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

1. 소스 코드

윈도우즈프로그래밍  
과제#11 계산기

namespace WinFormsApp1

{

public partial class Form1 : Form

{

Parser p = new Parser();

bool equalFlag = false;

public Form1()

{

InitializeComponent();

}

private void Form1\_Load(object sender, EventArgs e)

{

}

private void textBox1\_TextChanged(object sender, EventArgs e)

{

}

private void button10\_Click(object sender, EventArgs e)

{

if (equalFlag)

{

textBox1.Clear();

equalFlag = false;

}

textBox1.Text += "0";

}

private void button1\_Click(object sender, EventArgs e)

{

if (equalFlag)

{

textBox1.Clear();

equalFlag = false;

}

textBox1.Text += "1";

}

private void button2\_Click(object sender, EventArgs e)

{

if (equalFlag)

{

textBox1.Clear();

equalFlag = false;

}

textBox1.Text += "2";

}

private void button3\_Click(object sender, EventArgs e)

{

if (equalFlag)

{

textBox1.Clear();

equalFlag = false;

}

textBox1.Text += "3";

}

private void button4\_Click(object sender, EventArgs e)

{

if (equalFlag)

{

textBox1.Clear();

equalFlag = false;

}

textBox1.Text += "4";

}

private void button5\_Click(object sender, EventArgs e)

{

if (equalFlag)

{

textBox1.Clear();

equalFlag = false;

}

textBox1.Text += "5";

}

private void button6\_Click(object sender, EventArgs e)

{

if (equalFlag)

{

textBox1.Clear();

equalFlag = false;

}

textBox1.Text += "6";

}

private void button7\_Click(object sender, EventArgs e)

{

if (equalFlag)

{

textBox1.Clear();

equalFlag = false;

}

textBox1.Text += "7";

}

private void button8\_Click(object sender, EventArgs e)

{

if (equalFlag)

{

textBox1.Clear();

equalFlag = false;

}

textBox1.Text += "8";

}

private void button9\_Click(object sender, EventArgs e)

{

if (equalFlag)

{

textBox1.Clear();

equalFlag = false;

}

textBox1.Text += "9";

}

private void button19\_Click(object sender, EventArgs e)

{

textBox1.Clear();

textBox2.Clear();

}

private void button21\_Click(object sender, EventArgs e)

{

textBox2.Clear();

}

private void textBox2\_TextChanged(object sender, EventArgs e)

{

}

private void button22\_Click(object sender, EventArgs e)

{

if (textBox1.Text[0] != '-')

{

textBox1.Text = "-(" + textBox1.Text + ")";

}

else if (textBox1.Text.StartsWith("-(") && textBox1.Text.EndsWith(")"))

{

textBox1.Text = textBox1.Text.Substring(2, textBox1.Text.Length - 3);

}

}

//하나 지우기

private void button20\_Click(object sender, EventArgs e)

{

textBox1.Text = "" + textBox1.Text.Substring(0, textBox1.Text.Length - 1);

}

// =

private void button16\_Click(object sender, EventArgs e)

{

string exp = textBox1.Text;

double result = p.Evaluate(exp);

textBox2.Text = "" + result;

equalFlag = true;

}

// +

private void button12\_Click(object sender, EventArgs e)

{

textBox1.Text += "+";

}

// -

private void button15\_Click(object sender, EventArgs e)

{

textBox1.Text += "-";

}

//

private void button14\_Click(object sender, EventArgs e)

{

textBox1.Text += "\*";

}

// 나누기

private void button13\_Click(object sender, EventArgs e)

{

textBox1.Text += "/";

}

// 루트

private void button23\_Click(object sender, EventArgs e)

{

if (textBox2.Text != "")

{

textBox1.Text = "";

textBox2.Text = "" + Math.Sqrt(double.Parse(textBox2.Text));

}

}

// 제곱

private void button17\_Click(object sender, EventArgs e)

{

if (textBox2.Text != "")

{

textBox1.Text = "";

textBox2.Text = "" + Math.Pow(double.Parse(textBox2.Text), 2);

}

}

// 1/x

private void button18\_Click(object sender, EventArgs e)

{

if (textBox2.Text != "")

{

textBox1.Text = "";

textBox2.Text = "" + (1 / double.Parse(textBox2.Text));

}

}

// sin

private void button28\_Click(object sender, EventArgs e)

{

if (textBox2.Text != "")

{

textBox1.Text = "";

double radian = double.Parse(textBox2.Text) \* (Math.PI / 180);

textBox2.Text = "" + Math.Sin(radian);

}

}

private void button24\_Click(object sender, EventArgs e)

{

if (textBox2.Text != "")

{

textBox1.Text = "";

double radian = double.Parse(textBox2.Text) \* (Math.PI / 180);

textBox2.Text = "" + Math.Cos(radian);

}

}

private void button25\_Click(object sender, EventArgs e)

{

if (textBox2.Text != "")

{

textBox1.Text = "";

double radian = double.Parse(textBox2.Text) \* (Math.PI / 180);

textBox2.Text = "" + Math.Tan(radian);

}

}

private void button26\_Click(object sender, EventArgs e)

{

long fact(long n)

{

long result = 1;

for (int i = 1; i <= n; i++)

{

result \*= i;

}

return result;

}

if (textBox2.Text != "")

{

textBox1.Text = "";

long temp = long.Parse(textBox2.Text);

textBox2.Text = "" + fact(temp);

}

}

private void button27\_Click(object sender, EventArgs e)

{

if (textBox2.Text != "")

{

textBox1.Text = "";

double temp = double.Parse(textBox2.Text);

textBox2.Text = "" + Math.Log10(temp);

}

}

}

class Parser

{

private const int NONE = 0;

private const int OPERATOR = 1;

private const int NUMBER = 2;

private const string EOE = "\0";

private string exp;

private int expIdx;

private string token;

private int tokType;

public double Evaluate(string expstr)

{

double result;

exp = expstr;

expIdx = 0;

Lex();

if (token == EOE)

HandleError();

result = Expr();

if (token != EOE)

HandleError();

return result;

}

private double Expr()

{

char op;

double result;

double nextResult;

result = Term();

while ((op = token[0]) == '+' || op == '-')

{

Lex();

nextResult = Term();

switch (op)

{

case '+':

result += nextResult;

break;

case '-':

result -= nextResult;

break;

}

}

return result;

}

private double Term()

{

char op;

double result;

double nextResult;

result = Factor();

while ((op = token[0]) == '\*' || op == '/')

{

Lex();

nextResult = Factor();

switch (op)

{

case '\*':

result \*= nextResult;

break;

case '/':

if (nextResult == 0.0)

HandleError();

result /= nextResult;

break;

}

}

return result;

}

private double Factor()

{

double result;

char unaryOp = '\0';

if (token == "+" || token == "-")

{

unaryOp = token[0];

Lex();

}

if (token == "(")

{

Lex();

result = Expr();

if (token != ")")

HandleError();

Lex();

}

else

{

result = Number();

}

if (unaryOp == '-')

{

result = -result;

}

return result;

}

private double Number()

{

double result = 0.0;

if (tokType == NUMBER)

{

result = double.Parse(token);

Lex();

}

else

{

HandleError();

}

return result;

}

private void HandleError()

{

throw new Exception("Error");

}

private void Lex()

{

tokType = NONE;

token = "";

if (expIdx == exp.Length)

{

token = EOE;

return;

}

if (IsOperator(exp[expIdx]))

{

token += exp[expIdx];

expIdx++;

tokType = OPERATOR;

}

else if (char.IsDigit(exp[expIdx]) || exp[expIdx] == '.')

{

while (expIdx < exp.Length && (char.IsDigit(exp[expIdx]) || exp[expIdx] == '.'))

{

token += exp[expIdx];

expIdx++;

}

tokType = NUMBER;

}

else

{

HandleError();

}

}

private bool IsOperator(char c)

{

return "+-\*%^=()/".IndexOf(c) != -1;

}

}

}

프로그래밍 언어론 수업에서 학습한 재귀 하강 파서를 이용하여 수식에 대한 사칙연산을 구현함.

재귀 하강 파서란 각 수식에 대하여 재귀적으로 수식을 하나씩 분석해가는 파싱 기법이다.

코드의 Parser는 +, -, \*, / 사칙연산과, (, ), ± 에 대해 처리할 수 있다.

해당 코드에는 Parser 클래스를 정의 후 WinFormsApp 의 필드로 Parser의 객체 필드를 가진다. 계산기의 ‘=’ 버튼을 누르면 Parser의 Evaluate 메서드를 호출한다.

2. 실행 결과

텍스트, 스크린샷, 직사각형, 사각형이(가) 표시된 사진

자동 생성된 설명

프로그램의 최초 실행 화면

전자제품, 텍스트, 스크린샷, 디스플레이이(가) 표시된 사진

자동 생성된 설명

순서대로 1, +, 6, -, 2 를 입력

전자제품, 텍스트, 스크린샷, 사각형이(가) 표시된 사진

자동 생성된 설명

= 버튼을 누르면 수식에 대해 계산 결과를 두번째 텍스트 필드에 출력한다

전자제품, 텍스트, 스크린샷, 디스플레이이(가) 표시된 사진

자동 생성된 설명

C 버튼을 누르면 모두 지운다

스크린샷, 라인, 디자인이(가) 표시된 사진

자동 생성된 설명전자제품, 텍스트, 스크린샷, 번호이(가) 표시된 사진

자동 생성된 설명

CE 버튼을 누르면 계산창을   
지운다

전자제품, 텍스트, 스크린샷, 디스플레이이(가) 표시된 사진

자동 생성된 설명

± 버튼을 클릭하면 단항 연산자가 적용된다

현재 수식에 대하여 - 와 괄호를 씌운다

전자제품, 텍스트, 스크린샷, 디스플레이이(가) 표시된 사진

자동 생성된 설명

한번 더 ± 를 클릭하면

-와 괄호가 사라진다

전자제품, 텍스트, 스크린샷, 사각형이(가) 표시된 사진

자동 생성된 설명

단항 연산자에 대해 계산 결과

스크린샷, 라인, 직사각형, 디자인이(가) 표시된 사진

자동 생성된 설명전자제품, 텍스트, 스크린샷, 디스플레이이(가) 표시된 사진

자동 생성된 설명

루트 버튼을 누르면 계산 결과창의 값의 루트 값을 결과창에 출력한다

텍스트, 전자제품, 스크린샷, 번호이(가) 표시된 사진

자동 생성된 설명

루트 버튼을 여러 번 눌러도 동작한다.

해당 값은 3에서 루트 두 번 누른 것.

스크린샷, 라인, 직사각형, 디자인이(가) 표시된 사진

자동 생성된 설명전자제품, 텍스트, 스크린샷, 디스플레이이(가) 표시된 사진

자동 생성된 설명

거듭제곱 버튼을 누르면 계산 결과창의 값의 제곱 값을 결과창에 출력한다

스크린샷, 라인, 직사각형, 디자인이(가) 표시된 사진

자동 생성된 설명전자제품, 텍스트, 스크린샷, 번호이(가) 표시된 사진

자동 생성된 설명

1/x 버튼을 누르면 계산 결과창의 값의 1/x 값을 결과창에 출력한다

스크린샷, 폰트, 디자인이(가) 표시된 사진

자동 생성된 설명전자제품, 텍스트, 스크린샷, 번호이(가) 표시된 사진

자동 생성된 설명

cos 버튼을 누르면 결과창의 값에 cos을 적용한다.

해당 값은 60에 대해 cos 버튼을 누른 것

sin 버튼을 누르면 결과창의 값에 sin을 적용한다.

해당 값은 30에 대해 sin 버튼을 누른 것

스크린샷, 라인, 직사각형, 디자인이(가) 표시된 사진

자동 생성된 설명텍스트, 전자제품, 스크린샷, 사각형이(가) 표시된 사진

자동 생성된 설명

텍스트, 스크린샷, 폰트, 라인이(가) 표시된 사진

자동 생성된 설명전자제품, 텍스트, 스크린샷, 번호이(가) 표시된 사진

자동 생성된 설명

tan 버튼을 누르면 결과창의 값에 tan을 적용한다.

해당 값은 45에 대해 tan 버튼을 누른 것

라인, 스크린샷, 직사각형, 도표이(가) 표시된 사진

자동 생성된 설명전자제품, 텍스트, 스크린샷, 디스플레이이(가) 표시된 사진

자동 생성된 설명

n! 버튼을 누르면 결과창의 값에 팩토리얼을 적용한다.

해당 값은 4에 대해 n! 버튼을 누른 것

스크린샷, 폰트, 텍스트, 번호이(가) 표시된 사진

자동 생성된 설명전자제품, 텍스트, 스크린샷, 디스플레이이(가) 표시된 사진

자동 생성된 설명

log 버튼을 누르면 결과창의 값에 로그(밑 10)를 적용한다.

해당 값은 100에 대해 log 버튼을 누른 것

전자제품, 텍스트, 스크린샷, 디스플레이이(가) 표시된 사진

자동 생성된 설명

마지막으로 연산자 우선순위에 대해 정상적으로 잘 작동하는 것을 볼 수 있다