PAIR PROGRAMMING TRAINING

Pray2U : ash-hun x go-ring

INDEX

Category of Calculate Expression

1. 전위(prefix) 표기법

> 연산자를 먼저표시하고 피연산자를 나중에 표시
> ex) *+AB-CD

2. 중위(infix) 표기법

> 피연산자 사이에 연산자표기 주로 우리가 사용하는 표기법이다.
> ex) (A+B) * (C-D)

3. 후위(prefix) 표기법

> 피연산자를 먼저 표신한뒤 연산자를 나중에 표시
> ex) A B + C D - *

INFIX EXPRESSION

Infix Expression

2. 중위(infix) 표기법

> 피연산자 사이에 연산자표기 주로 우리가 사용하는 표기법이다.
>
> ex) (A+B) * (C-D)

PREFIX EXPRESSION

Prefix Expression

1. 전위(prefix) 표기법

> 연산자를 먼저표시하고 피연산자를 나중에 표시
> ex) *+AB-CD

Infix to Prefix

1. 먼저 연산자의 우선순위를 지정한다
   *,/가 +,- 보다 우선순위가 높다.
   코드를 짤 때는 아래와 같이 미리 우선순위를 설정해놓는것이 편하다.

```
priority = {'*':3,'/':3,'+':2,'-':2,'(':1}
```

2. 하위 표현식으로 만들어질 빈 리스트를 생성
3. 중위표현식을 왼쪽부터 순서대로 읽는다
4. 피연산자이면 2번에서 만들어 놓은 빈 리스트(lst)에 추가한다.
5. '('이면 stack에 추가하고, ')'이면 stack에서 '('가 나올때까지 스택을 pop()처리 해준 이후 pop()처리 한것들을 2번에서 생성해놓은 list에 추가한다.
6. 선택된게 연산자이면 stack의 위에서 선택된 연산자보다 같거나 높은 우선순위의 연산자들을 pop과 동시에 lst에 추가 해준다. 더 이상 자기보다 같거나 높은 우선순위를 가진 연산자가 없으면 stack에 추가 해준다.
7. 만약 선택된 연산자가 stack 젤 위에 있는 연산자 보다 우선순위가 높다면 다른 처리 없이 바로 stack에 추가해준다.
8. stack에 남아있는 연산자 모두 pop을 하여 lst에 추가 시켜준다.

POSTFIX EXPRESSION

Postfix Expression

3. 후위(prefix) 표기법

> 피연산자를 먼저 표신한뒤 연산자를 나중에 표시
> ex) A B + C D - *

Infix to Postfix

1. infix를 순서대로 읽는다.
2. 읽은 값이 여는 괄호 '('이면, 스택에 넣는다.
3. 읽은 값이 닫는 괄호 ')'이면, 스택에서 여는 괄호가 나올 때까지 pop을 수행하여 postfix에 출력한다.
4. 읽은 값이 숫자이면, postfix에 출력한다.
5. 읽은 값이 연산자이면, 스택에서 자기보다 낮은 연산자를 꺼내 postfix에 출력한다.
6. 마지막으로 스택에 남아있는 것을 순서대로 pop하여 postfix에 출력한다.