using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Calculatrice

{

public partial class Form1 : Form

{

double[,] TableauNormale;

public Form1()

{

InitializeComponent();

CB\_Cas.SelectedIndex = 0;

}

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

{

//on remplit le tableau des valeurs normales

TableauNormale = ValeurNormale();

}

//Gestion des événements

private void CB\_Cas\_SelectedIndexChanged(object sender, EventArgs e)

{

if (CB\_Cas.SelectedIndex == 0)

{

LB\_b.Visible = true;

TB\_b.Visible = true;

}

else

{

LB\_b.Visible = false;

TB\_b.Visible = false;

}

RefreshButton();

TB\_Rep.Text = string.Empty;

}

//retourne le tableau de la loi normale

private double[,] ValeurNormale()

{

double[,] valeur ={

// z 0,00 0,01 0,02 0,03 0,04 0,05 0,06 0,07 0,08 0,09

/\*0,0\*/{0.0000, 0.0040, 0.0080, 0.0120, 0.0160, 0.0199, 0.0239, 0.0279, 0.0319, 0.0359},

/\*0,1\*/{0.0398, 0.0438, 0.0478, 0.0517, 0.0557, 0.0596, 0.0636, 0.0675, 0.0714, 0.0753},

/\*0,2\*/{0.0793, 0.0832, 0.0871, 0.0910, 0.0948, 0.0987, 0.1026, 0.1064, 0.1103, 0.1141},

/\*0,3\*/{0.1179, 0.1217, 0.1255, 0.1293, 0.1331, 0.1368, 0.1406, 0.1443, 0.1480, 0.1517},

/\*0,4\*/{0.1554, 0.1591, 0.1628, 0.1664, 0.1700, 0.1736, 0.1772, 0.1808, 0.1844, 0.1879},

/\*0,5\*/{0.1915, 0.1950, 0.1985, 0.2019, 0.2054, 0.2088, 0.2123, 0.2157, 0.2190, 0.2224},

/\*0,6\*/{0.2257, 0.2291, 0.2324, 0.2357, 0.2389, 0.2422, 0.2454, 0.2486, 0.2517, 0.2549},

/\*0,7\*/{0.2580, 0.2611, 0.2642, 0.2673, 0.2704, 0.2734, 0.2764, 0.2794, 0.2823, 0.2852},

/\*0,8\*/{0.2881, 0.2910, 0.2939, 0.2967, 0.2995, 0.3023, 0.3051, 0.3078, 0.3106, 0.3133},

/\*0,9\*/{0.3159, 0.3186, 0.3212, 0.3238, 0.3264, 0.3289, 0.3315, 0.3340, 0.3365, 0.3389},

/\*1,0\*/{0.3413, 0.3438, 0.3461, 0.3485, 0.3508, 0.3531, 0.3554, 0.3577, 0.3599, 0.3621},

/\*1,1\*/{0.3643, 0.3665, 0.3686, 0.3708, 0.3729, 0.3749, 0.3770, 0.3790, 0.3810, 0.3830},

/\*1,2\*/{0.3849, 0.3869, 0.3888, 0.3907, 0.3925, 0.3944, 0.3962, 0.3980, 0.3997, 0.4015},

/\*1,3\*/{0.4032, 0.4049, 0.4066, 0.4082, 0.4099, 0.4115, 0.4131, 0.4147, 0.4162, 0.4177},

/\*1,4\*/{0.4192, 0.4207, 0.4222, 0.4236, 0.4251, 0.4265, 0.4279, 0.4292, 0.4306, 0.4319},

/\*1,5\*/{0.4332, 0.4345, 0.4357, 0.4370, 0.4382, 0.4394, 0.4406, 0.4418, 0.4429, 0.4441},

/\*1,6\*/{0.4452, 0.4463, 0.4474, 0.4484, 0.4495, 0.4505, 0.4515, 0.4525, 0.4535, 0.4545},

/\*1,7\*/{0.4554, 0.4564, 0.4573, 0.4582, 0.4591, 0.4599, 0.4608, 0.4616, 0.4625, 0.4633},

/\*1,8\*/{0.4641, 0.4649, 0.4656, 0.4664, 0.4671, 0.4678, 0.4686, 0.4693, 0.4699, 0.4706},

/\*1,9\*/{0.4713, 0.4719, 0.4726, 0.4732, 0.4738, 0.4744, 0.4750, 0.4756, 0.4761, 0.4767},

/\*2,0\*/{0.4772, 0.4778, 0.4783, 0.4788, 0.4793, 0.4798, 0.4803, 0.4808, 0.4812, 0.4817},

/\*2,1\*/{0.4821, 0.4826, 0.4830, 0.4834, 0.4838, 0.4842, 0.4846, 0.4850, 0.4854, 0.4857},

/\*2,2\*/{0.4861, 0.4864, 0.4868, 0.4871, 0.4875, 0.4878, 0.4881, 0.4884, 0.4887, 0.4890},

/\*2,3\*/{0.4893, 0.4896, 0.4898, 0.4901, 0.4904, 0.4906, 0.4909, 0.4911, 0.4913, 0.4916},

/\*2,4\*/{0.4918, 0.4920, 0.4922, 0.4925, 0.4927, 0.4929, 0.4931, 0.4932, 0.4934, 0.4936},

/\*2,5\*/{0.4938, 0.4940, 0.4941, 0.4943, 0.4945, 0.4946, 0.4948, 0.4949, 0.4951, 0.4952},

/\*2,6\*/{0.4953, 0.4955, 0.4956, 0.4957, 0.4959, 0.4960, 0.4961, 0.4962, 0.4963, 0.4964},

/\*2,7\*/{0.4965, 0.4966, 0.4967, 0.4968, 0.4969, 0.4970, 0.4971, 0.4972, 0.4973, 0.4974},

/\*2,8\*/{0.4974, 0.4975, 0.4976, 0.4977, 0.4977, 0.4978, 0.4979, 0.4979, 0.4980, 0.4981},

/\*2,9\*/{0.4981, 0.4982, 0.4982, 0.4983, 0.4984, 0.4984, 0.4985, 0.4985, 0.4986, 0.4986},

/\*3,0\*/{0.4987, 0.4987, 0.4987, 0.4988, 0.4988, 0.4989, 0.4989, 0.4989, 0.4990, 0.4990},

/\*3,1\*/{0.4990, 0.4991, 0.4991, 0.4991, 0.4992, 0.4992, 0.4992, 0.4992, 0.4993, 0.4993},

/\*3,2\*/{0.4993, 0.4993, 0.4994, 0.4994, 0.4994, 0.4994, 0.4994, 0.4995, 0.4995, 0.4995},

/\*3,3\*/{0.4995, 0.4995, 0.4995, 0.4996, 0.4996, 0.4996, 0.4996, 0.4996, 0.4996, 0.4997},

/\*3,4\*/{0.4997, 0.4997, 0.4997, 0.4997, 0.4997, 0.4997, 0.4997, 0.4997, 0.4997, 0.4998},

/\*3,5\*/{0.4998, 0.4998, 0.4998, 0.4998, 0.4998, 0.4998, 0.4998, 0.4998, 0.4998, 0.4998},

/\*3,6\*/{0.4998, 0.4998, 0.4999, 0.4999, 0.4999, 0.4999, 0.4999, 0.4999, 0.4999, 0.4999},

/\*3,7\*/{0.4999, 0.4999, 0.4999, 0.4999, 0.4999, 0.4999, 0.4999, 0.4999, 0.4999, 0.4999},

/\*3,8\*/{0.4999, 0.4999, 0.4999, 0.4999, 0.4999, 0.4999, 0.4999, 0.4999, 0.4999, 0.4999},

/\*3,9\*/{0.5000, 0.5000, 0.5000, 0.5000, 0.5000, 0.5000, 0.5000, 0.5000, 0.5000, 0.5000},

/\*4,0\*/{0.5000, 0.5000, 0.5000, 0.5000, 0.5000, 0.5000, 0.5000, 0.5000, 0.5000, 0.5000}};

return valeur;

}

private void Moy\_ET\_a\_b\_Text\_Changed(object sender, EventArgs e)

{

RefreshButton();

if (TB\_Moy.Text.StartsWith("."))

{

TB\_Moy.Text = "0" + TB\_Moy.Text;

TB\_Moy.Select(2, 0);

}

if (TB\_ET.Text.StartsWith("."))

{

TB\_ET.Text = "0" + TB\_ET.Text;

TB\_ET.Select(2, 0);

}

if (TB\_a.Text.StartsWith("."))

{

TB\_a.Text = "0" + TB\_a.Text;

TB\_a.Select(2, 0);

}

if (TB\_b.Text.StartsWith("."))

{

TB\_b.Text = "0" + TB\_b.Text;

TB\_b.Select(2, 0);

}

}

private void RefreshButton()

{

//si les chiffres sont valide ou pas vide

if ((TB\_b.Visible &&

TB\_Moy.Text != string.Empty &&

TB\_ET.Text != string.Empty &&

TB\_ET.Text != "0" &&

TB\_ET.Text != "0." &&

TB\_a.Text != string.Empty &&

TB\_b.Text != string.Empty) ||

(!TB\_b.Visible &&

TB\_Moy.Text != string.Empty &&

TB\_ET.Text != string.Empty &&

TB\_ET.Text != "0" &&

TB\_ET.Text != "0." &&

TB\_a.Text != string.Empty))

{

BTN\_Calculer.Enabled = true;

}

else

{

BTN\_Calculer.Enabled = false;

}

}

//Masque des entré usager (chifre seulement)

private void Moy\_ET\_a\_b\_KeyPress(object sender, KeyPressEventArgs e)

{

char ch = e.KeyChar;

if (TB\_Moy.Focused && ch == 46 && TB\_Moy.Text.IndexOf(".") != -1)

{

e.Handled = true;

return;

}

if (TB\_ET.Focused && ch == 46 && TB\_ET.Text.IndexOf(".") != -1)

{

e.Handled = true;

return;

}

if (TB\_a.Focused && ch == 46 && TB\_a.Text.IndexOf(".") != -1)

{

e.Handled = true;

return;

}

if (TB\_b.Focused && ch == 46 && TB\_b.Text.IndexOf(".") != -1)

{

e.Handled = true;

return;

}

if (!Char.IsDigit(ch) && ch != 8 && ch != 46)

{

e.Handled = true;

}

}

private void BTN\_Calculer\_Click(object sender, EventArgs e)

{

TB\_Rep.Text = Calculer() + "%";

}

private string Calculer()

{

//assignation des valeurs

double Rep = 0;

double EcartType = double.Parse(TB\_ET.Text, System.Globalization.CultureInfo.InvariantCulture);

double Moyenne = double.Parse(TB\_Moy.Text, System.Globalization.CultureInfo.InvariantCulture);

double a = double.Parse(TB\_a.Text, System.Globalization.CultureInfo.InvariantCulture);

double b = 0;

//si on doit calculer le z

if (TB\_b.Text != string.Empty)

b = double.Parse(TB\_b.Text, System.Globalization.CultureInfo.InvariantCulture);

//calculer cote Z du a

double Za = (a - Moyenne) / EcartType;

double Zb = 0;

if (CB\_Cas.SelectedIndex == 0)//si P(a<x<b)

{

Zb = (b - Moyenne) / EcartType;

if ((Zb<0 && Za>0)||(Zb>0 && Za<0))

{

Rep = GetProportion(Za) + GetProportion(Zb);

}

else

{

if (GetProportion(Za)<=GetProportion(Zb))

{

Rep = GetProportion(Zb) - GetProportion(Za);

}

else if(GetProportion(Za)>GetProportion(Zb))

{

Rep = GetProportion(Za) - GetProportion(Zb);

}

}

}

else if (CB\_Cas.SelectedIndex == 1)//si P(X<a)

{

if (Za > 0)

{

Rep = GetProportion(Za) + 0.5;

}

else

{

Rep = 0.5 - GetProportion(Za);

}

}

else if (CB\_Cas.SelectedIndex == 2)//si P(X>a)

{

if (Za < 0)

{

Rep = GetProportion(Za) + 0.5;

}

else

{

Rep = 0.5 - GetProportion(Za);

}

}

return (Rep\*100).ToString();

}

//recois la cote z et retourne la valeur du tableau

private double GetProportion(double nombre)

{

double i = nombre;

if (i < 0)

i = i \* -1;

int Ligne = Convert.ToInt32((i / 10).ToString("0.0000000").Substring(2, 2));

int Colonne = Convert.ToInt32((i / 10).ToString("0.0000000").Substring(4, 1));

return TableauNormale[Ligne,Colonne];

}

}

}