

2014 자료구조 실습과제 8

[실습 8] 스택의 응용 : 계산기 프로그램

스택을 이용하여 계산기 프로그램 구현하시오.

- 중위표기법으로 입력하면 먼저 이를 후위표기법으로 바꾸고, 이를 후위표기법 계산 방법으로 계산하시오.
- 단, 한자리가 아닌 큰 수를 사용할 수 있도록 해야 한다.
- 후위표기법 변환 : 교재 205~207쪽의 프로그램 5.7 참조
- 후위표기법 계산 : 교재 199~200쪽의 프로그램 5.6 참조
- 입력 예: $(1 + 2 * 3 / 4) * 5 / 6 + 8$
- 다음은 참고 코드이며, 독창적으로 작성하여도 됨.
- 첨부에 포함된 실행파일 참고

<main() 함수>

```
#include "Calculator.h"

void main()
{
    CCalculator cal;

    cal.Read( );
    cal.PrintInfix();

    cal.Infix2Postfix( );
    cal.PrintPostfix();

    double result = cal.Calculate();

    printf( "결과값은 ==> %f\n\n", result);
    getchar();
}
```

<CCalcObj 클래스> - 하나의 연산자 또는 피연산자를 저장하기 위한 클래스

```
#include <stdio.h>
#define T_BRACKET 0
#define T_OPERATOR 1
#define T_NUMBER 2

class CCalcObj
{
public:
    int m_type;
    char m_op;
    double m_number;

    CCalcObj(void) { }
    CCalcObj( double val ) { Set(val); }
    ~CCalcObj(void){ }

    void Set( char ch ) {
        m_op = ch;
        if( isBracket(ch) ) m_type = T_BRACKET;
        else if( isOperator( ch ) ) m_type = T_OPERATOR;
    }
    void Set( double val ) {
        m_type = T_NUMBER;
        m_number = val;
    }

    int Precedence() { return precedence( m_op ); }
    bool IsBracket() { return m_type == T_BRACKET; }
    bool IsOperator() { return m_type == T_OPERATOR; }
    bool IsNumber() { return m_type == T_NUMBER; }
    void Print() {
        if( IsBracket() || IsOperator() )
            printf( "\t%c ", m_op );
        else
            printf( "\t%4.1f ", m_number );
    }
    friend bool isBracket( char ch );
    friend bool isOperator( char ch );
    friend bool isNumber( char ch );
    friend int precedence( char op );
};

inline bool isBracket( char ch ) {
    switch (ch) {
        case '(':
        case ')':
        case '{':
        case '}':
        case '[':
        case ']': return true;
    }
    return false;
}

inline bool isOperator( char ch ) {
    switch (ch) {
        case '+':
        case '-':
        case '/':
        case '*': return true;
    }
    return false;
}

inline bool isNumber( char ch ) {
    return (ch>='0' && ch<='9' );
}

inline int precedence( char op ) {
    switch (op) {
        case '(':
        case ')': return 0;
        case '+':
        case '-': return 1;
        case '*':
        case '/': return 2;
    }
    return -1;
}
```

<Stack 클래스> CCalcObj를 넣고 빼기 위한 스택 클래스

```
#pragma once
#include "CalcObj.h"

#define STACKSIZE 10000

class CCalcStack
{
public:
    CCalcObj    m_obj[STACKSIZE];
    int         m_top;

    CCalcStack(void)    { m_top = 0; }
    ~CCalcStack(void)   { }

    bool isEmpty()      { return m_top==0 ; }
    bool isFull()       { return m_top==STACKSIZE ; }

    void push( CCalcObj &pt ) { ... }
    CCalcObj* pop () { ... }
    CCalcObj* peek () { ... }
};
```

<계산기 클래스> 실제 계산기 클래스

```
#include "CalcStack.h"

class CCalculator
{
public:
    CCalcObj    m_infix[1000];
    int         m_inCount;
    CCalcObj    m_postfix[1000];
    int         m_postCount;

    CCalcStack  m_stack;

    CCalculator(void);
    ~CCalculator(void);

    void Read( FILE *fp = stdin ) {
        printf("수식 : ");
        char  ch;
        double val;
        m_inCount = 0;

        while ( (ch=getc(fp)) != '\n' ) {
            if( isBracket(ch) || isOperator(ch) ) {
                m_infix[m_inCount++].Set( ch );
            }
            else if (isNumber(ch)) {
                ungetc( ch, fp );
                fscanf( fp, "%lf", &val );
                m_infix[m_inCount++].Set( val );
            }
        }
    }

    void PrintInfix() { ... }
    void PrintPostfix() { ... }
    void Write();

    void Infix2Postfix( ) { ... }

    double calculate( ) { ... }
};
```