Compiler #2

Scanner

Description:

Compiler #2 is a **case sensitive** Object Oriented Computer programming language Like C++-Language.

Lexical Analysis:

Compiler#2 Scanner is a lexical analysis stage used to locate and identify language's lexemes. It must recognize the following Lexemes and returns Return Token according to the following table:

Lexeme	Meaning	Return Token
Divisio	is the blueprint from which individual objects are created.	Class
InferedFrom	Inheritance in OOP	Inheritance
WhetherDo-Else	conditional statements	Condition
Ire	Integer type	Integer
Sire	Signed Integer type	SInteger
Clo	Character Type	Character
SetOfClo	Group of characters	String
FBU	Float type	Float
SFBU	Signed Float type	SFloat
NoneValue	Does not return a value	Void
TerminateThisNow	Break immediately from a loop	Break
RingWhen	repeatedly execute code as long as condition is true	Loop
BackedValue	Return a value from a function	Return
STT	grouped list of variables placed under one name	Struct
Check –CaseOf	To switch between many cases	Switch
Beginning	Program Starting Statement	Start Statement
End	Program Ending Statement	End Statement
(+, -, *, /,)	Used to add, subtract, multiply and divide respectively	Arithmetic Operation

(&&, ||, ~)	Used to and, or and not respectively	Logic operators
(==, <, >, !=, <=, >=)	Used to describe relations	relational operators
=	Used to describe Assignment operation	Assignment operator
.	Used in STT to access STT elements	Access Operator
{,},[,]	Used to group class statements, statements or array index respectively	Braces
[0-9] and any combination	Used to describe numbers	Constant
“,’	Used in defining strings and single character respectively	Quotation Mark
Using	Used to include one file in another	Inclusion
/#	Used to Comment some portion of code (Multiple Lines)	Comment
#/	Used to a matcher to Comment left side (Multiple Lines)	Comment
/-	Used to Comment some portion of code (Single Line)	Comment
@	Used as Token Delimiter	Delimiter
;	Used as Line Delimiter	Delimiter

Com#2_1: Tokens Description

The Scanner also recognizes identifiers. An identifier is a sequence of letters and digits, starting with a letter. The underscore ‘_’ counts as a letter. For each identifier, Compiler #2 Scanner returns the token IDENTIFIER. Compiler#2 language allows many identifiers to be identified by one type separated by comma (,)

Comments in Compiler#2:

Compiler #2 includes two types of comments single line comments are prefixed by /- and multiple line comment are written between /# and #/.Your scanner must ignore all comments and white.

Using file command:

In order to facilitate the inclusion of multiple files, your Compiler#2 scanner is also responsible for directly handling the include file command. When encountering the using directive placing at the first column of a given line, the scanner must open the file indicated by the file name in the directive and start processing its contents. Once the used file has been processed the scanner must return to processing the original file. Used file may also use another file and so forth. If the file names does not exist in the local directory you should simply ignore the using command and proceed with the tokens in the current file.

Compiler#2 Output format:

Compiler#2 builds a dictionary to save lexemes that are defined in Compiler#1 language. Dictionary structure is according to the following:

Line NO	Lexeme	Return Token	Word NO in Line	Matchability
				Matched/Not Matched

Total NO of Errors:

Note: Matchability must be either Matched or Not Matched

Compiler#2 Language Delimiters (Lexemes and lines):

The lexemes are delimited by At Sign (@) and lines are delimited by Semicolon (;).

Sample Input and output:

Input:

1-/-This is main function

2- **Ire**@decrease(){

3- **Ire**@3num=5;

4- **RingWhen** (counter<num){

5-reg3=reg3-1;} }

Scanner Output:

Line NO	Lexeme	Return Token	Lexeme NO in Line	matchability
1	/-	Comment	1	Matched
2	Ire	Integer type	1	Matched
2	@	Token Delimiter	2	Matched
2	Decrease	Identifier	3	Matched

----------Etc.

Total NO of errors:1

Best Wishes Prof.Amal Aboutabl , Eng.Ahmed Badawy