

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

**Український державний університет науки і технологій**

Кафедра «Комп’ютерні інформаційні технології»

**Лабораторна робота №1**

**з дисципліни «Комп’ютерна графіка»**

## на тему:

**«Вивід графічної інформації. Побудова графіків функції.»**

Виконав:

студент гр. ПЗ2011

Кулик Сергій Вадимович

Прийняла:

Шаповал І. В.

Дніпро, 2022

**Тема.** Вивід графічної інформації. Побудова графіків функції.

**Мета.** Отримати навички створення простих зображень та створювання графіка функції. Вивчити графічні можливості середи, в якій створювалась програма.

**Завдання**

В середовищі розробки MS Visual Studio розробити програму для малювання графіку функції.

Програма повинна надавати можливість користувачеві:

- задавати діапазон аргументу функції і значення коефіцієнтів функцій;

- кольори і стиль зображення і заливки різних елементів графіку(фону, осей, заливки і графіку функції, маркерів);

- задавати вид(круг або квадрат) і розмір маркерів.

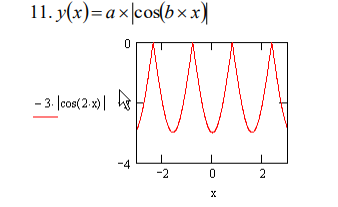
Вимоги до графіку функції :

- графік функції повинен повністю відображатися у вказаній області;

- графік повинен містити осі координат з підписами і розміткою;

- у вузлових точках на графіці повинні відображатися маркери.

**Індивідуальне завдання**



**Текст розробленої програми**

**Файл Form.cs**

using System;

using System.Drawing;

using System.Windows.Forms;

namespace ComputerGraphicsLB\_1

{

public partial class Form1 : Form

{

Pen graphicPen = new Pen(Color.Black, 1f);

Pen axisPen = new Pen(Color.Black, 1f);

Pen markPen = new Pen(Color.Black, 1f);

SolidBrush brush = new SolidBrush(Color.Black);

double a, b;

double from, to;

double minY, maxY;

double step;

public Form1()

{

InitializeComponent();

}

private double f(double x, double a, double b)

{

return a \* Math.Abs(Math.Cos(b \* x));

}

private void pictureBoxGraphic\_Paint(object sender, PaintEventArgs e)

{

DrawAxis(e);

a = floatTrackBarA.Value;

b = floatTrackBarB.Value;

int width = pictureBoxGraphic.Width;

int height = pictureBoxGraphic.Height;

try

{

from = double.Parse(textBoxFrom.Text);

to = double.Parse(textBoxTo.Text);

}

catch (Exception)

{

MessageBox.Show("Неверный формат ввода данных");

textBoxFrom.Text = "-1";

textBoxTo.Text = "1";

}

if (from > to)

{

double temp = from;

from = to;

to = temp;

string tempStr = textBoxFrom.Text;

textBoxFrom.Text = textBoxTo.Text;

textBoxTo.Text = tempStr;

}

else if (from == to)

{

MessageBox.Show("Початок і кінець відрізка мають бути різними!");

textBoxFrom.Text = "-1";

textBoxTo.Text = "1";

from = double.Parse(textBoxFrom.Text);

to = double.Parse(textBoxTo.Text);

}

minY = f(from, a, b);

maxY = minY;

step = (to - from) / 700;

for (double x = from; x <= to; x += step)

{

double y = f(x, a, b);

if (minY > y) minY = y;

if (maxY < y) maxY = y;

}

//масштабування графіку

for (double x = from; x <= to; x += step)

{

try

{

double y = f(x, a, b);

double y2 = f(x + step, a, b);

double x\_ = (x - from) \* width / (to - from) + 15;

double y\_ = (maxY - y) \* (height - 15) / (maxY - minY) +1 ;

double x2\_ = (x + step - from) \* width / (to - from) + 15;

double y2\_ = (maxY - y2) \* (height - 15) / (maxY - minY) + 1 ;

e.Graphics.DrawLine(graphicPen, (float)x\_, (float)y\_, (float)x2\_, (float)y2\_);

}

catch (Exception)

{

continue;

}

}

}

//малювання осей

void DrawAxis(PaintEventArgs e)

{

int width = pictureBoxGraphic.ClientSize.Width;

int height = pictureBoxGraphic.ClientSize.Height;

e.Graphics.DrawLine(axisPen, 15, height - 15, width - 15, height - 15); // axis Y

e.Graphics.DrawLine(axisPen, 15, -height, 15, height - 15);// axis X

const int DIVIDE = 6;

double iteratorX = Math.Sqrt(Math.Pow(to - from, 2)) / DIVIDE;

double iteratorY = Math.Sqrt(Math.Pow(maxY - minY, 2)) / DIVIDE;

double kX = from, kY = minY;

//штрихи осі Х

for (int i = 15; i <= width - 15; i += width / DIVIDE)

{

if (checkBoxMarkType.Checked)

{

e.Graphics.FillEllipse(brush, i, height - 20, 10, 10);

}

else

{

e.Graphics.DrawLine(markPen, i, height - 20, i, height - 10);

}

e.Graphics.DrawString(Math.Round(kX, 2).ToString(), fontDialog1.Font, Brushes.Black, i, height - 35);

kX += iteratorX;

}

//штрихи осі Y

for (int i = 0; i < height - 15; i += height / DIVIDE)

{

if (checkBoxMarkType.Checked)

{

e.Graphics.FillEllipse(brush, 10, i, 10, 10);

}

else

{

e.Graphics.DrawLine(markPen, 20, i, 10, i);

}

e.Graphics.DrawString(Math.Round(kY, 2).ToString(), fontDialog1.Font, Brushes.Black, 25 ,i);

kY += iteratorY;

}

}

void pictureBox1\_Resize(object sender, EventArgs e)

{

pictureBoxGraphic.Refresh();

}

void button2\_Click(object sender, EventArgs e)

{

colorDialog1.ShowDialog();

buttonColorBgr.BackColor = colorDialog1.Color;

pictureBoxGraphic.BackColor = colorDialog1.Color;

}

void loatTrackBarA\_ValueChanged(object sender, EventArgs e)

{

pictureBoxGraphic.Refresh();

labelAValue.Text = floatTrackBarA.Value.ToString();

}

void floatTrackBarB\_ValueChanged(object sender, EventArgs e)

{

pictureBoxGraphic.Refresh();

labelBValue.Text = floatTrackBarB.Value.ToString();

}

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

{

colorDialog1.ShowDialog();

buttonColorAxis.BackColor = colorDialog1.Color;

axisPen.Color = colorDialog1.Color;

pictureBoxGraphic.Refresh();

}

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

{

colorDialog1.ShowDialog();

buttonColorMarkers.BackColor = colorDialog1.Color;

markPen.Color = colorDialog1.Color;

brush.Color = colorDialog1.Color;

pictureBoxGraphic.Refresh();

}

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

{

fontDialog1.ShowDialog();

labelAValue.Font = fontDialog1.Font;

labelBValue.Font = fontDialog1.Font;

labelFrom.Font = fontDialog1.Font;

labelTo.Font = fontDialog1.Font;

labelA.Font = fontDialog1.Font;

labelB.Font = fontDialog1.Font;

label1.Font = fontDialog1.Font;

labelColor.Font = fontDialog1.Font;

labelMarkers.Font = fontDialog1.Font;

labelGraphic.Font = fontDialog1.Font;

label1.Text = fontDialog1.Font.Name;

labelAxis.Font = fontDialog1.Font;

labelBackground.Font = fontDialog1.Font;

labelPenSize.Font = fontDialog1.Font;

labelX.Font = fontDialog1.Font;

labelDiapason.Font = fontDialog1.Font;

labelSettings.Font = fontDialog1.Font;

checkBoxMarkType.Font = fontDialog1.Font;

buttonGenerate.Font = fontDialog1.Font;

pictureBoxGraphic.Refresh();

}

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

{

pictureBoxGraphic.Refresh();

}

void buttonGenerate\_Click(object sender, EventArgs e)

{

pictureBoxGraphic.Refresh();

}

private void trackBarPenGraphicSize\_ValueChanged(object sender, EventArgs e)

{

graphicPen.Width = trackBarPenGraphicSize.Value;

pictureBoxGraphic.Refresh();

}

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

{

colorDialog1.ShowDialog();

buttonColorGraphic.BackColor = colorDialog1.Color;

graphicPen.Color = colorDialog1.Color;

pictureBoxGraphic.Refresh();

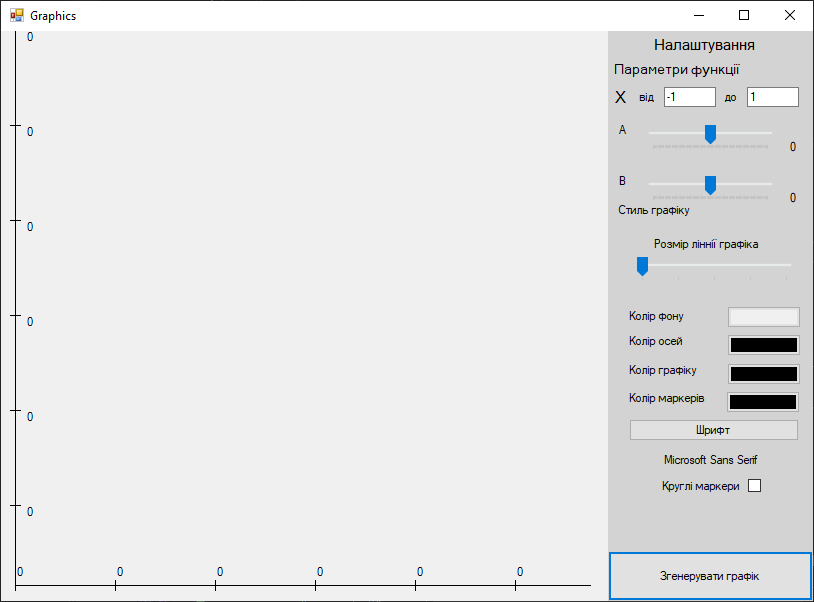
}

}

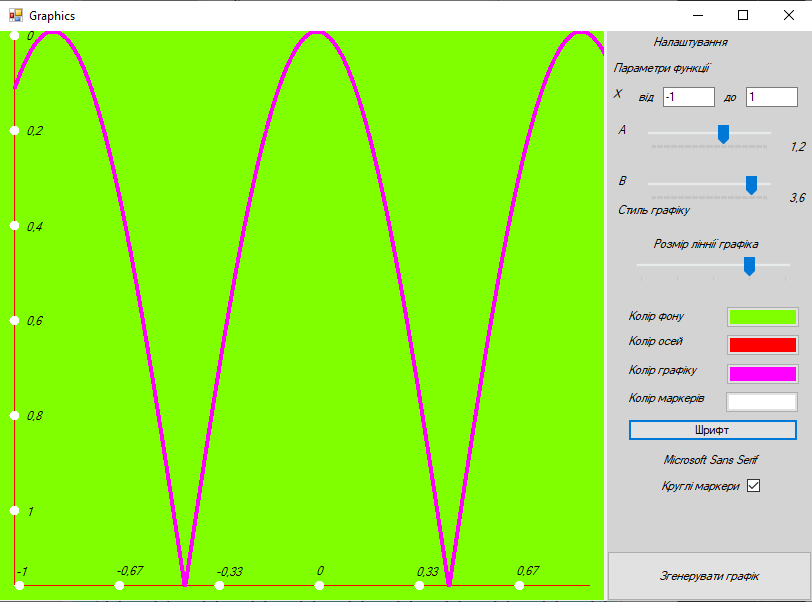
}

**Демонстрація роботи програми**

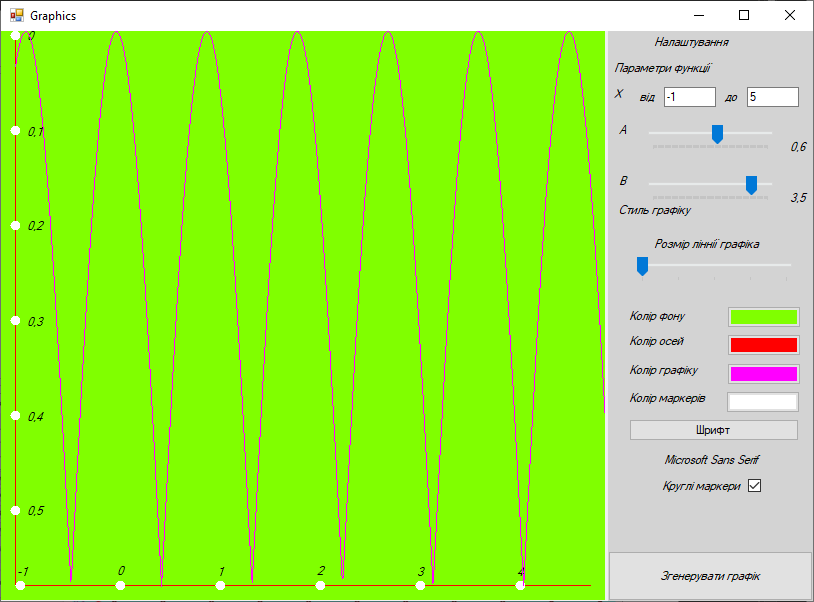
Значення за замовчуванням.



Зміна значень елементів



Зміна діапазону значень



**Висновок**

Під час виконання лабораторної роботи навчився розробляти програму для малювання графіку функції за допомогою елемента PictureBox. Так реалізував заміну шрифту, кольорів осей, графіку та маркерів. Графік у програмі відображається на всьому діапазону визначення функції в ширину та висоту.