МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

**«МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

**(МОСКОВСКИЙ ПОЛИТЕХ)**

09.03.01 Информатика и вычислительная техника

Образовательная программа (профиль)

«Интеграция и программирование в САПР»

Кафедра «СМАРТ технологии»

ОТЧЕТ

по дисциплине:

**Проектная деятельность**

на тему:

Решение задач методом численного интегрирования (метод Чебышева)

Преподаватель: / Толстиков А.В., к.т.н. /

*подпись ФИО, уч. Звание и степень*

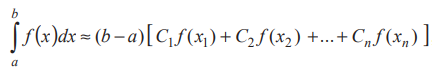
Студент: / Сергеев О. П., 201-323 /

*подпись ФИО, группа*

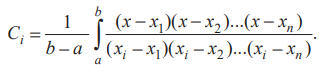
Москва, 2022 г.

**ТЕОРЕТИЧЕСКАЯ ЧАСТЬ**

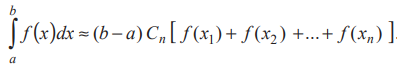
Интерполяционная формула Лагранжа дает возможность составить множество приближенных формул вида



где x1, x2, …, xn числа между a и b . Действительно, задав узлы интерполяции на отрезке [a; b] произвольным образом и заменив подынтегральную функцию полиномом Лагранжа степени n, получим (действуя так же, как при выводе формулы Котеса) для вычисления Сi формулу

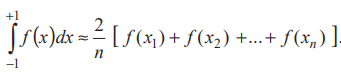


Однако вычислять коэффициенты Ci по этой формуле трудно. Чебышев поставил обратную задачу: задать не узлы x1, x2, …, xn, а коэффициенты Ci и искать соответствующие им узлы. При этом коэффициенты Ci задаются так, чтобы формула была как можно проще для вычислений, а это будет тогда, когда они все равны: C1 = C2 = C3 = …= Cn, тогда формула примет вид



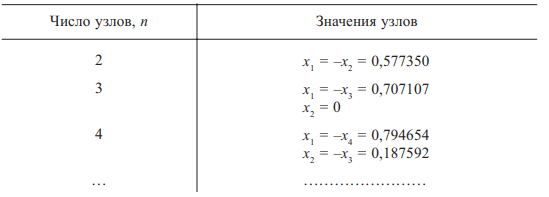
где

Таким образом, определен коэффициент Cn и узлы x1, x2, …, xn квадратурной формулы Чебышева

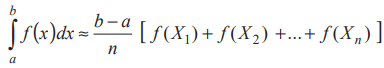


Составлены таблицы значений узлов при различных количествах узлов n в формуле Чебышева

Таблица 1 - Вычисленные узлы в формуле Чебышева



Если пределы интегрирования a и b, формула Чебышева принимает вид



где



а xi берется из таблицы.

**ПРАКТИЧЕСКАЯ ЧАСТЬ**

По теоритическому материалу напишем программу на языке программирования C# с использованием API для графического интерфейса Windows Forms и интегрированной среды разработки Visual Studio 2022.

Перед началом работы нам нужно создать проект Windows Forms (.NET). Выбираем из списка шаблонов проектов Приложение Windows Forms.

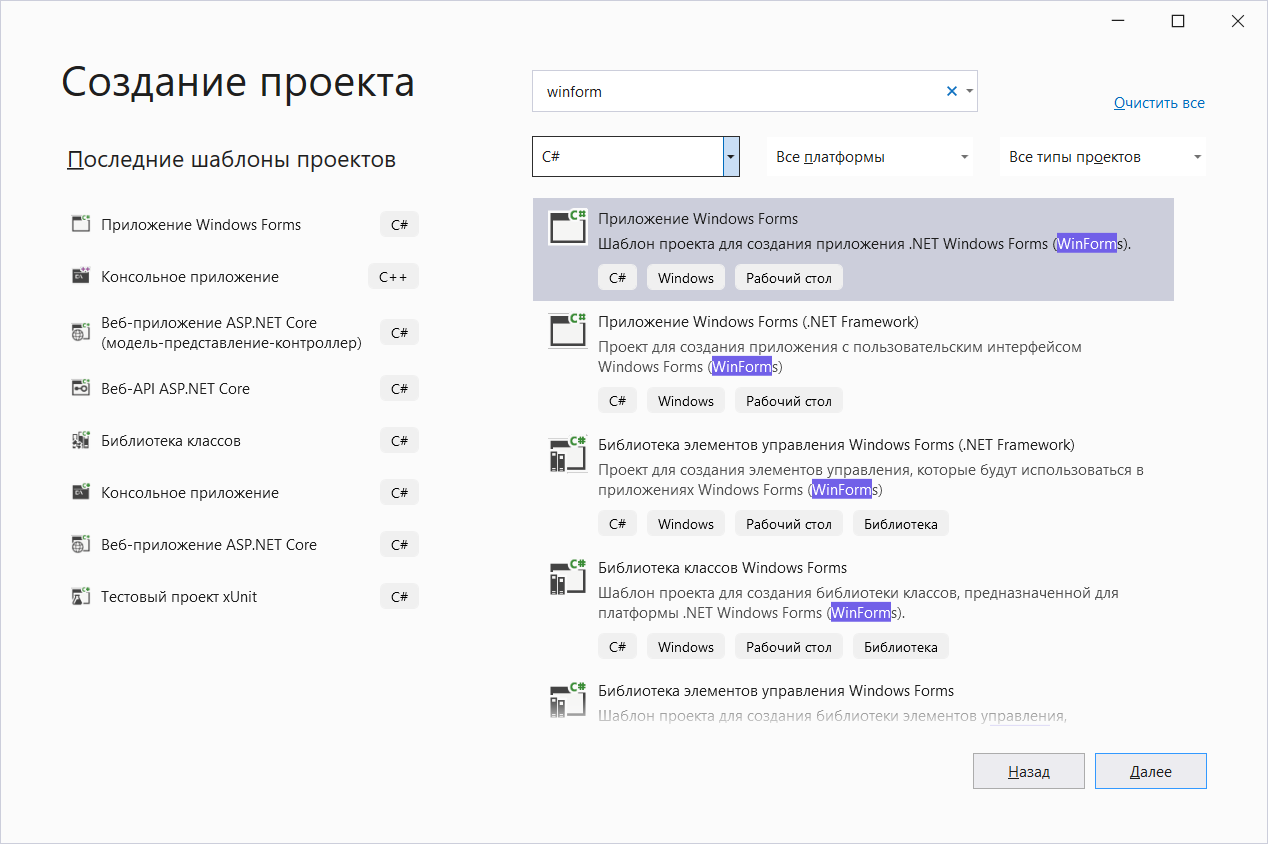


Рисунок 1 - Окно выбора шаблона проекта

Далее указываем имя проекту и расположение проекта и создаем сам проект.

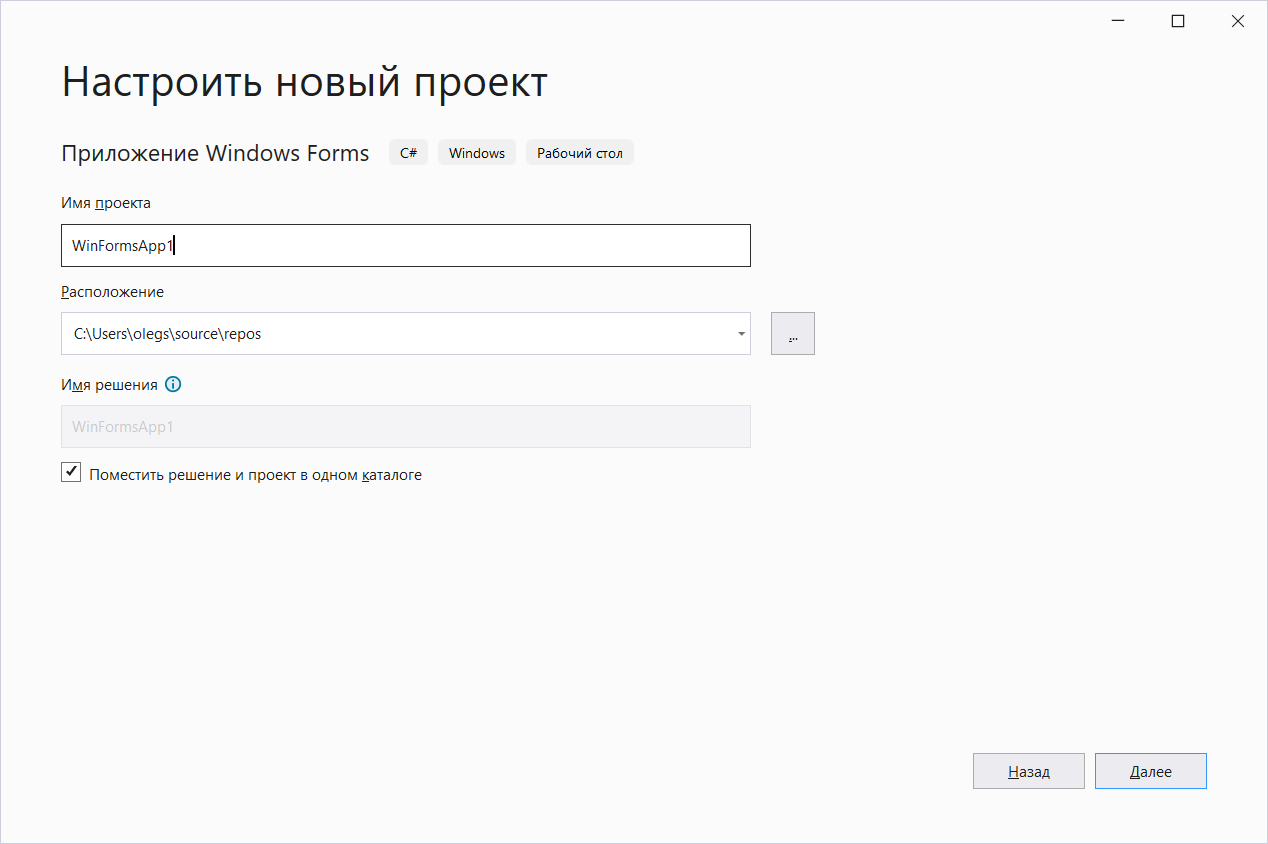


Рисунок 2 - Окно настройки нового проекта

В конструкторе форм с помощью вкладки «Панель элементов» на левой панели инструментов построим интерфейс приложения. В примере использовались такие элементы как Button, TextBox, Label, PictureBox, ComboBox.

Желательно давать осмысленные названия элементам интерфейса, чтобы в коде программы можно было быстрее и легче найти нужный элемент.

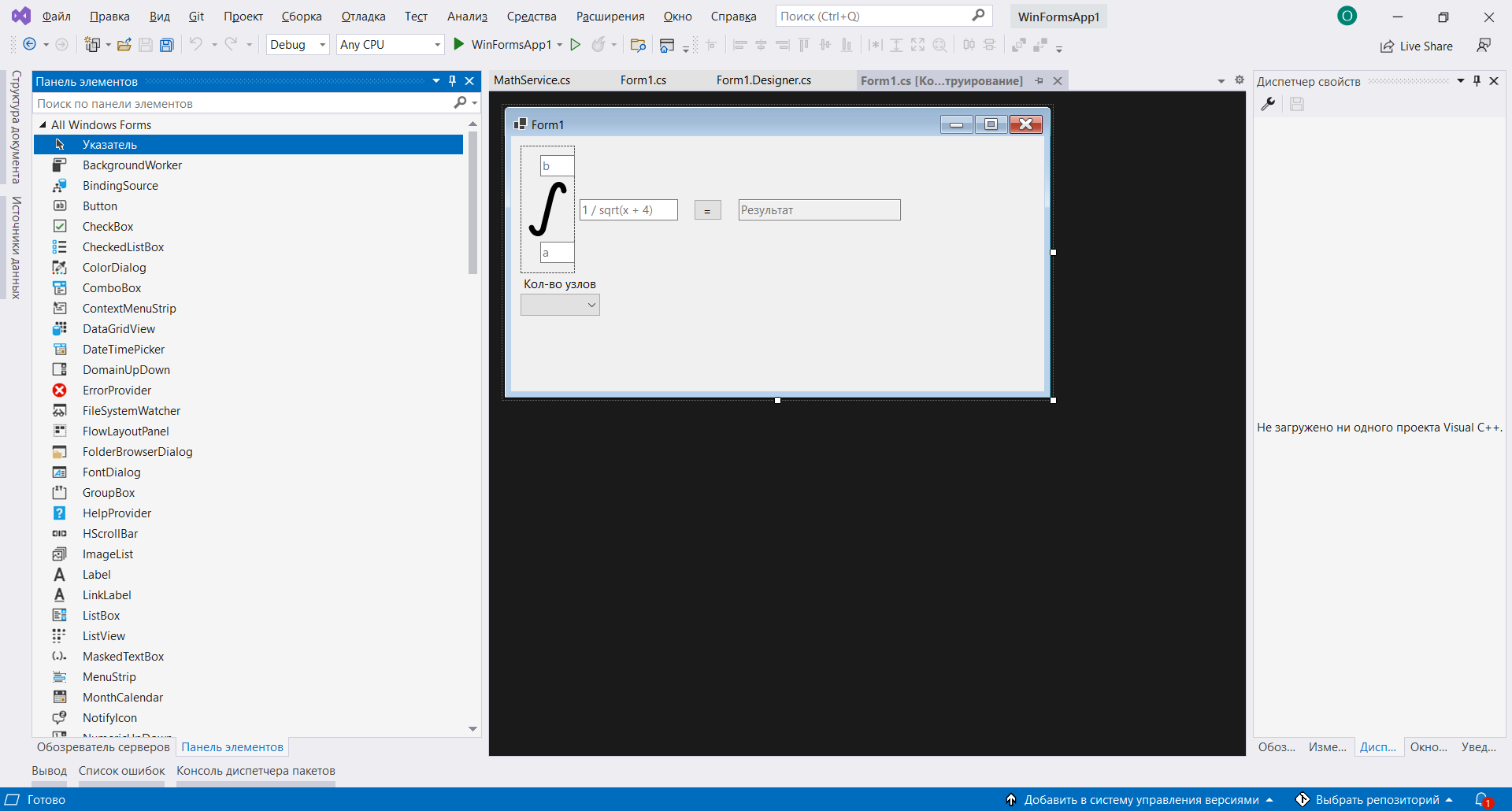


Рисунок 3 - Спроектированный интерфейс приложения

Нажимаем 2 раза по ранее добавленной кнопке для создания события «Button\_Calculate\_Click», также дополнительно создадим событие «Form\_FormClosed», найдя это событие в «Диспетчере свойств» справа, для корректного завершения работы приложения при выхода пользователя из него.

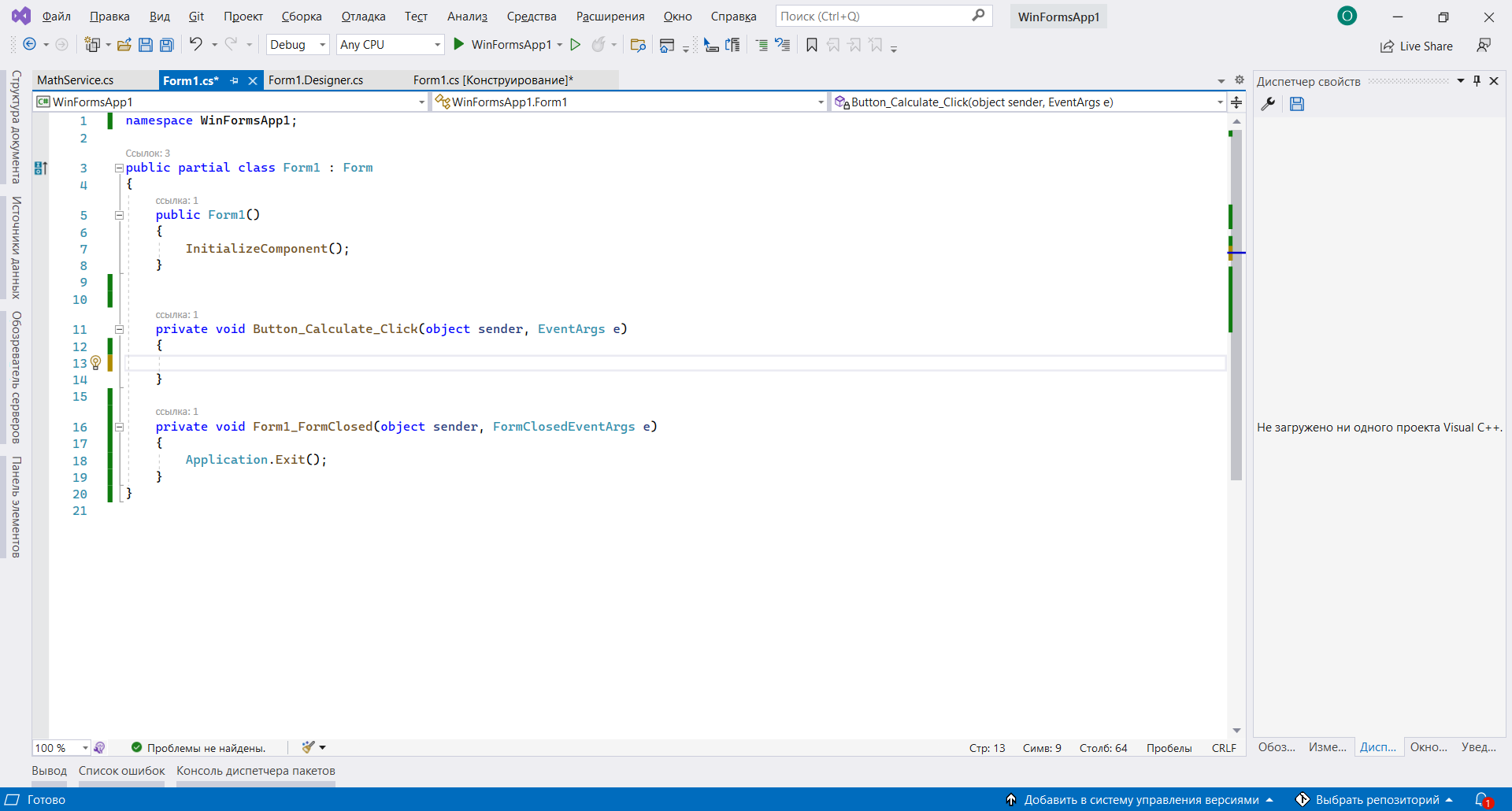
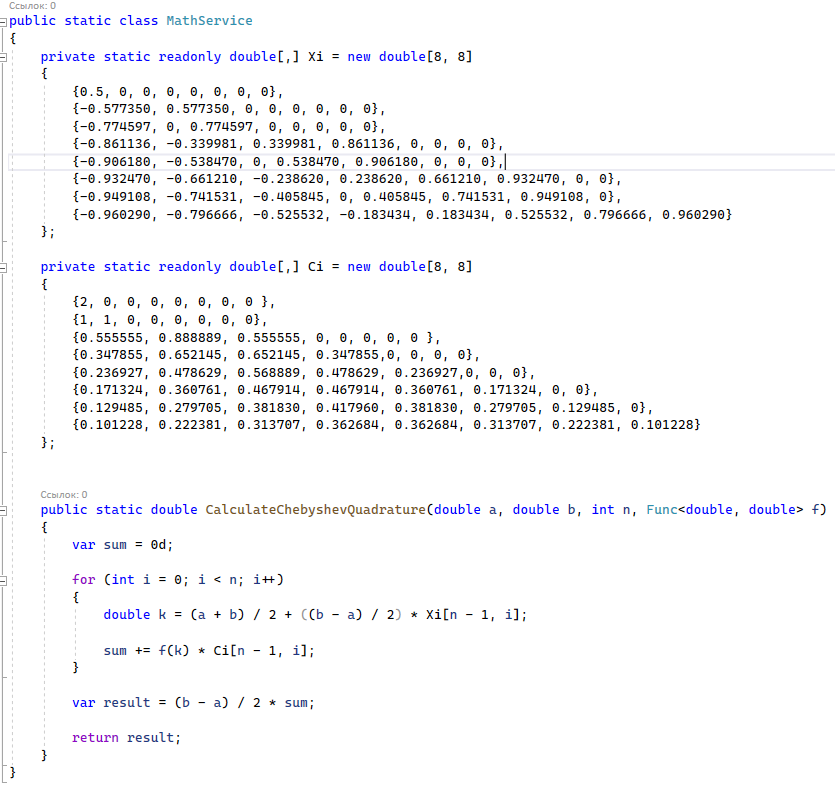


Рисунок 4 - Добавленные события

Создадим вспомогательный класс MathService, который нужен будет для непосредственной реализации метода Чебышева, и реализуем сам метод.

Вынесем заранее известные коэффициенты в 2 массива Xi и Ci для удобства, а также создадим метод «CalculateChebyshevQuadrature», который мы будем вызывать для получения результата вычислений

Листинг 1 - Реализация метода Чебышева



Теперь вернемся в файл Form.cs и добавим логику в метод нажатия кнопки. Чтобы программа работала корректно, мы добавляем условия на корректность введенных пользователем нижнего и верхнего предела интегрирования, также проверяем условие, что нижний предел меньше верхнего, и не забудем про корректность числа узлов (n).

После того, как все условия выполнены, мы вызываем метод CalculateChebyshevQuadrature и выводим результат на экран.

Листинг 2 - Проверка введенных данных и вызов метода расчета результата



Убедимся, что наш метод работает. В качестве примера взяты следующие данные:

, a = 0, b = 5, n = 8

Результат должен быть равен двум. Вводим данные в приложение и видим, что результат программы совпадает с предполагаемым результатом, значит программа работает корректно.

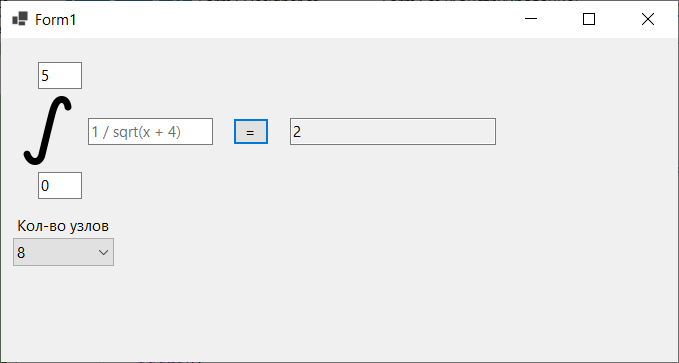


Рисунок 5 - Работа приложения

**РЕЗУЛЬТАТЫ**

В результате работы было спроектировано и реализовано приложение по решению задач методом численного интегрирования Чебышева. Работоспособность приложения была проверена вводом тестовых данных и сверкой их с ожидаемым результатом.

Полный листинг кода представлен в приложениях 1 и 2.

**Приложение 1.**

**Листинг Form.cs**

namespace WinFormsApp1;

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

private void Button\_Calculate\_Click(object sender, EventArgs e)

{

if (!double.TryParse(TextBox\_A.Text, out var a))

{

MessageBox.Show("Укажите нижний предел интеграла", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);

return;

}

if (!double.TryParse(TextBox\_B.Text, out var b))

{

MessageBox.Show("Укажите верхний предел интеграла", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);

return;

}

var accuracy = ComboBox\_Accuracy.SelectedIndex + 1;

if (accuracy <= 0)

{

MessageBox.Show("Укажите количество узлов", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);

return;

}

if (a > b)

MessageBox.Show("Нижний предел интеграла больше верхнего", "Внимание", MessageBoxButtons.OK, MessageBoxIcon.Warning);

Func<double, double> function = (x) => 1f / Math.Sqrt(x + 4);

var result = MathService.CalculateChebyshevQuadrature(a, b, accuracy, function);

TextBox\_Result.Text = result.ToString("0.######");

}

private void Form1\_FormClosed(object sender, FormClosedEventArgs e)

{

Application.Exit();

}

}

**Приложение 2.**

**Листинг MathService.cs**

namespace WinFormsApp1;

public static class MathService

{

private static readonly double[,] Xi = new double[8, 8]

{

{0.5, 0, 0, 0, 0, 0, 0, 0},

{-0.577350, 0.577350, 0, 0, 0, 0, 0, 0},

{-0.774597, 0, 0.774597, 0, 0, 0, 0, 0},

{-0.861136, -0.339981, 0.339981, 0.861136, 0, 0, 0, 0},

{-0.906180, -0.538470, 0, 0.538470, 0.906180, 0, 0, 0},

{-0.932470, -0.661210, -0.238620, 0.238620, 0.661210, 0.932470, 0, 0},

{-0.949108, -0.741531, -0.405845, 0, 0.405845, 0.741531, 0.949108, 0},

{-0.960290, -0.796666, -0.525532, -0.183434, 0.183434, 0.525532, 0.796666, 0.960290}

};

private static readonly double[,] Ci = new double[8, 8]

{

{2, 0, 0, 0, 0, 0, 0, 0 },

{1, 1, 0, 0, 0, 0, 0, 0},

{0.555555, 0.888889, 0.555555, 0, 0, 0, 0, 0 },

{0.347855, 0.652145, 0.652145, 0.347855,0, 0, 0, 0},

{0.236927, 0.478629, 0.568889, 0.478629, 0.236927,0, 0, 0},

{0.171324, 0.360761, 0.467914, 0.467914, 0.360761, 0.171324, 0, 0},

{0.129485, 0.279705, 0.381830, 0.417960, 0.381830, 0.279705, 0.129485, 0},

{0.101228, 0.222381, 0.313707, 0.362684, 0.362684, 0.313707, 0.222381, 0.101228}

};

public static double CalculateChebyshevQuadrature(double a, double b, int n, Func<double, double> f)

{

var sum = 0d;

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

{

double k = (a + b) / 2 + ((b - a) / 2) \* Xi[n - 1, i];

sum += f(k) \* Ci[n - 1, i];

}

var result = (b - a) / 2 \* sum;

return result;

}

}