

ДОДАТОК А

Код програми

```
using System;
using System.IO;
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;
using System.Diagnostics;

namespace WindowsFormsApp1
{
    public partial class Form1 : Form
    {

        public Form1()
        {
            InitializeComponent();
            button2.Enabled = false; //деактивуємо кнопку старт
        }

        private void OnTextChanged(object sender, EventArgs args)
        {
```

```
UpdateUserInterface(); //викликаємо функцію кожний раз коли змінюється текст
}
```

`private void UpdateUserInterface()` //функція що перевіряє чи введені значення в текстові поля

```
{
    if (textBox1 != null || textBox2 != null )
    {
        button2.Enabled = true;
    }
    else
    {
        button2.Enabled = false;
    }
}
```

`private void button1_Click(object sender, EventArgs e)`//генеруємо координати для датчиків

```
{
```

`int[] dataLength = new int[20];` //ініціалізуємо масив для зберігання координат датчиків

```
this.dataGridView1.Rows.Clear(); //видаляємо данні з таблиці
```

```
Random r = new Random();
```

```
foreach (var i in dataLength)
```

```
{
```

```

double x;
double y;
x = Math.Round((r.NextDouble() * (9 - 1) + 1), 1);
y = Math.Round((r.NextDouble() * (9 - 1) + 1), 1);

this.dataGridView1.Rows.Add(x,y);//додаємо координати в таблицю
}
}

```

```

private void newToolStripMenuItem_Click(object sender, EventArgs
e)//видалення даних з таблиці
{
    this.dataGridView1.Rows.Clear();
}

```

```

private void openToolStripMenuItem_Click(object sender, EventArgs
e)//завантаження даних з зовнішнього файлу
{
    Stream mystr = null;//ініціалізуємо потік
    if(openFileDialog1.ShowDialog() == DialogResult.OK)
    {
        if((mystr = openFileDialog1.OpenFile()) != null)
        {
            StreamReader myread = new StreamReader(mystr);//ініціалізуємо зчи-
тувач потоку
            string[] str;
            int num = 0;
            try

```

```
{
    string[] str1 = myread.ReadToEnd().Split('\n');//зчитуємо данні строк
з потоку розділяючи їх по переходу на нову строку
    num = str1.Count();//кількість строк
    dataGridView1.RowCount = num;
    for(int i = 0; i < num; i++)
    {
        str = str1[i].Split(' ');//розділяємо данні по пробілам
        for(int j = 0; j < dataGridView1.ColumnCount; j++)
        {
            try
            {
                dataGridView1.Rows[i].Cells[j].Value = str[j];
            }
            catch { }
        }
    }
}
catch (Exception ex) {
    MessageBox.Show(ex.Message);
}
finally
{
    myread.Close();//закриваємо потік
}
}
}
```

```

private void saveToolStripMenuItem_Click(object sender, EventArgs
e)//записуємо данні в файл
{
    Stream myStream;
    if (saveFileDialog1.ShowDialog() == DialogResult.OK)
    {
        if ((myStream = saveFileDialog1.OpenFile()) != null)
        {
            StreamWriter myWritet = new StreamWriter(myStream);
            try
            {
                for(int i = 0; i < dataGridView1.RowCount; i++)//зчитуємо данні з
                строчок
                {
                    for(int j = 0; j < dataGridView1.ColumnCount; j++)//зчитуємо
                    данні з !ячеек!
                    {
                        myWritet.Write(dataGridView1.Rows[i].Cells[j].Value.ToString());
                            if ((dataGridView1.ColumnCount - j) != 1) myWritet.Write("
");//розділяємо данні в строчці
                    }
                    myWritet.WriteLine();//записуємо данні в файл
                }
            }
            catch (Exception ex)
            {
                MessageBox.Show(ex.Message);
            }
            finally

```

```

    {
        myWritet.Close();
    }
    myStream.Close();
}
}
}

```

`private void` button2_Click(`object` sender, `EventArgs` e)//передаємо данні в другу форму

```

{
    double[,] result = new double[dataGridView1.Rows.Count, data-
GridView1.ColumnCount];//зчитуємо данні з таблиці в змінну
    for (int i = 0; i < dataGridView1.Rows.Count; i++)
    {
        for (int j = 0; j < dataGridView1.ColumnCount; j++)
        {
            result[i, j] = Convert.ToDouble(dataGridView1.Rows[i].Cells[j].Value);
        }
    }

    string costm = textBox1.Text;//зчитуємо данні з текстових полів в змінні
    string costunit = textBox2.Text;

```

`Form2` form = `new Form2`(costm, costunit, result);//відкриваємо другу форму и передаємо змінні

```
try
{
    form.ShowDialog();
}
catch
{
    MessageBox.Show("Some text", "Some title",
        MessageBoxButtons.OK, MessageBoxIcon.Error);
}
}
private void textBox1_KeyPress(object sender, KeyPressEventArgs e)
{
    if (char.IsNumber(e.KeyChar))
    {
    }
    else
    {
        e.Handled = e.KeyChar != (char)Keys.Back;
    }
}
}
```

```
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 DevExpress;
using System.Windows.Forms;
using System.Windows.Forms.DataVisualization.Charting;
using System.Threading;
using System.Diagnostics;

namespace WindowsFormsApp1
{
    public partial class Form2 : Form
    {

        public int SwitchCount = 3;
        Random r = new Random();
        double[,] sw = new double[6, 2];
        public double[,] sensors;
        public int costunit;
        public int costm;
        public double pathSum = 0;
        double[,] distance;
        public int[] Path;
        public Form2(string cost1, string cost2, double[,] result)
        {

            InitializeComponent();

            try
            {
```

```

sensors = result;
costm = Convert.ToInt32(cost1);
costunit = Convert.ToInt32(cost2);

for (int i = 1; i < 6; i++)//генеруємо координати для комутаторів
{
    sw[i, 0] = Math.Round((r.NextDouble() * (8 - 2) + 3), 1);
    sw[i, 1] = Math.Round((r.NextDouble() * (8 - 2) + 3), 1);
}
label2.Text = Convert.ToString(costunit * SwitchCount);
chart1.Series["Main"].Points.Add(new DataPoint(5, 5));//МАЛЮЄМО ГОЛОВ-
НИЙ КОМП'ЮТЕР
for (int i = 0; i < 20; i++)//МАЛЮЄМО ДАТЧИКИ
{
    chart1.Series["Sensors"].Points.Add(new DataPoint(result[i, 0], result[i,
1]));
}
for (int i = 1; i <= SwitchCount; i++)//МАЛЮЄМО КОМУТАТОРИ
{
    chart1.Series["Switch"].Points.Add(new DataPoint(sw[i, 0], sw[i, 1]));
}
label2.Text = Convert.ToString(costunit * SwitchCount);
drawToMain();
drawToSensors();
}
catch (Exception ex)
{
    //MessageBox.Show(ex.Message);
}

```



```

        dist = distTmp;
    }
    else if (distTmp < dist)
    {
        dist = distTmp;
        count++;
    }
    else if (i == SwitchCount)
    {
        chart1.Series["Connection"].Points.Add(new DataPoint(sensors[j,
0], sensors[j, 1]));
        chart1.Series["Connection"].Points.Add(new DataPoint(sw[count,
0], sw[count, 1]));
        chart1.Series["Connection"].ChartType = SeriesChartType.Line;
    }

    }
    catch
    {
    }
    //dist = Math.Sqrt(Math.Pow(this.x - P.x, 2) + Math.Pow(this.x - P.x, 2));
    }
    }
}

private void button1_Click(object sender, EventArgs e)//видаляємо всі з'єдна-
ння
{
    chart1.Series["Connection"].Points.Clear();
}

```

`private void` button2_Click(object sender, EventArgs e)//додаємо кількість комутаторів

```
{
    if (SwitchCount<5)
    {
        //chart1.Series["Connection"].Points.Clear();
        chart1.Series["Main Connections"].Points.Clear();
        SwitchCount++;
        chart1.Series["Switch"].Points.Clear();
        label2.Text = Convert.ToString(costunit * SwitchCount);
        for (int i = 1; i <= SwitchCount; i++)
        {
            chart1.Series["Switch"].Points.Add(new DataPoint(sw[i, 0], sw[i, 1]));
        }
        drawToMain();
        chart1.Series["Connection"].Points.Clear();
        drawToSensors();
    }
}
```

`private void` button3_Click(object sender, EventArgs e)//зменшуємо кількість комутаторів

```
{
    if (SwitchCount>1)
    {
        //chart1.Series["Connection"].Points.Clear();
        chart1.Series["Main Connections"].Points.Clear(); ;
        SwitchCount--;
        chart1.Series["Switch"].Points.Clear();
        label2.Text = Convert.ToString(costunit * SwitchCount);
    }
}
```

```
for (int i = 1; i <= SwitchCount; i++)
{
    chart1.Series["Switch"].Points.Add(new DataPoint(sw[i, 0], sw[i, 1]));
}
drawToMain();
chart1.Series["Connection"].Points.Clear();
drawToSensors();
}
}
private void label1_Click(object sender, EventArgs e)
{
}
}
}
```

ДОДАТОК Б

Презентація

