Федеральное государственное автономное

образовательное учреждение

высшего образования

«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт Космических и информационных технологий

институт

Кафедра «Информатика»

кафедра

**ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №1**

Основные конструкции языка C#

тема

Вариант 14

Преподаватель \_\_\_\_\_\_\_\_\_\_\_ А.А. Чикизов

подпись, дата инициалы, фамилия

Студент КИ15–16Б, 031510065 \_\_\_\_\_\_\_\_\_\_\_ Т.В. Радионов

номер группы, зачетной книжки подпись, дата инициалы, фамилия

Красноярск 2017

# Задание

Вычислить и вывести на экран в виде таблицы значения функции, заданной с помощью ряда Тейлора, на интервале от Хнач до Хкон с шагом dx с точностью e. Таблицу снабдить заголовком и шапкой. Каждая строка таблицы должна содержать значение аргумента, значение функции и количество просуммированных членов ряда.

# Исходный код

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 WindowsFormsApp1

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

private int Factorial(int n)

{

int result = 1;

int i = 1;

while (i <= n)

{

result = result \* i;

i++;

}

return result;

}

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

{

double x\_start = Convert.ToDouble(maskedTextBoxXStart.Text); // х начальное

double x\_end = Convert.ToDouble(maskedTextBoxXEnd.Text); // x конечное

double dx = Convert.ToDouble(maskedTextBoxDX.Text); // шаг

double accuracy = Convert.ToDouble(maskedTextBoxAccuracy.Text); // точность

double result = 0;

int counter = 1; // счетчик

// Очистка данных и создание скелета dataGridView1

dataGridViewResults.Rows.Clear();

dataGridViewResults.Columns.Clear();

dataGridViewResults.Columns.Add("x", "X");

dataGridViewResults.Columns.Add("f", "F(X)");

dataGridViewResults.Columns.Add("result", "Результат");

dataGridViewResults.Columns.Add("n", "Кол-во членов ряда");

dataGridViewResults.Rows.Add();

dataGridViewResults.Rows[0].Height = 20;

// Вычисление функции Тейлора

int row = 0;

for (double x = x\_start; x <= x\_end; x = x + dx)

{

double prev\_f;

double f = 0;

int n = 0;

do

{

prev\_f = f;

f = (Math.Pow(-1, n) \* Math.Pow(x, 2 \* n)) / Factorial(2 \* n);

result += f;

counter++;

n++;

} while (Math.Abs(f - prev\_f) > accuracy && n < 10);

dataGridViewResults.Rows[row].Cells[0].Value = x.ToString();

dataGridViewResults.Rows[row].Cells[1].Value = f.ToString();

dataGridViewResults.Rows[row].Cells[2].Value = result.ToString();

dataGridViewResults.Rows[row].Cells[3].Value = n.ToString();

dataGridViewResults.Rows.Add();

row++;

}

}

private void Form1\_Load(object sender, EventArgs e)

{

maskedTextBoxXStart.Text = Convert.ToString(1);

maskedTextBoxXEnd.Text = Convert.ToString(3);

maskedTextBoxDX.Text = Convert.ToString(1);

maskedTextBoxAccuracy.Text = Convert.ToString(0.01);

}

}

}

# Результы

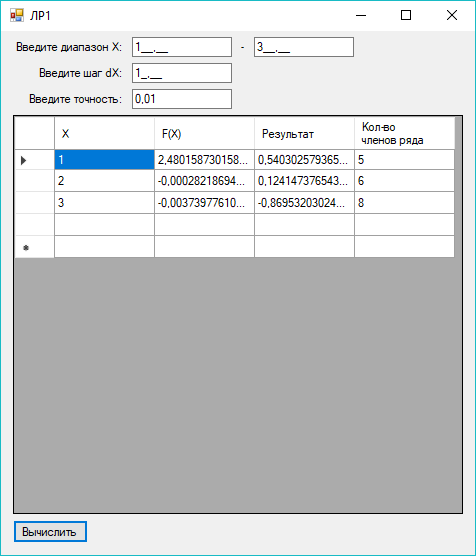


Рисунок 1 – Пример работы программы

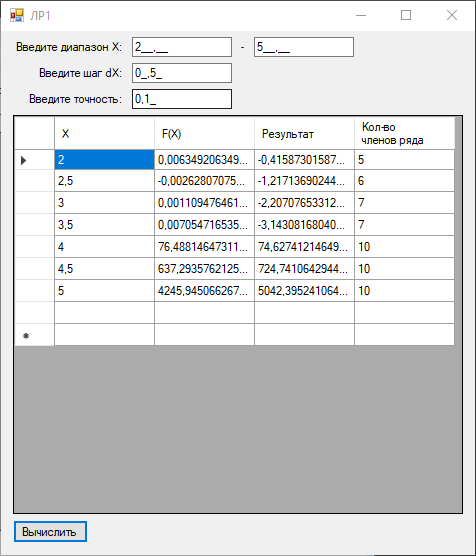


Рисунок 2 – Пример работы программы

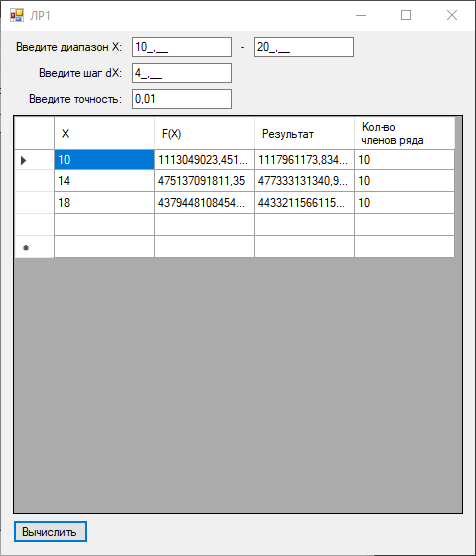


Рисунок 3 – Пример работы программы