

Лабораторна робота 12

Тема роботи: Використання та налаштування DataGridView Табличне введення даних. Об'єкт Chart.

Мета: вивчити прийоми роботи з таблицями з використанням об'єкту DataGridView. Засвоїти методи створення, збереження та відкриття таблиць у форматі файлу XML. Ознайомитись з роботою об'єкту DataSet у проектах Windows Forms у Visual C++. Навчитися створювати об'єкти DataTable для збереження даних у табличному вигляді.

Ознайомитись з прийомами побудови графіків з використанням

Хід роботи

Завдання до роботи

1. Виконати приклади наведені у роботі
2. Виконати індивідуальне завдання

Завдання для індивідуального виконання

Завдання Створити проект Windows Form для розв'язання задачі згідно варіанту.

5.	Задано матриця D (4,4). Якщо максимальний елемент матриці стоїть на головній діагоналі, то всі елементи головної діагоналі зробити рівними максимальному
----	--

Лістинг 12.1 – Код програми

```
#pragma once
```

```
// Підключаємо необхідні простори імен  
using namespace System;  
using namespace System::Windows::Forms;  
using namespace System::Data;  
using namespace System::Drawing; // Для Point, Size, Color  
using namespace System::IO;
```

```
namespace MatrixApp
{
    public ref class MyForm : public Form
    {
    public:
        // Конструктор форми
        MyForm(void)
        {
            InitializeComponent();
        }

    protected:
        // Деструктор форми
        ~MyForm()
        {
            if (components)
            {
                delete components;
            }
        }

    private:
        /// <summary>
        /// Потрібна змінна дизайнера.
        /// </summary>
        System::ComponentModel::Container^ components;
        System::Windows::Forms::DataGridView^ dataGridView1;
        System::Windows::Forms::Button^ buttonFill;
        System::Windows::Forms::Button^ buttonProcess;
        System::Windows::Forms::Button^ buttonSave;
        System::Windows::Forms::Button^ buttonLoad;
        System::Windows::Forms::Button^ buttonClear;
        System::Data::DataSet^ MatrDSet;
        System::Data::DataTable^ Tb1D;
        const int Msize = 4; // Розмір матриці 4x4
        void InitializeComponent(void)
        {
            this->Text = L"Матриця D (4x4) - Варіант 5";
            this->Width = 650;
            this->Height = 460;
            this->BackColor = System::Drawing::Color::LightGray;

            // 2. DataGridView
            this->dataGridView1 = gcnew
System::Windows::Forms::DataGridView();
```

```
this->dataGridView1->Location = System::Drawing::Point(10,
10);
this->dataGridView1->Size = System::Drawing::Size(400, 200);
this->dataGridView1->ColumnCount = Msize;
this->dataGridView1->RowCount = Msize;

// Налаштування стовпців DataGridView
for (int i = 0; i < Msize; i++)
{
    this->dataGridView1->Columns[i]->Name = L"C" + (i +
1).ToString();
    this->dataGridView1->Columns[i]->Width = 80; //
Встановлення ширини
}
this->dataGridView1->AllowUserToAddRows = false;
this->dataGridView1->RowHeadersVisible = false; // Приховати
ліві заголовки рядків

// 3. Кнопки
int buttonTop = 230; // Початкова позиція Top для першої кнопки

// Кнопка 'Заповнити'
this->buttonFill = gcnew System::Windows::Forms::Button();
this->buttonFill->Text = L"Заповнити";
this->buttonFill->Location = System::Drawing::Point(420,
buttonTop);
this->buttonFill->Click += gcnew System::EventHandler(this,
&MyForm::buttonFill_Click);
this->Controls->Add(this->buttonFill);
buttonTop += 40; // Зсув для наступної кнопки

// Кнопка 'Обробка'
this->buttonProcess = gcnew System::Windows::Forms::Button();
this->buttonProcess->Text = L"Обробка";
this->buttonProcess->Location = System::Drawing::Point(420,
buttonTop);
this->buttonProcess->Click += gcnew
System::EventHandler(this, &MyForm::buttonProcess_Click);
this->Controls->Add(this->buttonProcess);
buttonTop += 40;

// Кнопка 'Зберегти в XML'
this->buttonSave = gcnew System::Windows::Forms::Button();
this->buttonSave->Text = L"Зберегти XML";
this->buttonSave->Location = System::Drawing::Point(420,
buttonTop);
```

```
        this->buttonSave->Click += gcnew System::EventHandler(this,
&MyForm::buttonSave_Click);
        this->Controls->Add(this->buttonSave);
        buttonTop += 40;

        // Кнопка 'Завантажити з XML'
        this->buttonLoad = gcnew System::Windows::Forms::Button();
        this->buttonLoad->Text = L"Завантажити XML";
        this->buttonLoad->Location = System::Drawing::Point(420,
buttonTop);
        this->buttonLoad->Click += gcnew System::EventHandler(this,
&MyForm::buttonLoad_Click);
        this->Controls->Add(this->buttonLoad);
        buttonTop += 40;

        // Кнопка 'Очистити таблицю'
        this->buttonClear = gcnew System::Windows::Forms::Button();
        this->buttonClear->Text = L"Очистити";
        this->buttonClear->Location = System::Drawing::Point(420,
buttonTop);
        this->buttonClear->Click += gcnew System::EventHandler(this,
&MyForm::buttonClear_Click);
        this->Controls->Add(this->buttonClear);

        // 4. Об'єкти даних (DataSet/DataTable)
        this->MatrDSet = gcnew System::Data::DataSet(L"МатрицяD");
        this->TblD = gcnew System::Data::DataTable(L"MatrD");
        this->MatrDSet->Tables->Add(this->TblD);

        // Додавання DataGridView до форми
        this->Controls->Add(this->dataGridView1);
    }

    // =====
    //                                ОБРОБНИКИ ПОДІЙ
    // =====

    /// <summary>
    /// Заповнення DataGridView випадковими числами.
    /// </summary>
    void buttonFill_Click(System::Object^ sender, System::EventArgs^
e)
    {
        Random^ rnd = gcnew Random(); // Створення генератора
випадкових чисел
```

```
        for (int i = 0; i < Msize; i++)
        {
            for (int j = 0; j < Msize; j++)
            {
                // Заповнення числами від 1 до 100
                dataGridView1->Rows[i]->Cells[j]->Value = rnd-
>Next(1, 100).ToString();
            }
        }

    /// <summary>
    /// Обробка: Знаходження максимального елемента на головній
діагоналі та його позиції.
    /// Заміна всіх елементів рядка, де знаходиться максимум, на "0".
    /// </summary>
    void buttonProcess_Click(System::Object^ sender,
System::EventArgs^ e)
    {
        int maxVal = Int32::MinValue; // Початкове значення мінімальне
        int maxRow = -1; // Індекс рядка з максимумом
        int maxCol = -1; // Індекс стовпця з максимумом

        // 1. Пошук максимального елемента на головній діагоналі
        for (int i = 0; i < Msize; i++)
        {
            int val;
            // Конвертація значення комірки в int. Забезпечення, що
дані коректні.
            if (!Int32::TryParse(dataGridView1->Rows[i]->Cells[i]-
>Value->ToString(), val))
            {
                MessageBox::Show(L"Помилка конвертації даних у
комірці [" + i.ToString() + L", " + i.ToString() + L"]. Перевірте дані.",
L"Помилка", MessageBoxButtons::OK, MessageBoxIcon::Error);
                return;
            }

            // Перевірка на максимум
            if (val > maxVal)
            {
                maxVal = val;
                maxRow = i; // Оскільки на діагоналі, Row == Col
                maxCol = i;
            }
        }
    }
```

```
// 2. Обробка результату
if (maxRow != -1) // Якщо максимум знайдено
{
    // Заміна елементів у знайденому рядку на "0"
    for (int j = 0; j < Msize; j++)
    {
        dataGridView1->Rows[maxRow]->Cells[j]->Value = L"0";
    }

    MessageBox::Show(
        L"Максимальний елемент на діагоналі: " +
maxVal.ToString() +
        L".\nЗнаходиться в позиції [" + (maxRow +
1).ToString() + L", " + (maxCol + 1).ToString() + L"]." +
        L"\nРядок " + (maxRow + 1).ToString() + L" замінено
на нулі.",
        L"Результат обробки",
        MessageBoxButtons::OK,
        MessageBoxIcon::Information
    );
}
else // На випадок, якщо матриця була порожня або обробка не
відбулася
{
    MessageBox::Show(L"Максимум не знайдено. Нічого не
змінено.",
        L"Результат обробки",
        MessageBoxButtons::OK,
        MessageBoxIcon::Warning);
}
}

/// <summary>
/// Збереження даних з DataGridView в XML-файл.
/// </summary>
void buttonSave_Click(System::Object^ sender, System::EventArgs^
e)
{
    // 1. Очищення DataTable перед новим заповненням
    TblD->Clear();
    TblD->Columns->Clear();

    // 2. Створення стовпців у DataTable
    for (int j = 0; j < Msize; j++)
    {
        // Стовпці мають тип Int32 для коректної роботи з числами
```

```
TblD->Columns->Add(L"C" + (j + 1).ToString(),
Int32::typeid);
}

// 3. Заповнення DataTable даними з DataGridView
for (int i = 0; i < Msize; i++)
{
    DataRow^ row = TblD->NewRow();
    for (int j = 0; j < Msize; j++)
    {
        // Конвертація і збереження
        int val;
        if (Int32::TryParse(dataGridView1->Rows[i]->Cells[j]-
>Value->ToString(), val))
        {
            row[j] = val;
        }
        else
        {
            // Якщо не вдалося сконвертувати, записуємо нуль
            (або кидаємо помилку)
            row[j] = 0;
        }
    }
    TblD->Rows->Add(row);
}

// 4. Додавання DataTable до DataSet (якщо вона ще не там,
хоча в InitializeComponent ми вже її додали)
if (!MatrDSet->Tables->Contains(TblD->TableName))
{
    MatrDSet->Tables->Add(TblD);
}

// 5. Збереження DataSet у файл MatrD.xml
try
{
    MatrDSet->WriteXml(L"MatrD.xml",
XmlWriteMode::WriteSchema);
    MessageBox::Show(L"Дані успішно збережено у MatrD.xml",
L"Збереження", MessageBoxButtons::OK, MessageBoxIcon::Information);
}
catch (Exception^ ex)
{
    MessageBox::Show(L"Помилка при збереженні файлу: " + ex-
>Message, L"Помилка", MessageBoxButtons::OK, MessageBoxIcon::Error);
}
```

```
    }  
  }  
  
  /// <summary>  
  /// Завантаження даних з XML-файлу в DataGridView.  
  /// </summary>  
  void buttonLoad_Click(System::Object^ sender, System::EventArgs^  
e)  
  {  
    String^ fileName = L"MatrD.xml";  
  
    if (File::Exists(fileName))  
    {  
      // 1. Очищення DataSet та завантаження  
      MatrDSet->Clear();  
      try  
      {  
        MatrDSet->ReadXml(fileName);  
      }  
      catch (Exception^ ex)  
      {  
        MessageBox::Show(L"Помилка при читанні файлу: " + ex-  
>Message, L"Помилка", MessageBoxButtons::OK, MessageBoxIcon::Error);  
        return;  
      }  
  
      // 2. Перевірка, чи завантажилась потрібна таблиця  
      if (MatrDSet->Tables->Contains(L"MatrD"))  
      {  
        TblD = MatrDSet->Tables[L"MatrD"];  
        int rowCount = TblD->Rows->Count;  
        int colCount = TblD->Columns->Count;  
  
        // Очищення DataGridView та встановлення розмірів  
        dataGridView1->RowCount = rowCount;  
        dataGridView1->ColumnCount = colCount;  
  
        // 3. Заповнення DataGridView даними з DataTable  
        for (int i = 0; i < rowCount; i++)  
        {  
          for (int j = 0; j < colCount; j++)  
          {  
            // Оскільки в DataTable ми зберігали Int32,  
            конвертуємо назад у String
```



```
dataGridView1->Rows[i]->Cells[j]->Value =  
TblD->Rows[i]->default[j]->ToString();  
    }  
    }  
  
    MessageBox::Show(L"Дані успішно завантажено з  
MatrD.xml", L"Завантаження", MessageBoxButtons::OK,  
    MessageBoxIcon::Information);  
    }  
    else  
    {  
        MessageBox::Show(L"Файл MatrD.xml не містить таблиці  
'MatrD'.", L"Помилка завантаження", MessageBoxButtons::OK,  
        MessageBoxIcon::Error);  
    }  
    }  
    else  
    {  
        MessageBox::Show(L"Файл MatrD.xml не знайдено.",  
        L"Помилка завантаження", MessageBoxButtons::OK, MessageBoxIcon::Error);  
    }  
    }  
  
    /// <summary>  
    /// Очищення DataGridView.  
    /// </summary>  
    void buttonClear_Click(System::Object^ sender, System::EventArgs^  
e)  
    {  
        // Очищення всіх комірок  
        for (int i = 0; i < Msize; i++)  
        {  
            for (int j = 0; j < Msize; j++)  
            {  
                dataGridView1->Rows[i]->Cells[j]->Value = L""; //  
Встановлення порожнього рядка  
            }  
        }  
        MessageBox::Show(L"Таблицю очищено.", L"Очищення",  
        MessageBoxButtons::OK, MessageBoxIcon::Information);  
    }  
};  
}
```

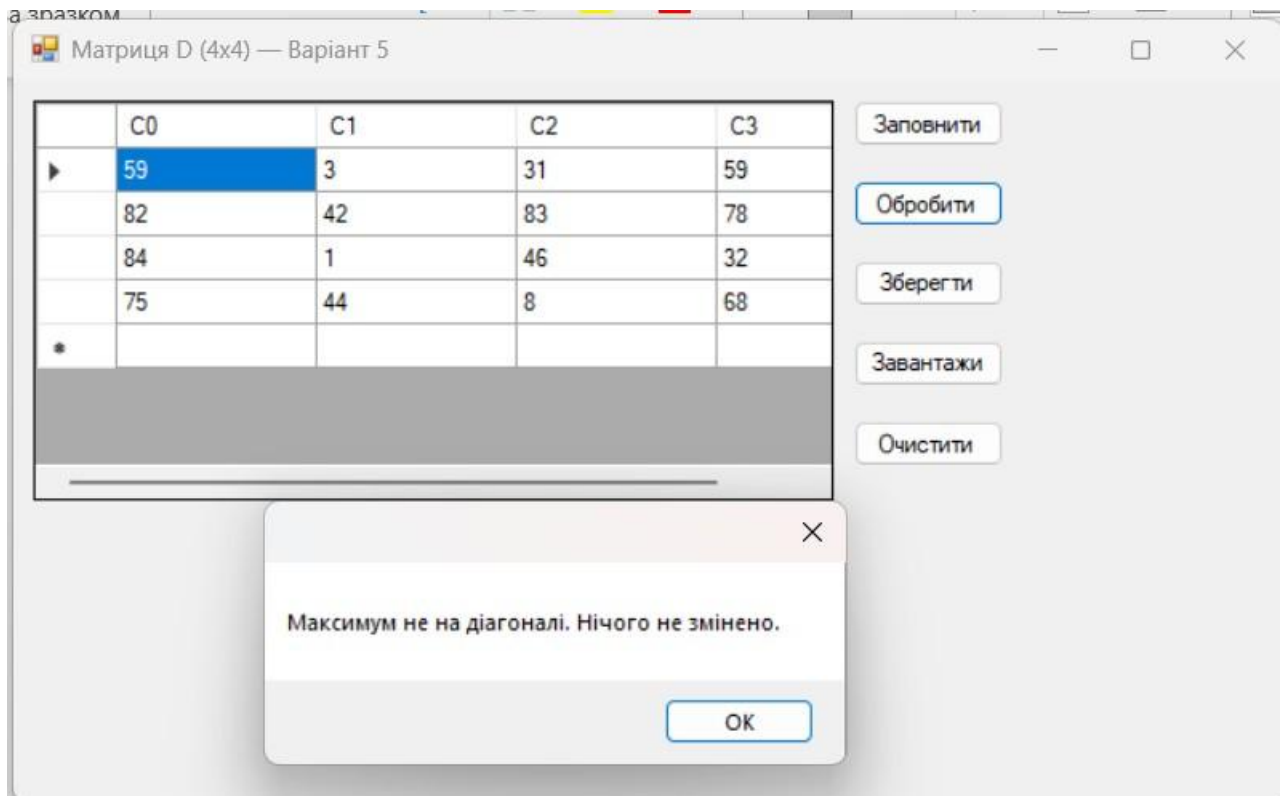


Рисунок 12.1 – Вікно результатів програми

Висновок: На цій лабораторній роботі я вивчила прийоми роботи з таблицями з використанням об'єкту DataGridView. Засвоїла методи створення, збереження та відкриття таблиць у форматі файлу XML. Ознайомилась з роботою об'єкту DataSet у проектах Windows Forms у Visual C++. Навчилась створювати об'єкти DataTable для збереження даних у табличному вигляді.