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 № 2

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

тема: **«СОЗДАНИЕ ПРИЛОЖЕНИЯ, ПОДДЕРЖИВАЮЩЕГО**

**ТЕХНОЛОГИЮ ООП»**

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

2019 учебный год

**ЛАБОРАТОРНАЯ РАБОТА № 3**

**СОЗДАНИЕ ПРИЛОЖЕНИЯ, ПОДДЕРЖИВАЮЩЕГО**

**ТЕХНОЛОГИЮ ООП**

Цель работы:

Изучить принципы построения классов.

Задание:

Разработать класс вычисления определенного интеграла четырьмя

методами (методами левых, правых, средних прямоугольников и

трапеций). В классе предусмотреть методы графического отображения

подинтегральной функции на расширенном интервале и геометрического

смысла интеграла.

Разработать приложение в соответствии с л.р. №1 и п.2.

Результаты выполнения задания:

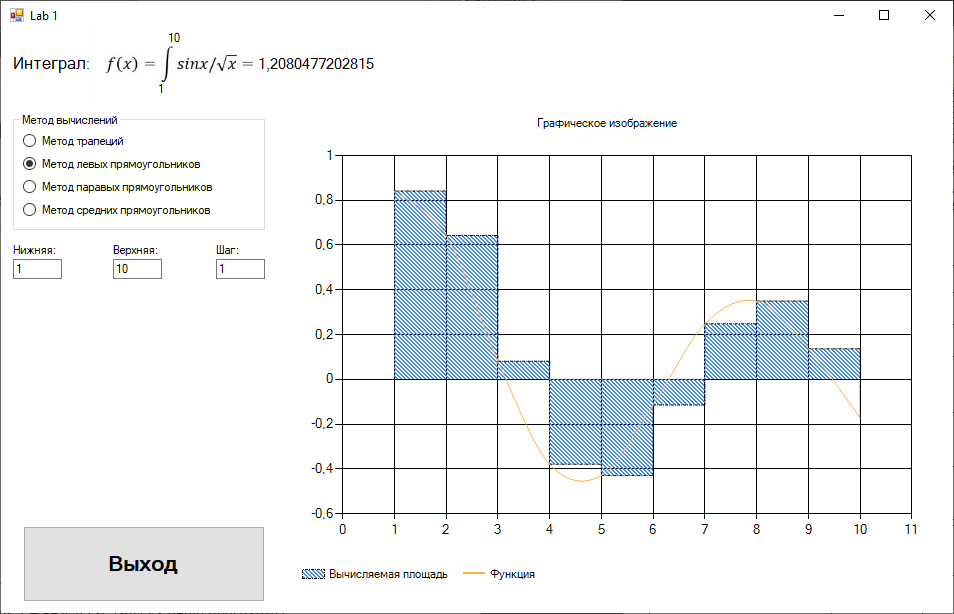


Рисунок 1 – Результат с методом левых прямоугольников.

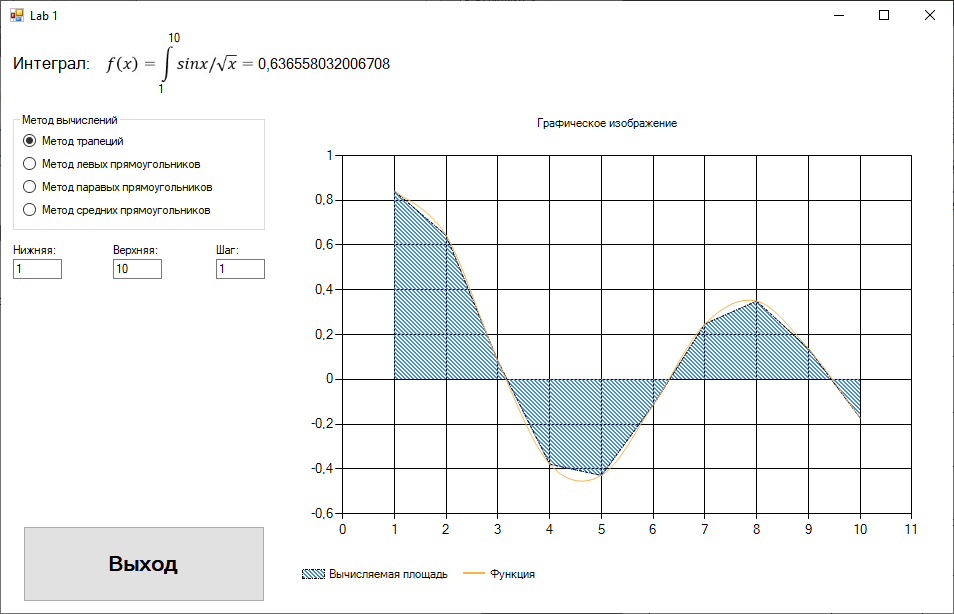


Рисунок 2 – Результат с методом трапеций.

*ПРИЛОЖЕНИЕ A*

Листинг исходных кодов программ

**Файл Lab2Form.cs**

using System;

using System.Windows.Forms;

namespace lab\_2

{

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;

}

System.Windows.Forms.DataVisualization.Charting.Series series;

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

Integrals integral = new Integrals

{

Lim1 = lim1,

Lim2 = lim2,

Step = step

};

switch (currentMod)

{

case Integrals.Mods.LEFT\_RECTANGLE\_METHOD:

{

label2.Text = integral.CalcLeftRectanglesSquare().ToString();

series = integral.CalcLeftRectanglesDots();

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

{

chart1.Series[0].Points.Add(series.Points[i]);

}

break;

}

case Integrals.Mods.MIDDLE\_RECTANGLE\_METHOD:

{

label2.Text = integral.CalcMiddleRectanglesSquare().ToString();

series = integral.CalcMiddleRectanglesDots();

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

{

chart1.Series[0].Points.Add(series.Points[i]);

}

break;

}

case Integrals.Mods.RIGHT\_RECTANGLE\_METHOD:

{

label2.Text = integral.CalcRightRectanglesSquare().ToString();

series = integral.CalcRightRectanglesDots();

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

{

chart1.Series[0].Points.Add(series.Points[i]);

}

break;

}

case Integrals.Mods.TRAPEZOID\_METHOD:

{

label2.Text = integral.CalcTrapezoidSquare().ToString();

series = integral.CalcTrapezoidDots();

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

{

chart1.Series[0].Points.Add(series.Points[i]);

}

break;

}

}

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

series = integral.CalcIntegralDots();

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

{

chart1.Series[1].Points.Add(series.Points[i]);

}

}

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;

using System.Windows.Forms.DataVisualization.Charting;

namespace lab\_2

{

class Integrals

{

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

public double lim1; //lower limit of integral

private double lim2; //upper limit of integral

private double step; //step of integral

public double Lim1 { get => lim1; set => lim1 = value; }

public double Lim2 { get => lim2; set => lim2 = value; }

public double Step { get => step; set => step = value; }

public static double Func(double x)

{

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

}

public double CalcLeftRectanglesSquare()

{

double area = 0;

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

{

double y = Func(x);

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

}

return area;

}

public Series CalcLeftRectanglesDots()

{

Series series = new Series();

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

{

double y = Func(x);

series.Points.AddXY(x, y);

if (x + step <= lim2)

{

series.Points.AddXY(x + step, y);

}

}

return series;

}

public double CalcMiddleRectanglesSquare()

{

double area = 0;

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

{

double y = Func(x + step / 2);

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

}

return area;

}

public Series CalcMiddleRectanglesDots()

{

Series series = new Series();

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

{

double y = Func(x + step / 2);

series.Points.AddXY(x, y);

if (x + step <= lim2)

{

series.Points.AddXY(x + step, y);

}

}

return series;

}

public double CalcRightRectanglesSquare()

{

double area = 0;

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

{

double y = Func(x + step);

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

}

return area;

}

public Series CalcRightRectanglesDots()

{

Series series = new Series();

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

{

double y = Func(x + step);

series.Points.AddXY(x, y);

if (x + step <= lim2)

{

series.Points.AddXY(x + step, y);

}

}

return series;

}

public double CalcTrapezoidSquare()

{

double area = 0;

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

{

double y = Func(x);

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

}

return area;

}

public Series CalcTrapezoidDots()

{

Series series = new Series();

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

{

double y = Func(x);

series.Points.AddXY(x, y);

}

return series;

}

public Series CalcIntegralDots()

{

Series series = new Series();

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

{

double x = i;

double y = Func(i);

series.Points.AddXY(x, y);

}

return series;

}

}

}