# **Практическое занятие №3**

*Дата*: 19.04.2023

*Тема:* «Процедуры и функции. Перегрузка функций (методов) и операторов»

*Цель:* Научиться разрабатывать функции. Реализовывать перегрузку функций.

Использовать перегруженные функции.

**Ход работы**

**Задание 1.**

**Условие:** Даны m и n - натуральные числа. Вычислите НОК(m, n).

**Исходный код:**

namespace \_1зад

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

private int GCD(int a, int b)

{

while (b != 0)

{

int temp = b;

b = a % b;

a = temp;

}

return a;

}

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

{

int m, n;

if (!int.TryParse(textBox1.Text, out m))

{

MessageBox.Show("Введите корректное значение m");

return;

}

if (!int.TryParse(textBox2.Text, out n))

{

MessageBox.Show("Введите корректное значение n");

return;

}

int gcd = GCD(m, n);

int lcm = m \* n / gcd;

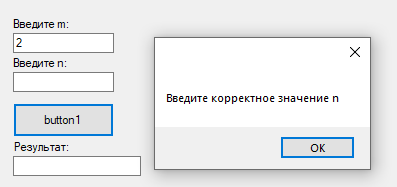
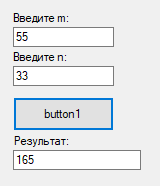
textBox3.Text = lcm.ToString();

}

}

}

**Результат:**



**Задание 2.**

**Условие:** Дана точка A с координатами (x, y) и мишень - 10 концентрических

кругов с центром в начале координат и радиусами R1, R2 ... R10. Постройте приложение, которое определяет количество выбитых очков. Предполагается, как обычно, что за попадание точки в круг самого малого радиуса начисляется 10 очков и так далее до одного очка. За попадание в "молоко" очки не начисляются.

**Исходный код:**

namespace \_2зад

{

public partial class Form1 : Form

{

private const int NumCircles = 10;

private readonly int[] \_circleRadii = { 20, 40, 60, 80, 100, 120, 140, 160, 180, 200 };

private readonly Brush[] \_circleBrushes = { Brushes.White, Brushes.Red, Brushes.Orange, Brushes.Yellow, Brushes.Green, Brushes.Blue, Brushes.Indigo, Brushes.Violet, Brushes.Gray, Brushes.Black };

public Form1()

{

InitializeComponent();

pictureBoxCanvas.Paint += pictureBoxCanvas\_Paint;

}

private int GetScore(double x, double y)

{

double r = Math.Sqrt(x \* x + y \* y);

if (r > \_circleRadii[NumCircles - 1]) // точка вне мишени

{

return 0;

}

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

{

if (r <= \_circleRadii[i]) // точка попала в круг

{

return NumCircles - i; // начисляем очки в зависимости от радиуса круга

}

}

return 0; // точка попала в молоко, очков не начисляется

}

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

{

// рисуем мишень на pictureBox

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

{

int diameter = \_circleRadii[i] \* 2;

int x = pictureBoxCanvas.Width / 2 - \_circleRadii[i];

int y = pictureBoxCanvas.Height / 2 - \_circleRadii[i];

e.Graphics.FillEllipse(\_circleBrushes[i], x, y, diameter, diameter);

}

}

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

{

double x, y;

if (!double.TryParse(textBoxX.Text, out x))

{

MessageBox.Show("Введите корректное значение координаты x");

return;

}

if (!double.TryParse(textBoxY.Text, out y))

{

MessageBox.Show("Введите корректное значение координаты y");

return;

}

int score = GetScore(x, y);

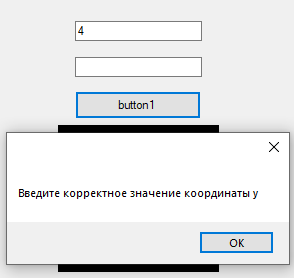
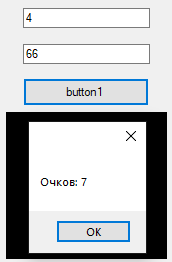
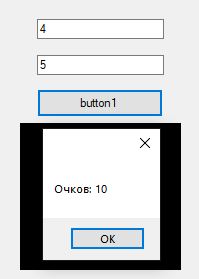
MessageBox.Show($"Очков: {score}");

}

}

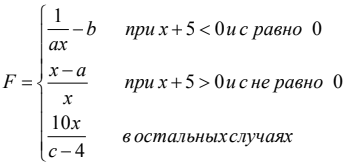
}

**Результат:**



**Задание 3.**

**Условие:** Вычислить и вывести на экран в виде таблицы значения функции F на интервале от Хнач. до Хкон. с шагом dХ.



где a, b, c — действительные числа.

Функция F должна принимать действительное значение, если выражение (Aц И Bц) ИЛИ (Bц И Сц) не равно нулю, и целое значение в противном случае. Через Aц, Bц и Сц обозначены целые части значений a, b, c, операции И и ИЛИ — поразрядные. Значения a, b, c, Хнач., Хкон., dХ ввести с клавиатуры. Предусмотреть обработку возможных ошибок ввода и вычислений.

**Исходный код:**

namespace \_3зад

{

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

{

double a, b, c, x, dx, fx;

int ac, bc, cc;

if (!double.TryParse(textBoxA.Text, out a))

{

MessageBox.Show("Некорректное значение a", "Ошибка");

return;

}

if (!double.TryParse(textBoxB.Text, out b))

{

MessageBox.Show("Некорректное значение b", "Ошибка");

return;

}

if (!double.TryParse(textBoxC.Text, out c))

{

MessageBox.Show("Некорректное значение c", "Ошибка");

return;

}

if (!double.TryParse(textBoxXFrom.Text, out x))

{

MessageBox.Show("Некорректное значение Xнач.", "Ошибка");

return;

}

if (!double.TryParse(textBoxXTo.Text, out double xTo))

{

MessageBox.Show("Некорректное значение Xкон.", "Ошибка");

return;

}

if (!double.TryParse(textBoxStep.Text, out dx))

{

MessageBox.Show("Некорректное значение dX", "Ошибка");

return;

}

dataGridViewResults.Rows.Clear();

for (double i = x; i <= xTo; i += dx)

{

ac = (int)Math.Truncate(a);

bc = (int)Math.Truncate(b);

cc = (int)Math.Truncate(c);

if ((ac & bc) != 0 || (bc & cc) != 0)

{

fx = 1.0 / (a \* i) - b;

}

else if (i + 5 > 0 && c != 0)

{

fx = (i - a) / i;

}

else

{

fx = 10 \* i / (c - 4);

}

if (Math.Truncate(fx) == fx)

{

dataGridViewResults.Rows.Add(i, (int)fx);

}

else

{

dataGridViewResults.Rows.Add(i, fx);

}

}

}

}

**Результат:**

