МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

Учреждение образования «Гомельский государственный технический

университет имени П.О. Сухого»

Факультет автоматизированных и информационных систем

Кафедра «Информационные технологии»

Отчёт по лабораторной работе №4

По дисциплине «Объектно-ориентированное программирование»

**«****Делегаты и события»**

Выполнил:

Студент группы ИТП-21

Коркуц С. И.

Принял:

ассистент Карабчикова Е.А.

Гомель 2020

**Цель работы:** изучить основы делегатов и событий.

**Задание:**

Разработать графическое приложение (WinForm) без использования конструктора. Интерфейс должен генерироваться динамически в коде программы.

События должны добавляться несколькими способами:

– лямбда-выражения;

– методы, реализующие делегат.

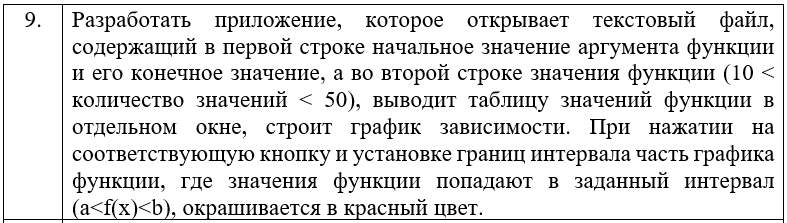


Рисунок 1 – Вариант задания

На рисунке 2 изображена структура решения.

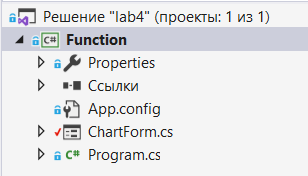


Рисунок 2 – Структура решения

При запуске программы открывается основное меню (Рисунок 3).

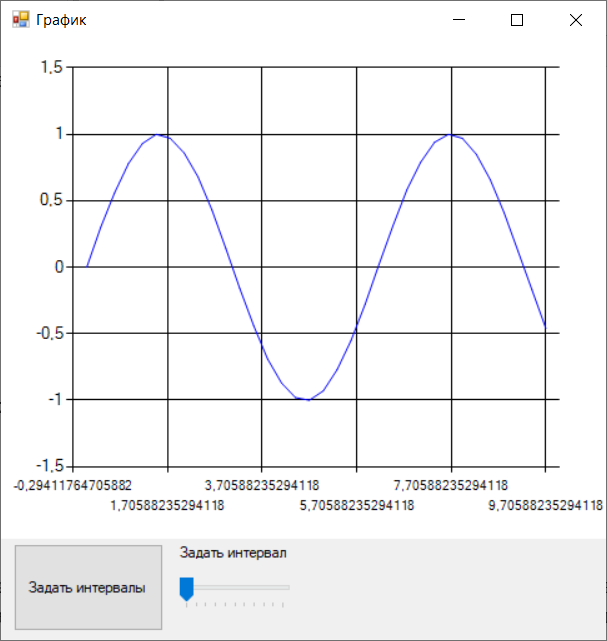


Рисунок 3 – Основное меню программы

При задании интервала и нажатии на кнопку “Задать интервал”, выбранная часть графика окрашивается в красный цвет (рисунок 4).

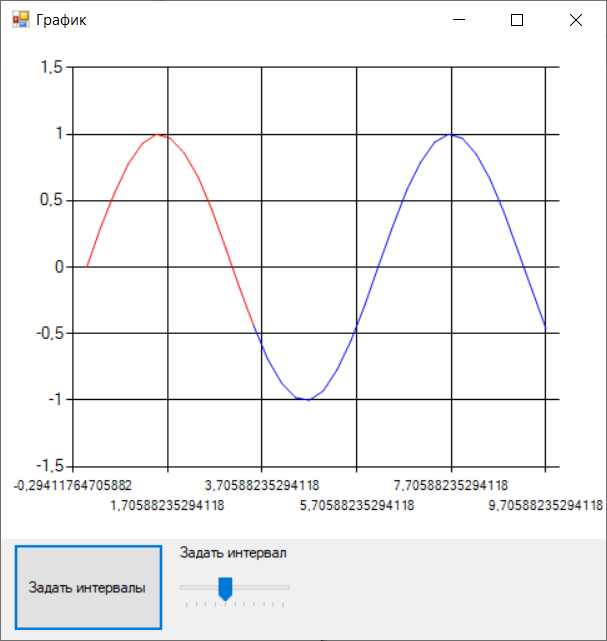


Рисунок 4 – график после задания интервала

**Вывод:** изучены основы синтаксиса объектно-ориентированного языка программирования, делегаты и события.

**ПРИЛОЖЕНИЕ А**

Листинг класса ChartForm.cs

using System;

using System.Windows.Forms.DataVisualization.Charting;

using System.IO;

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 Function

{

public partial class ChartForm : Form

{

double[] yStartStop;

double[] xValues;

Chart myChart = new Chart();

SplitContainer splitContainer = new SplitContainer();

Button button = new Button();

Label tBarLabel = new Label();

TrackBar tBar = new TrackBar();

public ChartForm()

{

ValuesInitialize();

FormInitialize();

ContainerInitialize();

ChartInitialize();

LineInitialize();

InterfaceInitialize();

}

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

{

//Изначально сделать цвет графика голубым

for (int i = 0; i < xValues.Length; i++)

{

myChart.Series["Sinus"].Points[i].Color = Color.Blue;

}

double a, b;

a = tBar.Minimum;

b = tBar.Value;

int count = (int)((b - a) \* xValues.Length / (yStartStop[1] - yStartStop[0]));

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

{

myChart.Series["Sinus"].Points[i].Color = Color.Red;

}

}

void InterfaceInitialize()

{

// Создаю кнопку

button.Parent = this.splitContainer.Panel2;

button.Text = "Задать интервалы";

button.Location = new Point(10, 0);

button.Size = new Size(120, 70);

// Привязываю событие ко кнопке

button.Click += new EventHandler(button\_Click);

// Создаю трэкбар для задания интервалов

tBarLabel.Parent = this.splitContainer.Panel2;

tBarLabel.Text = "Задать интервал";

tBarLabel.Location = new Point(140, 0);

tBar.Parent = this.splitContainer.Panel2;

tBar.Location = new Point(135, 25);

tBar.Maximum = (int)yStartStop[1];

tBar.Minimum = (int)yStartStop[0];

}

void LineInitialize()

{

Series sinusSeries = new Series("Sinus");

sinusSeries.ChartType = SeriesChartType.Line;

sinusSeries.ChartArea = "Sinus";

double value = yStartStop[0];

int i = 0;

while (value <= yStartStop[1])

{

sinusSeries.Points.AddXY(value, xValues[i]);

sinusSeries.Points[i].Color = Color.Blue;

value += yStartStop[1] / xValues.Length;

i += 1;

}

//Добавляем созданный набор точек в Chart

myChart.Series.Add(sinusSeries);

}

void ChartInitialize()

{

myChart.Parent = this.splitContainer.Panel1;

myChart.Dock = DockStyle.Fill;

myChart.ChartAreas.Add(new ChartArea("Sinus"));

}

void ContainerInitialize()

{

splitContainer.Orientation = Orientation.Horizontal;

splitContainer.Size = new Size(500, 500);

splitContainer.SplitterDistance = 400;

this.Controls.Add(splitContainer);

}

void FormInitialize()

{

this.Size = new Size(500, 520);

this.Text = "График";

}

void ValuesInitialize()

{

using (StreamReader sr = new StreamReader(@"D:\korkuts-itp21-oop\lab4\FuncValues.txt"))

{

yStartStop = Array.ConvertAll(sr.ReadLine().Split(' '), Double.Parse);

List<Double> xValuesList = new List<Double>();

String line;

int i = 0;

while ((line = sr.ReadLine()) != null)

{

i++;

xValuesList.Add(Double.Parse(line));

}

xValues = xValuesList.ToArray();

}

}

}

}