**ПРАКТИЧЕСКОЕ ЗАДАНИЕ 23**

**ЭЛЕМЕНТЫ УПРАВЛЕНИЯ**

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

количество нулей и единиц.

Листинг программы

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

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

{

listBox1.Items.Add(listBox1.Text);

int index = listBox1.SelectedIndex;

string str = (string)listBox1.Items[index];

int len = str.Length;

int count1 = 0, count2 = 0;

for (int i = 0; i < str.Length; i++)

{

if (str[i] == '1')

{

count1++;

}

}

for (int i = 0; i < str.Length; i++)

{

if (str[i] == '0')

{

count2++;

}

}

label2.Text = "Количество нулей = " + count2.ToString();

label1.Text = "Количество единиц = " + count1.ToString();

}

}

Анализ результатов:

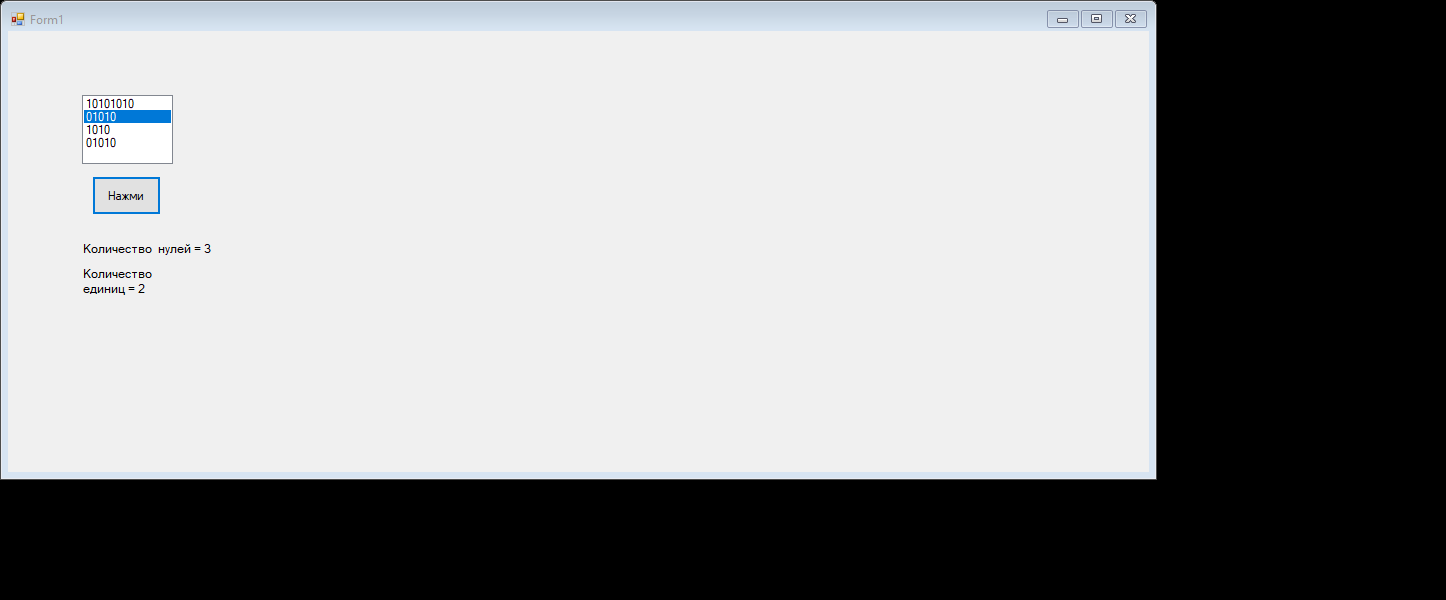


Рисунок 1.1 – Результат работы программы

Задание 2. Построить график функции. Таблицу данных получить путём

изменения параметра X с шагом h. Самостоятельно выбрать удобные параметры настройки.

1. y=ax+b

Листинг программы

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

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

{

double A = double.Parse(textBox1.Text);

double B = double.Parse(textBox2.Text);

double X = double.Parse(textBox3.Text);

int count = (int)Math.Ceiling((A \* X) / B)

+ 1;

double[] x = new double[count];

double[] y1 = new double[count];

double[] y2 = new double[count];

for (int i = 0; i < count; i++)

{

x[i] = A + B \* i;

y1[i] = Math.Sin(x[i]);

}

chart1.ChartAreas[0].AxisX.Minimum = A;

chart1.ChartAreas[0].AxisX.Maximum = B;

chart1.ChartAreas[0].AxisX.MajorGrid.Interval = X;

chart1.Series[0].Points.DataBindXY(x, y1);

}

}

Анализ результатов:

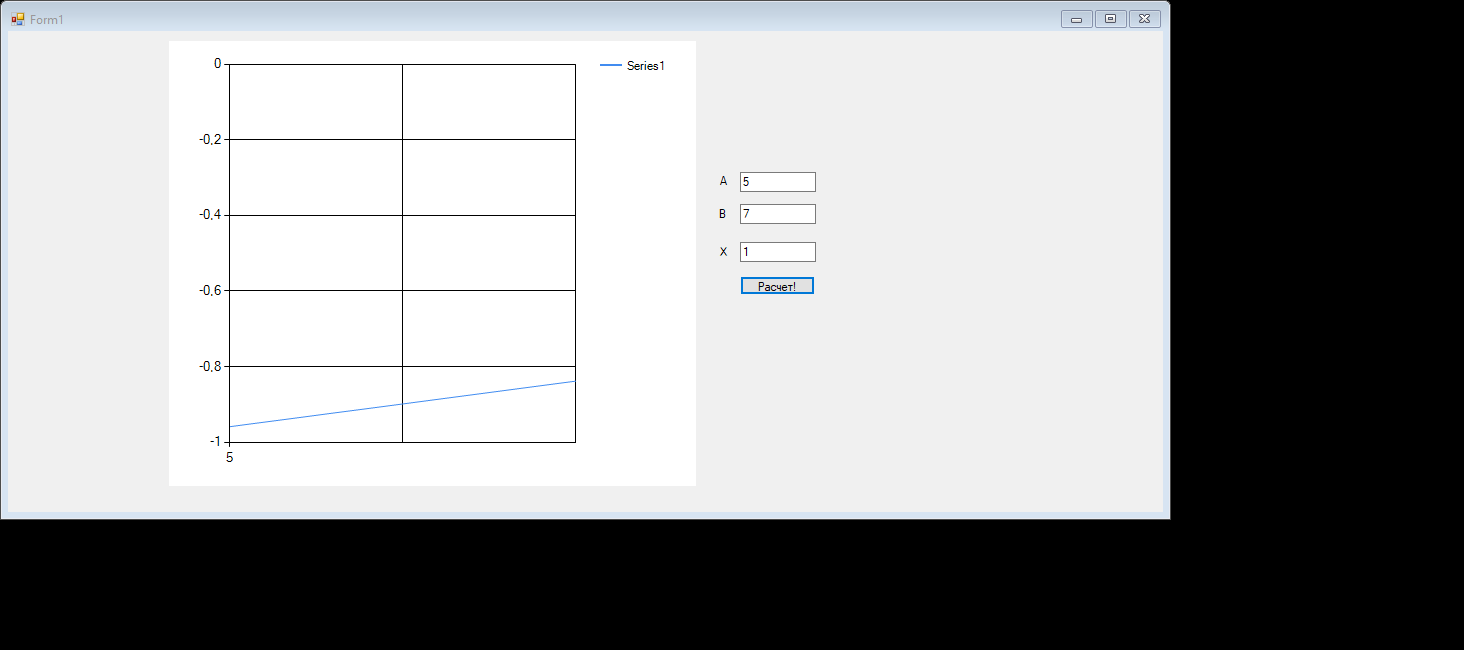


Рисунок 1.2 – Результат работы программы

Задание 3. Построить изображение при помощи фигур. Арбуз

Листинг программы

private void Form1\_Paint\_1(object sender, PaintEventArgs e)

{

Graphics g = e.Graphics;

g.Clear(Color.White);

g.FillRectangle(Brushes.Black, 240, 180, 10, 10);

SolidBrush B = new SolidBrush(Color.Red);

g.FillEllipse(B, 200, 100, 200, 100);

for (int i = 0; i < 1; i++)

g.DrawEllipse(new Pen(Brushes.DarkGreen, 4),

200, 100 \* i + 100, 200, 100 \* i + 100);

g.FillRectangle(Brushes.Black, 270, 140, 2, 10);

g.FillRectangle(Brushes.Black, 250, 160, 7, 10);

g.FillRectangle(Brushes.Black, 250, 144, 4, 10);

g.FillRectangle(Brushes.Black, 260, 180, 3, 10);

g.FillRectangle(Brushes.Black, 275, 120, 1, 10);

g.FillRectangle(Brushes.Black, 290, 170, 7, 10);

g.FillRectangle(Brushes.Black, 295, 190, 8, 10);

g.FillRectangle(Brushes.Black, 265, 180, 7, 10);

g.FillRectangle(Brushes.Black, 350, 170, 10, 10);

g.FillRectangle(Brushes.Black, 340, 160, 7, 10);

g.FillRectangle(Brushes.Black, 350, 110, 9, 10);

g.FillRectangle(Brushes.Black, 321, 180, 7, 10);

g.FillRectangle(Brushes.Black, 344, 170, 4, 10);

g.FillRectangle(Brushes.Black, 300, 160, 7, 10);

g.FillRectangle(Brushes.Black, 312, 140, 1, 10);

g.FillRectangle(Brushes.Black, 315, 140, 2, 10);

}

Анализ результатов:

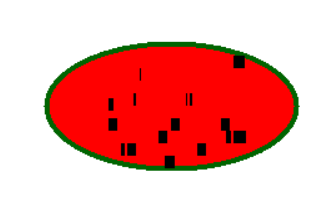


Рисунок 1.3 – Результат работы программы

Задание 4. Опишите одномерный массив. Создайте обработчики события для кнопок (код приведен ниже). Данная программа заменяет все отрицательные числа нулями. Протестируйте правильность выполнения программы. Модифицируйте программу в соответствии с индивидуальным заданием.

Листинг программы

public partial class Form1 : Form

{

int[] Mas = new int[20];

int j = 0;

public Form1()

{

InitializeComponent();

}

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

{

listBox1.Items.Clear();

Random rand = new Random();

for (int i = 0; i < 20; i++)

{

Mas[i] = rand.Next(-15, 15);

listBox1.Items.Add("Mas[" + i.ToString() + "]=" + Mas[i].ToString());

}

}

private void button2\_Click(object sender, EventArgs e)

{

int max = Mas.Max();

listBox2.Items.Clear();

for (int i = 0; i < 20; i++)

{

Mas[0] = max;

listBox2.Items.Add("Mas[" + Convert.ToString(i) + "]=" + Mas[i].ToString());

}

}

}

Анализ результатов:

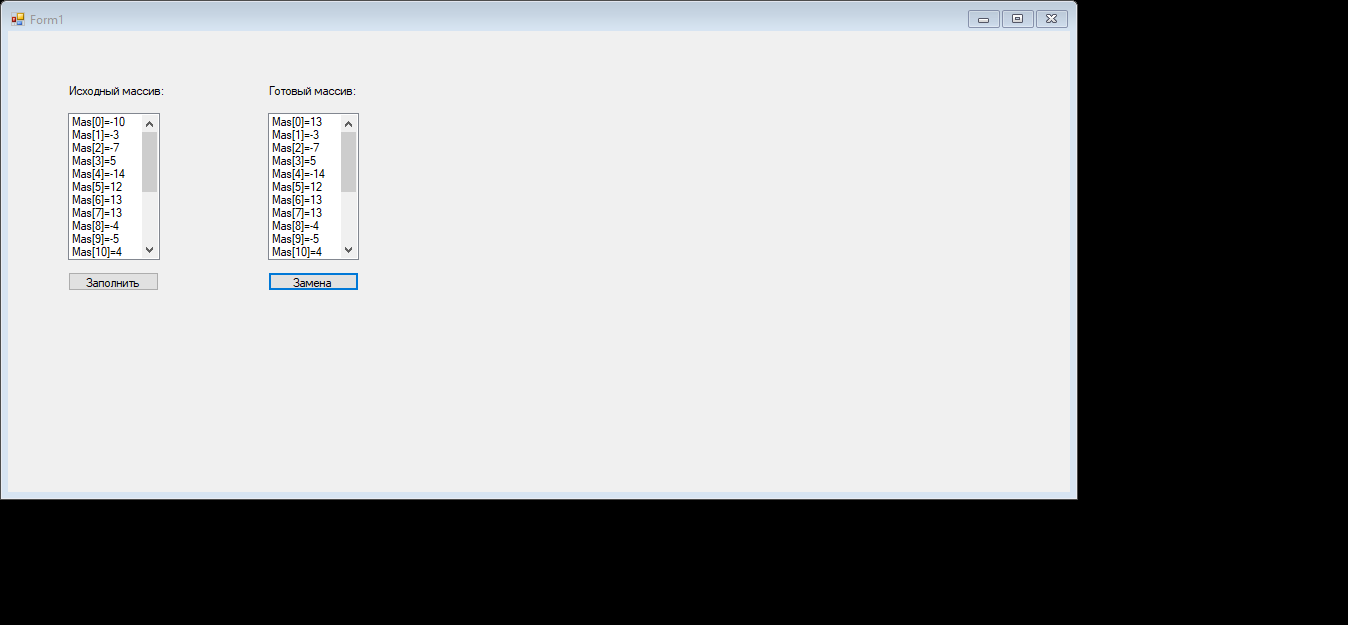


Рисунок 1.4 – Результат работы программы

Задание 5. Дана матрица A(3,4). Найти наименьший элемент в каждой строке

матрицы. Вывести исходную матрицу и результаты вычислений.

Листинг программы

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

{

dataGridView1.RowCount = 3;

dataGridView1.ColumnCount = 4;

int[,] mas = new int[3, 4];

int i, j;

Random random = new Random();

for (i = 0; i < 3; i++)

{

for (j = 0; j < 4; j++)

{

mas[i, j] = random.Next(10, 20);

}

}

for (i = 0; i < 3; i++)

{

for (j = 0; j < 4; j++)

{

dataGridView1.Rows[i].Cells[j].Value = mas[i, j].ToString();

}

}

for (i = 0; i < 3; i++)

{

int min = 100;

for (j = 0; j < 4; j++)

{

if (mas[i, j] < min)

{

min = mas[i, j];

}

}

textBoxResult.Text += min.ToString() + Environment.NewLine;

}

Анализ результатов:

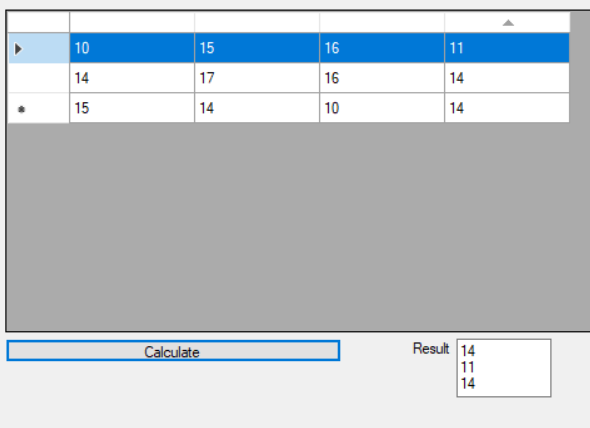


Рисунок 1.5 – Результат работы программы

Задание 6. Написать программу с кнопкой и тремя полями ввода. При нажатии на кнопку программа анализирует содержимое первого поля и динамически пораждает элемент управления. Если в первом поле ввода содержится буква «К», то на форму добавляется кнопка, если буква «П»-поле ввода, если «М»-метка. Во втором и третьем поле находятся координаты левого верхнего угла будущего элемента управления. При создании будущего элемента настроить его свойства (на ваш выбор (цвет, размер и т.п)).

Листинг программы:

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

{

try

{

string letter = textBoxLetter.Text;

double x = Convert.ToDouble(textBoxX.Text);

double y = Convert.ToDouble(textBoxY.Text);

if (letter == "П")

{

TextBox textBox = new TextBox();

textBox.MouseMove += MouseDownTB;

textBox.Location = new Point(Convert.ToInt32(x), Convert.ToInt32(y));

this.Controls.Add(textBox);

}

else if (letter == "К")

{

Button btn = new Button();

btn.Text = "Кнопка";

btn.MouseMove += MouseDownB;

btn.Location = new Point(Convert.ToInt32(x), Convert.ToInt32(y));

this.Controls.Add(btn);

}

else if (letter == "М")

{

Label lbl = new Label();

lbl.Text = "Метка";

lbl.MouseMove += MouseDownL;

lbl.Location = new Point(Convert.ToInt32(x), Convert.ToInt32(y));

this.Controls.Add(lbl);

}

else

{

MessageBox.Show("Элемент управления отсутствует", "Warning", MessageBoxButtons.OK, MessageBoxIcon.Warning);

}

}

catch (Exception ex)

{

MessageBox.Show(ex.StackTrace, "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void MouseDownTB(Object sender, MouseEventArgs e)

{

TextBox t = (TextBox)sender;

t.Dispose();

}

private void MouseDownB(Object sender, MouseEventArgs e)

{

Button b = (Button)sender;

b.Dispose();

}

private void MouseDownL(Object sender, MouseEventArgs e)

{

Label l = (Label)sender;

l.Dispose();

}

Анализ результатов:

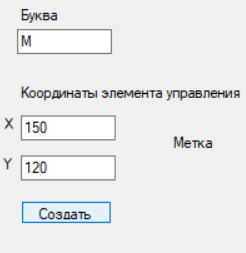


Рисунок 1.6 – Результат работы программы

Задание 7. Разместите на форме четыре кнопки (Button). Напишите для каждой обработчик события, который будет менять размеры и местоположение на окне других кнопок.

Листинг программы

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

{

button2.Size = new System.Drawing.Size(100, 30);

button2.Location = new Point(162, 49);

button3.Size = new System.Drawing.Size(70, 50);

button3.Location = new Point(182, 125);

button4.Size = new System.Drawing.Size(80, 20);

button4.Location = new Point(24, 149);

}

private void button2\_Click(object sender, EventArgs e)

{

button1.Size = new System.Drawing.Size(75, 23);

button1.Location = new Point(210, 108);

button3.Size = new System.Drawing.Size(85, 33);

button3.Location = new Point(94, 157);

button4.Size = new System.Drawing.Size(95, 43);

button4.Location = new Point(282, 170);

}

private void button3\_Click(object sender, EventArgs e)

{

button1.Size = new System.Drawing.Size(30, 50);

button1.Location = new Point(25, 27);

button2.Size = new System.Drawing.Size(30, 30);

button2.Location = new Point(256, 123);

button4.Size = new System.Drawing.Size(50, 20);

button4.Location = new Point(142, 96);

}

private void button4\_Click(object sender, EventArgs e)

{

button1.Size = new System.Drawing.Size(100, 50);

button1.Location = new Point(36, 228);

button2.Size = new System.Drawing.Size(88, 38);

button2.Location = new Point(46, 141);

button3.Size = new System.Drawing.Size(65, 20);

button3.Location = new Point(142, 59);

}

private void button5\_Click(object sender, EventArgs e)

{

button1.Size = new System.Drawing.Size(75, 23);

button1.Location = new Point(78, 79);

button2.Size = new System.Drawing.Size(75, 23);

button2.Location = new Point(210, 79);

button3.Size = new System.Drawing.Size(75, 23);

button3.Location = new Point(78, 199);

button4.Size = new System.Drawing.Size(75, 23);

button4.Location = new Point(210, 199);

}

Анализ результатов:

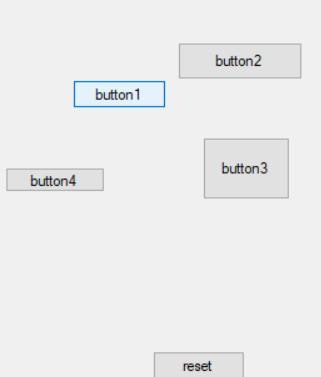


Рисунок 1.7 – Результат работы программы

Задание 8. Разместите на форме четыре кнопки (Button). Сделайте на кнопках следующие надписи: «красный», «зеленый», «синий», «желтый». Создайте четыре обработчика события нажатия на данные кнопки, которые будут менять цвет формы в соответствии с текстом на кнопках.

Листинг программы

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

{

this.BackColor = System.Drawing.Color.Red;

}

private void button2\_Click(object sender, EventArgs e)

{

this.BackColor = System.Drawing.Color.Green;

}

private void button3\_Click(object sender, EventArgs e)

{

this.BackColor = System.Drawing.Color.Blue;

}

private void button4\_Click(object sender, EventArgs e)

{

this.BackColor = System.Drawing.Color.Yellow;

}

Анализ результатов:

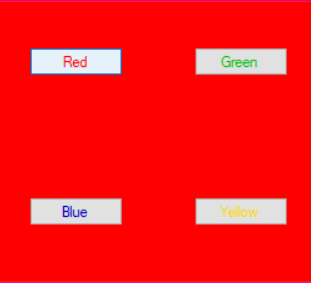


Рисунок 1.8 – Результат работы программы