

## 23 ЭЛЕМЕНТЫ УПРАВЛЕНИЯ

Задание 1. Дана строка, состоящая из групп нулей и единиц. Посчитать количество нулей и единиц.

Листинг программы:

```
private void button1_Click(object sender, EventArgs e)
{
    try
    {
        int index = listBox1.SelectedIndex;
        string str = (string)listBox1.Items[index];
        int len = str.Length;
        int countOnes = 0;
        int countZeros = 0;
        int i = 0;
        while (i < len)
        {
            if (str[i] == '1')
            {
                countOnes += 1;
            }
            else if (str[i] == '0')
            {
                countZeros += 1;
            }
            i += 1;
        }
        labelOnes.Text = $"Result 1: {Convert.ToString(countOnes)}";
        labelZeros.Text = $"Result 0: {Convert.ToString(countZeros)}";
    }
    catch (Exception ex)
    {
        MessageBox.Show($"Some error! ({ex.Message})");
    }
}
```

					УП 2-40 01 01.31 ТП.2471.22.23			
Изм.	Лист	№ докум	Подп.	Дата				
Разраб.		Мушинский М.С			ЭЛЕМЕНТЫ УПРАВЛЕНИЯ		Лист	Лист
Пров.		Толочко П.С.						
Н.контр.							Гродненский ГКТТид	
Утв.								

}

Таблица 23.1 – Входные и выходные данные

Входные данные	Выходные данные
Набор нулей и единиц	Количество нулей и единиц

Анализ результатов:

The screenshot shows a program window with a text area containing the following binary strings:

```

101010111000
0000000000
111111111110101
111111111111
1 0
0 1
  
```

Below the text area is a button labeled "Calculate". At the bottom of the window, there are two labels: "Result 1:" and "Result 0:", which are currently empty.

Рисунок 23.1 – Результат работы программы  
Источник: собственная разработка

Задание 2. Построить график функции  $y = ax + b$ . Таблицу данных получить путём изменения параметра  $X$  с шагом  $h$ . Самостоятельно выбрать удобные параметры настройки.

Листинг программы:

```

private void button1_Click(object sender, EventArgs e)
{
    try
    {
        double Xmin = double.Parse(textBoxXmin.Text);
        double Xmax = double.Parse(textBoxXmax.Text);
        double Step = double.Parse(textBoxStep.Text);
        double a = double.Parse(textBoxA.Text);
        double b = double.Parse(textBoxB.Text);
        int count = (int)Math.Ceiling((Xmax - Xmin) / Step) + 1;
        double[] x = new double[count];
    }
}
  
```

```

double[] y1 = new double[count];
for (int i = 0; i < count; i++)
{

    x[i] = Xmin + Step * i;
    y1[i] = (a * x[i]) + b;
}
chart1.ChartAreas[0].AxisX.Minimum = Xmin;
chart1.ChartAreas[0].AxisX.Maximum = Xmax;
chart1.ChartAreas[0].AxisX.MajorGrid.Interval = Step;
chart1.Series[0].Points.DataBindXY(x, y1);
}
catch (Exception ex)
{
    MessageBox.Show(ex.Message, "Error", MessageBoxButtons.OK,
    MessageBoxIcon.Error);
}
}

```

Таблица 23.2 – Входные и выходные данные

Входные данные	Выходные данные
1, 10, 1, 1, 10	График функции

Анализ результатов:

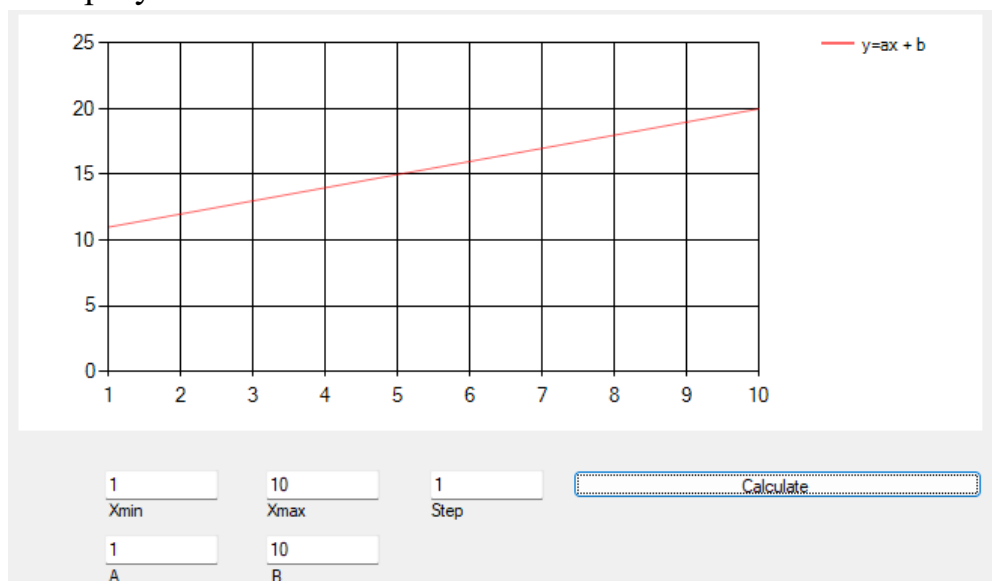


Рисунок 23.2 – Результат работы программы  
Источник: собственная разработка