Compte rendu Calculatrice

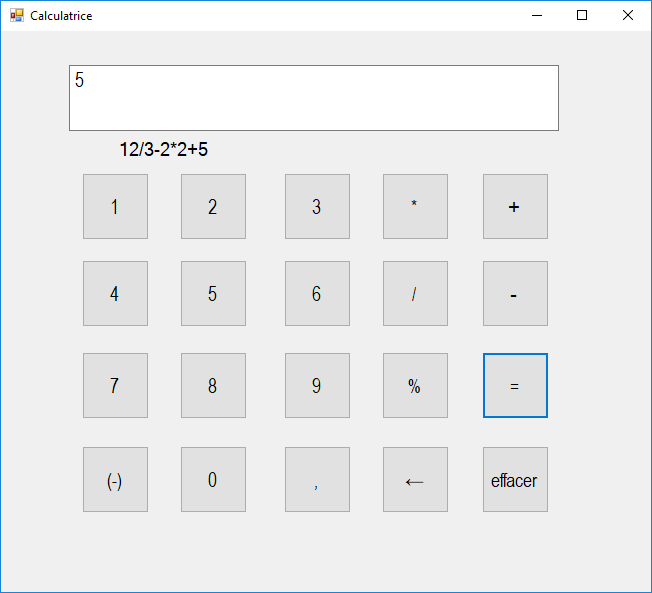
SLAM 2

PLANCHOT Matthieu

Mon programme se décompose en trois parties :

* Le design,
* Les boutons d’event
* Les méthodes de calcul

# Le design



(-) permet de mettre le nombre courant négatif, par exemple si l’on tape 12 puis sur ce bouton, il deviendra -12.

La flèche ← supprime le caractère précédent.

% correspond au modulo.

\* correspond au signe multiplié.

# Les boutons d’event

Ils sont au nombre de 7 dont les deux premiers button1\_Click et button13\_Click qui acceptent respectivement l’ajout des chiffres et l’ajout des opérateurs.

public partial class Form1 : Form

{

private Calcul calcul;

private string nb;

public Form1()

{

InitializeComponent();

calcul = new Calcul();

}

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

{

//ajout de chiffre

textBox1.Text += sender.ToString().Last();

nb += sender.ToString().Last();

}

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

{

//ajout d'un opérateur

calcul.setNb(nb);

nb = "";

textBox1.Text += sender.ToString().Last();

calcul.setOp(sender.ToString().Last().ToString());

}

private void button11\_Click(object sender, EventArgs e)

{

textBox1.Text += ",";

nb += ",";

}

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

{

//- met le nombre en négatif

textBox1.Text = textBox1.Text.Remove(textBox1.Text.Length - nb.Length, nb.Length);

nb = "-" + nb;

textBox1.Text += nb;

}

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

{

//effacer tout le textBox, les nb stockés et les opérateurs

textBox1.Text = "";

nb = "";

calcul.clear();

}

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

{

//= affiche le résultat et garde en mémoire l’entrée dans le label en dessous

if (nb != "")

calcul.setNb(nb);

label1.Text = textBox1.Text;

textBox1.Text = calcul.egal(calcul.nbList(), calcul.opList());

nb = textBox1.Text;

calcul.clear();

}

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

{

// effacer un caractère

if (textBox1.Text.Last() == '+' || textBox1.Text.Last() == '-' || textBox1.Text.Last() == '\*' || textBox1.Text.Last() == '/' || textBox1.Text.Last() == '%')

{

calcul.opList().RemoveAt(calcul.opList().Count - 1);

nb = Convert.ToString(calcul.nbList().ElementAt(calcul.nbList().Count - 1));

calcul.nbList().RemoveAt(calcul.nbList().Count - 1);

textBox1.Text = textBox1.Text.Remove(textBox1.Text.Length - 1, 1);

}

else if (textBox1.Text != "") {

textBox1.Text = textBox1.Text.Remove(textBox1.Text.Length - 1, 1);

nb = nb.Remove(nb.Length - 1, 1);

}

}

}

# Le calcul

J’ai utilisé deux listes nb et op pour stocker les nombres et les opérateurs.

class Calcul

{

private List<double> nb = new List<double>();

private List<string> op = new List<string>();

private double total = 0;

public void setNb(string nb)

{

this.nb.Add(Convert.ToDouble(nb));

}

public List<double> nbList()

{

return nb;

}

public void setOp(string op)

{

this.op.Add(op);

}

public List<string> opList()

{

return op;

}

// pour calculer le total d’une opération on supprime les 2 nombres qui ont servit ainsi que l’opérateur et on insère le total à la place vacante

public void RemoveInsert(int i, double total)

{

nb.RemoveAt(i);

nb.RemoveAt(i);

op.RemoveAt(i);

nb.Insert(i, total);

egal(nb, op);

}

public string egal(List<double> nb, List<string> op)

{

// la multiplication et la division sont prioritaire sur les autres opérateurs

// on les place dans une boucle séparé avant les autres

for (int i = 0; i < op.Count; i++)

{

if (op.ElementAt(i) == "\*")

{

total = nb.ElementAt(i) \* nb.ElementAt(i + 1);

RemoveInsert(i, total);

}

else if (op.ElementAt(i) == "/")

{

try

{

if (nb.ElementAt(i + 1) == 0)

throw new ArithmeticException("division impossible par 0");

total = nb.ElementAt(i) / nb.ElementAt(i + 1);

RemoveInsert(i, total);

}

catch (ArithmeticException e)

{

MessageBox.Show(e.ToString());

}

}

}

for (int i = 0; i < op.Count; i++)

{

if (op.ElementAt(i) == "+")

{

total = nb.ElementAt(i) + nb.ElementAt(i + 1);

RemoveInsert(i, total);

}

else if (op.ElementAt(i) == "-")

{

total = nb.ElementAt(i) - nb.ElementAt(i + 1);

RemoveInsert(i, total);

}

else if (op.ElementAt(i) == "%")

{

total = nb.ElementAt(i) % nb.ElementAt(i + 1);

RemoveInsert(i, total);

}

}

return Convert.ToString(total);

}

// appelé par button20\_Click : efface les nb stockés et les opérateurs

public void clear()

{

nb.Clear();

op.Clear();

}

}