

# **프로그래밍언어**

## **[과제2 TOY Language (version1)]**

과목명 : 프로그래밍언어

교수명 : 양승민

학과 : 컴퓨터학부

학번 : 20160548

이름 : 이승현

## ● 구현 자료구조 및 함수 설명

### ➤ Main.cpp

```
void run(); // TOY언어 인터프리터를 실행하는 함수

void printMenu(); // 메뉴를 출력하는 함수

int inputNumber(); // 숫자를 입력받는 함수

char *getFileName(); // 파일명을 입력받는 함수

void printFileContents(FILE* fp); // 파일의 내용을 출력하는 함수

char* getTerm(); // 명령어를 입력받는 함수

char* loadTermsFromFile(); // 파일에서 명령어를 읽어오는 함수

bool executeFromFile(FILE* fp); // 파일의 명령어들을 실행시키는 함수

void executeInterpreter(); // 인터프리터를 실행시키는 함수


struct termNode { // 명령어를 트리로 변환해서 저장한다. 그 트리의 노드 구조체

    const char* type; // 함수, 변수, 정수 세가지 타입이 있다

    termNode* param1; // 함수인 경우 첫번째 피연산자를 여기에 저장한다

    termNode* param2; // 함수인 경우 두번째 피연산자를 여기에 저장한다

    int value; // 정수인 경우 그 값을 여기에 저장

    termNode() :type(NULL), param1(NULL), param2(NULL), value(0) {} // 생성자


    void print() { // 디버깅을 위해 노드의 내용을 출력해보고 싶을 때 사용

        cout << "type : " << type << endl;

        cout << "param1 : " << param1 << endl;

        cout << "param2 : " << param2 << endl;

        if (!strcmp(INTEGER, type))
```

```

        cout << "value : " << value << endl;

    }

};

```

### ➤ **Compiler.cpp**

bool compile(char\* terms); // 명령어를 컴파일하는 함수. 컴파일이란 명령어를 후위연산식으로 변환하고, 변환된 명령어를 텍스트 형식으로 파일에 출력해 중간 코드를 생성하는 것을 의미.

bool isWrongCharacter(char c); // 해당 문자가 TOY 언어에서 사용되면 안되는 문자인지 검사하여 사용하면 안되는 문자이면 true 리턴.

char\* removeLeftWhitespace(char\* terms); // 명령어 앞쪽의 공백을 제거하는 함수. 공백이 아닌 첫 번째 문자의 주소를 리턴한다.

void removeRightWhitespace(char\* terms); // 명령어 뒤쪽의 공백을 제거하는 함수.

termNode\* termToNode(char\*\* terms); // 명령어를 트리로 변환하는 함수 재귀호출 함수.

bool checkTermsIncludingWrongCharacter(char\* terms, char \*wrongChar);

bool isInteger(char\* term); // 전달받은 문자열이 정수 형식인지 확인하는 함수. 정수이면 true 리턴

void printAllTerms(termNode\* node, FILE\* fp); // 트리의 내용을 후위 연산식 형식으로 파일에 출력하는 함수

bool saveIntermediateCode(termNode\* node); // 중간코드를 저장하는 함수

void freeAllNode(termNode\* node); // 전달인자를 head로 하는 트리의 모든 노드를 free하는 함수

bool checkInvalidNumber(char\* term); // 숫자가 입력되어야 할 자리에 "-"가 여러 번 연속으로 입력된 경우를 확인하는 함수. 숫자가 입력되어야 할 자리에 "-"가 여러 번 연속으로 입력되었으면 true 리턴

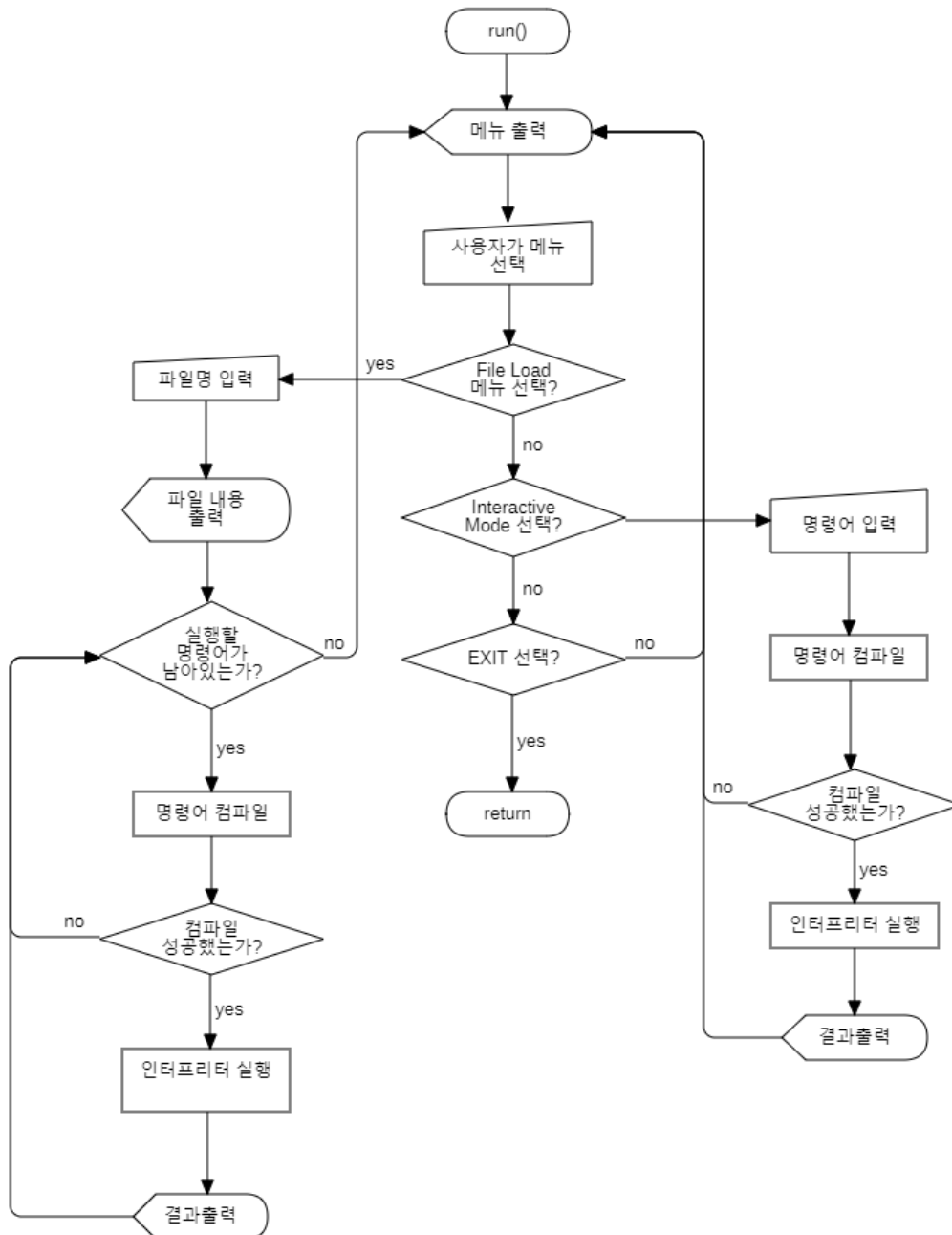
bool checkParenthesis(char\* terms); // 괄호의 짝이 맞는지 확인하는 함수 괄호의 짝이 맞으면 true 리턴

### ➤ **Interpreter.cpp**

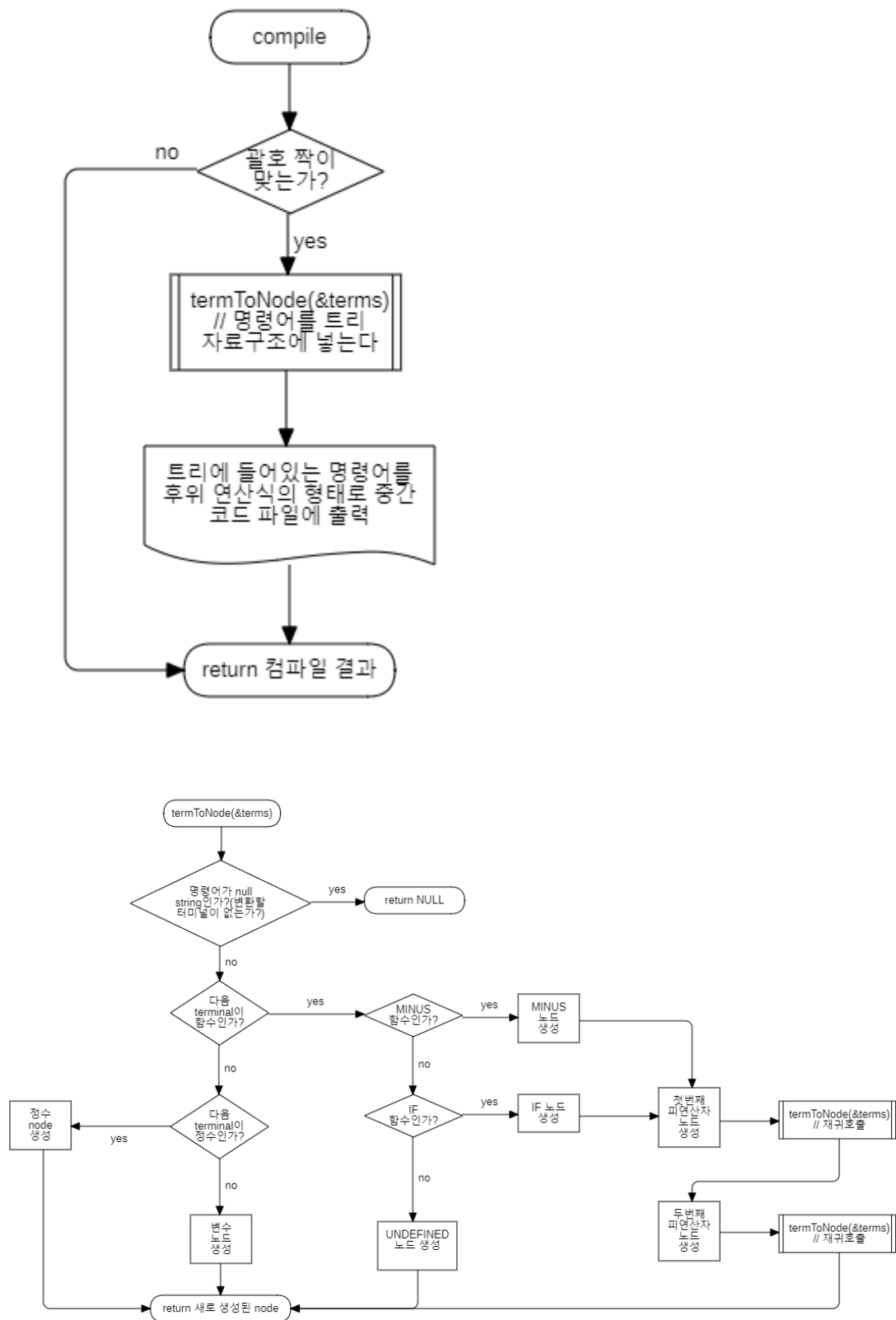
void interpreter(char\* terms); // 후위 연산식으로 변환된 명령어를 인자로 받아 실행하는 함수

- 흐름도

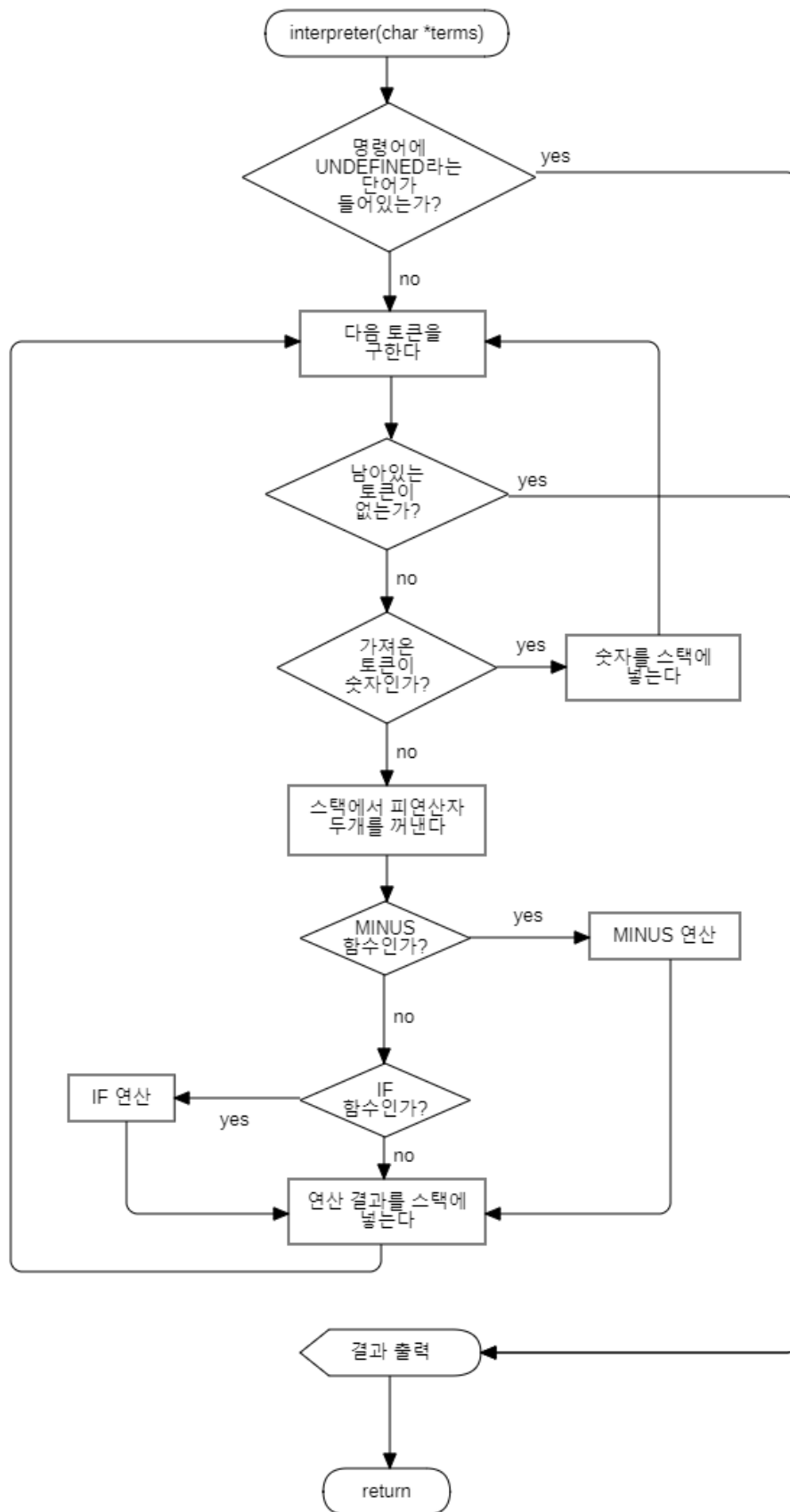
- Main.cpp



➤ Compiler.cpp



➤ Interpreter.cpp



## ● 실행 결과

```
=====
1. File Load
2. Interactive Mode
3. Exit
=====
```

메뉴를 선택하세요 >> 1

파일명을 입력하세요 >> wrongfilename

파일이 존재하지 않습니다.

파일명을 입력하세요 >> 1

파일이 존재하지 않습니다.

파일명을 입력하세요 >> test.txt

파일 내용은

```
=====
(MINUS 2 1)
(IF 1 4)
(MINUS (MINUS 2 (IF 1 4)) 1)
(MINUS (MINUS 2 1) 1)
(MINUS 2 (MINUS 2 1))
(MINUS 2 (MINUS ---2 1))
(MINUS (MINUS 2 (IF (IF 3 3) (IF 55 (MINUS 66 66)))) 1)
  (IF 1 4)
  (IF 1 4)
  (IF 1 4))
  (IF 1 4(
  (IF 1 4
  IF 1 4
  fdf232
  -----abf
  (NONE 1 4)
  13123.323
  #@#@#
=====
```

입니다.

```
결과 : 1
결과 : 4
결과 : -3
결과 : 0
결과 : 1
숫자가 입력되어야 할 자리에 "-"가 여러번 연속으로 입력되었습니다 - 컴파일 에러
결과 : 1
결과 : 4
결과 : 4
" ) "의 위치가 잘못되었습니다. - 컴파일 에러
" ( "의 위치가 잘못되었습니다. - 컴파일 에러
" " "의 위치가 잘못되었습니다. - 컴파일 에러
결과 : undefined
결과 : undefined
결과 : undefined
결과 : undefined
" " " 사용할 수 없는 문자입니다. - 컴파일 에러
" # " 사용할 수 없는 문자입니다. - 컴파일 에러
=====
```

```
=====
1. File Load
2. Interactive Mode
3. Exit
=====
```

메뉴를 선택하세요 >> 2

문장을 입력하세요.

(MINUS 2 1)

결과 : 1

⇒ 파일 로드 기능

⇒ 입력한 파일명이 폴더 내에 존재하지 않는 경우 에러 메시지를 출력 후 다시 입력 받는다.

⇒ 파일 내용 출력

⇒ 파일 내에 문장이 여러 개 있는 경우 각각의 결과 값을 모두 출력시켜준다

⇒ 문법에 맞지 않는 문장이 입력되면 어디가 잘못되었는지 체크하여 출력하도록 구현

⇒ 프로그램 작성 시 직접 코드를 작성하고 엔터를 누르면 인터프리터를 실행시킨다.

```
=====
1. File Load
2. Interactive Mode
3. Exit
=====
```

메뉴를 선택하세요 >> 2

문장을 입력하세요.

(MINUS --1 2)

숫자가 입력되어야 할 자리에 "-"가 여러번 연속으로 입력되었습니다 - 컴파일 에러

```
=====
1. File Load
2. Interactive Mode
3. Exit
=====
```

메뉴를 선택하세요 >> 2

문장을 입력하세요.

(MINUS ---4 1)

숫자가 입력되어야 할 자리에 "-"가 여러번 연속으로 입력되었습니다 - 컴파일 에러

```
=====
1. File Load
2. Interactive Mode
3. Exit
=====
```

메뉴를 선택하세요 >> 2

문장을 입력하세요.

(MINUS (IF 1 3) (MINUS (MINUS -5 1) 1))

결과 : 10

```
=====
1. File Load
2. Interactive Mode
3. Exit
=====
```

메뉴를 선택하세요 >> 3

프로그램을 종료합니다.

C:\Users\ddgh04\source\repos\PL hw2\Debug\PL hw2.exe(19268 프로세스)이(가) 0 코드로 인해 종료되었습니다.  
이 창을 닫으려면 아무 키나 누르세요.

⇒ 숫자가 입력되어야 할 자리에  
“-”가 여러 번 연속으로 입력된  
경우 에러 메시지 출력

⇒ 복잡한 문장의 해석이 가능하  
도록 구현

⇒ 프로그램 종료를 선택 시 프  
로그램을 종료한다