Министерство образования Республики Беларусь

Белорусский Национальный Технический Университет

Факультет информационных технологий и робототехники

Кафедра «Программное обеспечение вычислительной техники

и автоматизированных систем»

**Отчёт**

по лабораторной работе № 1

по дисциплине ***«РАЗРАБОТКА ПРИЛОЖЕНИЙ В ВИЗУАЛЬНЫХ СРЕДАХ»***

тема: **«ПРИНЦИПЫ ПОСТРОЕНИЯ МОДУЛЬНОЙ ПРОГРАММЫ»**

|  |  |  |
| --- | --- | --- |
| Исполнитель: |  | студент группы 10701217  Голованов Павел Андреевич |
| Преподаватель: |  | Гурский Николай Николаевич |

2019 учебный год

**ЛАБОРАТОРНАЯ РАБОТА № 1**

**ПРИНЦИПЫ ПОСТРОЕНИЯ МОДУЛЬНОЙ ПРОГРАММЫ**

Цель работы:

Изучить возможности создания приложения в Delphi с разделением интерфейсной и математической задач проекта.

Задание:

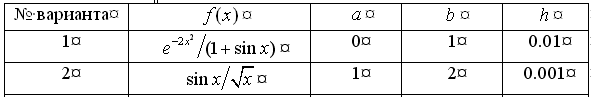
Разработать программу, состоящую из главной формы и отдельного Unit. Главная форма обеспечивает интерфейсную

поддержку. Unit должен содержать набор процедур и функций для вычисления определенного интеграла, графическое

представление подинтегральной функции и геометрическую интерпретацию определенного интеграла. Примерный вид приложения

показан на рис.5.





Результаты выполнения задания:

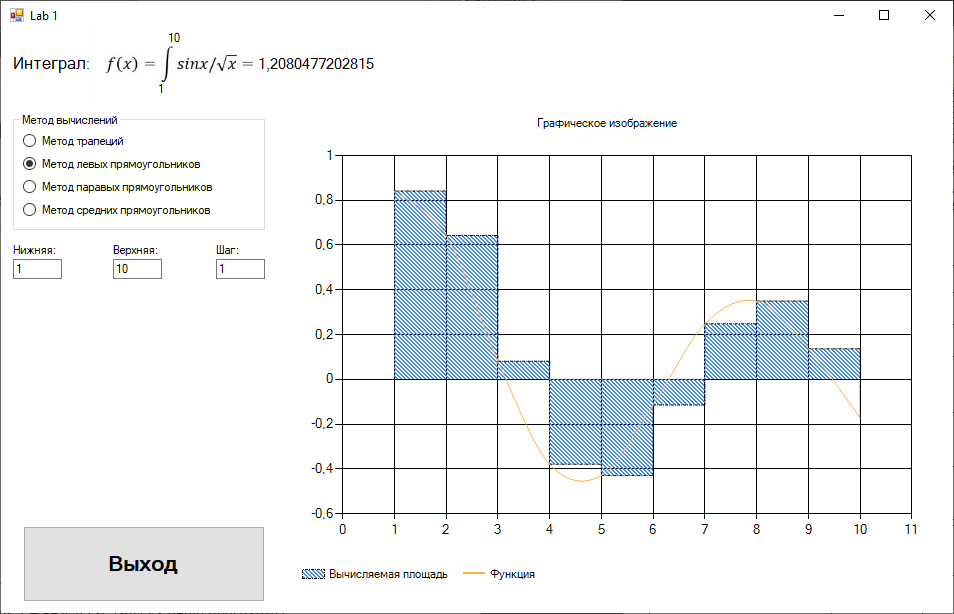


Рисунок 1 – Результат с методом левых прямоугольников.

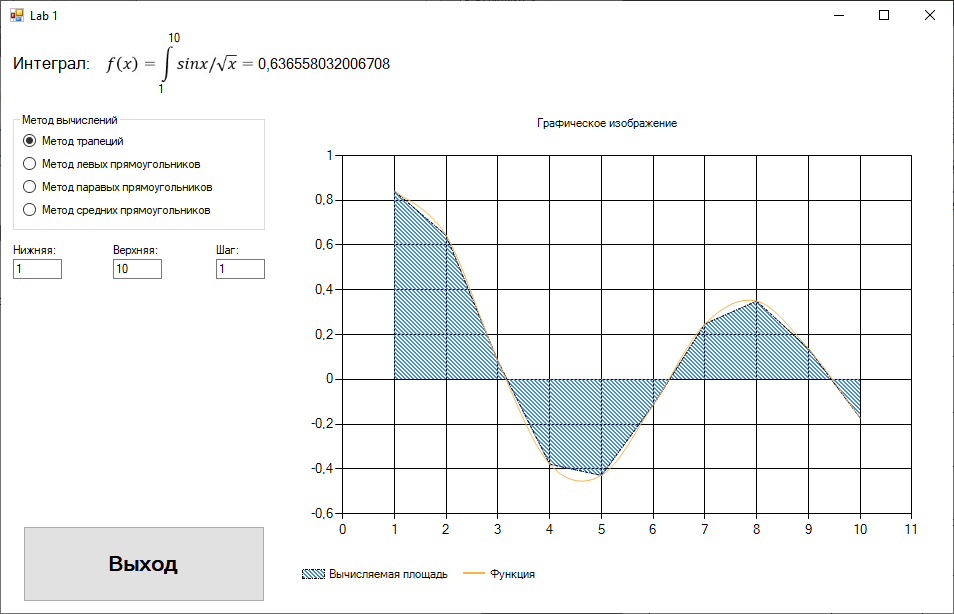


Рисунок 2 – Результат с методом трапеций.

*ПРИЛОЖЕНИЕ A*

Листинг исходных кодов программ

**Файл Lab1Form.cs**

using System;

using System.Windows.Forms;

namespace lab\_1

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

private static Integrals.Mods currentMod = Integrals.Mods.TRAPEZOID\_METHOD;

private void TextBox123\_TextChanged\_Or\_ModChanged(object sender, EventArgs e)

{

label8.Text = "";

label7.Text = textBox1.Text;

label6.Text = textBox2.Text;

double.TryParse(textBox1.Text, out double lim1);

double.TryParse(textBox2.Text, out double lim2);

double.TryParse(textBox3.Text, out double step);

if (lim1 >= lim2)

{

label8.Text = "Нижняя граница больше чем\nверхняя, не удаётся\nпостроить график!";

return;

}

if (step <= 0)

{

label8.Text = "Шаг должен быть больше нуля!";

return;

}

label2.Text = Integrals.CalcAndDraw(lim1, lim2, step, currentMod, ref chart1).ToString();

Integrals.DrawIntegral(ref chart1, lim1, lim2, step);

}

private void RadioButton1\_CheckedChanged(object sender, EventArgs e)

{

currentMod = Integrals.Mods.TRAPEZOID\_METHOD;

TextBox123\_TextChanged\_Or\_ModChanged(sender, e);

}

private void RadioButton2\_CheckedChanged(object sender, EventArgs e)

{

currentMod = Integrals.Mods.LEFT\_RECTANGLE\_METHOD;

TextBox123\_TextChanged\_Or\_ModChanged(sender, e);

}

private void RadioButton3\_CheckedChanged(object sender, EventArgs e)

{

currentMod = Integrals.Mods.RIGHT\_RECTANGLE\_METHOD;

TextBox123\_TextChanged\_Or\_ModChanged(sender, e);

}

private void RadioButton4\_CheckedChanged(object sender, EventArgs e)

{

currentMod = Integrals.Mods.MIDDLE\_RECTANGLE\_METHOD;

TextBox123\_TextChanged\_Or\_ModChanged(sender, e);

}

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

{

Application.Exit();

}

}

}

**Файл Integrals.cs**

using System;

namespace lab\_1

{

class Integrals

{

public enum Mods { LEFT\_RECTANGLE\_METHOD, MIDDLE\_RECTANGLE\_METHOD, RIGHT\_RECTANGLE\_METHOD, TRAPEZOID\_METHOD }

public static double Func(double x)

{

return Math.Sin(x)/Math.Pow(x, 0.5);

}

public static double CalcAndDraw(double lim1, double lim2, double step, Mods mod,

ref System.Windows.Forms.DataVisualization.Charting.Chart chart1)

{

double area = 0;

chart1.Series[0].Points.Clear();

double y;

for (double x = lim1; x <= lim2; x += step)

{

switch (mod)

{

case Mods.LEFT\_RECTANGLE\_METHOD:

{

y = Func(x);

chart1.Series[0].Points.AddXY(x, y);

if (x + step <= lim2)

{

chart1.Series[0].Points.AddXY(x + step, y);

}

area += y \* step; //левых прямоугольников left rectangle

break;

}

case Mods.MIDDLE\_RECTANGLE\_METHOD:

{

y = Func(x + step / 2);

chart1.Series[0].Points.AddXY(x, y);

if (x + step <= lim2)

{

chart1.Series[0].Points.AddXY(x + step, y);

}

area += y \* step; //средних прямоугольников middle rectangle

break;

}

case Mods.RIGHT\_RECTANGLE\_METHOD:

{

y = Func(x + step);

chart1.Series[0].Points.AddXY(x, y);

if (x + step <= lim2)

{

chart1.Series[0].Points.AddXY(x + step, y);

}

area += y \* step; //правых прямоугольников right rectangle

break;

}

case Mods.TRAPEZOID\_METHOD:

{

y = Func(x);

chart1.Series[0].Points.AddXY(x, y);

area += step\*(Func(x) + Func(x + step)) / 2; //трапеций trapezoid

break;

}

}

}

return area;

}

public static void DrawIntegral(ref System.Windows.Forms.DataVisualization.Charting.Chart chart1,

double lim1, double lim2, double step)

{

int counter = 0;

chart1.Series[1].Points.Clear();

for (double i = lim1; i <= lim2 && counter < short.MaxValue; i += step)

{

double x = i;

double y = Func(i);

chart1.Series[1].Points.AddXY(x, y);

counter++;

}

}

}

}