Pascal ANTLR4 Scanner

The purpose of this assignment is to give you practice using ANTLR4, the scanner and parser generator that we will be using this semester.

> *ANTLR (ANother Tool for Language Recognition) is a powerful parser generator for reading, processing, executing, or translating structured text or binary files. It's widely used to build languages, tools, and frameworks. From a grammar, ANTLR generates a parser that can build and walk parse trees.*

For this assignment, you will only be creating a Lexer (scanner) and not the parser. Soon, we will be creating both.

Requirements

Start by pulling the latest from the class repository and opening the Simple4Scanner maven project. When you build the project, it will download the necessary ANTLR4 packages. Build the project and familiarize yourself with the two-step process whereby the project first generates the Lexer class from the G4 grammar file and then compiles the code.

You have been provided with a partially completed G4 grammar file. You should be able to successfully build the application and run the tests. *Note, all five tests will fail.*

Your assignment is to complete the grammar so that all five tests pass.

Additional Information

You are expected to draw upon what you have previously learned regarding regular expression and grammars. However, you will have to do some self-learning on the specifics of ANTLR4 syntax. Use the ANTLR website and online documentation. At this stage, you should not need the ANTLR command line tools, but it is recommended to follow the instructions, install them locally and start becoming familiar with them. Both IntelliJ and VSCode have additional support for ANTLR that we will take advantage of very soon when we start building the parser.

Comments

The tokens and output are *almost* identical to previous assignment. However, there are some differences since I tweaked the handling of certain situations for clarity.

To pass all the tests you should not need to change any code. You should only have to change the G4 file.

To use the program from the command line, you can build the jar using the following maven command from the project root:

```
mvn clean compile test install -DskipTests
```

This will create the jar file in the target directory. You can then run the program from the command line against the test input files (found in `target/test-classes/input`) or any other input file:

```
java -jar target/Simple4Scanner-1.jar test.txt
```

Submission

- ✓ Check in your working code to your YU GitHub repository in your **Compilers** branch.
- ✓ Check in the G4 file to the **/hw3-simple4scanner** directory.