|  |  |
| --- | --- |
| 이름 | 신동욱 |
| 학번 | 20204101 |
| 학년 | 3 |
| 과목 | 프로그래밍언어론 |



프로그래밍언어론

보고서#4

1. 재귀 하강 파서 소개
2. 프로그램의 BNF
3. 코드 소개
   1. lex()
   2. expr()
   3. term()
   4. factor()
   5. atom()
   6. GUI
4. 실행 결과 및 테스트
5. **재귀 하강 파서 소개**

재귀 하강 파서(Recursive Descent)란, 주어진 문법을 기반으로 입력 수식을 분석하고 구문 트리를 생성하는 알고리즘, 즉 하향식 구문 분석이다. 이 알고리즘은 재귀적으로 작동하며 수식이 더 이상 처리될 수 없는 기본 단위까지 재귀적인 절차를 가지며 분석된다.

일반적으로 BNF를 기술하는 방식과 이 알고리즘의 진행 방식은 매우 유사하며, 따라서 이러한 알고리즘을 통해 계산기를 구현하는 데에 유용하다.

1. **프로그램의 BNF**

ㅇㅇㅇ

<expression> ::= <term> { ("+" | "-") <term> }

<term> ::= <factor> { ("\*" | "÷") <factor> }

<factor> ::= <atom> | "(" <expression> ")"

<atom> ::= <number>

<number> ::= (digit {digit} ) | (digit {digit} "." digit {digit})

<digit> ::= "0" | "1" | "2" | "3" | "4" | "5" | "6" | "7" | "8" | "9"

1. **코드 소개**
2. lex()

이 함수는 주어진 수식 문자열에 대해 토큰으로 분할하는 역할을 한다. 현재 위치에서 다음 토큰을 결정하며, 각 토큰은 NUMBER, DELIMITER 로 구분된다.

1. **실행 결과 및 테스트**

ㅇㅇ