

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
 - MyInt는 Int를 보관함과 동시에 정상적인 값인지지를 식별한다.
 - 식별자들의 값을 저장하는 map이다
- std::string currentLine
 - parsing중인 line을 저장하는 string이다.
- TokenInformationNode* tree
 - parse tree의 최상단 노드이다. driver.cpp에서 printTree를 호출하면 parsing된 Tree를 시각적으로 확인 할 수 있다.

Functions

- void readInFile(std::string fileName)
 - parsing할 input파일을 향한 stream을 열어주는 함수
- void addChar(CharType nc)
 - CharType은 char변수와 CharClass enum으로 구성되어 있다.
 - char변수에 해당되는 character를 lexeme의 끝에 추가한다.
- CharType getChar()
 - CharType의 종류를 결정하고(LETTER,DIGIT,UNKNOWN) stream으로 부터 char변수에 character를 받아온다
- CharType getNonBlank()
 - getChar의 return의 character가 blank가 아닐때까지 반복해서 받아온후 blank가 아니라면 return한다.
- TokenCodes lex()
 - lexeme(string)을 구성하고 nextToken을 결정한다. 또한 ID나 CONST임이 식별되면 각각의 count를 증가시킨다.
- TokenCodes lookup(char nextChar)
 - lex()안에서 사용되는 함수이다. nextChar가 Unknown이라면 어떤 TokenCode인지 식별해서 return한다.
- CharClass classifyChar(char nextChar)
 - nextChar의 타입을 결정한다.(LETTER,DIGIT,UNKNOWN)

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 켠다.
2. g++ -o 실행파일이름 driver.cpp Lexical.cpp 를 입력한다.(생성된 실행파일 이름을 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.