**Занятие № 3**

**Дата выполнения работы:** 19.04.2023

# **Тема работы:** Процедуры и функции. Перегрузка функций (методов) и операторов.

**Ход работы**

**Задание 1**

Даны m и n - натуральные числа. Вычислите НОД(m, n). При

вычислениях не используйте операций умножения и деления.

**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 App1

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

static int GCD(int a, int b)

{

while (b != 0)

{

int remainder = a % b;

a = b;

b = remainder;

}

return a;

}

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

{

int m = Convert.ToInt32(textBox1.Text);

int n = Convert.ToInt32(textBox2.Text);

int nod = GCD(m, n);

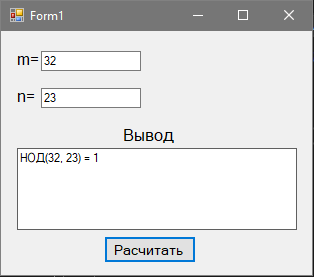
textBox3.Text = $"НОД({m}, {n}) = {nod}";

}

}

}

**Результат:**



**Задание 2**

Дана точка A с координатами (x, y) и два прямоугольника Q1 и Q2 со

сторонами, параллельными осям координат. Каждый из

прямоугольников задается парой точек Q1(p1, p2), Q2(p3, p4),

определяющих левый нижний и правый верхний углы прямоугольника.

Постройте приложение, которое определяет, принадлежит ли точка

A(x, y) хотя бы одному из прямоугольников Q1 и Q2. Если точка

принадлежит прямоугольнику, то следует сообщить, какому именно

прямоугольнику, если принадлежит двум прямоугольникам, то и этот

факт должен быть отражен в результирующем сообщении. Если точка

принадлежит границе прямоугольника, то и это должно быть

отображено в сообщении.

**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 App2

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

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

{

int x = Convert.ToInt32(textBox1.Text);

int y = Convert.ToInt32(textBox2.Text);

int q1x1 = 2;

int q1y1 = 3;

int q1x2 = 7;

int q1y2 = 6;

int q2x1 = 4;

int q2y1 = 1;

int q2x2 = 9;

int q2y2 = 4;

bool inQ1 = (x >= q1x1 && x <= q1x2 && y >= q1y1 && y <= q1y2);

bool inQ2 = (x >= q2x1 && x <= q2x2 && y >= q2y1 && y <= q2y2);

if (inQ1 && inQ2)

{

textBox3.Text = $"Точка A({x}, {y}) принадлежит и прямоугольнику Q1 и прямоугольнику Q2";

}

else if (inQ1)

{

textBox3.Text = $"Точка A({x}, {y}) принадлежит прямоугольнику Q1";

}

else if (inQ2)

{

textBox3.Text = $"Точка A({x}, {y}) принадлежит прямоугольнику Q2";

}

else

{

textBox3.Text = $"Точка A({x}, {y}) не принадлежит ни одному из прямоугольников Q1 и Q2";

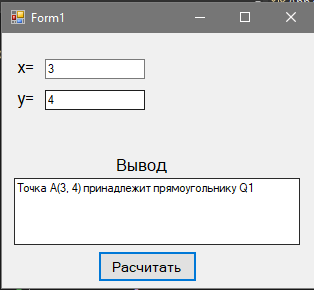
}

}

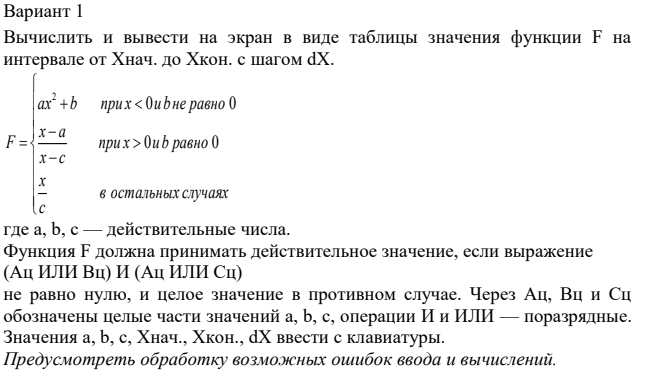
}

}

**Результат:**

****

**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 App3

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

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

{

try

{

int a = Convert.ToInt32(textBox1.Text);

int b = Convert.ToInt32(textBox2.Text);

int c = Convert.ToInt32(textBox3.Text);

double xStart = Convert.ToDouble(textBox4.Text);

double xEnd = Convert.ToDouble(textBox5.Text);

double dX = Convert.ToDouble(textBox6.Text);

textBox7.Text = " x | F(x) ";

textBox7.Text = "------------";

for (double x = xStart; x <= xEnd; x += dX)

{

double result;

int aInt = (int)a;

int bInt = (int)b;

int cInt = (int)c;

if (((aInt & bInt) != 0) || ((aInt & cInt) != 0))

{

result = a \* Math.Pow(x, 2) + b;

}

else if ((x > 0) && (b == 0))

{

result = (x - a) / (x - c);

}

else

{

result = x / c;

}

if (result % 1 == 0)

{

textBox7.Text = $"{x} | {result:F0}";

}

else

{

textBox7.Text = $"{x} | {result:F2}";

}

}

}

catch (Exception ex)

{

textBox7.Text = $"Ошибка: {ex.Message}";

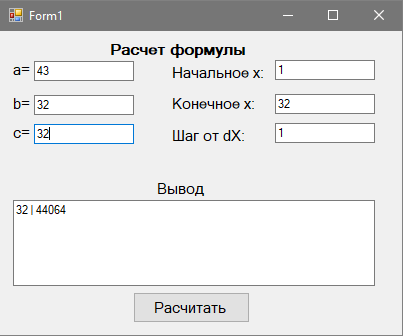
}

}

}

}

**Результат:**

****

**Вывод:** после изучения темы "Процедуры и функции. Перегрузка функций (методов) и операторов" я усвоил следующие основные концепции:

1. Процедуры и функции — это основные инструменты структурирования кода, которые позволяют разделить код на более мелкие и логически связанные части, упрощая его понимание, тестирование и обслуживание.
2. Процедуры — это набор инструкций, выполняющих определенную операцию, но не возвращающих результат. Они могут быть вызваны из других частей программы и могут иметь параметры, передаваемые при вызове.
3. Функции — это набор инструкций, возвращающих результат выполнения операции. Они могут быть вызваны из других частей программы, и после выполнения возвращают значение, которое может быть использовано далее в программе.
4. Перегрузка функций (методов) и операторов — это возможность определить несколько функций (методов) или операторов с одинаковым именем, но с разными параметрами или входными аргументами. Компилятор или интерпретатор определяют, какую функцию (метод) или оператор использовать на основе типов аргументов, переданных при вызове.

Понимание этих основных концепций позволяет более эффективно использовать процедуры, функции и перегрузку функций (методов) и операторов в своих программах, создавая более структурированный, гибкий и поддерживаемый код.