

федеральное государственное бюджетное образовательное учреждение высшего образования

«Санкт-Петербургский государственный технологический институт (технический университет)»

**Дисциплина: «Разработка ПС»**

**Отчет по практической работе №3**

**Вариант№9**

# «Знакомство с WinForms»

**Выполнил студент группы №475:**

Овчинников Роман Сергеевич

**Проверили:**

Ст. преп. Федин Алексей Константинович

Доц. Корниенко Иван Григорьевич

Санкт-Петербург

2019

**1. Постановка задачи:**

Необходимо написать приложение с использованием технологии WinForms для построения графика функции и вывода таблицы значений функции. Пользователь задает правую и левую границу, шаг, коэффициенты (при их наличии). При невозможности построить график функции в заданном интервале пользователю выдается предупреждение об этом с предложением сменить границы построения. Если график функции из-за коэффициентов вырождается в точку или не может быть построен пользователь также видит предупреждение.

**2. Исходные данные**

В качестве исходных данных программа использует параметры, вводимые пользователем в поля для ввода

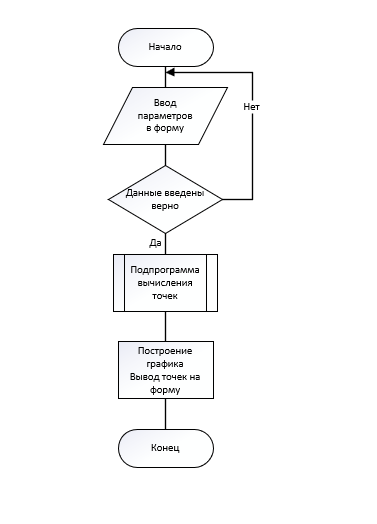
**3. Особые ситуации**

Рассмотреть ситуации, когда график вырождается в точку. Ситуации, связанные с неправильным форматом данных

**4. Методы решения задачи**

В форме был создан график и таблица, с помощью которых и будет построен график, а также выведены точки, по которым график был построен.

Формула для построения графика:

**5. Блок-схема**

*Рисунок 1- Блок-схема*

**6. Форматы представленных данных**

*Таблица 1 - Форматы представления данных*

|  |  |  |
| --- | --- | --- |
| **Имя** | **Тип** | **Описание** |
| A | string | Параметр а |
| Left | string | Левая граница Х |
| Right | string | Правая граница Х |
| X | double[] | Массив значений Х |
| Y | double[] | Массив значений Y |

**7. Структура программы**

Программа состоит из 4 модулей: Main, Calculate, Form1 и Program. Каждый из них выполняет определённую функцию.

Модуль Main точка входа в программу

Модуль Form1 отвечает за отображение информации на форме

Модуль Calculate отвечает за вычисления и работу с внешними источниками.

Модуль Program связывает вычисления и представление через их интерфейсы и выполняет основную работу программы.

View:

*Таблица 2 - Методы интерфейса IView для Form1*

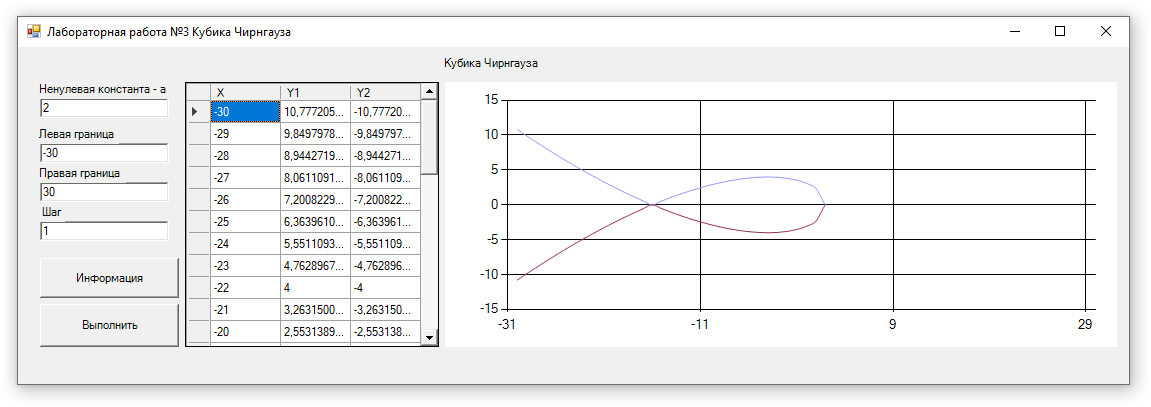
|  |  |  |
| --- | --- | --- |
| Имя члена класса | Тип | Описание |
| WrongData() | void | Выводит сообщение о некорректном вводе |
| WrongInterval() | void | Выводит сообщение о неверном интервале |
| SetChart() | void | Строит график |
| SetDotsList() | void | Выводит точки |
| Start() | void | Запускает приложение |
| DrawChart() | void | Рисовка графика |
| ShowPoints() |  | Показ точек в листе |

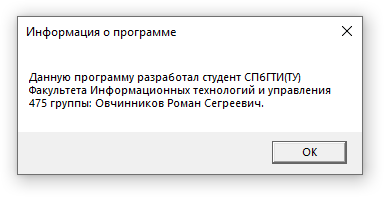
*Таблица 3 - методы класса Program*

|  |  |  |
| --- | --- | --- |
| Имя члена класса | Тип | Описание |
| Run() | void | Начало действий после нажатия на кнопку “Выполнить” |
| GenerateDots | void | Проверка введённых значений и вызов функций находящих координаты точек |
| GenerateX GeneratePositiveY  GenerateNegativeY | List<double> | Генерация точек |

*Таблица 4 - Методы интерфейса ICalculate для Calculate*

|  |  |  |
| --- | --- | --- |
| Имя члена класса | Тип | Описание |
| PositiveEquation | double | Вычисление положительного графика |
| NegativeEquation | double | Вычисление отрицательного графика |

**8. Результаты работы программы**

*****Рисунок 2- Результат работы программы*

*Рисунок 3- Информация о программе*

**9 Исходный текст программы**

[Начало блока IView]

using System;

using System.Collections.Generic;

namespace Lab33.View

{

public interface IView

{

string A { get; }

string MinX { get;}

string MaxX { get; }

string Period { get; }

void DrawChart(List<double> X, List<double> Y1, List<double> Y2);

void ShowPoints(List<double> X, List<double> Y1, List<double> Y2);

void WrongInterval();

void WrongData();

void Start();

event EventHandler buttonClick;

event EventHandler helpClick;

}

}

[Конец блока IView]

[Начало блока Program.cs]

using System;

using System.Collections.Generic;

using Lab33.View;

using Lab33.Service;

namespace Lab33.Programs

{

public class Program

{

private readonly IView view;

private readonly ICalculate model;

public Program(IView view, ICalculate model)

{

this.view = view;

this.model = model;

}

public void Run()

{

view.buttonClick += View\_buttonClick;

view.Start();

}

private void View\_buttonClick(object sender, EventArgs e)

{

GenerateDots();

}

public void GenerateDots()

{

const int maxleft = -100;

const int maxright = 100;

double left = 0, right = 0, a = 0;

double per = 0;

bool isCorrectData = false;

try

{

left = Convert.ToDouble(view.MinX);

right = Convert.ToDouble(view.MaxX);

a = Convert.ToDouble(view.A);

per = Convert.ToDouble(view.Period);

isCorrectData = true;

}

catch(FormatException e)

{

view.WrongData();

}

if((left >= right) ||(left > maxright) || (left < maxleft) || (right > maxright) || (right < maxleft))

{

isCorrectData = false;

view.WrongInterval();

} else if(per <= 0)

{

isCorrectData = false;

view.WrongData();

}

if (isCorrectData)

{

List<double> x = GenerateX(left, right, per);

List<double> positiveY = GeneratePositiveY(left, right, a, per);

List<double> negativeY = GenerateNegativeY(left, right, a, per);

view.DrawChart(x, positiveY, negativeY);

view.ShowPoints(x, positiveY, negativeY);

}

}

public List<double> GenerateX(double left, double right,double period)

{

List<double> X = new List<double>();

for(double i = left; i < right; i+=period)

{

X.Add(i);

}

return X;

}

public List<double> GeneratePositiveY(double left, double right, double a,double period)

{

List<double> Y = new List<double>();

for (double i = left; i < right; i+=period)

{

Y.Add(model.PositiveEquation(a, i));

}

return Y;

}

public List<double> GenerateNegativeY(double left, double right, double a, double period)

{

List<double> Y = new List<double>();

for (double i = left; i < right; i+=period)

{

Y.Add(model.NegativeEquation(a, i));

}

return Y;

}

}

}

[Конец блока Program.cs]

[Начало блока Calculate.cs]

using System;

namespace Lab33.Service

{

public class Calculate: ICalculate

{

public double NegativeEquation(double A, double x)

{

return -1 \* Math.Sqrt((A - x) \* (8 \* A + x) \* (8 \* A + x) / (A \* 27));

}

public double PositiveEquation(double A, double x)

{

return Math.Sqrt((A - x) \* (8 \* A + x) \* (8 \* A + x) / (A \* 27));

}

}

}

[Конец блока Calculate.cs]

[Начало блока Form1.cs]

using System;

using System.Collections.Generic;

using Lab33.View;

using System.Windows.Forms;

namespace Lab33

{

public partial class Form1 : Form,IView

{

public string A { get { return ParamA.Text; } }

public string MinX { get { return LeftBorder.Text; } }

public string MaxX { get { return RightBorder.Text; } }

public string Period { get { return textBox1.Text; } }

public event EventHandler buttonClick;

public event EventHandler helpClick;

public void DrawChart(List<double> X, List<double> Y1, List<double> Y2)

{

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

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

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

{

chart1.Series[0].Points.AddXY(X[i], Y1[i]);

chart1.Series[1].Points.AddXY(X[i], Y2[i]);

}

}

public void ShowPoints(List<double> X, List<double> Y1, List<double> Y2)

{

DotsList.Rows.Clear();

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

{

if (!double.IsNaN(Y1[i]) || !double.IsNaN(Y2[i]))

{

DotsList.Rows.Add(X[i], Y1[i], Y2[i]);

}

}

}

public void WrongData()

{

string message = "Пожалуйста, вводите числа!";

string caption = "Неверный формат данных";

MessageBox.Show(message, caption, MessageBoxButtons.OK);

}

public void WrongInterval()

{

string message = "Неверно указан интервал\nЛевая граница должны быть меньше правой\nВводите числа от -100 до 100";

string caption = "Неверный интервал";

MessageBox.Show(message, caption, MessageBoxButtons.OK);

}

public void Start()

{

this.Show();

}

public new void Show()

{

Application.Run(this);

}

private void button1\_Click(object sender, EventArgs e)

{

buttonClick?.Invoke(this, null);

}

public Form1()

{

InitializeComponent();

}

private void helpToolStripMenuItem\_Click\_1(object sender, EventArgs e)

{

helpClick?.Invoke(this, null);

}

private void Button2\_Click(object sender, EventArgs e)

{

MessageBox.Show("Данную программу разработал студент СПбГТИ(ТУ)\nФакультета Информационных технологий и управления\n475 группы: Овчинников Роман Сегреевич.", "Информация о программе");

}

}

}

[Конец блока Form1.cs]