**LL(1) Parser Report**

|  |  |
| --- | --- |
|  |  |
| 과 목 | 프로그래밍 언어론  Principles of Programming Languages Autumn, 2023 |
| 담당 교수 | 김진성 |
| 학부 | 소프트웨어학부 |
| 팀 | 20202203 박호근  20204946 이규성 |

Table of Contents

[-Internal Documents 3](#_Toc150290494)

[1. utils.h 3](#_Toc150290495)

[2. utils.cpp 3](#_Toc150290496)

[3. treenode.h 4](#_Toc150290497)

[4. treenode.cpp 5](#_Toc150290498)

[5. parser.h 5](#_Toc150290499)

[6. parser.cpp 6](#_Toc150290500)

[7. main.cpp 7](#_Toc150290501)

[8. Error check 8](#_Toc150290502)

[-External Documents 11](#_Toc150290503)

[1. Build 11](#_Toc150290504)

[2. Run 11](#_Toc150290505)

[3. Test environment 11](#_Toc150290506)

[4. OS 11](#_Toc150290507)

[5. Compiler 11](#_Toc150290508)

# -Internal Documents

## 1. utils.h

텍스트, 스크린샷, 폰트, 번호이(가) 표시된 사진

자동 생성된 설명

Class Token, Lexer, SymbolTable, 내부 method와 attribute 선언.

## 2. utils.cpp

텍스트, 스크린샷, 디스플레이, 소프트웨어이(가) 표시된 사진

자동 생성된 설명

Utils.h에서 선언한 class들의 선언자와 내부 method body 작성.

Lexer::getNextToken : 현재 position의 character값을 받아 const, ident, op로 분류하고 분류한 token과 lexeme을 반환.

SymbolTable::set : symbol의 name과 value을 받아 symbols array에 저장.

SymbolTable::get : symbol의 name과 value의 주소를 받아 symbols array에 존재하는지 확인 후 존재한다면 value의 주소에 symbols array에 저장된 value값을 저장하고 true를 반환. 존재하지 않는다면 false를 반환.

SymbolTable::exists : symbol의 name을 받아 symbols array에 존재한다면 true를 반환.

SymbolTable::print\_result : parse 결과값을 출력.

## 3. treenode.h

텍스트, 스크린샷, 소프트웨어, 폰트이(가) 표시된 사진

자동 생성된 설명

LL(1)Parser의 문법에 존재하는 nonterminal을 class로 선언, node형식으로 정의하여 parse tree의 생성에 사용.

## 4. treenode.cpp

텍스트, 스크린샷, 소프트웨어이(가) 표시된 사진

자동 생성된 설명

treenode.h에서 선언한 class들의 생성자들과 결과값을 구하기 위해 parse tree의 해당 노드에서 해야하는 연산을 calculate함수로 정의, get\_op 함수로 nonterminal <add\_op>와 <mult\_op>를 결정하여 calculate 함수에 반환.

## 5. parser.h

텍스트, 스크린샷, 폰트, 번호이(가) 표시된 사진

자동 생성된 설명

class Parser 선언.

## 6. parser.cpp

텍스트, 스크린샷, 폰트, 소프트웨어이(가) 표시된 사진

자동 생성된 설명

해당 노드의 RHS 호출과 Lexer 에서 반환된 token과 lexeme으로 LHS 생성하여 반환.

## 7. main.cpp

텍스트, 스크린샷, 디스플레이, 소프트웨어이(가) 표시된 사진

자동 생성된 설명

main에서 file을 읽어 file 내부의 string을 읽어 input string변수에 저장해 parser class instance 생성. ProgramNode 포인터 rootNode를 생성해 parser.Program()으로 start symbol로 input data parsing. rootNode->calculate\_statements();로 parsing된 결과를 연산하여 symbolTable에 저장.

symbolTable.print\_result()로 input data 연산한 결과 출력.

## 8. Error check

1. 선언되지 않은 ident 사용

텍스트, 폰트, 스크린샷, 블랙이(가) 표시된 사진

자동 생성된 설명

1. 연산자 중복

텍스트, 폰트, 스크린샷이(가) 표시된 사진

자동 생성된 설명

1. 괄호 쌍 체크

텍스트, 스크린샷, 폰트, 디자인이(가) 표시된 사진

자동 생성된 설명

텍스트, 스크린샷, 폰트, 블랙이(가) 표시된 사진

자동 생성된 설명

텍스트, 스크린샷, 폰트, 블랙이(가) 표시된 사진

자동 생성된 설명

텍스트, 스크린샷, 폰트, 디자인이(가) 표시된 사진

자동 생성된 설명

1. 두 op, ident 사이 연산자 생략

텍스트, 폰트, 스크린샷, 블랙이(가) 표시된 사진

자동 생성된 설명

1. RHS 생략

텍스트, 폰트, 스크린샷, 블랙이(가) 표시된 사진

자동 생성된 설명

1. File.txt의 마지막 줄에 ; 존재

텍스트, 폰트, 스크린샷, 타이포그래피이(가) 표시된 사진

자동 생성된 설명

# -External Documents

## 1. Build

$ make

## 2. Run

$> make test

## 3. Test environment

- macOS 14.0 arm64

## 4. OS

- macOS Sonoma 14.0

## 5. Compiler

- Apple clang version 14.0.3 (clang-1403.0.22.14.1)

- Target: arm64-apple-darwin23.0.0