SOFTWARE DESIGN DOCUMENT

MINI-PASCAL COMPILER

Version 0.2

William Mork

Augsburg University

PROJECT OVERVIEW

Introduction

This project, written in Java 8, is a compiler which parses Mini-Pascal code to generate MIPS assembly code. Please refer to the “Project Structure” section for clarity on the files used in the project.

Module 1: Scanner

The scanner module reads a Mini-Pascal text file and scans each line. Keywords and symbols which are recognized as valid (listed below) by the scanner are converted into “tokens”, which will later be handled by the parser module.

Scanner.java is a file which has been generated by JFlex, a lexical analyzer (scanner) generator. The generator uses a specified set of token types, expected patterns, and lexical rules to create a deterministic finite automata (DFA) which is used to construct the aforementioned token stream.

Token.java defines a token object containing the token lexeme and type.

TokenType.java enumerates the list of valid keywords and symbols.

Valid keywords:

AND ARRAY BEGIN DIV DO ELSE END FUNCTION IF INTEGER MOD NOT OF OR PROCEDURE PROGRAM REAL THEN VAR WHILE READ WRITE RETURN

Valid symbols (token type is listed first, followed by the symbol itself):

SEMI ; COMMA , PERIOD . COLON : LBRACE [RBRACE] LPAREN (RPAREN) PLUS + MINUS - EQUAL = NOTEQ <> LTHAN < LTHANEQ <= GTHAN > GTHANEQ >= ASTERISK * FSLASH / ASSIGN :=

Module 2: Parser

The recognizer / parser module employs an instance of the scanner class to iterate through tokens of an input stream and match them with the production rules articulated in the MicroPascal grammar (Grammar.pdf).

To use the current version of the recognizer class, create an instance and then call the top-level function, <code>exp()</code>. If the function returns without encountering an error, the file is an acceptable MicroPascal program or contains an acceptable expression.

The Recognizer class can be used to read the token stream of a pascal file or a provided input String.

Project Structure

- MorkCompiler
  - documentation
    - Grammar.pdf
    - SDD.pdf
    - User Manual.pdf
  - src
    - pascal
      - simple.pas
      - simplest.pas
    - scanner
      - LookupTable.java
      - Scanner.flex
      - Scanner.java
      - ScannerTest.java
      - Token.java
      - TokenType.java
    - parser
      - Recognizer.java
      - RecognizerTest.java
    - symbol table
      - SymbolTable.java
  - .gitignore
  - README.md

Master Changelog

Commit ID	Commit Tag	Version	Description	Date
Initial commit				
5B22F20	N/A	0.0.0		1/17/2019
Initial commit (remote)				
6A7EEAD	N/A	0.0.0		1/27/2019
Initialized files for module 1 - scanner				
46D4E1D	N/A	0.0.0		1/27/2019
Imported old files				
98ED5E0	N/A	0.0.0		1/27/2019
Final commit for Module 1				
56892B8	Scanner	0.0.1	Finished Scanner module.	1/31/2019
Final commit for Module 2				
	Recognizer	0.0.2	Finished Recognizer module.	2/25/2019