

Лабораторна робота № 10

Відображення довільних даних у табличній формі за допомогою компонента DataGridView.

Мета. Ознайомлення принципами виведення масиву даних за допомогою класу DataGridView.

Теоретичні відомості

Компонент DataGridView дозволяє відобразити колекції складних даних, що можуть бути зчитані з файлів бази даних або розраховані за допомогою певних алгоритмів, у табличному виді. Застосування компонента DataGridView розглянемо на прикладі.

Приклад. Розробити застосунок Windows Forms, який розраховуватиме заданий математичний вираз з двома змінними $f(x_1, x_2) = x_1 + x_2$ в певному діапазоні для обох змінних. Програма повинна забезпечувати ввід мінімальних $x1_{min}$, $x2_{min}$ та максимальних $x1_{max}$, $x2_{max}$ значень, а також кроку зміни $dx1$, $dx2$. Всі розраховані значення виразу слід вивести у вікно в табличній формі за допомогою елемента керування DataGridView.

Порядок виконання завдання

В середовищі Microsoft Visual Studio створити новий проект застосунку Windows Forms.

Розробка головного вікна застосунку

1. Змінити назву файла, який описує головне вікно, на fMain.
2. Автоматично створена форма представлятиме головне вікно програми. Привести форму до вигляду згідно рис. 10.1. Для цього розмістити на ній:

- 6 компонентів Label
- 6 компонентів Textbox
- компонент DataGridView

- 3 компоненти Button

3. Задати назви (властивість Name) елементів керування TextBox відповідно до призначення:

- tbx1min
- tbx1max
- tbdx1
- tbx2min
- tbx2max
- tbdx2

4. Задати назву (властивість Name) елемента керування DataGridView:

- gv

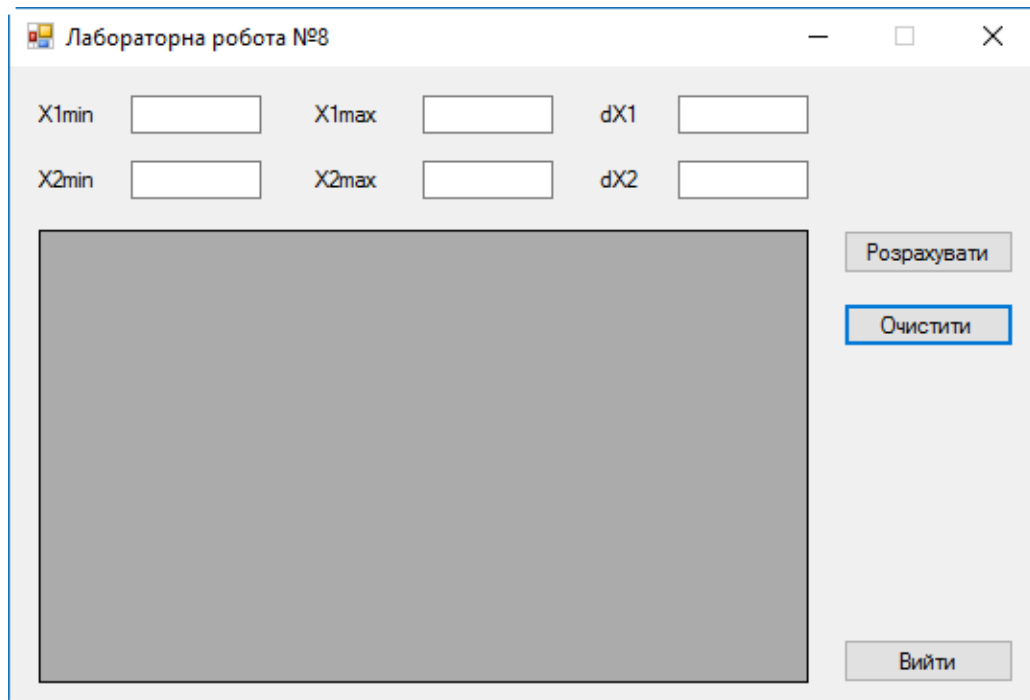


Рис. 10.1. Головна форма застосунку

5. Задати назви (властивість Name) елементів керування Button:

- btnCalc (Розрахувати)
- btnClear (Витерти)
- btnExit (Вийти)

6. Продумати алгоритм програми. Алгоритм для виконання завдання такий:

- 1) Отримати значення $x1_{min}$, $x1_{max}$, $dx1$, $x2_{min}$, $x2_{max}$, $dx2$.
- 2) Відповідно до введених даних, обчислити кількість стовпчиків та рядків таблиці.
- 3) У циклі пройти по всіх комірках стовпчика із заголовками рядків та заповнити його значеннями змінної $x1$.
- 4) У циклі пройти по всіх комірках рядка із заголовками стовпчиків та заповнити його значеннями змінної $x2$.
- 5) $rw = 0$ (змінна rw представляє поточний індекс рядка).
- 6) $x1 = x1_{min}$ (змінна $x1$ представляє поточне значення $x1$).
- 7) У циклі, поки $x1 \leq x1_{max}$:
 - $x2 = x2_{min}$ (змінна $x2$ представляє поточне значення $x2$).
 - $cl = 0$ (змінна cl представляє поточний індекс стовпчика).
 - У циклі, поки $x2 \leq x2_{max}$:
 - Розрахувати значення функції для поточних значень $x1$ та $x2$ і вивести його у комірку.
 - Збільшити $x2$ на $dx2$, збільшити cl на 1.
 - Збільшити $x1$ на $dx1$, збільшити rw на 1.

7. У оброблювач події Click для кнопки btnCalc ("Розрахувати") записати код з лістингу 10.1.

Лістинг 10.1

```
// Перетворення текстових рядків, які ввів користувач,  
  
// у змінні числового типу double  
x1min = double.Parse(tbx1min.Text); double  
x1max = double.Parse(tbx1max.Text);  
double x2min = double.Parse(tbx2min.Text);
```

```
double x2max = double.Parse(tbx2max.Text);
double dx1 = double.Parse(tbdx1.Text);
double dx2 = double.Parse(tbdx2.Text);

// Обчислення кількості рядків та стовпчиків таблиці
gv.ColumnCount = (int)Math.Truncate((x2max - x2min) / dx2) + 1;
gv.RowCount = (int)Math.Truncate((x1max - x1min) / dx1) + 1;

// Вивід заголовків рядків та стовпців таблиці
for (int i = 0; i < gv.RowCount; i++)
    gv.Rows[i].HeaderCell.Value = (x1min + i * dx1).ToString("0.000");
gv.RowHeadersWidth = 80;

for (int i = 0; i < gv.ColumnCount; i++)
{
    gv.Columns[i].HeaderCell.Value = (x2min + i *
dx2).ToString("0.000");
    gv.Columns[i].Width = 60;
}

// Для автоматичного підлаштування розмірів стовпчиків та рядків
// можна використовувати ці методи
//gv.AutoSizeColumns();
//gv.AutoSizeRows();

int cl, rw;
double x1, x2, y;

// Розрахунок і вивід езультатів
```

```
rw = 0; x1 = x1min; while  
(x1 <= x1max)  
{
```

```

        x2 = x2min; cl = 0;
while (x2 <= x2max)
    {
        y = x1 + x2;
        gv.Rows[rw].Cells[cl].Value = y.ToString("0.000");
x2 += dx2;    cl++;
    }    x1
+= dx1;    rw++;
    }

```

8. У оброблювач події Click для кнопки btnClear ("Очистити") записати наступний код з лістингу 10.2.

Лістинг 10.2

```

tbx1min.Text = ""; tbx1max.Text = ""; tbx2min.Text = "";
tbx2max.Text = ""; tbdx1.Text = ""; tbdx2.Text = "";
gv.Rows.Clear();

for (int Cl = 0; Cl < gv.ColumnCount; Cl++)
    gv.Columns[Cl].HeaderCell.Value = "";

```

9. У оброблювач події Click для кнопки btnExit ("Вихід") записати код з лістингу 10.3.

Лістинг 10.3

```

if (MessageBox.Show("Закрити програму?", "Вихід з програми",
    MessageBoxButtons.OKCancel, MessageBoxIcon.Question) ==
    DialogResult.OK)
    Application.Exit();

```

10. Запустити застосунок (F5) і перевірити його функціонал. Приклад вікна програми з розрахунковими даними приведено на рис. 10.2.

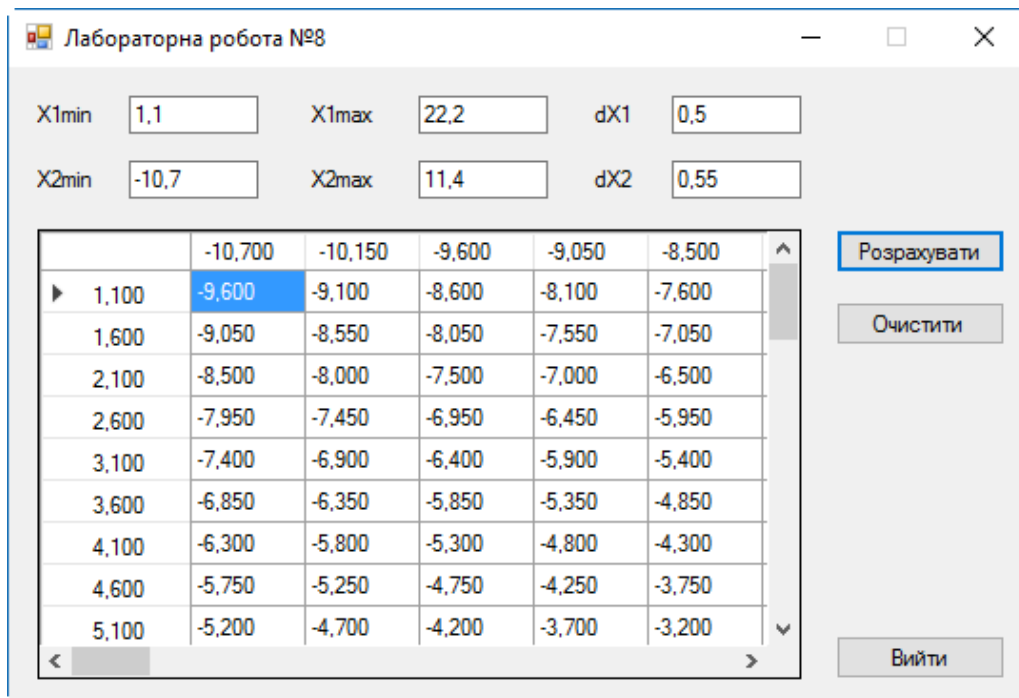


Рис. 10.2. Приклад вікна програми з розрахунковими даними

Завдання для самостійного опрацювання

- Вирішити розглянуте раніше завдання для заданої функції $f(x_1, x_2)$ згідно варіанту (30+№ за переліком).
- Додатково до попереднього завдання, обчислити та вивести на екран:
 - для варіантів, номери яких закінчуються на 0 чи 5 – суму всіх від'ємних розрахованих проміжних значень $f(x_1, x_2)$;
 - для варіантів, номери яких закінчуються на 1 чи 6 – добуток всіх додатних розрахованих проміжних значень $f(x_1, x_2)$;
 - для варіантів, номери яких закінчуються на 2 чи 7 – суму від'ємних синусів всіх розрахованих проміжних значень $f(x_1, x_2)$;
- для варіантів, номери яких закінчуються на 3 чи 8 – суму додатних косинусів всіх розрахованих проміжних значень $f(x_1, x_2)$;
- для варіантів, номери яких закінчуються на 4 чи 9 – суму квадратів всіх від'ємних розрахованих проміжних значень $f(x_1, x_2)$;

4. В кінці до таблиці з результатами додати порожній рядок (знизу) та стовпчик (справа) і заповнити їх:

- для парних варіантів – сумами відповідних стовпчиків та рядків;
- для непарних варіантів – середніми значеннями відповідних стовпчиків та рядків.

В ході роботи було розроблено форму із GridView з використанням формули із таблиці(варіант 10). Було додано додаткові стовбці для підрахунку суми від’ємних значень та підрахунку суми всіх елементів певного рядка. Для цього в цикл, для додавання елементів було додано змінну, в яку додається кожний елемент. (в залежності від того підрахунок це в стовпці чи рядку це відбувається у різних циклах.)

Код програми:

```
using System;
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 Lab10
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void button3_Click(object sender, EventArgs e)
        {
            Application.Exit();
        }

        private void btnCalc_Click(object sender, EventArgs e)
        {
            // Перетворення текстових рядків, які ввів користувач,
            // у змінні числового типу
            double x1min = double.Parse(tbx1min.Text);
            double x1max = double.Parse(tbx1max.Text);
            double x2min = double.Parse(tbx2min.Text);
            double x2max = double.Parse(tbx2max.Text);
            double dx1 = double.Parse(tbdx1.Text);
            double dx2 = double.Parse(tbdx2.Text);
        }
    }
}
```



```

// Обчислення кількості рядків та стовпчиків таблиці
gv.ColumnCount = (int)Math.Truncate((x2max - x2min) / dx2) +
3;
gv.RowCount = (int)Math.Truncate((x1max - x1min) / dx1) + 2;
// Вивід заголовків рядків та стовпців таблиці
for (int i = 0; i < gv.RowCount - 1; i++)
    gv.Rows[i].HeaderCell.Value = (x1min + i *
dx1).ToString("0.000");
gv.RowHeadersWidth = 80;
for (int i = 0; i < gv.ColumnCount - 1; i++)
{
    gv.Columns[i].HeaderCell.Value = (x2min + i *
dx2).ToString("0.000");
    gv.Columns[i].Width = 60;
}
gv.Columns[gv.ColumnCount - 2].HeaderCell.Value = "Sum
negative";
gv.Columns[gv.ColumnCount - 1].HeaderCell.Value = "Sum";
gv.Rows[gv.RowCount - 1].HeaderCell.Value = "Sum";
// Для автоматичного підлаштування розмірів стовпчиків та
рядків
// можна використовувати ці методи
//gv.AutoSizeColumns();
//gv.AutoSizeRows();
int cl, rw;
double x1, x2, y;
// Розрахунок і вивід результатів
double[] columnSums = new double[gv.ColumnCount - 2];
Double sum = 0;
Double totalRowSum = 0;
rw = 0; x1 = x1min;
while (x1 <= x1max)
{
    x2 = x2min; cl = 0;
    while (x2 <= x2max)
    {
        y = Math.Pow(Math.Sin((x1 * (x2 / (x1 + 53 *
Math.Pow(x2, 2))))), 2);
        if (y < 0)
        {
            sum += y;
        }
        columnSums[cl] += y;
        totalRowSum += y;
        gv.Rows[rw].Cells[cl].Value = y.ToString("0.000");
        x2 += dx2; cl++;
    }
    gv.Rows[rw].Cells[gv.ColumnCount - 2].Value =
sum.ToString("0.000");
    gv.Rows[rw].Cells[gv.ColumnCount - 1].Value =
totalRowSum.ToString("0.000");
    totalRowSum = 0;
    sum = 0;
    x1 += dx1;
    rw++;
}
for (int i = 0; i < gv.ColumnCount - 2; i++)
{

```

```

        gv.Rows[gv.RowCount - 1].Cells[i].Value =
columnSums[i].ToString("0.000");
    }
}
}
}

```

Результат роботи програми зображено на рисунку 10.1.

	5,000	5,500	6,000	6,500	7,000	Sum negative	Sum
► 10,000	0,001	0,001	0,001	0,001	0,001	0,000	0,005
10,500	0,002	0,001	0,001	0,001	0,001	0,000	0,006
11,000	0,002	0,001	0,001	0,001	0,001	0,000	0,006
11,500	0,002	0,002	0,001	0,001	0,001	0,000	0,007
12,000	0,002	0,002	0,001	0,001	0,001	0,000	0,007
12,500	0,002	0,002	0,002	0,001	0,001	0,000	0,008
13,000	0,002	0,002	0,002	0,001	0,001	0,000	0,009
* Sum	0,013	0,011	0,009	0,008	0,007	0,031	0,027

Рисунок 10.1 – Модифікована програма з формулою по варіанту і додатковими стовпцями/рядком

Варіант	Завдання	Варіант	Завдання
31.	$y = \frac{\sqrt{x_1^3 + x_2^5}}{1000 \cdot \sqrt{x_1 + x_2^5}} + 65$	32.	$y = 23 \cos^2(x_1^3 x_2^5) + 2x_1$
33.	$y = \frac{\sqrt{\cos^3(x_1) + x_2}}{x_1^{13} + 3/\cos(x_2)}$	34.	$y = \sin(x_1 - x_2^3 + \sqrt{x_1}) - 1.3x_1^3$
35.	$y = \frac{4 \sin(3 + x_1 x_2)}{34 - 9x_2^3}$	36.	$y = \sqrt{56x_1 + \frac{x_1 + x_2 + \sin(x_1 x_2)}{5 - \cos(x_2^2)}}$
37.	$y = \sqrt[5]{0.1x_1 \sin x_2 \cos x_1^2 + 55x_1 x_2}$	38.	$y = \sqrt{\frac{\cos(2x_2) + x_1/x_2}{16x_2 x_1}}$
39.	$y = \cos(\sqrt{x_2} + 34x_1) - 4 \sin(x_2)$	40.	$y = \sin^2\left(x_1 \frac{x_2}{x_1 + 53x_2^2}\right)$
41.	$y = \frac{\sqrt{x_1^3 + x_2^5}}{1000} + 65$	42.	$y = \exp(x_1 - x_2^2) + 31.55x_2 x_1^2$
43.	$y = \frac{6 - \cos(3 + x_1)}{34 - 9x_2^3 + x_2}$	44.	$y = \cos^4\left(x_1 - \sqrt{\frac{x_2}{x_1 + 53x_2^2}}\right)$
45.	$y = 23 \sin^2(x_1^3 x_2^5) + 2x_1 + \cos(x_1 x_2)$	46.	$y = \sqrt{56 + \frac{x_1 + x_2 + \sin(x_1 x_2)}{5 \cos(x_2^2)}}$
47.	$y = 45x_1 \sin x_2 + \sqrt{9x_2 x_1^3}$	48.	$y = \frac{1}{4 + x_2} \cdot \sqrt{\cos^2\left(\frac{x_2}{x_1}\right)}$
49.	$y = \frac{\cos^3(x_1) + 45 + x_2}{x_1^{13} + \cos(x_2)}$	50.	$y = \frac{5\sqrt{x_1^3 + x_2^5 - \cos(x_2)}}{\exp(x_1)}$
51.	$y = \frac{\cos^2\left(\lg_{10} \frac{x_2}{x_1}\right)}{45 + x_2}$	52.	$y = 45 \sin(x_1 + x_2 + \lg_{10}(x_1 x_2^2))$
53.	$y = \sqrt{\frac{x_2^2 + x_1/x_2}{16x_2 x_1}}$	54.	$y = \frac{\ln(x_2)}{\sqrt[5]{0.6x_1 \sin x_2 \cos x_1^4}}$
55.	$y = \cos^3\left(\exp\left(\frac{x_1 + 2x_2 + 9}{0.666}\right)\right)$	56.	$y = \frac{3x_2 - x_1^2}{\cos^3\left(\frac{x_1 + 2x_2 + 9}{0.37}\right)}$
57.	$y = \cos(\sqrt{x_2} + 34 \cdot \sin(x_1)) - 4 \sin(x_2)$	58.	$y = \sqrt{\frac{x_2^2 + x_1/x_2}{\cos(x_1^3 x_2^5) + 2x_1}}$
59.	$y = 0.1x_1 \sin x_2 \cos x_1^4 + 55$	60.	$y = \frac{\sin^3(x_1) + 45 + x_2}{2x_1^4 + 4x_2}$