**23 Элементы управления**

Задание 1. Лабораторная работа 6, стр. 50.

Листинг программы:

namespace Task1

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

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

{

try

{

int index = listBox1.SelectedIndex;

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

char[] chars = str.ToCharArray();

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

{

if ((int)chars[i] >= 97 && (int)chars[i] <= 122)

{

chars[i] = '+';

}

}

str = String.Join("", chars);

label1.Text = str;

}

catch (Exception ex)

{

MessageBox.Show(ex.Message, "Something goes wrong", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

}

}

Анализ результатов:

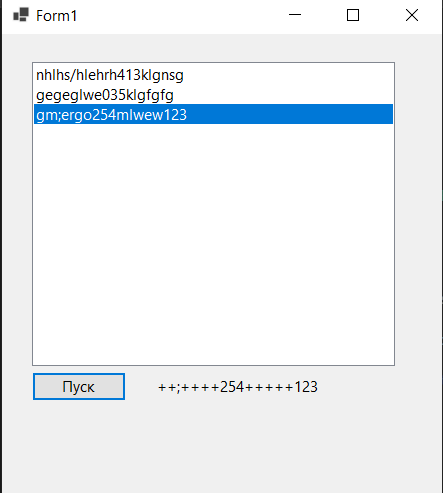


Рисунок 23.1 – Результат работы программы

Источник: собственная разработка

Задание 2. Построить график функции. Таблицу данных получить путём изменения параметра X с шагом h. Самостоятельно выбрать удобные параметры настройки.

Листинг программы:

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 Task2

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

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

{

try

{

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

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

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

int count = (int)Math.Ceiling((Xmax - Xmin) / Step)

+ 1;

double[] x = new double[count];

double[] y = new double[count];

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

{

x[i] = Xmin + Step \* i;

y[i] = Math.Abs(x[i]);

}

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, y);

}

catch (Exception ex)

{

MessageBox.Show(ex.Message, "Something goes wrong", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

}

}

Анализ результатов:

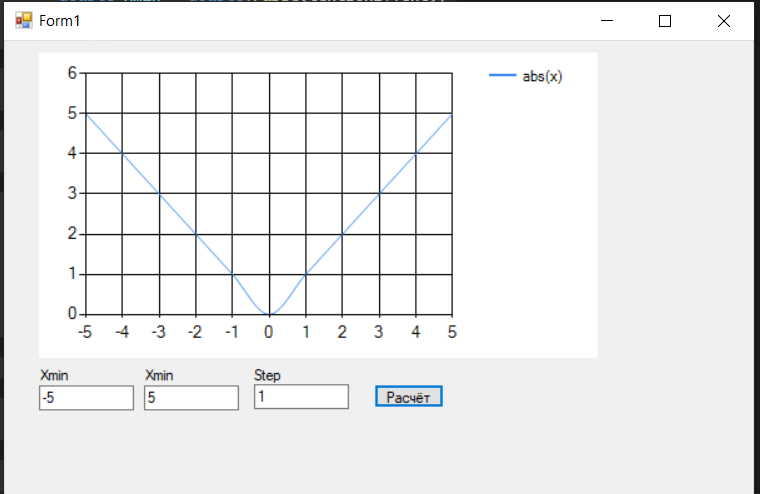


Рисунок 23.2 – Результат работы программы

Источник: собственная разработка

Задание 3. Лабораторная работа No10, стр. 70. Построить изображение при помощи фигур.

Листинг программы:

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 Additonal3

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

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

{

dataGridView1.RowCount = 10;

dataGridView1.ColumnCount = 20;

int[,] arr = new int[10, 20];

Random rnd = new Random();

for (int i = 0; i < arr.GetLength(0); i++)

for (int j = 0; j < arr.GetLength(1); j++)

{

arr[i, j] = rnd.Next(-20, 20);

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

}

dataGridView1.AutoResizeColumns();

int[] arrPositive = new int[arr.GetLength(0)];

for (int i = 0; i < arr.GetLength(0); i++)

for (int j = 0; j < arr.GetLength(1); j++)

if (arr[i,j] >= 0)

arrPositive[i]++;

label1.Text = "Количество положительных элементов в каждой строке: " + string.Join(" ", arrPositive);

}

}

}

Анализ результатов:

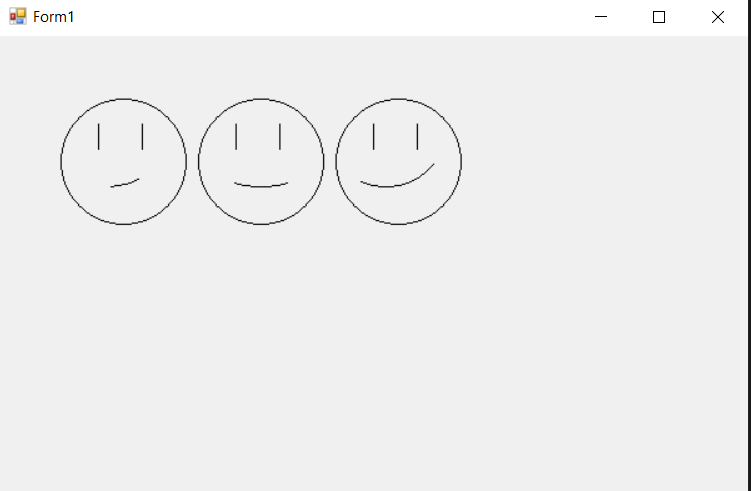


Рисунок 23.3 – Результат работы программы

Источник: собственная разработка

Задание 4. 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)).

При наведении курсора на элемент управления он должен быть удален с формы.

Листинг программы:

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 Task4

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

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

{

if (textBox1.Text == "К")

{

Button btn = new Button();

btn.Text = "Кнопка";

btn.MouseMove += ButtonMouseMove;

btn.Location = new Point(Convert.ToInt32(textBox2.Text), Convert.ToInt32(textBox3.Text));

this.Controls.Add(btn);

}

if (textBox1.Text == "П")

{

TextBox text = new TextBox();

text.MouseMove += TextBoxMouseMove;

text.Location = new Point(Convert.ToInt32(textBox2.Text), Convert.ToInt32(textBox3.Text));

this.Controls.Add(text);

}

if (textBox1.Text == "М")

{

Label lbl = new Label();

lbl.Text = "Метка";

lbl.MouseMove += LabelMouseMove;

lbl.Location = new Point(Convert.ToInt32(textBox2.Text), Convert.ToInt32(textBox3.Text));

this.Controls.Add(lbl);

}

}

private void ButtonMouseMove(Object sender, MouseEventArgs e)

{

Button btn = (Button)sender;

btn.Dispose();

}

private void TextBoxMouseMove(Object sender, MouseEventArgs e)

{

TextBox text = (TextBox)sender;

text.Dispose();

}

private void LabelMouseMove(Object sender, MouseEventArgs e)

{

Label lbl = (Label)sender;

lbl.Dispose();

}

}

}

Анализ результатов:

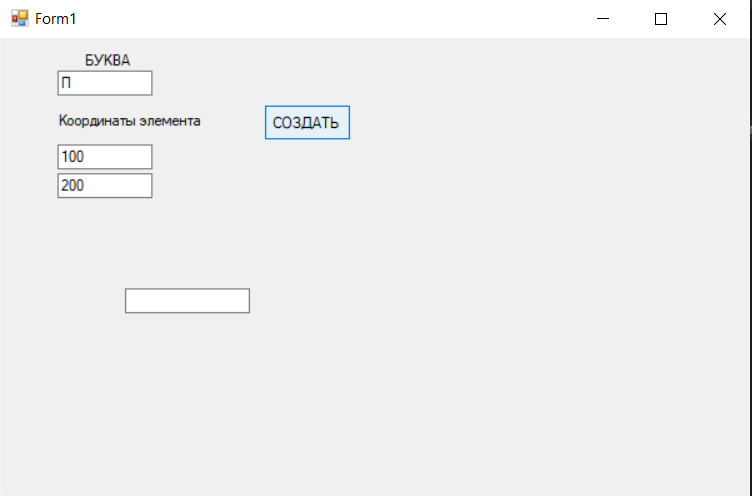


Рисунок 23.4 – Результат работы программы

Источник: собственная разработка

Дополнительное задание1.1.

Листинг программы:

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 Additional1.\_1

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

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

{

label1.Text = textBox1.Text;

}

private void Form1\_MouseDown(object sender, MouseEventArgs e)

{

this.BackColor = Color.Green;

label1.Text = "Начало работы";

textBox1.Text = "";

}

}

}

Анализ результатов:

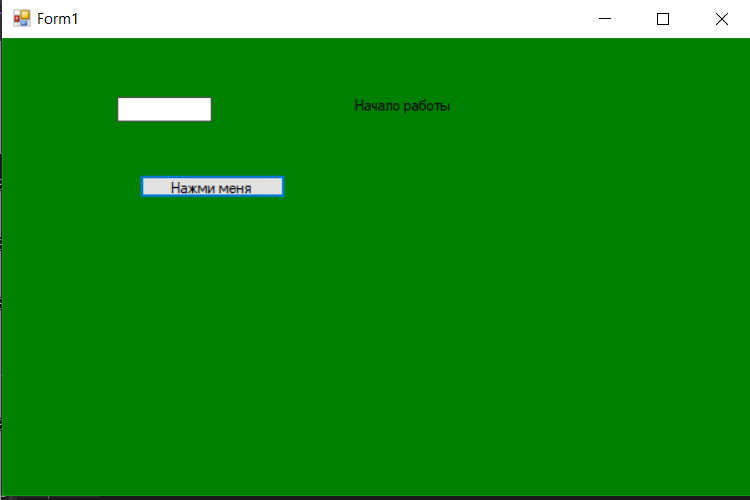


Рисунок 23.5 – Результат работы программы

Источник: собственная разработка

Дополнительное задание 1.2.

Листинг программы:

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 Additional1.\_2

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

private void Form1\_MouseDown(object sender, MouseEventArgs e)

{

this.BackColor = Color.Red;

textBox1.Visible = false;

button1.Visible = false;

button2.Visible = false;

}

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

{

textBox1.ReadOnly = true;

}

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

{

textBox1.ReadOnly = false;

}

}

}

Анализ результатов:

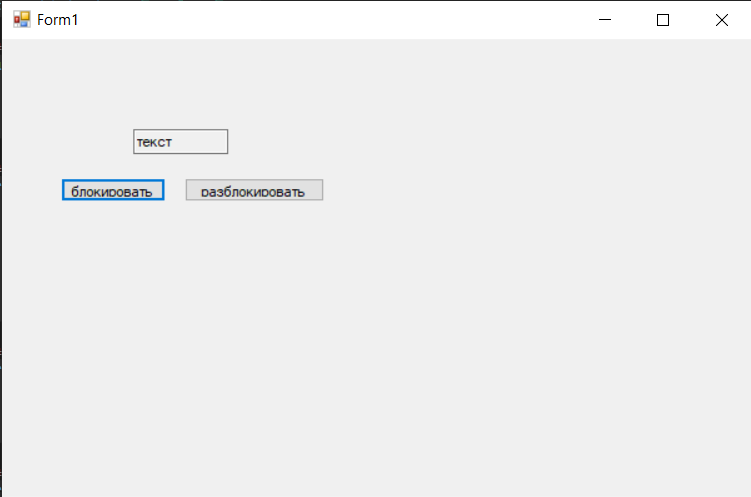


Рисунок 23.6 – Результат работы программы

Источник: собственная разработка

Дополнительное задание 2.

Листинг программы:

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 Additional2

{

public partial class Form1 : Form

{

int[] arr = new int[15];

public Form1()

{

InitializeComponent();

}

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

{

Random rnd = new Random();

for (int i = 0; i < 15; i++)

{

arr[i] = rnd.Next(50) - 20;

listBox1.Items.Add($"Mas[{i}] = {arr[i]}");

}

}

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

{

for (int i = 0; i < 15; i++)

{

if (arr[i] > 0)

listBox2.Items.Add($"Mas[{i}] = {arr[i] \* arr[i]}");

if (arr[i] < 0)

listBox2.Items.Add($"Mas[{i}] = {arr[i] \* 2}");

}

}

}

}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 Additional2

{

public partial class Form1 : Form

{

int[] arr = new int[15];

public Form1()

{

InitializeComponent();

}

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

{

Random rnd = new Random();

for (int i = 0; i < 15; i++)

{

arr[i] = rnd.Next(50) - 20;

listBox1.Items.Add($"Mas[{i}] = {arr[i]}");

}

}

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

{

for (int i = 0; i < 15; i++)

{

if (arr[i] > 0)

listBox2.Items.Add($"Mas[{i}] = {arr[i] \* arr[i]}");

if (arr[i] < 0)

listBox2.Items.Add($"Mas[{i}] = {arr[i] \* 2}");

}

}

}

}

Анализ результатов:

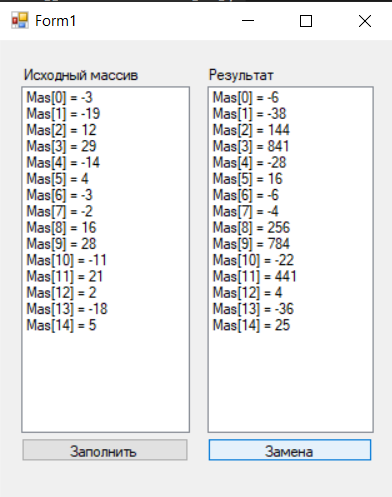


Рисунок 23.7 – Результат работы программы

Источник: собственная разработка

Дополнительное задание 3.

Листинг программы:

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 Additonal3

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

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

{

dataGridView1.RowCount = 10;

dataGridView1.ColumnCount = 20;

int[,] arr = new int[10, 20];

Random rnd = new Random();

for (int i = 0; i < arr.GetLength(0); i++)

for (int j = 0; j < arr.GetLength(1); j++)

{

arr[i, j] = rnd.Next(-20, 20);

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

}

dataGridView1.AutoResizeColumns();

int[] arrPositive = new int[arr.GetLength(0)];

for (int i = 0; i < arr.GetLength(0); i++)

for (int j = 0; j < arr.GetLength(1); j++)

if (arr[i,j] >= 0)

arrPositive[i]++;

label1.Text = "Количество положительных элементов в каждой строке: " + string.Join(" ", arrPositive);

}

}

}

Анализ результатов:



Рисунок 23.8 – Результат работы программы

Источник: собственная разработка