Kevin Abrahams

CMSC330 Project 1

1/27/21

# Introduction and Programming Process

In this assignment, I have implemented the lexical analyzer and parser for the given grammar to build the GUI. The assignment started off with me beginning with the grammar to define the expression and then taking the expression represented by the input file and converting it into a GUI like the phenomenon of parsing a string expression with a parsing tree. The heavy, substantial, and pivotal task at hand was to know how to generate the GUI based upon the input file and knowing how to convert the contents of the input file into appropriate java code. To help facilitate this task, I created a helper utility known as Token which would store an enumeration or collection of tokens representing appropriate values which the user could enter. This helped with my lexical analyzer and parser tremendously in terms of translating and converting the contents of the input file. To accomplish this programming task, I separated various key components of the programming approach to various classes, including a lexical analyzer, a parser, a custom named syntax exception, a main class to run the program and select the input file storing the contents by which to create the GUI, and a token representing valid entries in the input file which formed the syntactic structure of the grammar. To fulfill each critical objective, I had my parser structure the various parts, assets, and segments of the input file. The combination of the lexical analyzer and the parser was crucial and imperative to achieve the translation and conversion of the contents of the input file to java code and GUI.

# Test Cases

1. For the first test case, I first start the program which causes a JFileChooser to appear in the screen prompting the user to select a file. Using the JFileChooser, the input file I select for this first test case is called inputfileA.txt. The following is the content of the input file called inputfileA.txt.

Window "Calculator" (300, 300) 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";

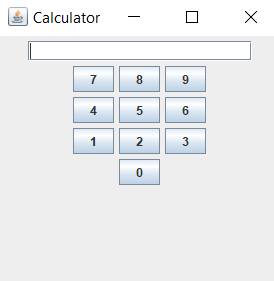
End;

End.

From the above input file, I have tested Window Generation, Grid Layout, Button Widgets, Label Widgets, Panel Widgets, and Textfield Widgets.

Below, a screenshot of the produced GUI from inputfileA.txt is shown below.

**Screenshot**



1. For the second input file called inputfileB.txt, I also select it using the JFileChooser which initially prompts the user at the beginning of the program once it starts. This indicates my additional test with some other GUI grammar. For this test case, I provide examination of the Window Generation, Flow Layout, Grid Layout, Button Widgets, Group Widgets, Radio Button Widgets, Label Widgets, Panel Widgets, Textfield Widgets, and Nested Panels. The following, shown below, indicates the contents of the second input file which is called inputfileB.txt. To reiterate, the user must select this file using the JFileChooser which appears once the program first commences.

Window "OtherWidgetsTest" (400,400) Layout Flow:Panel Layout Grid(5,1):

Button "Button";

Label "Label";

Textfield 10;

Panel Layout Flow:

Panel Layout Grid(1, 6, 5, 5):

Group

Radio "1" ;

Radio "2" ;

Radio "3";

Radio "4" ;

Radio "5" ;

Radio "6";

End;

End;

End;

Panel Layout Flow:

Button "Double";

Button "Nested";

Button "Panel";

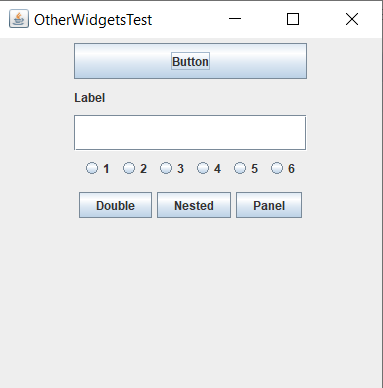
End;

End;

End.

Below shows a screenshot of the GUI created by the input file called inputfileB.txt, selected from the JFileChooser at the start of the program.

**Screenshot**



1. In this third test case, I have also tested for syntax errors. I have modified the given calculator grammar to test the syntax error. This third input file, loaded as well from the JFileChooser, is known as inputfileC.txt. The contents of inputfileC.txt are shown below.

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 "";

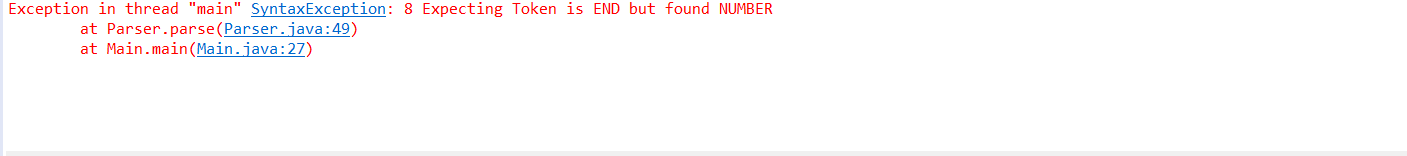
Button "0";

End;

End.

The following shows a screenshot of the syntax error triggered by the program because of the missing quotations around 5 in the line that states Button 5.

**Screenshot**



1. As a fourth test case, I provide a more explicit example of syntax error due to a blatantly obvious violation of the grammar set forth by the programming task. This fourth input file is called inputfileD.txt. I show extreme violation of the grammar, as the input file consists of plain English, indicating a similar result to the third test case.

The following indicates the contents of the fourth input file, inputfileD.txt.

Hello, world!

This is an input file with wrong grammar.

The following shows the unwelcome result of the syntax error issued by the compiler and program due to the grammar not being followed. Thus, the syntax error works as expected.

**Screenshot**

****

# Conclusion and Lessons Learned

In this assignment, I learnt about how the lexical analysis and parsing is done from a given grammar. The parser was difficult to implement. The assignment was a bit challenging. I enjoyed the assignment. It taught me more about recursive descent regarding parsing, which I have now come to better understand, as this programming exercise served as a pragmatic example and teaching tool. This lesson connects well with our discussion about parsers and parsing trees which relates to the process of a program’s compilation. Thus, I have a better working knowledge of the backbone of a program and its inner workings. This assignment extended my programming ability and built off of old previous knowledge. It was difficult to translate the grammar to actual java code, but it was invaluable as well. It was interesting to see how many of my former java knowledge pertained to and helped in accomplishing this programming task, such as creating custom named exceptions, building GUI by hand through java code, and traversing through and navigating an input file.