1. **Элементы управления**

Задание 1. Дана строка символов, состоящая из произвольного текста на английском языке, слова разделены пробелами. Заменить все буквы латинского алфавита на знак «+».

Листинг программы:

namespace Task\_1

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

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

{

char[] alf = "qwertyuiopasdfghjklzxcvbnm".ToCharArray();

string text = textBox1.Text;

text.ToLower();

string newText = "";

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

{

for (int j = 0; j < alf.Length; j++)

{

if (text[i] == ' ')

{

newText += " ";

break;

}

else if (text[i] == alf[j])

{

newText += "+";

break;

}

}

}

textBox2.Text = newText;

}

}

}

Таблица 1.1 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| Csdf b gbty ntbr bty | ++++ + ++++ ++++ +++ |

Анализ результатов:

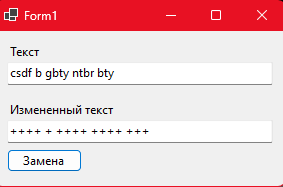


Рисунок 1.1 – Результат работы программы

Источник: собственная разработка

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

Листинг программы:

namespace Task\_2

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

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

{

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

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

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

double q = 2;

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] = GetVal(Sh(x[i]), x[i], q);

}

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);

}

private double GetVal(double f, double x, double q)

{

if (Math.Abs(x \* q) > 10.0)

return Math.Log(Math.Abs(f) + Math.Abs(q));

if (Math.Abs(x \* q) < 10.0)

return Math.Exp(f + q);

return f + q;

}

public double Sh(double x)

{

return (Math.Exp(x) - Math.Exp(-x)) / 2.0;

}

}

}

Таблица 2.1 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| 3 10 0,2 |  |

Анализ результатов:

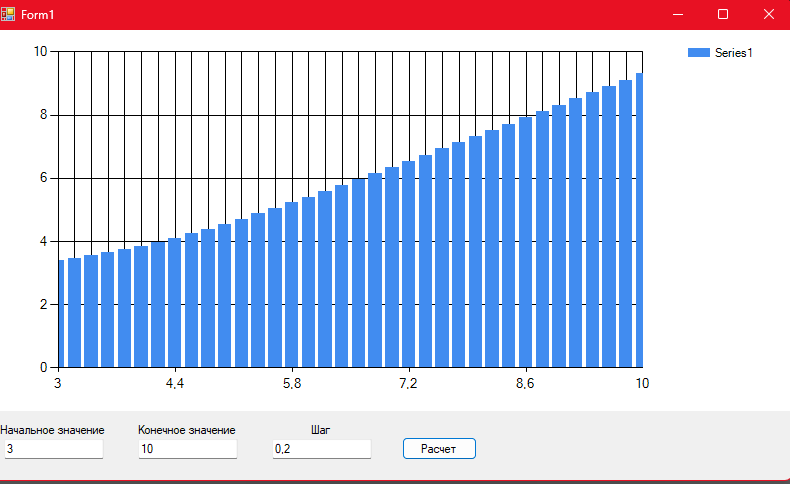


Рисунок 2.1 – Результат работы программы

Источник: собственная разработка

Задание 3. Построить изображение при помощи фигур. Три смайлика.

Листинг программы:

namespace Task\_3

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

Graphics g;

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

{

g = CreateGraphics();

g.Clear(Color.Azure);

g.DrawEllipse(Pens.Yellow, 50, 100, 200, 200);

g.DrawEllipse(Pens.Yellow, 70, 125, 50, 70);

g.FillEllipse(Brushes.Black, 80, 125, 30, 40);

g.DrawArc(Pens.Gold, 100, 200 ,100, 100, 0, 100);

g.DrawEllipse(Pens.Red, 300, 100, 200, 200);

g.DrawEllipse(Pens.Red, 330, 125, 50, 50);

g.FillEllipse(Brushes.Black, 340, 145, 30, 30);

g.DrawEllipse(Pens.Red, 420, 125, 50, 50);

g.FillEllipse(Brushes.Black, 430, 145, 30, 30);

g.DrawArc(Pens.Gold, 350, 200 ,100, 100, 180, 180);

g.DrawEllipse(Pens.Green, 550, 100, 200, 200);

g.DrawEllipse(Pens.Red, 580, 135, 50, 50);

g.FillEllipse(Brushes.Black, 590, 145, 30, 30);

g.DrawEllipse(Pens.Red, 670, 135, 50, 50);

g.FillEllipse(Brushes.Black, 680, 145, 30, 30);

g.DrawLine(Pens.Gold, 600, 250, 700, 250);

}

}

}

Таблица 3.1 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
|  |  |

Анализ результатов:

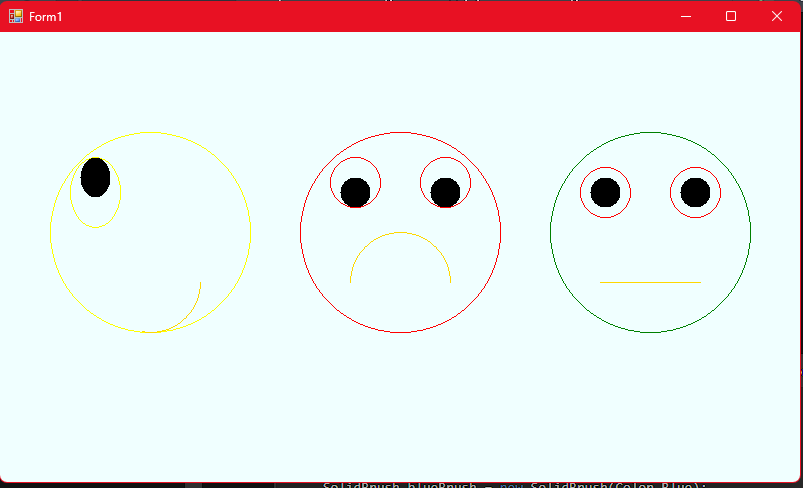


Рисунок 3.1 – Результат работы программы

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:

namespace Task\_4

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

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

{

char alf = char.Parse(textBox3.Text);

int x = int.Parse(textBox2.Text);

int y = int.Parse(textBox1.Text);

Control newControl = new Control();

switch (alf)

{

case 'K':

case 'К':

case 'k':

case 'к':

newControl = new Button() { Location = new Point(x, y), Text = "Кнопка" };

break;

case 'M':

case 'М':

case 'м':

case 'm':

newControl = new Label() { Location = new Point(x, y), Text = "Метка" };

break;

case 'П':

case 'п':

newControl = new TextBox() { Location = new Point(x, y), Text = "Текст бокс" };

break;

default:

MessageBox.Show("Такого элемента нет");

break;

}

this.Controls.Add(newControl);

newControl.MouseHover += new EventHandler((s, ev) =>

{

this.Controls.Remove(newControl);

});

}

}

}

Таблица 4.1 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| П 145 100  К 45 222  К 145 222  м 451 22 | Текст бокс  Кнопка  Кнопка  Метка |

Анализ результатов:

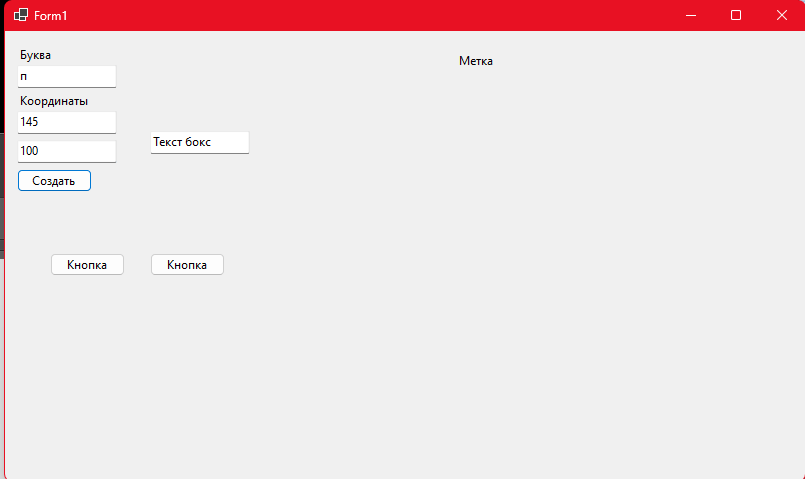


Рисунок 4.1 – Результат работы программы

Источник: собственная разработка

Задание 5. Дан массив, содержащий 15 элементов. Все положительные элементы возвести в квадрат, а отрицательные умножить на 2. Вывести исходный и полученный массив.

Листинг программы:

namespace Task\_5

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

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

{

listBox1.Items.Clear();

Random rnd = new Random();

for (int i = 0; i < 10; i++)

{

listBox1.Items.Add(rnd.Next(-20, 50));

}

}

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

{

listBox2.Items.Clear();

for (int i = 0; i < listBox1.Items.Count; i++)

{

int num = (int)listBox1.Items[i];

if (num >= 0)

{

listBox2.Items.Add(Math.Pow(num, 2));

}

else

{

listBox2.Items.Add(num \* 2);

}

}

}

}

}

Таблица 5.1 – Входные и выходные данные

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
|  |  |

Анализ результатов:

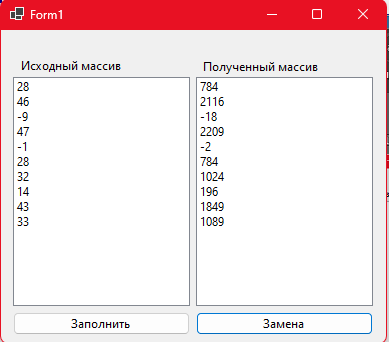


Рисунок 5.1 – Результат работы программы

Источник: собственная разработка