using System;

using System.Collections.Generic;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Input;

using System.Windows.Media;

using System.Numerics;

using System.Numerics.MPFR;

namespace Moon\_Calculator\_03

{

public partial class MainWindow : Window

{

private bool opFlag = false; // true:연산자버튼이 클릭시, false:아님

private bool enterFlag = false; // true:엔터버튼이 클릭시, false:아님

private bool HexFlag = false; // true:엔터버튼이 클릭시, false:아님

private bool opInput = false; // true:엔터버튼이 클릭시, false:아님

private string op = "", op2;

string txtExp1 = "", txtExp2 = "";

BigFloat x1 = new BigFloat("0", precision: 137); // prec 133 = 40digits \* 3.3219280948873623478; 177 = 133 + 4(여유분)

BigFloat x2 = new BigFloat("0", precision: 137); // prec 133 = 40digits \* 3.3219280948873623478;

BigFloat x3 = new BigFloat("0", precision: 137); // prec 133 = 40digits \* 3.3219280948873623478;

BigFloat memory = new BigFloat("0", precision: 137); // Mc 클릭시 txtResult 값 저장, Mv 클릭시 paste \*/

private void btnEntryClear\_Click(object sender, RoutedEventArgs e)

{

txtResult.Text = "0";

txtExp.Text = "";

op = "";

opFlag = false;

}

private void btnClear\_Click(object sender, RoutedEventArgs e)

{

txtResult.Text = "0";

enterFlag = true;

}

public MainWindow() // 시작시 초기화

{

InitializeComponent();

txtExp.Foreground = Brushes.Red;

History.Foreground = Brushes.BlueViolet;

op = "+";

}

private void Exit\_Click(object sender, RoutedEventArgs e)

{

this.Close();

}

private void btnCopy\_Click(object sender, RoutedEventArgs e)

{

Button btn = sender as Button;

op2 = btn.Content.ToString();

string temp;

switch (op2)

{

case "Mc":

temp = txtResult.Text.Replace(",", "");

BigFloat.Set(memory, temp, 10);

btnMcopy.Background = Brushes.YellowGreen;

break;

case "Mp":

if (opFlag == false)

{

txtExp.Text = "";

x1 = memory;

txtExp1 = x1.ToString();

txtExp.Text += x1.ToString();

}

if (opFlag == true)

{

x2 = memory;

txtExp2 = x2.ToString();

txtExp.Text += x2.ToString();

}

btnMcopy.Background = Brushes.WhiteSmoke;

op\_cal(op);

break;

}

} // Mc-Copy, Mv-Paste

private void Button(string Button\_No) //Neumeric 입력 및 계산 update

{

if (enterFlag == true)

{

enterFlag = false;

txtExp1 = "";

txtExp2 = "";

opFlag = false;

}

if (enterFlag == false)

{

txtExp.Foreground = Brushes.Blue;

txtExp.Text = "";

}

if (opFlag == false)

{

txtExp1 += Button\_No;

BigFloat.Set(x1, txtExp1, 10);

}

if (opFlag == true )

{

txtExp2 += Button\_No;

BigFloat.Set(x2, txtExp2, 10);

}

if (opFlag == true) txtExp.Text = txtExp1 + txtExp2 + " ";

if (opFlag == false ) txtExp.Text = txtExp1 + " ";

op\_cal(op);

}

// 숫자 입력

private void btn\_Click(object sender, RoutedEventArgs e) //Neumeric 입력

{

Button btn = sender as Button;

opInput = false; // 연산자 입력 후 숫자버튼 누르면 false로 전환

string btn\_No;

btn\_No = btn.Content.ToString();

Button(btn\_No);

}

// 소수점 입력

private void Button\_dot()

{

if (opFlag == false)

{

if (txtExp1.Contains(".")) return;

txtExp1 += ".";

txtExp.Text = txtExp1 + " ";

}

if (opFlag == true)

{

if (txtExp2.Contains(".")) return;

txtExp2 += ".";

txtExp.Text = txtExp1 + " " + op + " " + txtExp2 + " ";

}

} // "." 입력

private void btndot\_Click(object sender, RoutedEventArgs e)

{

Button\_dot();

} // "." 입력

// Sign 처리

private void btnsign\_Click(object sender, RoutedEventArgs e)

{

if (opFlag == false)

{

if(txtExp1.Contains("-")) txtExp1 = txtExp1.Replace("-","");

else txtExp1 = "-" + txtExp1;

txtExp.Text = txtExp1;

BigFloat.Set(x1, txtExp1, 10);

}

if (opFlag == true)

{

if (txtExp2.Contains("-")) txtExp2 = txtExp2.Replace("-", "");

else txtExp2 = "-" + txtExp2;

txtExp.Text = txtExp1 + " " + op + " " + txtExp2;

BigFloat.Set(x2, txtExp2, 10);

}

op\_cal(op);

} // 부호 +, - 변경

private void Enter()

{

//if (enterFlag == true || opFlag == false) return;

if (enterFlag == true) return;

txtExp.Foreground = Brushes.Red;

enterFlag = true;

opFlag = false;

op = "+";

txtExp.Text += " = ";

History.Text = History.Text + txtExp.Text + txtResult.Text + " " + "\n";

BigFloat.Set(x1, "0", 10);

BigFloat.Set(x2, "0", 10);

txtExp1 = "";

txtExp2 = "";

} // 엔터키 입력시 수식 및 결과값 확정, History에 저장

private void btnEnter\_Click(object sender, RoutedEventArgs e)

{

Enter();

} // 엔터키 입력시 수식 및 결과값 확정, History에 저장

private void Operator(string op)

{

if (enterFlag == true && opFlag == false)

{

enterFlag = false;

opFlag = true;

txtExp1 = txtExp.Text;

txtExp2 = "";

BigFloat.Set(x1, txtResult.Text, 10);

}

opFlag = true;

} // 연산자 입력시

private void btnFun2\_Click(object sender, RoutedEventArgs e)

{

opFlag = true;

Button btn = sender as Button;

op = btn.Content.ToString();

if (opInput == true) // 연산자(+,-,\*,÷) 반복 입력시 지막 연산자로 대체

{

txtExp1 = txtExp1.Substring(0, txtExp1.Length - 3);

txtExp1 += " " + op + " ";

txtExp.Text = txtExp1;

return;

}

if (opFlag == false) txtExp1 = txtExp1 + " ";

if (opFlag == true)

{

if (op == "xⁿ") txtExp1 += "^" + txtExp2;

if (op == "ⁿ√") txtExp1 += "";

if (op != "ⁿ√" && op != "xⁿ")

{

txtExp1 += " " + txtExp2 + " " + op + " ";

txtExp2 = "";

}

}

txtExp.Text = txtExp1;

BigFloat.Set(x1, x3);

if (enterFlag == true)

{

enterFlag = false;

txtExp.Text = "";

}

Operator(op);

opInput = true; // 연산자 입력되면 True로 전환

} // 연산자 입력시

private void op\_cal(string op)

{

Int64 a1, a2, a3;

switch (op)

{

case "+":

BigFloat.Add(x3, x1, x2);

txtResult.Text = x3.ToString("d40");

break;

case "-":

BigFloat.Sub(x3, x1, x2);

txtResult.Text = x3.ToString("d40");

break;

case "\*":

BigFloat.Mul(x3, x1, x2);

txtResult.Text = x3.ToString("d40");

break;

case "÷":

BigFloat.Div(x3, x1, x2);

txtResult.Text = x3.ToString("d40");

break;

case "mod":

a1 = x1.ToInt64();

a2 = x2.ToInt64();

a3 = (int)a1 % (int)a2;

x3.Set(a3);

txtResult.Text = x3.ToString("d40");

break;

case "xⁿ":

BigFloat.Pow(x3, x1, x2);

txtResult.Text = x3.ToString("d40");

break;

case "ⁿ√":

ulong long\_temp;

long\_temp = ulong.Parse(txtExp1);

BigFloat.Root(x3, x2, long\_temp);

txtExp.Text = txtExp1 + " " + "yRoot" + " " + txtExp2;

txtResult.Text = x3.ToString("d40");

break;

}

txtResult.Text = Display(txtResult.Text);

} // 기본 연산자 수행

private void btnConst\_Click(object sender, RoutedEventArgs e)

{

Button btn = sender as Button;

op = btn.Content.ToString();

switch (op)

{

case "π":

x3.ConstPi();

txtResult.Text = x3.ToString("d40");

txtExp.Text = " π";

enterFlag = true;

break;

case "e":

x3 = new BigFloat("2.71828182845904523536028747135266249775724709369995", precision: 133); //

txtResult.Text = x3.ToString("d40");

txtExp.Text = " e";

break;

case "%":

txtExp.Text = txtExp1 + " \* " + txtExp2 + "%";

BigFloat.Div(x3, x3, (double)100);

txtResult.Text = x3.ToString("d40");

enterFlag = true;

break;

}

txtResult.Text = Display(txtResult.Text);

History.Text = History.Text + txtExp.Text + " = " + txtResult.Text + " " + "\n";

op = "+";

} // pi, e 등 상수 함수

private void btnFun1\_Click(object sender, RoutedEventArgs e)

{

Button btn = sender as Button;

op = btn.Content.ToString();

if (enterFlag == false) x1 = x3;

if (enterFlag == true) x3 = new BigFloat(txtResult.Text, precision: 137);

txtExp.Text = " " + op + "( ";

opFlag = true;

switch (op) // 삼각함수ian 입력값

{

case "sin":

BigFloat.Sin(x3, x1);

txtExp.Text = op + "(" + txtExp1 + ")";

if (enterFlag == true) txtExp1 = " " + op + "( ";

if (enterFlag == false) txtExp1 = op + "(" + txtExp1 + ")";

txtResult.Text = x3.ToString("d40");

break;

case "cos":

BigFloat.Cos(x3, x1);

txtExp.Text = op + "(" + txtExp1 + ")";

if (enterFlag == true) txtExp1 = " " + op + "( ";

if (enterFlag == false) txtExp1 = op + "(" + txtExp1 + ")";

txtResult.Text = x3.ToString("d40");

break;

case "tan":

BigFloat.Tan(x3, x1);

txtExp.Text = op + "(" + txtExp1 + ")";

if (enterFlag == true) txtExp1 = " " + op + "( ";

if (enterFlag == false) txtExp1 = op + "(" + txtExp1 + ")";

txtResult.Text = x3.ToString("d40");

break;

case "asin":

BigFloat.Asin(x3, x1);

txtExp.Text = op + "(" + txtExp1 + ")";

if (enterFlag == true) txtExp1 = " " + op + "( ";

if (enterFlag == false) txtExp1 = op + "(" + txtExp1 + ")";

txtResult.Text = x3.ToString("d40");

break;

case "acos":

BigFloat.Acos(x3, x1);

txtExp.Text = op + "(" + txtExp1 + ")";

if (enterFlag == true) txtExp1 = " " + op + "( ";

if (enterFlag == false) txtExp1 = op + "(" + txtExp1 + ")";

txtResult.Text = x3.ToString("d40");

break;

case "atan":

BigFloat.Atan(x3, x1);

txtExp.Text = op + "(" + txtExp1 + ")";

if (enterFlag == true) txtExp1 = " " + op + "( ";

if (enterFlag == false) txtExp1 = op + "(" + txtExp1 + ")";

txtResult.Text = x3.ToString("d40");

break;

case "sinh":

BigFloat.Sinh(x3, x1);

txtExp.Text = op + "(" + txtExp1 + ")";

if (enterFlag == true) txtExp1 = " " + op + "( ";

if (enterFlag == false) txtExp1 = op + "(" + txtExp1 + ")";

txtResult.Text = x3.ToString("d40");

break;

case "cosh":

BigFloat.Cosh(x3, x1);

txtExp.Text = op + "(" + txtExp1 + ")";

if (enterFlag == true) txtExp1 = " " + op + "( ";

if (enterFlag == false) txtExp1 = op + "(" + txtExp1 + ")";

txtResult.Text = x3.ToString("d40");

break;

case "tanh":

BigFloat.Tanh(x3, x1);

txtExp.Text = op + "(" + txtExp1 + ")";

if (enterFlag == true) txtExp1 = " " + op + "( ";

if (enterFlag == false) txtExp1 = op + "(" + txtExp1 + ")";

txtResult.Text = x3.ToString("d40");

break;

case "asinh":

BigFloat.Asinh(x3, x1);

txtExp.Text = op + "(" + txtExp1 + ")";

if (enterFlag == true) txtExp1 = " " + op + "( ";

if (enterFlag == false) txtExp1 = op + "(" + txtExp1 + ")";

txtResult.Text = x3.ToString("d40");

break;

case "acosh":

BigFloat.Acosh(x3, x1);

txtExp.Text = op + "(" + txtExp1 + ")";

if (enterFlag == true) txtExp1 = " " + op + "( ";

if (enterFlag == false) txtExp1 = op + "(" + txtExp1 + ")";

txtResult.Text = x3.ToString("d40");

break;

case "atanh":

BigFloat.Atanh(x3, x1);

txtExp.Text = op + "(" + txtExp1 + ")";

if (enterFlag == true) txtExp1 = " " + op + "( ";

if (enterFlag == false) txtExp1 = op + "(" + txtExp1 + ")";

txtResult.Text = x3.ToString("d40");

break;

case "√":

BigFloat.Sqrt(x3, x1);

txtExp.Text = op + "(" + txtExp1 + ")";

if (enterFlag == true) txtExp1 = " " + op + "( ";

if (enterFlag == false) txtExp1 = op + "(" + txtExp1 + ")";

txtResult.Text = x3.ToString("d40");

break;

case "x²":

BigFloat.Sqr(x3, x1);

txtExp.Text = "(" + txtExp1 + ")" + "²";

if (enterFlag == true) txtExp1 = " " + "²" + "( ";

if (enterFlag == false) txtExp1 = "(" + txtExp1 + ")" + "²";

txtResult.Text = x3.ToString("d40");

break;

case "2ⁿ":

BigFloat.Set(x2, 2);

BigFloat.Pow(x3, x2, x1);

txtExp.Text = "2^" + txtExp1;

txtResult.Text = x3.ToString("d40");

break;

case "10ⁿ":

BigFloat.Set(x2, 10);

BigFloat.Pow(x3, x2, x1);

txtExp.Text = "10^" + txtExp1;

txtResult.Text = x3.ToString("d40");

break;

case "n!":

ulong long\_temp;

long\_temp = ulong.Parse(txtExp1);

BigFloat.Fac(x3, long\_temp);

txtExp.Text = txtExp1 + "!";

txtResult.Text = x3.ToString("d40");

break;

case "log2":

BigFloat.Log2(x3, x1);

txtExp.Text = "log2(" + txtExp1 + ")";

txtResult.Text = x3.ToString("d40");

break;

case "log10":

BigFloat.Log10(x3, x1);

txtExp.Text = "log10(" + txtExp1 + ")";

txtResult.Text = x3.ToString("d40");

break;

case "eⁿ":

BigFloat.Set(x2, "2.71828182845904523536028747135266249775724709369995", 10);//

BigFloat.Pow(x3, x2, x1);

txtExp.Text = "e^" + txtExp1;

txtResult.Text = x3.ToString("d40");

break;

case "1/x":

BigFloat.Set(x3, "1.0", 10);

BigFloat.Set(x2, txtExp1, 10);

BigFloat.Div(x3, x3, x2);

txtExp.Text = "1/" + txtExp1;

txtResult.Text = x3.ToString("d40");

break;

}

txtResult.Text = Display(txtResult.Text);

} // 삼각함수 등

private void Number\_Delete()

{

if (opFlag == false)

{

txtExp1 = txtExp1.Substring(0, txtExp1.Length - 1);

BigFloat.Set(x1, txtExp1, 10);

txtExp.Text = txtExp1 + " ";

}

if (opFlag == true)

{

txtExp2 = txtExp2.Substring(0, txtExp2.Length - 1);

BigFloat.Set(x2, txtExp2, 10);

txtExp.Text = txtExp1 + " " + op + " " + txtExp2 + " ";

}

op\_cal(op);

} // Back space 기능

private void btnDelete\_Click(object sender, RoutedEventArgs e)

{

Number\_Delete();

}

private void Window\_KeyDown(object sender, KeyEventArgs e)

{

string number;

int key\_value;

key\_value = (int)e.Key;

if (key\_value > 73 && key\_value < 85)

{

number = (key\_value - 74).ToString();

Button(number);

return;

}

if (e.Key == Key.Decimal) { Button\_dot(); return; }

if (e.Key == Key.Delete) { Number\_Delete(); return; }

if (e.Key == Key.Add) { op = "+"; Operator(op); return; }

if (e.Key == Key.Subtract) { op = "-"; Operator(op); return; }

if (e.Key == Key.Multiply) { op = "\*"; Operator(op); return; } // 내 PC에서 미작동

if (e.Key == Key.Divide) { op = "÷"; Operator(op); return; }

if ((Keyboard.IsKeyDown(Key.LeftCtrl) && Keyboard.IsKeyDown(Key.V)))

{

if (opFlag == false)

{

txtExp1 = Clipboard.GetText();

BigFloat.Set(x1, txtExp1, 10);

txtExp.Text = txtExp1;

}

if (opFlag == true)

{

txtExp2 = Clipboard.GetText();

BigFloat.Set(x2, txtExp2, 10); // 정상 작동

txtExp.Text = txtExp1 + " " + op + " " + txtExp2;

}

op\_cal(op);

}

} // 숫자키,

private void btnHEX\_Click(object sender, RoutedEventArgs e)

{

if (HexFlag == false)

{

HexFlag = true;

HexResult.Foreground = Brushes.DarkOliveGreen;

HexResult.Text = " Hex-Decimal : " + HexDisplay(x3.ToString("b16")) + " ";

return;

}

if (HexFlag == true) HexFlag = false;

HexResult.Foreground = Brushes.IndianRed;

HexResult.Text = " Hex-Decimal : ";

} // 16진수로 출력 클릭

private string HexDisplay(string str)

{

long str\_start, str\_length, E\_position, exp, exp2, mod3;

string str\_exp, sign, str\_temp = "", str\_no;

E\_position = str.LastIndexOf("@");

str\_length = str.Length;

str\_exp = str.Substring((int)E\_position, (int)(str\_length - E\_position));

exp = long.Parse(str\_exp.Substring(2, (int)(str\_length - E\_position - 2)));

if (str\_exp.Substring(1, 1) == "-") exp = -exp;

if (str.Substring(0, 1) == "-")

{

str\_start = 3;

sign = "-";

}

else

{ str\_start = 2; sign = ""; }

str\_no = "";

if (exp > 0) // exp > 1 일때는 exp>E-Position, exp<E-Position로 분리

{

if (exp > 0)

{

str\_no = "";

str\_no += sign;

for (long i = 0; i < exp; i++) str\_temp += str.Substring((int)(str\_start + i), 1);

str\_length = str\_temp.Length;

mod3 = str\_length % 4;

exp2 = exp;

for (long i = 0; i < exp; i++)

{

str\_no += str\_temp.Substring((int)i, 1);

if ((i + 1) % 4 == mod3 && i < (exp - 1)) str\_no += " ";

exp2--;

}

if ((E\_position - str\_start - exp) > 0) str\_no += ".";

str\_no += str.Substring((int)(str\_start + exp), (int)(E\_position - str\_start - exp));

if (exp2 > 0)

{

str\_no += "@+";

str\_no += exp2.ToString();

}

}

else

str\_no = str;

}

str\_temp = str\_no.ToUpper();

return str\_temp;

} // 16진수로 출력

private string Display(string str)

{

long digit\_temp, digits=40;

long str\_start = 0, str\_length, E\_position, exp, mod3 = 0;

string str\_exp, sign, str\_temp = "", str\_no = "";

E\_position = str.LastIndexOf("E");

str\_length = str.Length;

str\_exp = str.Substring((int)E\_position, (int)(str\_length - E\_position));

exp = long.Parse(str\_exp.Substring(2, (int)(str\_length - E\_position - 2)));

if (str\_exp.Substring(1, 1) == "-") exp = -exp;

if (str.Substring(0, 1) == "-")

{ str\_start = 3; sign = "-"; }

else

{ str\_start = 2; sign = ""; }

str\_no = "";

if (exp < 0 || -exp >= digits)

{

if (exp > -7 && exp < 0)

{

str\_no += "0.";

for (long i = 0; i > exp; i--) str\_no += "0";

str\_no += str.Substring((int)str\_start, (int)(E\_position - str\_start));

}

else

str\_no = str; // exp < 0 일때는 스트링 그대로 표시

}

if (exp == 0) str\_no = str.Substring(0, (int)E\_position); // exp == 0 일때는 스트링에서 지수 부분만 제거

if (exp == 1) // exp == 1 일때는 스트링에서 지수 부분 및 왼쪽으로 1자리수 시프트

{

str\_no += sign;

str\_no += str.Substring((int)str\_start, 1);

if (E\_position - 1 > str\_start) str\_no += ".";

str\_no += str.Substring((int)str\_start + 1, (int)(E\_position - str\_start - 1));

}

if (exp > 1 && exp < digits) // exp > 1 일때는 exp>E-Position, exp<E-Position로 분리

{

digit\_temp = E\_position - str\_start;

if (digit\_temp < exp)

{

for (long i = 0; i < digit\_temp; i++) str\_temp += str.Substring((int)(str\_start + i), 1);

for (long i = digit\_temp; i < exp; i++) str\_temp += "0";

str\_length = str\_temp.Length;

mod3 = str\_length % 3;

str\_no = "";

for (long i = 0; i < str\_length; i++)

{

str\_no += str\_temp.Substring((int)i, 1);

if ((i + 1) % 3 == mod3 && i < (exp - 1)) str\_no += ",";

}

}

if (digit\_temp >= exp) // exp >= (E\_position - str\_start)

{

str\_no = "";

mod3 = exp % 3;

for (long i = 0; i < exp; i++)

{

str\_no += str.Substring((int)(i + str\_start), 1);

if ((i + 1) % 3 == mod3 && i < (exp - 1)) str\_no += ",";

}

if (E\_position - str\_start > exp) str\_no += ".";

str\_no += str.Substring((int)(exp + str\_start), (int)(digit\_temp - exp));

}

str\_no = sign + str\_no;

}

if (exp > digits) str\_no = str;

return str\_no + " ";

} // 10진수로 출력

}

}