Antonio E. Ramirez

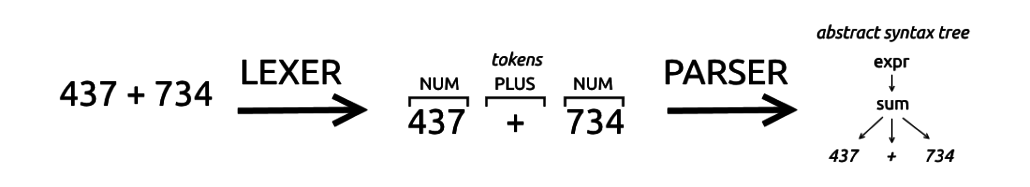
15 September 1, 2019

Project 1

CMSC 330, 6381, University of Maryland University College

# Process

This project was deceivingly simple at first glance. The given instructions were short and simple. Create a program that parses a GUI from a predefined language. We were given the grammar and a test case with the expected output. I did not have any experience with parsing or lexers prior to this project. My first step was to get a good understanding on the topics. In my research I found 2 helpful sites that I will reference below. I also found a picture that sums up the difference nicely and brought it all together for me.



After this initial research I developed a basic idea of how I wanted to tackle this project. I knew I would have to take the given grammar and create a swing GUI from that. With that goal in mind, I then investigated how I would read and analyze the input file. I found a helpful article that presented 2 options, I could take the file and read it line by line, using the String.split() method to break each line apart for analysis. My other option was the StreamTokenizer class. I had never worked with this class, so I read through the documentation to see what kind of capabilities I would have available to me. I now had a basic idea of how I wanted to solve the problem. I would read the file, break it into tokens, analyze each token, and depending on the value create a component to be added to the GUI. During my research I came across a GitHub repository, referenced below, that also solved this problem. I looked at their approach and found it also used a StreamTokenizer approach and implemented a lexer and a parser to create the GUI. I was influenced heavily by this example and used a lot of the same elements in my solution. After creating a program that would create the expected GUI from the given test case, I know had to develop another test case to ensure my code was functional. I also wanted to purposefully pass files with incorrect grammar to see if the correct error message would be generated.

Resources:

<https://github.com/CrutchTheClutch/CMSC-330-Project-1>

<http://savage.net.au/Ron/html/graphviz2.marpa/Lexing.and.Parsing.Overview.html>

<https://stackoverflow.com/questions/2842809/lexers-vs-parsers>

<https://crunchify.com/java-stringtokenizer-and-string-split-example/>

# Lessons Learned

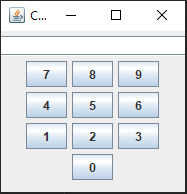
By implementing the StreamTokenizer class, I was able to learn a new class and its capabilities. I also was unaware of the correlation between char types and tokens. Both are fairly easy to convert to int type (<https://www.javatpoint.com/java-char-to-int#targetText=Java%20Convert%20char%20to%20int&targetText=If%20we%20direct%20assign%20char,.valueOf(char)%20method.>). This proved to be very useful within one of my switch cases in particular. Another area of learning was enums. I have read about them previously but had never used them in a project before. This were useful in checking the token values and ensuring it was part of the given grammar. I would like to get more comfortable with using enums. While I do not have too much experience with swing, since my previous GUIs were designed using JavaFX, I thankfully had experience with JFileChooser and all the basic swing elements used in this project. I would like to get more familiar with swing and building GUIs in general.

# Test Cases

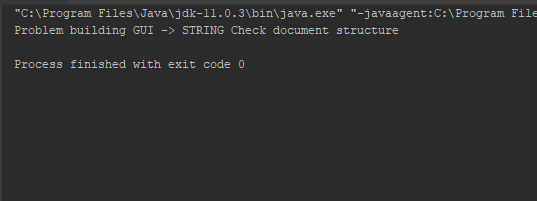
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Number | Test Scenario | Input | Expected Output | Pass/Fail | Screenshot Number |
| 1 | Given test case | Window "Calculator" (200, 200) Layout Flow:  Textfield 20;  Panel Layout Grid(4, 3, 5, 5):  Button "7";  Button "8";  Button "9";  Button "4";  Button "5";  Button "6";  Button "1";  Button "2";  Button "3";  Label "";  Button "0";  End;  End. | Proper GUI (See Screenshot) | Pass | 1 |
| 2 | Incorrect grammar | Window ;"Calculator" (200, 200) Layout Flow:  Textfield 20;  Panel Layout Grid(4, 3, 5, 5):  Button "7";  Button "8";  Button "9";  Button "4";  Button "5";  Button "6";  Button "1";  Button "2";  Button "3";  Label "";  Button "0";  End;  End. | Error message in console (See Screenshot) | Pass | 2 |
| 3 | Different GUI (Double Number Panel) | Window "Calculator" (200, 200) Layout Flow:  Textfield 20;  Panel Layout Grid(4, 3, 5, 5):  Button "7";  Button "8";  Button "9";  Button "4";  Button "5";  Button "6";  Button "1";  Button "2";  Button "3";  Label "";  Button "0";  End;  Panel Layout Grid(4, 3, 5, 5):  Button "7";  Button "8";  Button "9";  Button "4";  Button "5";  Button "6";  Button "1";  Button "2";  Button "3";  Label "";  Button "0";  End;  End. | GUI with 2 sets of numbers, in the same format. | Pass | 3 |

# Screenshots

Screenshot 1:



Screenshot 2:



Screenshot 3: Window expanded to fit all numbers

