**Дніпровський національний університет імені Олеся Гончара**

**ФАКУЛЬТЕТ ПРИКЛАДНОЇ МАТЕМАТИКИ**

**КАФЕДРА КОМП'ЮТЕРНИХ ТЕХНОЛОГІЙ**

**Звіт**

**з навчальної практики: комп’ютерно-технічної**

на тему: «Візуальне програмування в середовищі Microsoft Visual Studio»

Студентки 3 курсу групи ПА-20-1з

Мовсісян Лаури Ростомівни

спеціальності: Прикладна математика

керівник: ас. Сірик Світлана Федорівна

Кількість балів\_\_\_\_\_\_\_\_\_\_\_\_

Національна шкала \_\_\_\_\_\_\_\_

Оцінка ECTS \_\_\_\_\_\_\_\_\_\_\_\_\_

Члени комісії: \_\_\_\_\_\_\_\_\_\_\_\_\_в.о. зав. каф. Зайцева Т.А.\_\_\_\_\_

(підпис) (прізвище та ініціали)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_Сірик С.Ф.\_\_\_\_\_..

(підпис) (прізвище та ініціали)

\_\_\_\_\_\_\_\_\_\_\_\_ \_\_Лисиця Н.М.\_

(підпис) (прізвище та ініціали)

Дніпро – 2022р.

**Зміст**

[**Вступ** 3](#_Toc121931805)

[**Постанова задачі** 4](#_Toc121931806)

[**Початок роботи** 5](#_Toc121931807)

[**1 Підготовка середовища програмування** 5](#_Toc121931808)

[**2 Макетування** 7](#_Toc121931809)

[**3 Реалізація графічного інтерфейсу програми** 8](#_Toc121931810)

[**3.1 Панель інтервалу** 14](#_Toc121931811)

[**3.2 Панель вибору функції** 15](#_Toc121931812)

[**3.3 Панель вибору** 15](#_Toc121931813)

[**3.4 Панель графіку** 15](#_Toc121931814)

[**3.5 Кнопка початку роботи програми** 16](#_Toc121931815)

[**4 Етапи розробки графіків** 17](#_Toc121931816)

[***4.1 Додання декількох графіків*** 17](#_Toc121931817)

[***4.2 Стрілки на вісях*** 18](#_Toc121931818)

[***4.3*** ***Початок системи координат*** 19](#_Toc121931819)

[***4.4*** ***Розмітка головних вісей*** 19](#_Toc121931820)

[***4.5 Назва координатних вісей*** 19](#_Toc121931821)

[***4.6 Стиснення та розтягнення*** 19](#_Toc121931822)

[***4.7 Масштабування*** 19](#_Toc121931823)

[**5 Додаткові форми** 20](#_Toc121931824)

[***5.1 Формування файлу*** 20](#_Toc121931825)

[**Додатки** 21](#_Toc121931826)

[**Висновки** 22](#_Toc121931827)

[**Список використаних джерел** 23](#_Toc121931828)

# **Вступ**

З 05.12 по 17.12 2022-го року, я, студентка групи ПА-20-1з Мовсісян Лаура, пройшла навчальну практику з теми «Візуальне програмування в середовищі Microsoft Visual Studio» під керівництвом Сірик Світлани Федорівни.

Під час практики я вчилася працювати з візуальним інтерфейсом програми та фіксувала знання, які я отримала за 3 курс навчання. Моєю метою була розробка дизайну для цієї програми, її програмування та тестування роботи програми.

У якості завдання нашій групі дали реалізувати побудову графіку функції.

В ході роботи кожен мав змогу обрати для свого проекту комфортну для нього мову програмування. Я обрала мову С#, оскільки завдяки неї, легше реалізувати візуальний інтерфейс програми без зайвих налаштувань.

Кожен день, під час проходження практики, ми вели письмовий звіт у форматі щоденника , в який робили записи:

-хода роботи над програмою;

-що було зроблено за день;

-які труднощі виникли.

# **Постанова задачі**

## **Зміст роботи:**

1. Опанувати головні функції візуального програмування.
2. Вивчити головні можливості програмування графіків.
3. Створити програму для виведення на екран грфіка функції, а також програму для підготовки даних (номер варіанта Індивідуального завдання збігається з номером в студентському журналі)
4. Підготувати звіт про успішне виконання завдання для навчальної практики: комп'ютерно – технологічної.

Індивідуальне завдання: варіант 13

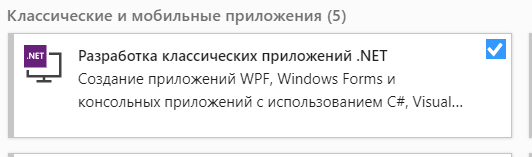
## **Технічні вимоги:**

* побудова графіка
* вісі (розмітка вісей , підписи вісей , оформлення вісей абсциси та ординати)
* можливість виведення кількох графіків фунцкій
* вивести графік функції, заданої в текстовому файлі поточно, тобто значеннями функції в точках
* масштабування
* переміщення графіка (паралельний перенос за віссю ординат і абсцис)
* стиснення / розтягнення щодо осі абсцис / ординат

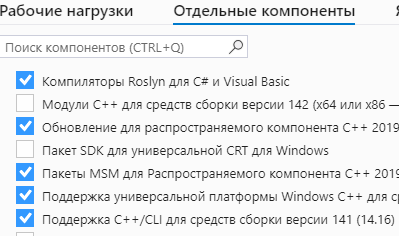
# **Початок роботи**

## **1 Підготовка середовища програмування**

Перед тим, як починати виконувати завдання в середовищі Visual Studio 2019, я мала доповнити підтримку мови програмування С#/CLI. Для цього я запустила VS Installer та завантажила додаткову окрему підтримку ( мал. 1) та відповідні розширення ( мал. 2 ):

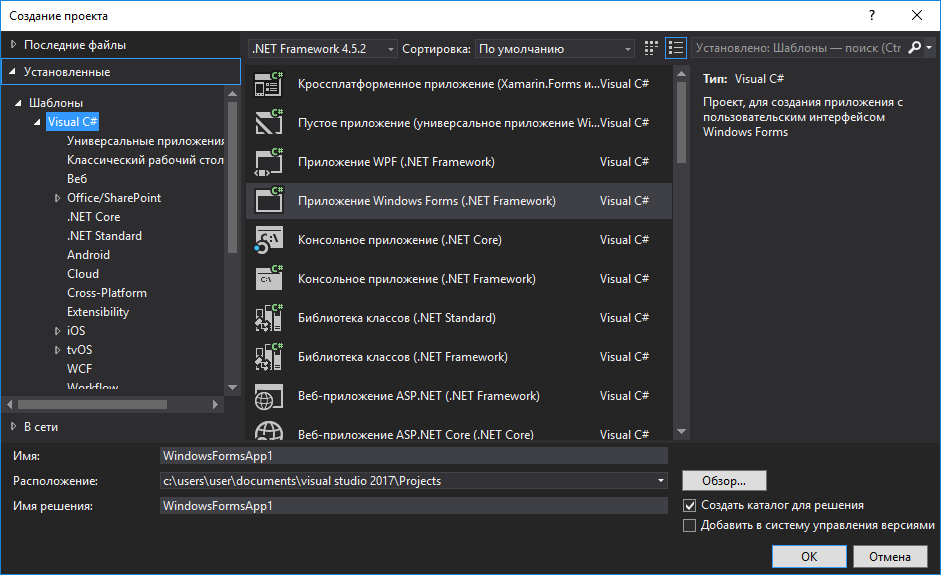


Малюнок 1-окрема підтримка VS Installer



Малюнок 2 - розширення VS Installer

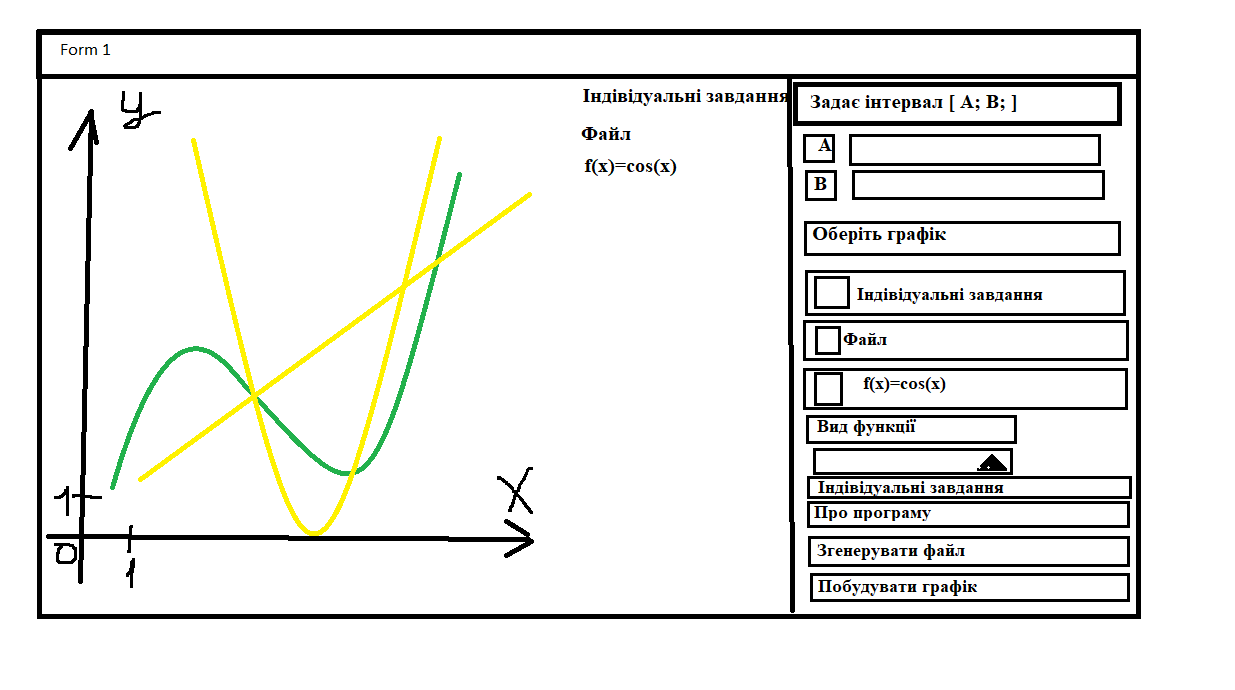
Під час роботи з графічним інтерфейсом в Visual Studio 2019 необхідно при створенні проекту обрати Додаток Windows Form (.NET Framework) ( мал.3).



Малюнок 3 - створення проекту

# **2 Макетування**

**Малюємо макет інтерфейсу майбутньої програми.(мал. 4)**

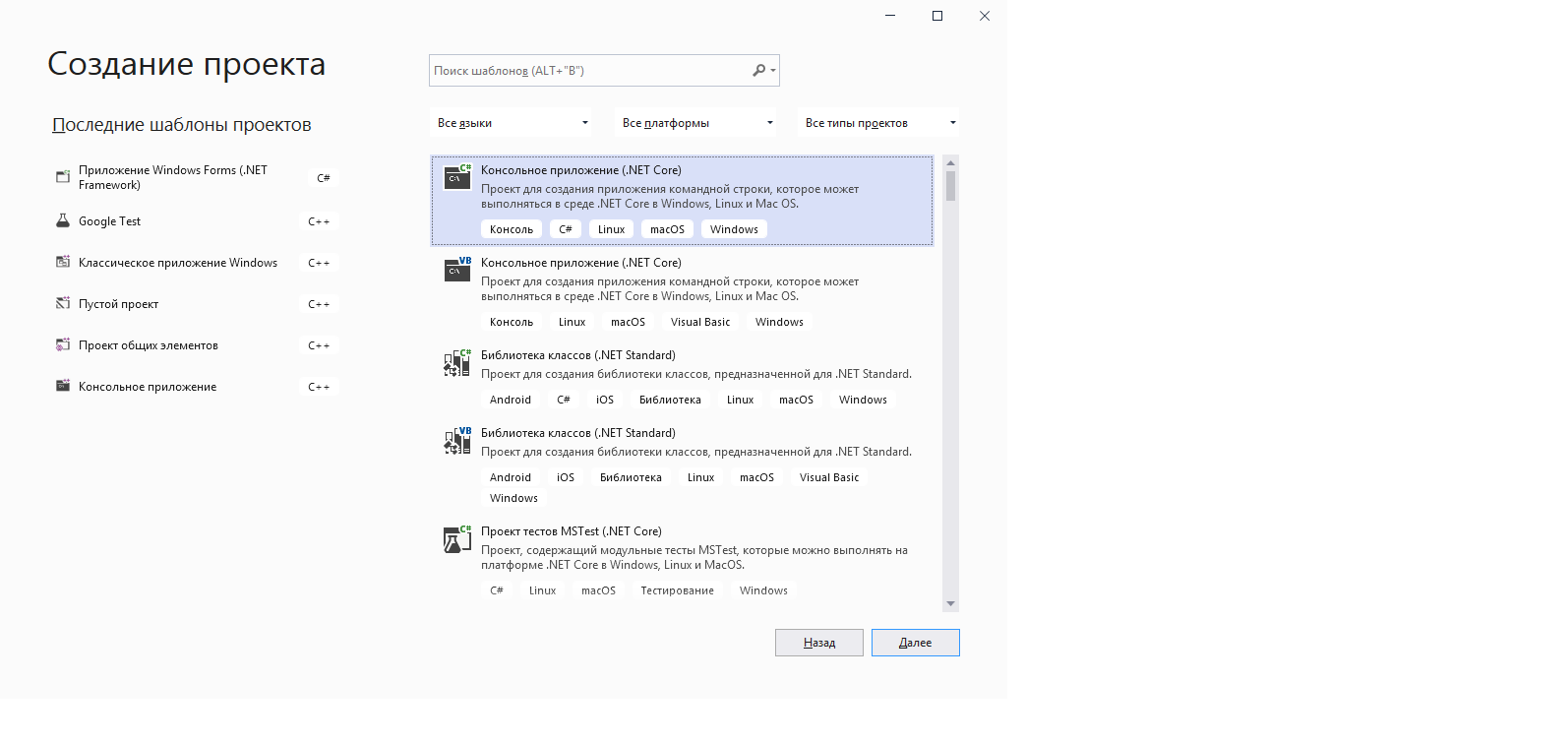


Малюнок 4 - макет інтерфейсу

На макеті зображено усі основні компоненти головного вікна, які буде містити наша програма.

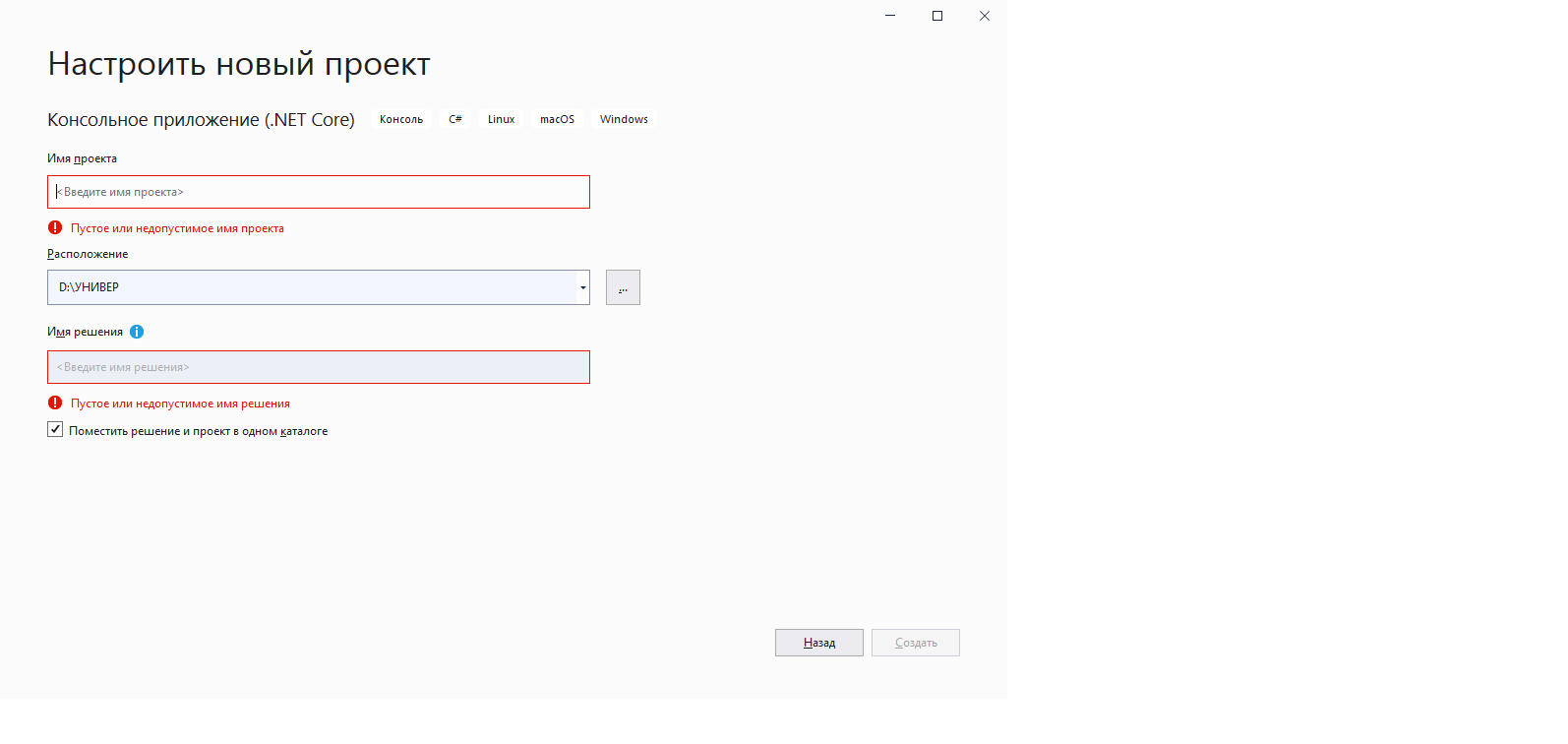
# **3** **Реалізація графічного інтерфейсу програми**

Запускаємо VS та створюємо нову Windows Forms (мал. 5) (C#):



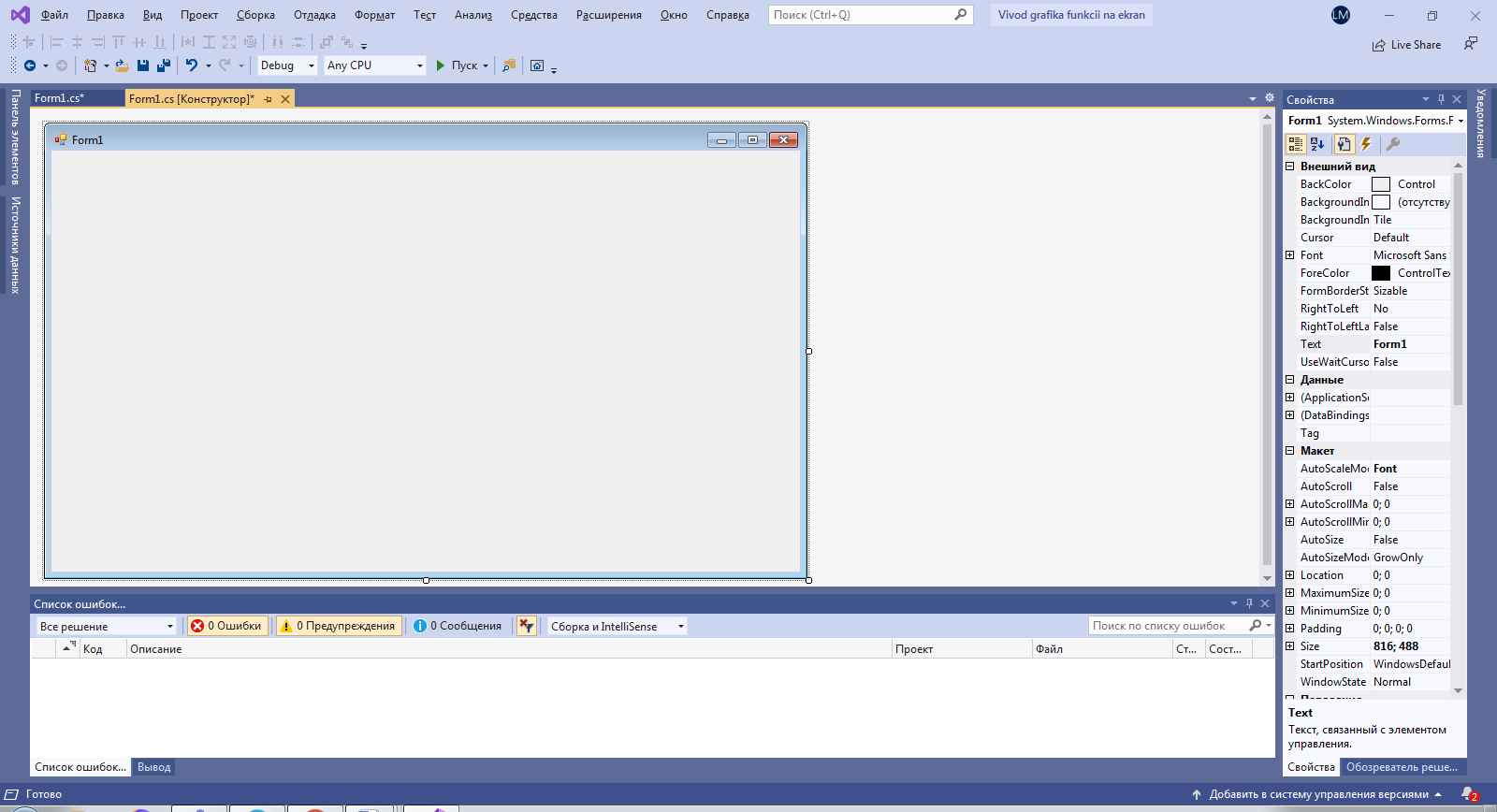
Малюнок 5 - Створення Windows Forms

Даємо назву проекту (мал. 6) :



Малюнок 6 - назва проекту

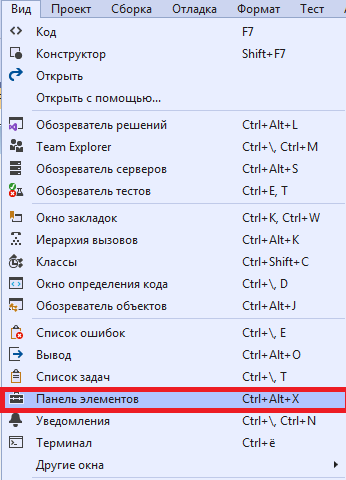
Тиснемо « Создать » і отримуємо порожній проект ( мал. 7 ) та форму:



Малюнок 7 - порожній проект

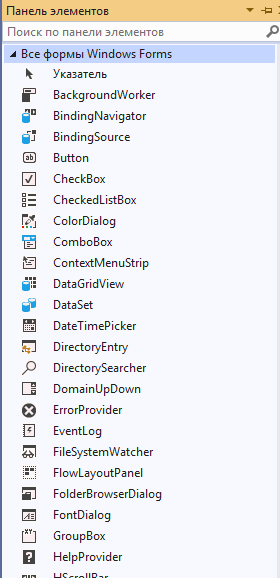
Після створення порожнього проекту, додаємо до нього елементи з Toolbox.

Для їх пошуку, треба знайти верхню позначку «Вид», після натиснути на «Панель элементів »( мал. 8 ).

**

Малюнок 8 - панель елементів

В результаті ми отримуємо вікно (мал. 9) , в якому знаходятться наступні інструменти:

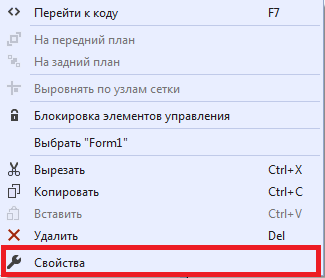


Малюнок 9 - Вікно з інструментами

У роботі використано елементи типу:

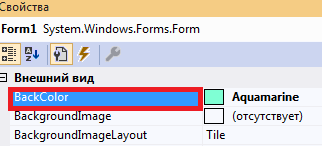
* Chart –побудова вікна з графіком.
* Bottom – побудова кнопок.
* Check Box – вибір графіків для виведення на екран.
* Combo Box – вибір одного з варіантів виводу графіка.
* Label – написи.
* Text Box – зміна тексту.
* Picture Box– вставка фото

Для редагування властивості елемента, наприклад, зміни напису на кнопці, треба натиснути на той елемент лівою клавішею миші та натиснути на «Свойства», після з’явиться вікно (мал. 10 ) . В ньому є можливість редагування властивостей елемента.



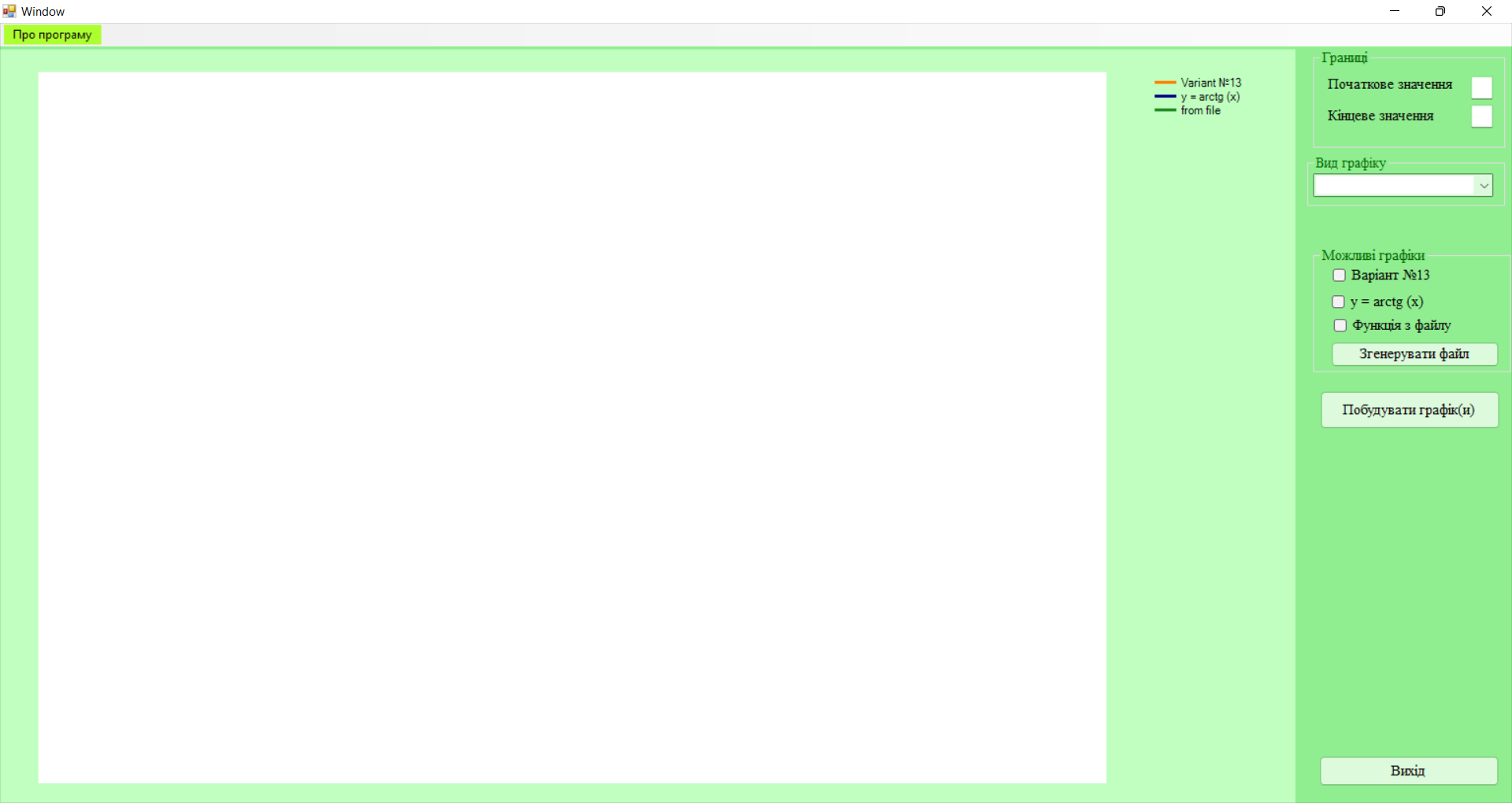
Малюнок 10 - Властивості елемента

Для зміни фону програми, у властивостях знаходимо елементи, що відповідають за зовнішній вигляд програми, і змінюємо фон (мал. 11) у Back Color , на один з кольорів на вибір.

**

Малюнок 10 - Зміна фону

Кінцевий інтерфейс (мал. 11 ) виглядає так:

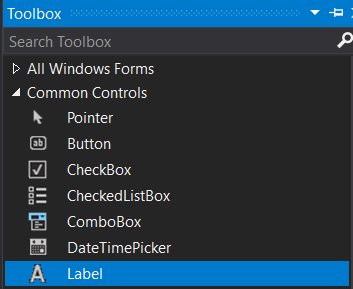


Малюнок 11 - Готовий інтерфейс

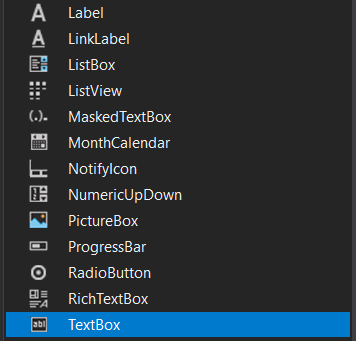
Отже відповідно до макету, було приєднано кнопки, додано для зручності користувача написи, які підказують що потрібно робити, створила написи на кнопках та створила фон програми.

## **3.1 Панель інтервалу**

Для того, щоб користувач мав змогу вводити інтервал, я використовувала у своїй роботі такі елементи, як Label ( мал. 12.1 ) та Text Box (мал 12.2 ). У вікні Label, ми написали користувачеві, що він має вводити в Text Box.



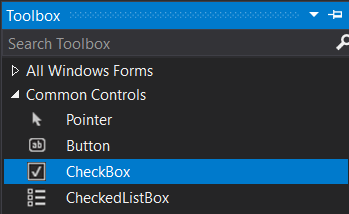
Малюнок 12.1 - Label 1



Малюнок 12.2 -TextBox 1

## **3.2 Панель вибору функції**

Для реалізації вибору функції я додала три елементи Check Box ( мал 12.3 ) кожний із назвою необхідного графіка. Програма почне малювати графіки, залежно від того які з компонентів будуть відмічені користувачем, після чого виведе на екран відповідні графіки.



Малюнок 12.3 - Check Box

## **3.3 Панель вибору**

Для реалізації вибору типу графіку використовуємо Combo Box

( мал.12.4 ). Користувач має обрати один із запропонованих варіантів.

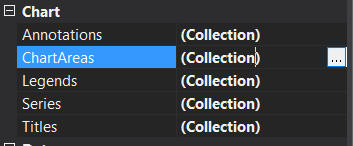
Після чого програма змінює вид графіка. В іншому випадку програма автоматично малює графік за допомогою ліній.

## https://cdn.discordapp.com/attachments/383323213112934432/725443951376466001/unknown.png

Малюнок 12.4 - ComboBox

## **3.4 Панель графіку**

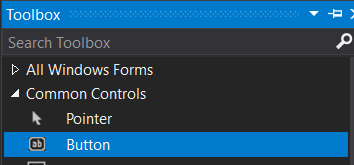
Для реалізації побудови графіків використовуємо елемент Chart (мал. 12.5). Має бути можливість відображення одночасно декількох графіків. Для кожного графіка обираємо окремий колір, додаємо розмітку вісей, надписи на них та направляючі стрілки. Прибираємо сітку. Важливо задати початок системи координат в точці (0.0).Даємо можливість користувачеві масштабувати графік за власним бажанням.



Малюнок 12.5 – ChartAreas

## **3.5 Кнопка початку роботи програми**

Для реалізації кнопки запуску програми ми використовували елемент Button ( мал. 12.6 ). Ми перейшли до коду програми двічі натиснувши на елемент. Коли користувач натискає кнопку перевіряються помилкові ситуації. Якщо користувач ввів усі параметри, то програма запускається і будує відповідний графік вказаного типу на вказаному інтервалі.



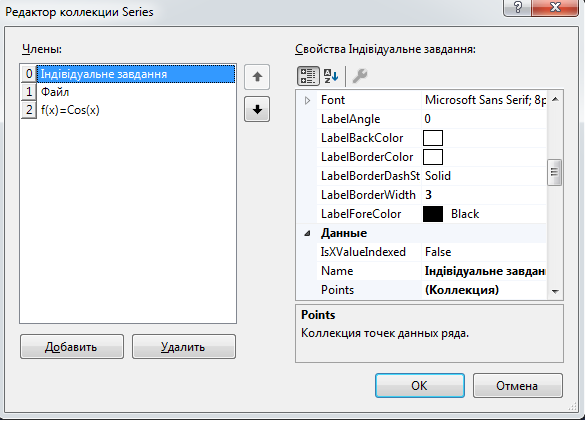
Малюнок 12.6 - Button 1

# **4 Етапи розробки графіків**

## **4****.1 Додання декількох графіків**

Для того, щоб одночасно намалювати декілька графіків, у властивостях елементу Chart знаходимо колекцію Series (рис. 15 ), натискаємо на кнопку «Добавить», додаємо графіки та налаштовуємо їх функціонал.

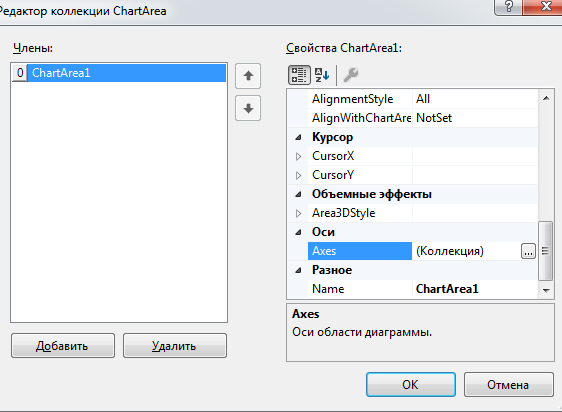
33



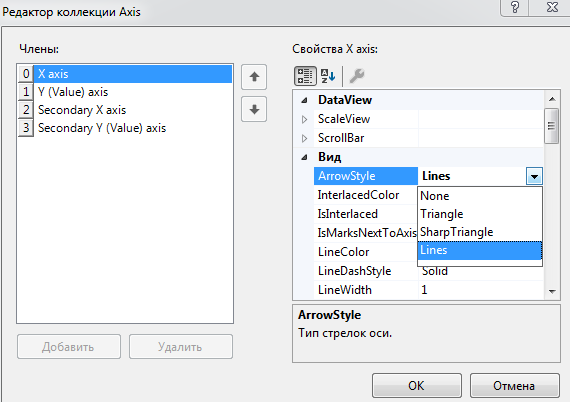
Малюнок 15 - додавання функцій

## **4.2 Стрілки на вісях**

В колекції Chart Areas знаходимо колекцію Axes ( (мал.16). Вона відповідає за властивості координат вісей. Для відображення стрілочок на координатних вісях міняємо властивість Arrow Style на Lines для вісі і X і Y( мал. 17).

****

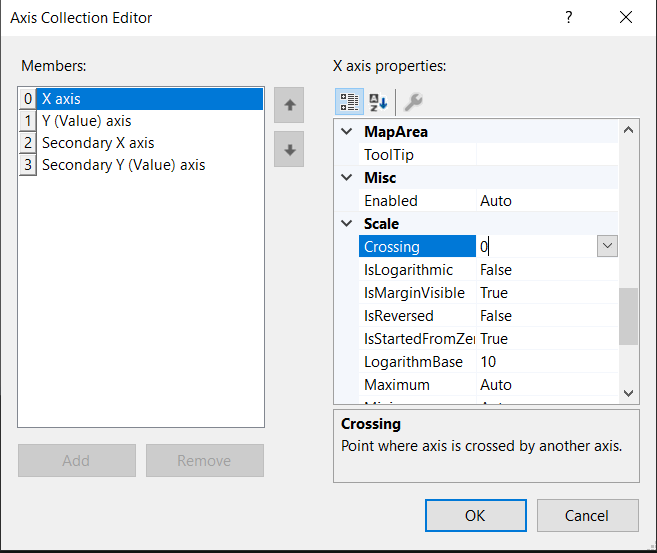
Малюнок 16 – Axes



Малюнок 17 - Arrow Style

## **Початок системи координат**

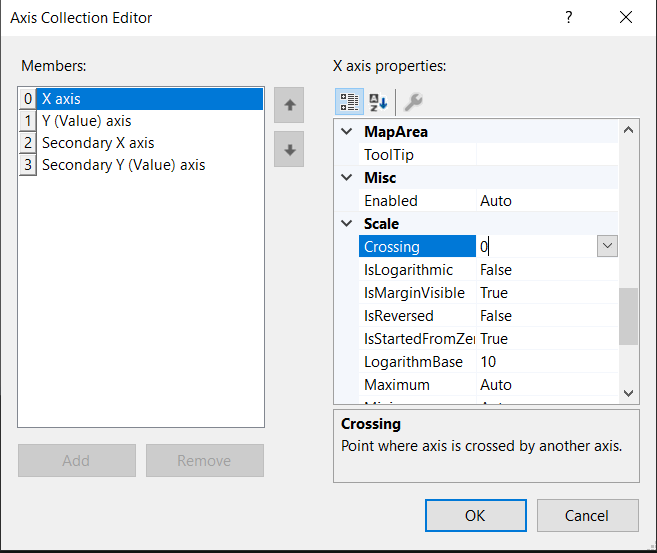
Для створення початку координат у властивості Scale змінемо властивість Crossing (мал. 18) з Auto на 0 по координаті і Х і У.



Малюнок 18 - Crossing 1

## **Розмітка головних вісей**

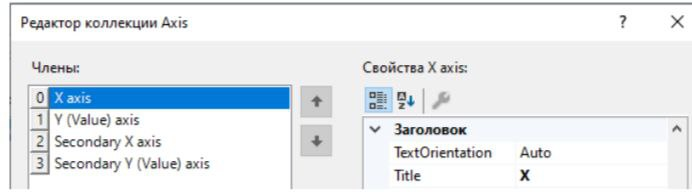
Для відображення розміток на координатних вісях змінимо властивість Is MarginVisible ( мал. 19) на True по вісі Х та У.



Малюнок 19 - IsMarginvisible 1

## **4.5 Назва координатних вісей**

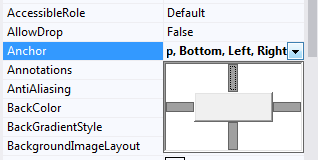
Для додання написів на координатних вісях додаємо назву вісей в властивості Title (мал. 20 )для кожної вісі окремо.



Малюнок 20 - Title

## **4.6 Стиснення та розтягнення**

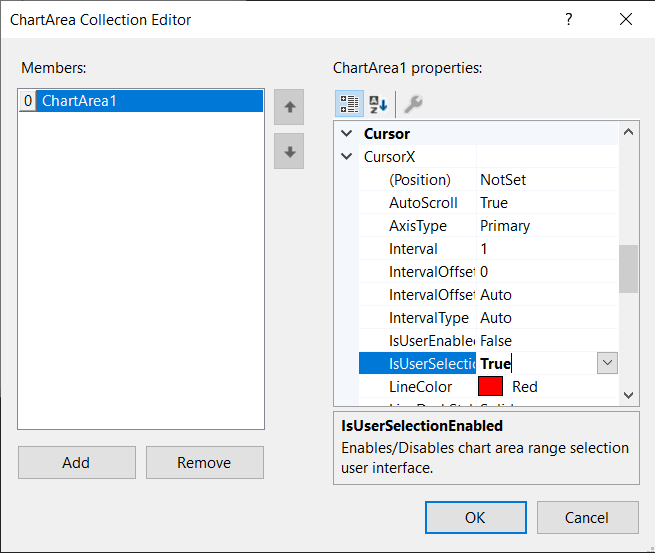
Для стиснення та розтягнення графіків використовуємо властивість Anchor (мал. 21) для кожного компонента, та налаштовуємо її в залежності від того, де знаходиться компонент і як він має розтягуватись:



Малюнок 21 - зміна Anchor

## **4.7 Масштабування**

Для масштабування в колекції Chart Areas потрібно змінити властивість Is User Enabled та Is User Selection ( мал. 22 ) на true для Cursorx та Cursory.



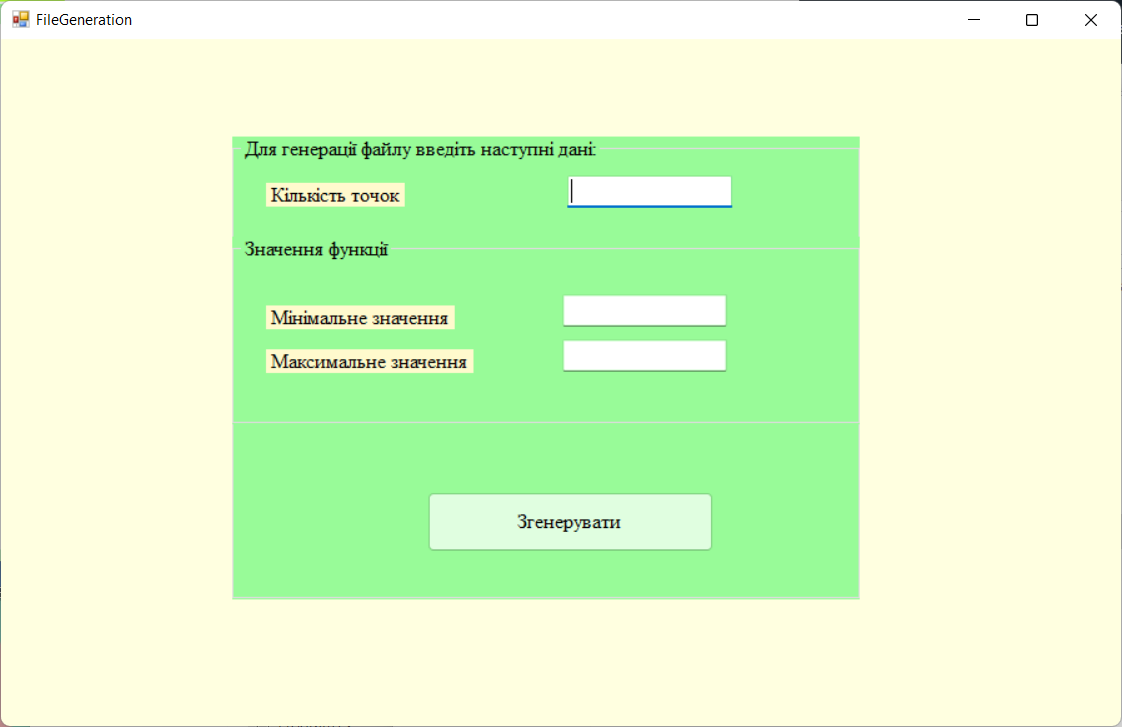
Малюнок 22 - IsUserSelection

# **5 Додаткові форми**

В процесі роботи я додала до головної сторінки додаткову форму:

## **5.1 Формування файлу**

При виборі графіка побудованого на основі файлу, користувач також має згенерувати файл, натиснувши на кнопку «Згенерувати файл». В формі, яка з'явиться ( мал. 23) користувач повинен заповнити інформацію для подальшої роботи, а саме кількість точок та максимальне і мінімальне значення, в якому будуть генеруватися точки по вісі ОУ.

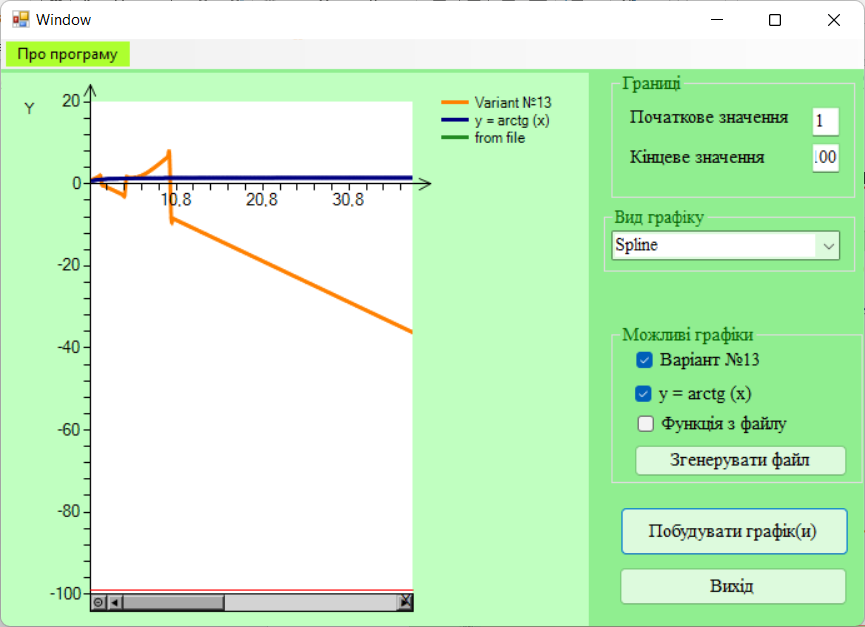


Малюнок 23 - форма для генерації файлу

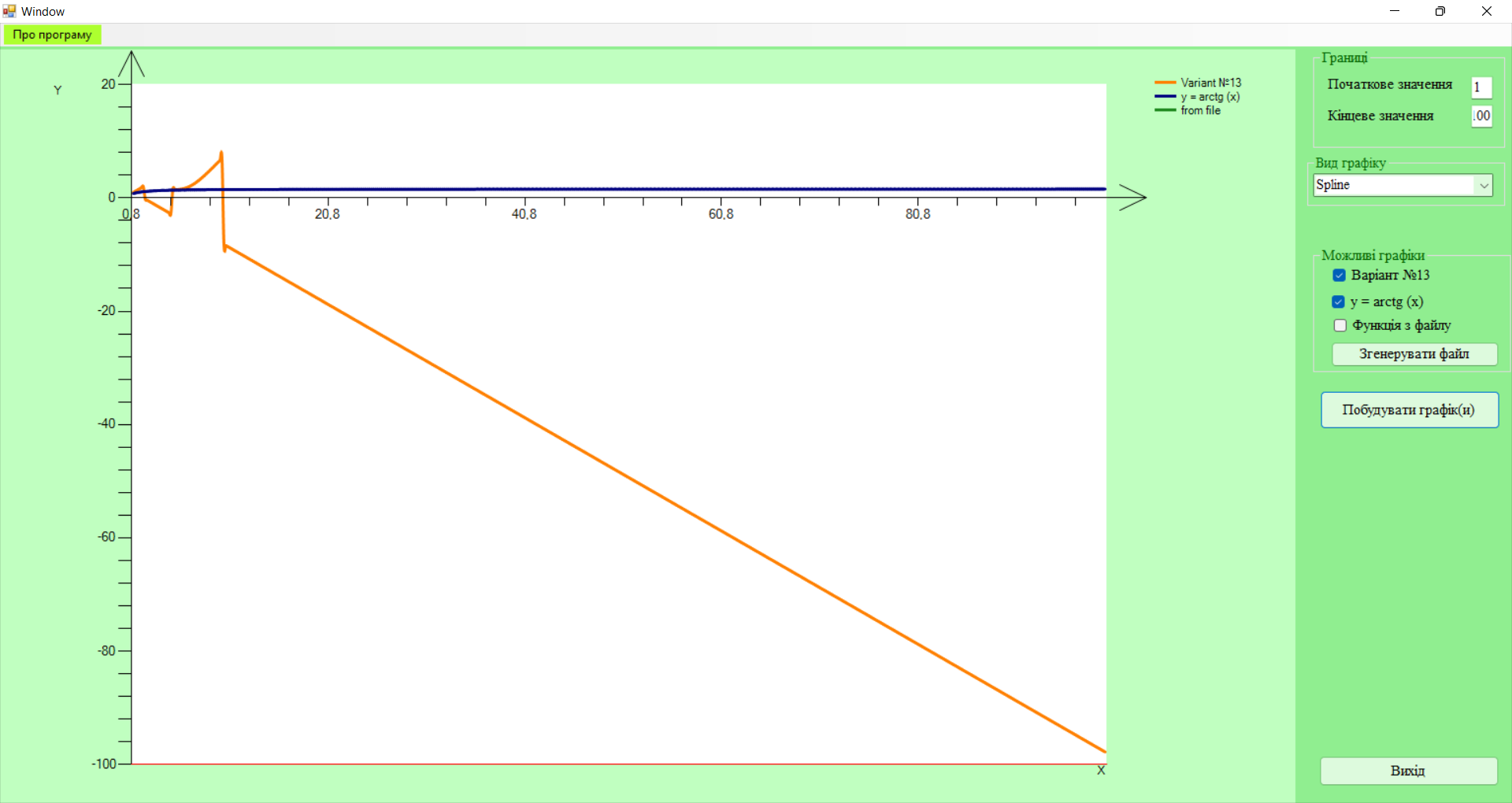
Після натиснення кнопки «Згенерувати файл»  з’явиться форма, яку користувач повинен заповнити. Якщо якесь з полів буде пустим, то програма видасть відповідне повідомлення. Якщо дані введені некоректно (кількість точок < 0 або нижня границя більша за верхню) то програма змінить дані на коректні сам, а від’ємну кількість змінить на додатню або змінить місцями.

Потім програма згенерує файл test.txt, в якому будуть знаходитися координати точок, за якими буде побудований відповідний графік.

# **Додатки**



Малюнок 24 – Стиснення



Малюнок 25 – Розтягнення

# **Висновки**

Під час виконання роботи по навчальній практиці ми навчилися:

- базовим можливостям візуального програмування;

- можливості програмування графіка.

Ознайомилися з базовими елементами їх властивостями. Під час проходження практики, я ознайомилася та отримала практичні навички в програмуванні на С#.

Навчилися додавати до проекту декілька форм та виводити їх на екран.

Розглянули елемент Chart та його базові можливості, а саме:

- масштабування

- редагування осей графіка

- додавання декількох графіків одночасно.

Використовувала його складові для реалізації свого проекту. Реалізувала вивід декількох графіків одночасно та окремо, завдяки Сhack Box. Я навчилася створювати простий дизайн для програм.

Робота з візуальним інтерфейсом була дуже цікава . Я здобула новий досвід в програмуванні, який неодмінно стану у пригоді в подальшому навчанні.

# **Список використаних джерел**

1. Н. А. Гук, Т. А. Зайцева, Н. М. Лисиця. Методичні вказівки настанови до проходження переддипломної практики студентами спеціальності 113 – Прикладна математика, РВВ ДНУ, ПП «Ліра ЛТД», 2018 р. 1,7 арк.. <http://repository.dnu.dp.ua:1100/?page=inner_material&id=10717>
2. Гук Н.А., Єгошкін Д.І., Сірик С.Ф. Алгоритм класифікації на базі нечіткої логіки з розширюваною кількістю виводів //  Зб. наук. праць «Питання прикладної математики і математичного моделювання». – Дніпро. – 2018. – Вип. 18. – С. 33-41.
3. http://msdn.microsoft.com/ru-ru/library/system.drawing.graphics.aspx

Текст програми

using System;

using System.IO;

using System.Windows.Forms;

using System.Windows.Forms.DataVisualization.Charting;

namespace Temp

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

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

{

DrawGraphics();

}

public void DrawGraphics()

{

//спробувати зробити частину кода в try, якщо виникне помилка виконається код в catch

try

{

var textA = textBox1.Text;

var textB = textBox2.Text;

if (string.IsNullOrEmpty(textA) || string.IsNullOrEmpty(textB))

{

MessageBox.Show("Введіть початкове і кінцеве значення інтервалу!");

}

else

{

var a = Convert.ToDouble(textBox1.Text);

var b = Convert.ToDouble(textBox2.Text);

// якщо початкове значення більше за кінцеве - змінюємо значення

if (a > b)

{

var k = a;

a = b;

b = k;

MessageBox.Show("Ви ввели неправильні межі інтервалу! Змінюю їх місцями!");

}

Series graphic1 = chart1.Series["Variant №13"];

Series graphic2 = chart1.Series["y = arctg (x)"];

Series graphic3 = chart1.Series["from file"];

// очищаємо всі раніше додані точки

if ((!checkBox1.Checked) && (!checkBox2.Checked) && (!checkBox3.Checked))

{

MessageBox.Show("Оберіть графік, який бажаєте побудувати!");

}

graphic1.Points.Clear();

graphic2.Points.Clear();

graphic3.Points.Clear();

// дістаємо графік

// якщо чекбокс обрано

if (checkBox1.Checked)

{// додаємо точки

for (double i = a; i <= b; i += 0.2)

{

double x = i;

double y = Myfunc(x);

graphic1.Points.AddXY(x, y);

}

}

if (checkBox2.Checked)

{

for (double i = a; i <= b; i += 0.2)

{

double x = i;

double y = Atan(x);

graphic2.Points.AddXY(x, y);

}

}

if (checkBox3.Checked)

{

//зчитуємо файл у вигляді масиву рядків

var fileContent = File.ReadAllLines(@"c://Users/асус/Desktop/temp/test.txt");

//розділяємо рядок Х і Y

var stringOfX = fileContent[0];

var stringOfY = fileContent[1];

//розділяємо Х і У через пробіл

var listOfX = stringOfX.Split(' ');

var listOfY = stringOfY.Split(' ');

for (var i = 0; i < listOfX.Length; i++)

{

//перетворюємо Х і У, записані у вигляді рядку у тип double

var x = Convert.ToDouble(listOfX[i]);

var y = Convert.ToDouble(listOfY[i]);

graphic3.Points.AddXY(x, y);

}

}

}

}

catch

{

}

}

// функція індивідуального завдання

public double Myfunc(double x)

{

if (x > 5 && x < 10)

{

double e = 2.71828182846;

return Math.Pow(Math.Sin(x / 2), 2) + Math.Pow(Math.Pow(e, 1.3 \* x) + Math.Pow(e, -1.3 \* x), 1.0 / 3) \* (1 / Math.Abs(x + 5.0 / 2));

}

else if (x > -5 && x <= 2)

{

return x;

}

else

{

return 2-x;

}

}

public double Atan(double x)

{

return Math.Atan(x);

}

private void comboBox1\_SelectedIndexChanged(object sender, EventArgs e)

{

//object - базовий тип, від якого відходять усі ссилочні типи

//привести об'єкт типу object до типу combobox

var comboBox = (ComboBox)sender;

//дістаємо обраний елемент

var selectedType = comboBox.SelectedItem.ToString();

//проходимось по всім графікам

var graphics = chart1.Series;

foreach (var graphic in graphics)

{

switch (selectedType)

{

case "Spline":

{

graphic.ChartType = SeriesChartType.Spline;

break;

}

case "Point":

{

graphic.ChartType = SeriesChartType.Point;

break;

}

}

}

}

private void button2\_Click(object sender, EventArgs e)

{

try

{

var a = Convert.ToDouble(textBox1.Text);

var b = Convert.ToDouble(textBox2.Text);

var childForm = new FileGeneration

{

A = a,

B = b

};

childForm.Show();

}

catch

{

}

}

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

{

this.Close();

}

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

{

var about=new About();

about.Show();

}

}

}

Генерація файлу

using System;

using System.Collections.Generic;

using System.IO;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Temp

{

public partial class FileGeneration : Form

{

public double A;

public double B;

public struct Point

{

public int X;

public int Y;

}

public FileGeneration()

{

InitializeComponent();

}

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

{

// srand(time(NULL)); взводим стётчик

var random = new Random();

//если текстбоск равен нулю или ""

if (string.IsNullOrEmpty(textBox1.Text) || string.IsNullOrEmpty(textBox2.Text) || string.IsNullOrEmpty(textBox3.Text))

{

MessageBox.Show("Ви ввели не всі дані!");

}

else

{

//считываем все поля ввода

var countOfPoint = Convert.ToInt32(textBox1.Text);

var lowerBorder = Convert.ToInt32(textBox2.Text);

var upperBorder = Convert.ToInt32(textBox3.Text);

if(lowerBorder > upperBorder)

{

var tmp = lowerBorder;

lowerBorder = upperBorder;

upperBorder = tmp;

MessageBox.Show("Ви ввели неправильні межі інтервалу! Змінюю їх місцями!");

}

if(countOfPoint<-0)

{

countOfPoint = Math.Abs(countOfPoint);

}

//создаём пустые списки чисел для X и Y

var listOfX = new List<double>();

var listOfY = new List<int>();

//считаем длину шага (длина отрезка)/(количество точек -1)

var step = (B - A) / (countOfPoint - 1);

//идём от левого края до правого, с шагом ^^^

for (var i = A; i <= B; i += step)

{

listOfX.Add(i);

//LowerBorder до UpperBorder; C++: -LowerBorder + rand () % (upperBorder - lowerBorder)

var y = lowerBorder + RandomNumber(random, 0, (upperBorder - lowerBorder));

listOfY.Add(y);

}

//удаляем файл, чтобы не было колизий(

File.Delete(@"c://Users/асус/Desktop/temp/test.txt");

//создаём новый текстовый файл

var file = File.CreateText(@"c://Users/асус/Desktop/temp/test.txt");

//выводим список X, разделенных между собой пробелами

file.WriteLine(string.Join(" ", listOfX));

//выводим список Y, разделенных между собой пробелами

file.WriteLine(string.Join(" ", listOfY));

//закрываем файл для записи

file.Close();

//закрываем форму

Close();

}

}

public int RandomNumber(Random random, int min, int max)

{

//возвращает рандомное число от min до max

//min>0 max>0 min<max

return random.Next(min, max);

}

}

}

Про програму

namespace Temp

{

partial class About

{

/// <summary>

/// Обязательная переменная конструктора.

/// </summary>

private System.ComponentModel.IContainer components = null;

/// <summary>

/// Освободить все используемые ресурсы.

/// </summary>

protected override void Dispose(bool disposing)

{

if (disposing && (components != null))

{

components.Dispose();

}

base.Dispose(disposing);

}

#region Код, автоматически созданный конструктором форм Windows

/// <summary>

/// Требуемый метод для поддержки конструктора — не изменяйте

/// содержимое этого метода с помощью редактора кода.

/// </summary>

private void InitializeComponent()

{

this.groupBox1 = new System.Windows.Forms.GroupBox();

this.label2 = new System.Windows.Forms.Label();

this.label1 = new System.Windows.Forms.Label();

this.groupBox1.SuspendLayout();

this.SuspendLayout();

//

// groupBox1

//

this.groupBox1.BackColor = System.Drawing.Color.LightCyan;

this.groupBox1.Controls.Add(this.label2);

this.groupBox1.Controls.Add(this.label1);

this.groupBox1.Font = new System.Drawing.Font("Times New Roman", 12F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(0)));

this.groupBox1.ForeColor = System.Drawing.Color.Olive;

this.groupBox1.Location = new System.Drawing.Point(16, 15);

this.groupBox1.Margin = new System.Windows.Forms.Padding(4);

this.groupBox1.Name = "groupBox1";

this.groupBox1.Padding = new System.Windows.Forms.Padding(4);

this.groupBox1.Size = new System.Drawing.Size(629, 214);

this.groupBox1.TabIndex = 1;

this.groupBox1.TabStop = false;

this.groupBox1.Text = "Про програму";

//

// label2

//

this.label2.AutoSize = true;

this.label2.Font = new System.Drawing.Font("Times New Roman", 16.2F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(0)));

this.label2.ForeColor = System.Drawing.Color.DarkGreen;

this.label2.Location = new System.Drawing.Point(162, 157);

this.label2.Margin = new System.Windows.Forms.Padding(4, 0, 4, 0);

this.label2.Name = "label2";

this.label2.Size = new System.Drawing.Size(317, 33);

this.label2.TabIndex = 1;

this.label2.Text = "Мітолап Аліна Василівна";

this.label2.TextAlign = System.Drawing.ContentAlignment.MiddleCenter;

//

// label1

//

this.label1.AutoSize = true;

this.label1.Font = new System.Drawing.Font("Times New Roman", 16.2F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(0)));

this.label1.ForeColor = System.Drawing.Color.DarkGreen;

this.label1.Location = new System.Drawing.Point(61, 25);

this.label1.Margin = new System.Windows.Forms.Padding(4, 0, 4, 0);

this.label1.Name = "label1";

this.label1.Size = new System.Drawing.Size(518, 132);

this.label1.TabIndex = 0;

this.label1.Text = "Ця програма написана в якості звіту \r\nпро комп\'ютерно - технологічну практику \r\n" +

"з дисципліни \"Програмування\".\r\nРоботу виконала студентка групи ПА-19-2";

this.label1.TextAlign = System.Drawing.ContentAlignment.MiddleCenter;

//

// About

//

this.AutoScaleDimensions = new System.Drawing.SizeF(8F, 16F);

this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;

this.BackColor = System.Drawing.Color.PaleGreen;

this.BackgroundImage = global::Temp.Properties.Resources.фон;

this.BackgroundImageLayout = System.Windows.Forms.ImageLayout.None;

this.ClientSize = new System.Drawing.Size(662, 463);

this.Controls.Add(this.groupBox1);

this.FormBorderStyle = System.Windows.Forms.FormBorderStyle.FixedDialog;

this.Margin = new System.Windows.Forms.Padding(4);

this.MaximizeBox = false;

this.MinimizeBox = false;

this.Name = "About";

this.Padding = new System.Windows.Forms.Padding(12, 11, 12, 11);

this.ShowIcon = false;

this.ShowInTaskbar = false;

this.StartPosition = System.Windows.Forms.FormStartPosition.CenterParent;

this.Text = "About";

this.groupBox1.ResumeLayout(false);

this.groupBox1.PerformLayout();

this.ResumeLayout(false);

}

#endregion

private System.Windows.Forms.GroupBox groupBox1;

private System.Windows.Forms.Label label2;

private System.Windows.Forms.Label label1;

}

}