МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ БЕЛАРУСЬ

БЕЛОРУССКИЙ НАЦИОНАЛЬНЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

***Факультет информационных технологий и робототехники***

Кафедра программного обеспечения информационных систем и технологий

**Отчет по лабораторной работе № 12**

по дисциплине: ”Разработка приложений в визуальных средах ”

# **на тему: *”* Создание приложения, использующего технологию COM-объектов*”***

Вариант 1

Выполнил**:** студент группы 10701322 Бородкин Д.В.

Принял**:** доц. Гурский Н. Н.

Минск 2024

**Лабораторная работа № 12.**

**Цель лабораторной работы:** изучить принципы создания COM-сервера и COM-приложения.

**Задание:** разработать приложение, реализующее вычисление определенного интеграла. Визуальные компоненты, предназначенные для задания исходных данных и вывода результатов, создавать на главной форме динамически.

**Текст программы:**

COMServer:

using System;

using System.Collections.Generic;

using System.Drawing;

using System.Runtime.InteropServices;

using System.Windows.Forms.DataVisualization.Charting;

namespace COMServer

{

[ComVisible(true)]

[InterfaceType(ComInterfaceType.InterfaceIsDual)]

[Guid("3B39D717-A888-4152-86DD-21C213FA91A0")]

public interface IIntegralCalculator

{

// Метод для вычисления интеграла методом трапеций

double Trap();

// Метод для вычисления интеграла методом левых прямоугольников

double Left();

// Метод для вычисления интеграла методом правых прямоугольников

double Right();

// Метод для вычисления интеграла методом средних прямоугольников

double Middle();

}

[ComVisible(true)]

[ProgId("CalcCOM")]

[Guid("D4852481-9898-41FD-8B8B-637824F4AF61")]

[ClassInterface(ClassInterfaceType.None)]

public class IntegralCalculator : IIntegralCalculator

{

//Поля

public double a; // нижняя граница

public double b; // верхняя граница

public double h; // шаг

public double rez; // результат

public Chart chart;

public List<double> X, Y;

//Метод для функции

public double F(double x)

{

return Math.Sin(x) / Math.Sqrt(x);

}

//Метод для трапеций

public double Trap()

{

X = new List<double>();

Y = new List<double>();

double sum = (F(a) + F(b)) / 2.0;

X.Add(a);

Y.Add(F(a));

for (double x = a + h; x < b; x += h)

{

X.Add(x);

Y.Add(F(x));

sum += F(x);

}

rez = sum \* h;

return rez;

}

//Метод для левых прямоугольников

public double Left()

{

X = new List<double>();

Y = new List<double>();

double sum = 0.0;

for (double x = a; x < b; x += h)

{

sum += F(x);

X.Add(x);

Y.Add(F(x));

}

rez = sum \* h;

return rez;

}

//Метод для правых прямоугольников

public double Right()

{

X = new List<double>();

Y = new List<double>();

double sum = 0.0;

for (double x = a + h; x <= b; x += h)

{

sum += F(x);

X.Add(x);

Y.Add(F(x));

}

rez = sum \* h;

return rez;

}

//Метод для средних прямоугольников

public double Middle()

{

double sum = 0.0;

X = new List<double>();

Y = new List<double>();

for (double x = a + h / 2; x <= b; x += h)

{

sum += F(x);

X.Add(x);

Y.Add(F(x));

}

rez = sum \* h;

return rez;

}

// Метод для рисования функции

public void DrawF(SeriesChartType tp)

{

// Очищаем существующие данные на графике

chart.Series.Clear();

// Создание нового объекта класса Series для графика линии

Series series = new Series

{

Name = "График",

ChartType = SeriesChartType.Spline

};

// Установка свойств графика (например, подписей осей)

chart.ChartAreas[0].AxisX.Title = "Ось X";

chart.ChartAreas[0].AxisY.Title = "Ось Y";

// Расчет значения функции для каждой точки

for (double x = a; x <= b; x += h)

{

double y = Math.Sin(x) / Math.Sqrt(x);

series.Points.AddXY(x, y);

}

// Добавление созданного объекта Series на график

chart.Series.Add(series);

chart.Series["График"].Color = Color.Red;

chart.Series["График"].BorderWidth = 3;

// Легенда

chart.Legends["Legend1"].Title = "Легенда";

chart.Legends["Legend1"].Docking = Docking.Right;

// Заголовок

chart.Titles["Title1"].Text = "Графическое изображение функции";

// Создание нового объекта класса Series для графика площади

Series series2 = new Series

{

Name = "Площадь",

ChartType = tp

};

// Заполнение значения функции для каждой точки, которые были рассчитаны при интегрировании

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

{

series2.Points.AddXY(X[i], Y[i]);

}

// Добавление созданного объекта Series на график

chart.Series.Add(series2);

chart.Series["Площадь"].Color = Color.Blue;

chart.Series["Площадь"].BackHatchStyle = ChartHatchStyle.ForwardDiagonal; // Косая штриховка

chart.Series["Площадь"].BorderWidth = 0; // Убрать границы

// Установка ширины столбцов

chart.Series["Площадь"]["PointWidth"] = "1.1";

}

// Метод для рисования левых прямоугольников

public void DrawLeft()

{

DrawF(SeriesChartType.Column);

}

// Метод для рисования правых прямоугольников

public void DrawRight()

{

DrawF(SeriesChartType.Column);

}

// Метод для рисования средних прямоугольников

public void DrawMiddle()

{

DrawF(SeriesChartType.Column);

}

// Метод для рисования трапецией

public void DrawTrap()

{

DrawF(SeriesChartType.Area);

}

}

}  
  
COMClient:

using System;

using System.Windows.Forms;

namespace COMClient

{

public partial class Form1 : Form

{

dynamic CalcCom; // получить указатель на интерфейс

// Конструктор

public Form1()

{

InitializeComponent();

radioButton1.Checked = true;

textBox1.Text = "1";

textBox2.Text = "2";

textBox3.Text = "0,1";

}

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

{

//CalcCom = new COMServer.IntegralCalculator();

var progID = "CalcCOM";

// Получение типа объекта по ProgID

var calcCom = Type.GetTypeFromProgID(progID);

CalcCom = Activator.CreateInstance(calcCom);

}

private void Button1\_Click(object sender, EventArgs e)

{

CalcCom.a = double.TryParse(textBox1.Text, out double aValue) ? aValue : 1;

CalcCom.b = double.TryParse(textBox2.Text, out double bValue) ? bValue : 2;

CalcCom.h = double.TryParse(textBox3.Text, out double hValue) ? hValue : 0.1;

CalcCom.chart = chart1;

RezCalc();

if (radioButton1.Checked)

{

CalcCom.DrawTrap();

}

else if (radioButton2.Checked)

{

CalcCom.DrawLeft();

}

else if (radioButton3.Checked)

{

CalcCom.DrawRight();

}

else if (radioButton4.Checked)

{

CalcCom.DrawMiddle();

}

}

// Метод вывода результатов подсчёта в текст бокс 4

private void RezCalc()

{

double rez = 0;

if (radioButton1.Checked)

{

rez = CalcCom.Trap();

}

else if (radioButton2.Checked)

{

rez = CalcCom.Left();

}

else if (radioButton3.Checked)

{

rez = CalcCom.Right();

}

else if (radioButton4.Checked)

{

rez = CalcCom.Middle();

}

// Результирующая строка для отображения пользовательских данных

string rezText = "Ваши введённые данные: " + Environment.NewLine;

rezText += label1.Text + textBox1.Text + Environment.NewLine;

rezText += label2.Text + textBox2.Text + Environment.NewLine;

rezText += label3.Text + textBox3.Text + Environment.NewLine;

rezText += "Результаты вычисления : " + rez.ToString("F5") + Environment.NewLine;

textBox4.Text = rezText.ToString() + Environment.NewLine;

//Результат в статус строку

toolStripStatusLabel1.Text = "Результаты вычисления : " + rez.ToString("F5");

}

}

}

**Результаты выполнения программы:**

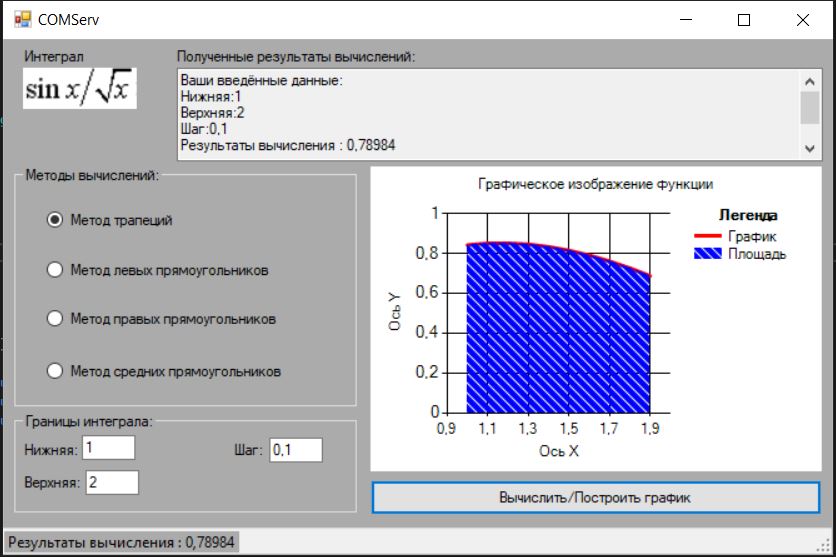


Рисунок 1. Результат работы программы.

**Вывод:** изучил принципы создания COM-сервера и COM-приложения.