

Cory W. Cordell
Michael Nguyen
February 26, 2016

Assignment 2

Ruby

Project Design

This program consists of two classes, a lexer and an analyzer. The lexer first has a hash of all the different kinds of tokens in order to keep track of the type of token. The `getTokenKind()` function uses an array and a counter in order to return the kind of token. The `getTokenText(token)` function returns the text representation of the token using a key. There is a `parsefile` function that keeps track of the line number while parsing the file and a `parseline` function that parses each line token by token and scans each token to see which token kind each token is or if there is an error. The analyzer defines the function program and determines whether a file is a SimPL program or not and determines where the error is. The program function checks if the SimPL program ends in a semicolon or has an error and also checks if the function stmts is syntactically correct. The function stmttss checks if there are any errors or and goes down a step to check if the function stmt is correct or not. The function stmt checks whether the next token is an identifier, an 'if', or a 'while' token and handles each appropriately according to the SimPL language. The functions addop and mulop add checks to make sure that addop and mulop are syntactically correct when used according to the SimPL language. The factor function makes sure that the next token is an integer, an identifier, or another addop. The lexpr and lterm functions add checks to make sure that lexpr and lterm are syntactically correct in accordance with the SimPL language. The lfactor function checks the next token is 'true', 'false', or a relop function. The relop function handles greater than, less than, and equal to checks that are syntactically aligned with the SimPL language. A run.rb program was added in order to test files with the analyzer. There is a readme.txt file that explains how to use the run.rb program.

Collaborators

Michael Nguyen and Cory Cordell both worked on the code and had input into this report. Cory came up with the overall design of the program, while Michael did testing.

Difficulties

Some of the difficulties with this assignment included parsing a file while including checks for end of lines and end of files. Another difficulty was understanding the SimPL language. For the analyzer, including checks for each of the functions in order to make sure all the tokens are syntactically correct in accordance with the SimPL language.

Like About Ruby

Ruby is really well documented and fairly easy to learn. The idea that everything is an object and a class in ruby is really interesting. Arrays, loops, and conditional statements in ruby were easy to learn. The object oriented concept of the language made learning the language easier to learn as well. Lastly, Ruby is really easy to read and understand even without knowing much about the language.

Dislike About Ruby

The IRB commands are confusing at times. The syntax of ruby was different and took some time to figure out.