

ЛАБОРАТОРНА РОБОТА №6

Тема: Робота з багатовимірними масивами.

Мета роботи

Дослідити роботу операторів циклу при роботі з двовимірними масивами. Закріпити навички використання компоненту DataGridView.

Постановка завдання

Розробити проект з інтерфейсом користувача, використовуючи необхідні компоненти та оброблювач подій мовою C#.

Теоретичні відомості

1. Багатовимірний масив

Масиви можуть мати декілька вимірів. Наприклад, матриця – це двовимірний масив, а об'єкти у тривимірному просторі можна описати у тривимірному масиві. Ось як можна їх оголошувати та ініціалізовувати:

```
int[,] array = new int[3,2];
int[,] array2D = { // такий масив має розмір [2,5]
    {1, 2, 3, 4, 5}, {6, 7, 8, 9, 10} };
int[,,] array3D = new int[3,3,3];
sbyte[,,] cubes = { // масив розміром [3,9,3]
    { {-1, 1, -1}, { 0, 1, -1}, { 1, 1, -1},
      {-1, 0, -1}, { 0, 0, -1}, { 1, 0, -1},
      {-1, -1, -1}, { 0, -1, -1}, { 1, -1, -1} },
    { {-1, 1, 0}, { 0, 1, 0}, { 1, 1, 0},
      {-1, 0, 0}, { 0, 0, 0}, { 1, 0, 0},
      {-1, -1, 0}, { 0, -1, 0}, { 1, -1, 0} },
    { {-1, 1, 1}, { 0, 1, 1}, { 1, 1, 1},
      {-1, 0, 1}, { 0, 0, 1}, { 1, 0, 1},
      {-1, -1, 1}, { 0, -1, 1}, { 1, -1, 1} } };
```

Доступ до елементів здійснюється аналогічно одновимірним масивам: спочатку в квадратних дужках вказується індекс по нульовому виміру¹, потім по першому і так далі.

2. Масиви масивів

Масиви масивів – це масив, елементами якого є інші масиви, причому не обов'язково рівні за розмірами. Саме цим вони відрізняються від

¹ Індеси також нумеруються з нуля. Зверніть на це увагу при визові функції GetLength.

багатовимірних масивів. Кожен елемент-масив потрібно ініціалізовувати окремо. Приклад ініціалізації масиву:

```
int[][] jaggedArray = {           // Масив містить три
    new int[] {1, 3, 5, 7, 9},    // елементи-масиви:
    new int[] {0, 2, 4, 6},      // у першому 5 елементів
    new int[] {8, 13}           // у другому – 4,
};                               // а у третьому – 2.
// Для доступу до елементів масиву використовуємо запис:
jaggedArray[0][2] = 14;
```

3. Використання оператора foreach

Для послідовного перебору всіх значень багатовимірного масиву можна використовувати оператор foreach аналогічно одновимірним масивам. Але для масиву масивів так не вийде, адже його елементи це масиви. Для них потрібно створювати вкладений цикл для перебору елементів:

```
foreach (int[] arrayElement in jaggedArray)
    foreach (int a in arrayElement)
        // змінна a набуває всіх значень масиву масивів
```

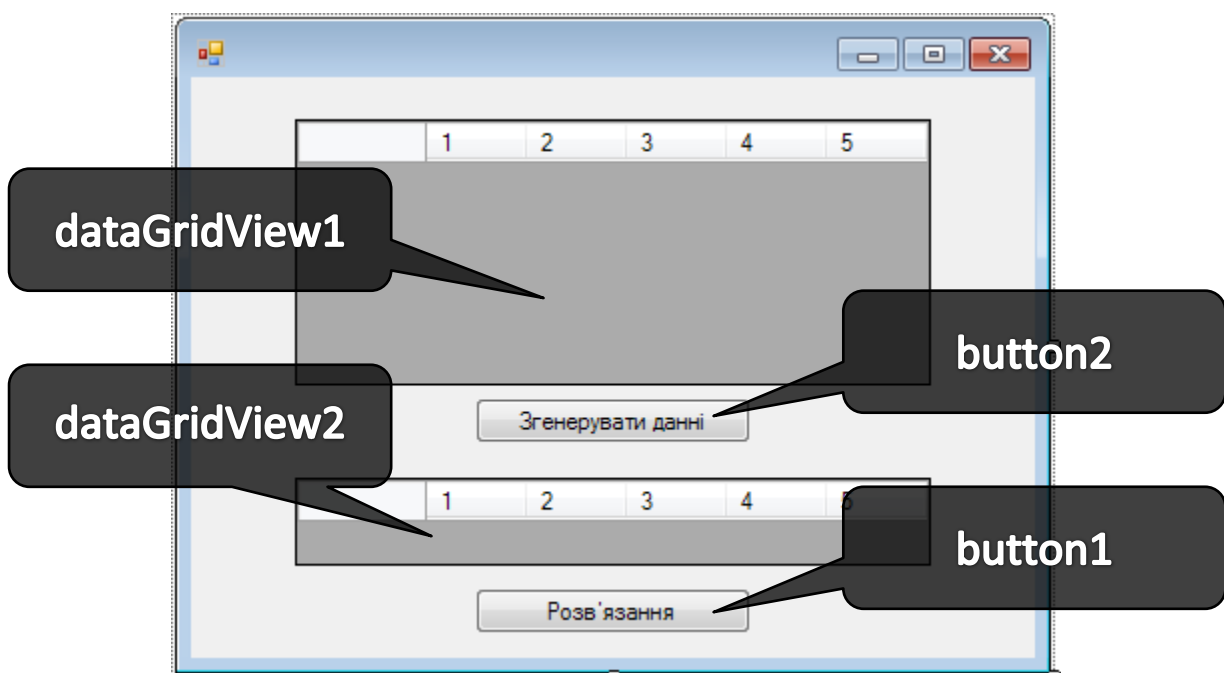
Приклади виконання завдань

Приклад 1

Задано масив A(5,5) дійсного типу. Заповнити масив B(5) мінімальними по стовпцях елементами масиву A.

Розв'язання

Створимо форму з двома компонентами DataGridView та кнопками:



У обох DataGridView виставте властивість **RowHeadersWidth** рівною 65. Інші властивості залиште як у попередніх прикладах (див. лаб. роб. №5). Код програми:

```
public partial class Form1 : Form
{
    public Form1()
    {
        InitializeComponent();
        for (byte i = 1; i <= 5; i++)
            dataGridView1.Rows[dataGridView1.Rows.Add()].
                HeaderCell.Value = i.ToString();
        dataGridView1.TopLeftHeaderCell.Value = "Данні";
        dataGridView1.TopLeftHeaderCell.Style.Alignment =
            DataGridViewContentAlignment.MiddleCenter;
        dataGridView2.Rows.Add();
        dataGridView2.TopLeftHeaderCell.Value = "Результат";
        dataGridView2.TopLeftHeaderCell.Style.Alignment =
            DataGridViewContentAlignment.MiddleCenter;
    }
    // Оброблювач події натискання на кнопку "Розв'язання"
    private void button1_Click(object sender, EventArgs e)
    {
        float[,] a = new float[5, 5];
        float[] b = new float[5];
        for (byte j = 0; j < 5; j++)
            for (byte i = 0; i < 5; i++)
                a[i, j] = Convert.ToSingle(dataGridView1[i, j].Value);
        for (byte i = 0; i < 5; i++)
        {
            b[i] = a[i, 0];
            for (byte j = 1; j < 5; j++)
                if (a[i, j] < b[i]) b[i] = a[i, j];
        }
        for (byte i = 0; i < 5; i++)
            dataGridView2[i, 0].Value = b[i];
    }
    // Оброблювач події натискання на кнопку "Згенерувати данні"
    private void button2_Click(object sender, EventArgs e)
    {
        Random r = new Random(); // генератор випадкових чисел
        for (byte j = 0; j < 5; j++)
            for (byte i = 0; i < 5; i++)
                dataGridView1[i, j].Value = (float)r.Next(-100, 100) / 10;
    }
}
```

Результат роботи програми:

Данні	1	2	3	4	5
1	-0.7	1.4	2.8	7.3	-0.7
2	-3	9.8	2	1.4	-3.7
3	6.3	4.9	9.4	8.7	-4
4	-0.4	1.2	8.8	-3.2	9.7
5	5.6	5.8	8.6	-8.2	-5

Згенерувати данні

Результат	1	2	3	4	5
1	-3	1.2	2	-8.2	-5

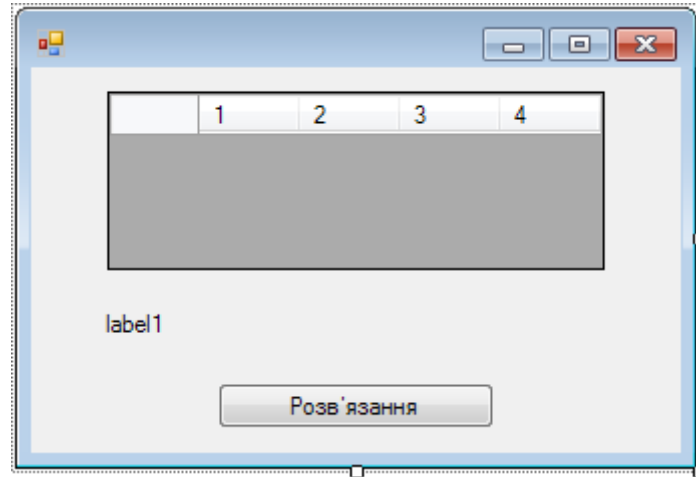
Розв'язання

Приклад 2

У цілому масиві A(4,3) знайти рядок з лише додатними елементами.

Розв'язання

Створимо форму з компонентами DataGridView, Button та Label:



Та напишемо наступний код:

```
public partial class Form1 : Form
{
    public Form1()
    {
        InitializeComponent();
        for (byte i = 1; i <= 3; i++)
            dataGridView1.Rows[dataGridView1.Rows.Add()].
                HeaderCell.Value = i.ToString();
        dataGridView1.TopLeftHeaderCell.Value = "Данні";
        dataGridView1.TopLeftHeaderCell.Style.Alignment =
            DataGridViewContentAlignment.MiddleCenter;
    }
    // Оброблювач події натискання на кнопку "Розв'язання"
    private void button1_Click(object sender, EventArgs e)
    {
        int[][] a = { new int[4], new int[4], new int[4] };
        for (byte j = 0; j < 3; j++)
            for (byte i = 0; i < 4; i++)
                a[j][i] = Convert.ToInt32(dataGridView1[i, j].Value);
        label1.Text = "";
        for (byte j = 0; j < 3; j++)
            if (Array.Find(a[j], x => x < 0) == 0)
                label1.Text += (j+1).ToString() + " ";
        label1.Text += (label1.Text == "") ? "Такого рядка немає" : "рядок";
    }
}
```

`Array.Find` – функція, що шукає елемент масиву, який задовольняє умові, якщо такого елемента немає, то повертається значення 0 (для типу `int`).

`x => x < 0` – лямбда-вираз, яким описується ця умова. Він означає функцію, яка приймає (у даному випадку) по чергово всі елементи масиву `a[j]` (це ж масив масивів) і повертає значення `true` якщо виконується умова `x < 0`.