

# 프로그래밍 언어론 Assignment

20215229 이호진

## Internal

### class Lexical

#### Variable

과제 문서에서 요구한 `next_token`, `token_string`, 함수 `lexical()`은 각각 `TokenCodes` `nextToken`, `std::string` `lexeme`, `lex()`로 대체했다.

- `std::ifstream` `inputFile`
  - Input stream이다.
- `std::map<std::string, MyInt>` `identifiers`
  - `MyInt`는 `Int`를 보관함과 동시에 정상적인 값인지를 식별한다.
  - 식별자들의 값을 저장하는 `map`이다
- `std::string` `currentLine`
  - parsing 중인 `line`을 저장하는 `string`이다.
- `TokenInformationNode*` `tree`
  - `parse tree`의 최상단 노드이다. `driver.cpp`에서 `printTree`를 호출하면 `parsing`된 `Tree`를 시각적으로 확인 할 수 있다.

#### Functions

- `void readInFile(std::string fileName)`
  - parsing할 input파일을 향한 `stream`을 열어주는 함수
- `void addChar(CharType nc)`
  - `CharType`은 `char`변수와 `CharClass` `enum`으로 구성되어있다.
  - `char`변수에 해당되는 `character`를 `lexeme`의 끝에 추가한다.
- `CharType` `getChar()`
  - `CharType`의 종류를 결정하고(`LETTER`,`DIGIT`,`UNKWOWN`) `stream`으로 부터 `char`변수에 `character`를 받아온다
- `CharType` `getNonBlank()`
  - `getChar`의 `return`의 `character`가 `blank`가 아닐때까지 반복해서 받아온후 `blank`가 아니라면 `return`한다.
- `TokenCodes` `lex()`
  - `lexeme(string)`을 구성하고 `nextToken`을 결정한다. 또한 `ID`나 `CONST`임이 식별되면 각각의 `count`를 증가시킨다.
- `TokenCodes` `lookup(char nextChar)`
  - `lex()`안에서 사용되는 함수이다. `nextChar`가 `Unknown`이라면 어떤 `TokenCode`인지 식별해서 `return`한다.
- `CharClass` `classifyChar(char nextChar)`
  - `nextChar`의 타입을 결정한다.(`LETTER`,`DIGIT`,`UNKWON`)

## Analysis functions

program(), statements(TokenInformationNode\* node), statement(TokenInformationNode\* node), expression(TokenInformationNode\* node), termTail(TokenInformationNode\* node, bool& add), term(TokenInformationNode\* node), factorTail(TokenInformationNode\* node, bool& mult), factor(TokenInformationNode\* node)는 각각 문법에 따른 parsing을 한다.

## External

### 컴파일 및 실행 방법

MinGW의 G++ Compiler를 이용했다.

1. 소스코드들이 있는 파일에서 cmd를 켜다.
2. g++ -o 실행파일이름 driver.cpp Lexical.cpp 를 입력한다.(생성된 실행파일이름을 a.exe라고 가정한다.)
3. 소스코드들이 있는 파일에 입력파일을 포함한다.(input.txt라 가정하겠다.)
4. a input.txt를 입력한다

### 오류 처리 목록

1. MULT\_OPERATION 또는 ADD\_OPERATION 뒤에 ADD\_OPERATION, MULT\_OPERATION, ASSIGNMENT\_OPERATION이 오면 WARNING을 띄우고 생략하도록 구성하였다.
  - a. 에러메세지 : (Warning) 중복연산자 제거
2. 선언되지 않은 식별자가 참조되면 ERROR를 출력하도록 하였다.
  - a. 에러메세지 : (Error) 정의되지 않은 식별자 [식별자이름] 참조됨.