state": "NY",  
        "postalCode": 10021  
    }  
}
```

Only the whitespace has changed. Each indent level is 4 spaces.

Task 3. Calculator

For this task, you must take an input arithmetic expression, and compute the result. For example, if this is your input:

6 + 5 * 2

Then your output should be:

16

Note that the grammar you were provided builds a parse tree that handles associativity and precedence in a sensible way.

Hint: You'll need to convert from Java Strings to integers. Here is documentation on how to do that:

<http://docs.oracle.com/javase/tutorial/java/data/converting.html>