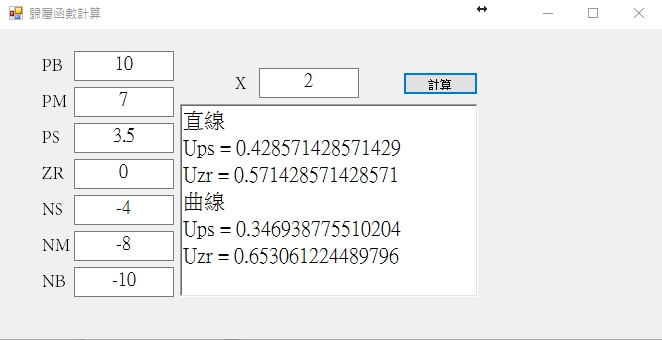
房志剛-1103105345-W01-20170323

結果



程式碼

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

{

}

private void button\_submit\_Click(object sender, EventArgs e)

{

if (CheckRange())

{

CalculateMF();

}

else

{

MessageBox.Show("數值範圍錯誤", "錯誤");

}

}

private bool CheckRange()

{

try

{

if (Convert.ToDouble(textBox\_PB.Text) < Convert.ToDouble(textBox\_PM.Text))

return false;

if (Convert.ToDouble(textBox\_PM.Text) < Convert.ToDouble(textBox\_PS.Text))

return false;

if (Convert.ToDouble(textBox\_PS.Text) < Convert.ToDouble(textBox\_ZR.Text))

return false;

if (Convert.ToDouble(textBox\_ZR.Text) < Convert.ToDouble(textBox\_NS.Text))

return false;

if (Convert.ToDouble(textBox\_NS.Text) < Convert.ToDouble(textBox\_NM.Text))

return false;

if (Convert.ToDouble(textBox\_NM.Text) < Convert.ToDouble(textBox\_NB.Text))

return false;

if (Convert.ToDouble(textBox\_PB.Text) == Convert.ToDouble(textBox\_X.Text))

return false;

if (Convert.ToDouble(textBox\_PM.Text) == Convert.ToDouble(textBox\_X.Text))

return false;

if (Convert.ToDouble(textBox\_PS.Text) == Convert.ToDouble(textBox\_X.Text))

return false;

if (Convert.ToDouble(textBox\_ZR.Text) == Convert.ToDouble(textBox\_X.Text))

return false;

if (Convert.ToDouble(textBox\_NS.Text) == Convert.ToDouble(textBox\_X.Text))

return false;

if (Convert.ToDouble(textBox\_NM.Text) == Convert.ToDouble(textBox\_X.Text))

return false;

if (Convert.ToDouble(textBox\_NB.Text) == Convert.ToDouble(textBox\_X.Text))

return false;

}

catch (Exception e)

{

return false;

}

return true;

}

private void CalculateMF()

{

int Scope = -1;

List<Double> num = new List<double>();

Double PB = Convert.ToDouble(textBox\_PB.Text);

Double PM = Convert.ToDouble(textBox\_PM.Text);

Double PS = Convert.ToDouble(textBox\_PS.Text);

Double ZR = Convert.ToDouble(textBox\_ZR.Text);

Double NS = Convert.ToDouble(textBox\_NB.Text);

Double NM = Convert.ToDouble(textBox\_NM.Text);

Double NB = Convert.ToDouble(textBox\_NB.Text);

Double X = Convert.ToDouble(textBox\_X.Text);

num.Add(PB);

num.Add(PM);

num.Add(PS);

num.Add(ZR);

num.Add(NS);

num.Add(NM);

num.Add(NB);

Double line\_a = 0, line\_b = 0, curve\_a = 0, curve\_b = 0;

for (int i = 0; i < num.Count - 1; i++)

{

if (num[i] > X && num[i + 1] < X)

{

Scope = i + 1;

line\_a = (num[i] - X) / (num[i] - num[i + 1]);

line\_b = (X - num[i + 1]) / (num[i] - num[i + 1]);

Double tmp = 0;

if ((num[i] + num[i + 1]) <= X)

{

tmp = Math.Pow((X - num[i]) / (num[i] - num[i + 1]), 2) \* 2;

}

else

{

tmp = Math.Pow((X - num[i + 1]) / (num[i] - num[i + 1]), 2) \* 2;

}

curve\_a = 1 - tmp;

curve\_b = tmp;

}

}

if (Scope == -1)

{

if (X > PB)

{

Scope = 0;

line\_a = 1;

line\_b = 0;

curve\_a = 1;

curve\_b = 0;

}

else

{

Scope = 9;

line\_a = 0;

line\_b = 1;

curve\_a = 1;

curve\_b = 0;

}

}

String A, B;

switch (Scope)

{

case 0: //X > PB

A = "pb";

B = "pb";

break;

case 1: // PB > X > PM

A = "pb";

B = "pm";

break;

case 2: // PM > X > PS

A = "pm";

B = "ps";

break;

case 3: // PS > X > ZR

A = "ps";

B = "zr";

break;

case 4: // ZR > X > NS

A = "zr";

B = "ns";

break;

case 5: // NS > X > NM

A = "ns";

B = "nm";

break;

case 6: // NM > X > NB

A = "nm";

B = "nb";

break;

default: //NB > X

A = "nb";

B = "nb";

break;

}

richTextBox\_out.Clear();

richTextBox\_out.Text = richTextBox\_out.Text + "直線" + Environment.NewLine;

richTextBox\_out.Text = richTextBox\_out.Text + "U" + A + " = " + line\_a + Environment.NewLine;

richTextBox\_out.Text = richTextBox\_out.Text + "U" + B + " = " + line\_b + Environment.NewLine;

richTextBox\_out.Text = richTextBox\_out.Text + "曲線" + Environment.NewLine;

richTextBox\_out.Text = richTextBox\_out.Text + "U" + A + " = " + curve\_a + Environment.NewLine;

richTextBox\_out.Text = richTextBox\_out.Text + "U" + B + " = " + curve\_b;

}