# SOFTWARE DESIGN DOCUMENT

#### **MINI-PASCAL COMPILER**

Version 0.1

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

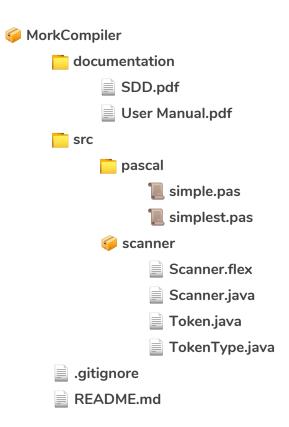
Valid symbols (token type is listed first, followed by the symbol itself):

```
SEMI; COMMA, PERIOD. COLON: LBRACE [ RBRACE ] LPAREN ( RPAREN )

PLUS + MINUS - EQUAL = NOTEQ <> LTHAN < LTHANEQ <= GTHAN >

GTHANEQ >= ASTERISK * FSLASH / ASSIGN :=
```

### **Project Structure**



## **Master Changelog**

Commit ID	Commit Tag	Version	Description	Date
Initial commit				
5B22F20	N/A	0.0.0		1/17/2018
Initial commit (remote)				
6A7EEAD	N/A	0.0.0		1/27/2018
Initialized files for module 1 - scanner				
46D4E1D	N/A	0.0.1		1/27/2018
Imported old files				
98ED5E0	N/A	0.0.1		1/27/2018