МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ



**Дніпровський національний університет  
залізничного транспорту імені академіка В. Лазаряна**

Кафедра «Комп’ютерні інформаційні технології»

**Лабораторна робота №6**

**з дисципліни «Основи програмної інженерії»**

**на тему: «Розробка програм із графічним інтерфейсом користувача»**

Виконав: студент гр. ПЗ1911

Сіньков Г.О.

Прийняла: ас. каф. КІТ

Куроп'ятник О. С.

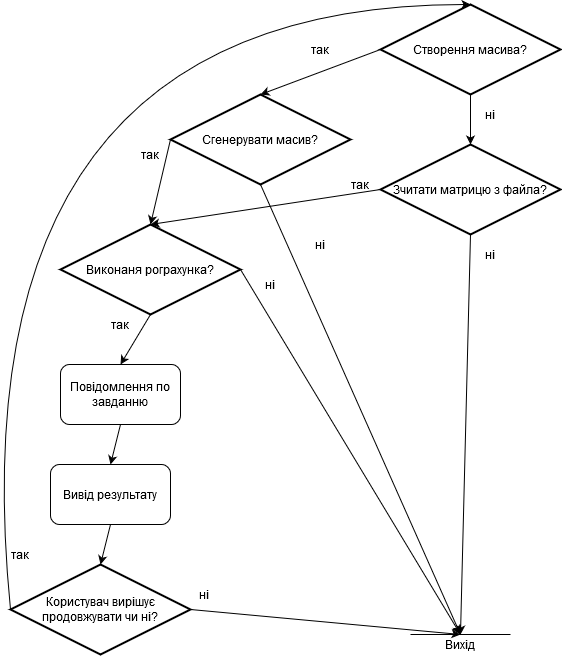
Дніпро, 2020

**Лабораторна робота №6**

Тема. Розробка програм із графічним інтерфейсом користувача.

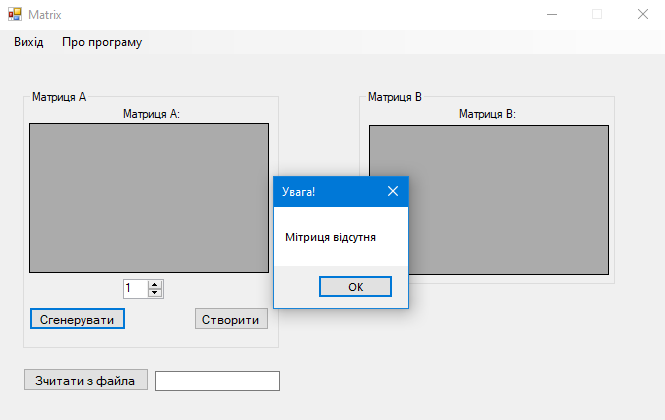
Мета. Отримати практичні навички розробки програм із графічним інтерфейсом користувача на основі механізму WindowsForms.

**Сценарій діалогу програми**

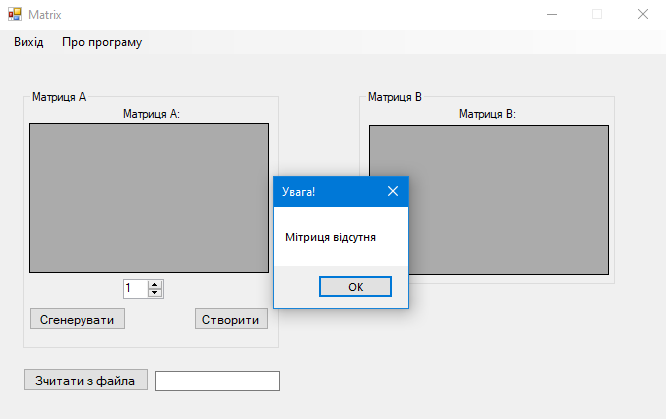


**Тести**

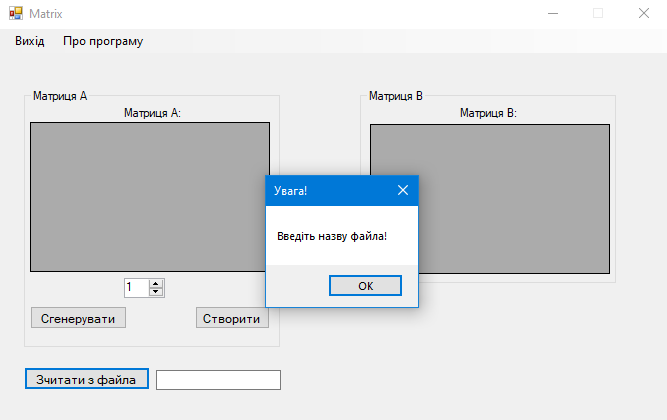
Генерація не створеного масива:



Виповнення дії над не створеною матрицею:

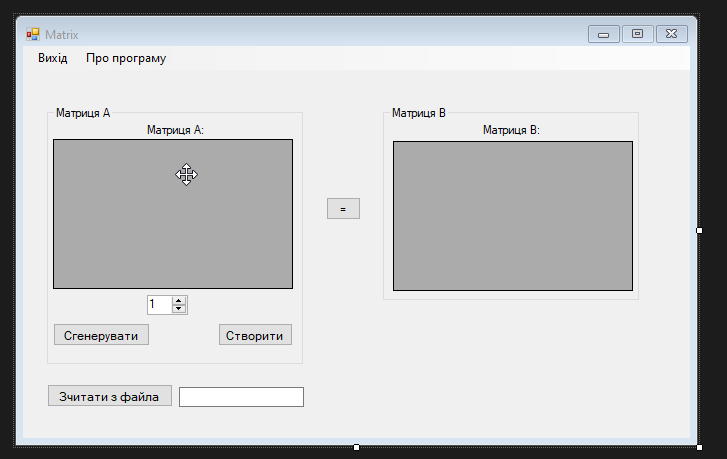


Зчитування назви яка небула написана:



**Текст програми**

Файл MyForm.h [Конструктор]



Файл MyForm.h

#pragma once

namespace Example\_1 {

using namespace System;

using namespace System::ComponentModel;

using namespace System::Collections;

using namespace System::Windows::Forms;

using namespace System::Data;

using namespace System::Drawing;

/// <summary>

/// Сводка для MyForm

/// </summary>

public ref class MyForm : public System::Windows::Forms::Form

{

public:

MyForm(void)

{

InitializeComponent();

//

//TODO: добавьте код конструктора

//

}

protected:

/// <summary>

/// Освободить все используемые ресурсы.

/// </summary>

~MyForm()

{

if (components)

{

delete components;

}

}

private: System::Windows::Forms::MenuStrip^ menuStrip1;

protected:

private: System::Windows::Forms::ToolStripMenuItem^ вихідToolStripMenuItem;

private: System::Windows::Forms::ToolStripMenuItem^ проПрограмуToolStripMenuItem;

private: System::Windows::Forms::ContextMenuStrip^ contextMenuStrip1;

private: System::Windows::Forms::Label^ label1;

private: System::Windows::Forms::DataGridView^ dataGridView1;

private: System::Windows::Forms::NumericUpDown^ numericUpDownSize;

private: System::Windows::Forms::Button^ buttonCreate;

private: System::Windows::Forms::Button^ buttonRandom;

private: System::Windows::Forms::GroupBox^ groupBox1;

private: System::Windows::Forms::GroupBox^ groupBox2;

private: System::Windows::Forms::DataGridView^ dataGridViewResult;

private: System::Windows::Forms::Label^ label2;

private: System::Windows::Forms::Button^ buttonReading;

private: System::Windows::Forms::TextBox^ textBoxReading;

private: System::Windows::Forms::Button^ buttonActions;

private: System::ComponentModel::BackgroundWorker^ backgroundWorker1;

private: System::ComponentModel::IContainer^ components;

private:

/// <summary>

/// Обязательная переменная конструктора.

/// </summary>

#pragma region Windows Form Designer generated code

/// <summary>

/// Требуемый метод для поддержки конструктора — не изменяйте

/// содержимое этого метода с помощью редактора кода.

/// </summary>

void InitializeComponent(void)

{

this->components = (gcnew System::ComponentModel::Container());

this->menuStrip1 = (gcnew System::Windows::Forms::MenuStrip());

this->вихідToolStripMenuItem = (gcnew System::Windows::Forms::ToolStripMenuItem());

this->проПрограмуToolStripMenuItem = (gcnew System::Windows::Forms::ToolStripMenuItem());

this->contextMenuStrip1 = (gcnew System::Windows::Forms::ContextMenuStrip(this->components));

this->label1 = (gcnew System::Windows::Forms::Label());

this->dataGridView1 = (gcnew System::Windows::Forms::DataGridView());

this->numericUpDownSize = (gcnew System::Windows::Forms::NumericUpDown());

this->buttonCreate = (gcnew System::Windows::Forms::Button());

this->buttonRandom = (gcnew System::Windows::Forms::Button());

this->groupBox1 = (gcnew System::Windows::Forms::GroupBox());

this->groupBox2 = (gcnew System::Windows::Forms::GroupBox());

this->dataGridViewResult = (gcnew System::Windows::Forms::DataGridView());

this->label2 = (gcnew System::Windows::Forms::Label());

this->buttonReading = (gcnew System::Windows::Forms::Button());

this->textBoxReading = (gcnew System::Windows::Forms::TextBox());

this->buttonActions = (gcnew System::Windows::Forms::Button());

this->backgroundWorker1 = (gcnew System::ComponentModel::BackgroundWorker());

this->menuStrip1->SuspendLayout();

(cli::safe\_cast<System::ComponentModel::ISupportInitialize^>(this->dataGridView1))->BeginInit();

(cli::safe\_cast<System::ComponentModel::ISupportInitialize^>(this->numericUpDownSize))->BeginInit();

this->groupBox1->SuspendLayout();

this->groupBox2->SuspendLayout();

(cli::safe\_cast<System::ComponentModel::ISupportInitialize^>(this->dataGridViewResult))->BeginInit();

this->SuspendLayout();

//

// menuStrip1

//

this->menuStrip1->Items->AddRange(gcnew cli::array< System::Windows::Forms::ToolStripItem^ >(2) {

this->вихідToolStripMenuItem,

this->проПрограмуToolStripMenuItem

});

this->menuStrip1->Location = System::Drawing::Point(0, 0);

this->menuStrip1->Name = L"menuStrip1";

this->menuStrip1->Size = System::Drawing::Size(667, 24);

this->menuStrip1->TabIndex = 0;

this->menuStrip1->Text = L"menuStrip1";

//

// вихідToolStripMenuItem

//

this->вихідToolStripMenuItem->Name = L"вихідToolStripMenuItem";

this->вихідToolStripMenuItem->Size = System::Drawing::Size(48, 20);

this->вихідToolStripMenuItem->Text = L"Вихід";

this->вихідToolStripMenuItem->Click += gcnew System::EventHandler(this, &MyForm::вихідToolStripMenuItem\_Click);

//

// проПрограмуToolStripMenuItem

//

this->проПрограмуToolStripMenuItem->Name = L"проПрограмуToolStripMenuItem";

this->проПрограмуToolStripMenuItem->Size = System::Drawing::Size(99, 20);

this->проПрограмуToolStripMenuItem->Text = L"Про програму";

this->проПрограмуToolStripMenuItem->Click += gcnew System::EventHandler(this, &MyForm::проПрограмуToolStripMenuItem\_Click);

//

// contextMenuStrip1

//

this->contextMenuStrip1->Name = L"contextMenuStrip1";

this->contextMenuStrip1->Size = System::Drawing::Size(61, 4);

//

// label1

//

this->label1->AutoSize = true;

this->label1->Location = System::Drawing::Point(97, 17);

this->label1->Name = L"label1";

this->label1->Size = System::Drawing::Size(64, 13);

this->label1->TabIndex = 2;

this->label1->Text = L"Матриця A:";

//

// dataGridView1

//

this->dataGridView1->ColumnHeadersHeightSizeMode = System::Windows::Forms::DataGridViewColumnHeadersHeightSizeMode::AutoSize;

this->dataGridView1->Location = System::Drawing::Point(6, 33);

this->dataGridView1->Name = L"dataGridView1";

this->dataGridView1->Size = System::Drawing::Size(240, 150);

this->dataGridView1->TabIndex = 3;

//

// numericUpDownSize

//

this->numericUpDownSize->Location = System::Drawing::Point(100, 189);

this->numericUpDownSize->Minimum = System::Decimal(gcnew cli::array< System::Int32 >(4) { 1, 0, 0, 0 });

this->numericUpDownSize->Name = L"numericUpDownSize";

this->numericUpDownSize->Size = System::Drawing::Size(41, 20);

this->numericUpDownSize->TabIndex = 4;

this->numericUpDownSize->Value = System::Decimal(gcnew cli::array< System::Int32 >(4) { 1, 0, 0, 0 });

this->numericUpDownSize->ValueChanged += gcnew System::EventHandler(this, &MyForm::numericUpDownSize\_ValueChanged);

//

// buttonCreate

//

this->buttonCreate->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif", 9, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,

static\_cast<System::Byte>(204)));

this->buttonCreate->Location = System::Drawing::Point(171, 217);

this->buttonCreate->Name = L"buttonCreate";

this->buttonCreate->Size = System::Drawing::Size(75, 23);

this->buttonCreate->TabIndex = 6;

this->buttonCreate->Text = L"Створити";

this->buttonCreate->UseVisualStyleBackColor = true;

this->buttonCreate->Click += gcnew System::EventHandler(this, &MyForm::buttonCreate\_Click);

//

// buttonRandom

//

this->buttonRandom->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif", 9, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,

static\_cast<System::Byte>(204)));

this->buttonRandom->Location = System::Drawing::Point(6, 217);

this->buttonRandom->Name = L"buttonRandom";

this->buttonRandom->Size = System::Drawing::Size(97, 23);

this->buttonRandom->TabIndex = 7;

this->buttonRandom->Text = L"Сгенерувати";

this->buttonRandom->UseVisualStyleBackColor = true;

this->buttonRandom->Click += gcnew System::EventHandler(this, &MyForm::buttonRandom\_Click);

//

// groupBox1

//

this->groupBox1->Controls->Add(this->dataGridView1);

this->groupBox1->Controls->Add(this->buttonRandom);

this->groupBox1->Controls->Add(this->label1);

this->groupBox1->Controls->Add(this->buttonCreate);

this->groupBox1->Controls->Add(this->numericUpDownSize);

this->groupBox1->Location = System::Drawing::Point(24, 60);

this->groupBox1->Name = L"groupBox1";

this->groupBox1->Size = System::Drawing::Size(256, 259);

this->groupBox1->TabIndex = 8;

this->groupBox1->TabStop = false;

this->groupBox1->Text = L"Матриця А";

//

// groupBox2

//

this->groupBox2->Controls->Add(this->dataGridViewResult);

this->groupBox2->Controls->Add(this->label2);

this->groupBox2->Location = System::Drawing::Point(360, 60);

this->groupBox2->Name = L"groupBox2";

this->groupBox2->Size = System::Drawing::Size(256, 195);

this->groupBox2->TabIndex = 9;

this->groupBox2->TabStop = false;

this->groupBox2->Text = L"Матриця В";

//

// dataGridViewResult

//

this->dataGridViewResult->ColumnHeadersHeightSizeMode = System::Windows::Forms::DataGridViewColumnHeadersHeightSizeMode::AutoSize;

this->dataGridViewResult->Location = System::Drawing::Point(10, 35);

this->dataGridViewResult->Name = L"dataGridViewResult";

this->dataGridViewResult->Size = System::Drawing::Size(240, 150);

this->dataGridViewResult->TabIndex = 3;

//

// label2

//

this->label2->AutoSize = true;

this->label2->Location = System::Drawing::Point(97, 17);

this->label2->Name = L"label2";

this->label2->Size = System::Drawing::Size(64, 13);

this->label2->TabIndex = 2;

this->label2->Text = L"Матриця B:";

//

// buttonReading

//

this->buttonReading->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif", 9, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,

static\_cast<System::Byte>(204)));

this->buttonReading->Location = System::Drawing::Point(24, 338);

this->buttonReading->Name = L"buttonReading";

this->buttonReading->Size = System::Drawing::Size(126, 23);

this->buttonReading->TabIndex = 10;

this->buttonReading->Text = L"Зчитати з файла";

this->buttonReading->UseVisualStyleBackColor = true;

this->buttonReading->Click += gcnew System::EventHandler(this, &MyForm::buttonReading\_Click);

//

// textBoxReading

//

this->textBoxReading->Location = System::Drawing::Point(156, 341);

this->textBoxReading->Name = L"textBoxReading";

this->textBoxReading->Size = System::Drawing::Size(125, 20);

this->textBoxReading->TabIndex = 11;

//

// buttonActions

//

this->buttonActions->Location = System::Drawing::Point(303, 151);

this->buttonActions->Name = L"buttonActions";

this->buttonActions->Size = System::Drawing::Size(35, 23);

this->buttonActions->TabIndex = 13;

this->buttonActions->Text = L"=";

this->buttonActions->UseVisualStyleBackColor = true;

this->buttonActions->Click += gcnew System::EventHandler(this, &MyForm::buttonActions\_Click);

//

// MyForm

//

this->AutoScaleDimensions = System::Drawing::SizeF(6, 13);

this->AutoScaleMode = System::Windows::Forms::AutoScaleMode::Font;

this->ClientSize = System::Drawing::Size(667, 392);

this->Controls->Add(this->buttonActions);

this->Controls->Add(this->textBoxReading);

this->Controls->Add(this->buttonReading);

this->Controls->Add(this->groupBox2);

this->Controls->Add(this->groupBox1);

this->Controls->Add(this->menuStrip1);

this->MainMenuStrip = this->menuStrip1;

this->MaximizeBox = false;

this->Name = L"MyForm";

this->StartPosition = System::Windows::Forms::FormStartPosition::CenterScreen;

this->Text = L"Matrix";

this->menuStrip1->ResumeLayout(false);

this->menuStrip1->PerformLayout();

(cli::safe\_cast<System::ComponentModel::ISupportInitialize^>(this->dataGridView1))->EndInit();

(cli::safe\_cast<System::ComponentModel::ISupportInitialize^>(this->numericUpDownSize))->EndInit();

this->groupBox1->ResumeLayout(false);

this->groupBox1->PerformLayout();

this->groupBox2->ResumeLayout(false);

this->groupBox2->PerformLayout();

(cli::safe\_cast<System::ComponentModel::ISupportInitialize^>(this->dataGridViewResult))->EndInit();

this->ResumeLayout(false);

this->PerformLayout();

}

#pragma endregion

private: System::Void вихідToolStripMenuItem\_Click(System::Object^ sender, System::EventArgs^ e);

private: System::Void проПрограмуToolStripMenuItem\_Click(System::Object^ sender, System::EventArgs^ e);

private: System::Void buttonRandom\_Click(System::Object^ sender, System::EventArgs^ e);

private: System::Void buttonCreate\_Click(System::Object^ sender, System::EventArgs^ e);

private: System::Void buttonReading\_Click(System::Object^ sender, System::EventArgs^ e);

private: System::Void numericUpDownSize\_ValueChanged(System::Object^ sender, System::EventArgs^ e);

private: void Show();

private: void ShowResult(int \*\*res);

private: System::Void buttonActions\_Click(System::Object^ sender, System::EventArgs^ e);

};

}

Файл MyForm.cpp

#include <string>

#include "MyForm.h"

#include "matrix.h"

using namespace System;

using namespace System::Windows::Forms;

[STAThreadAttribute]

int main(array<String^>^ args) {

Application::EnableVisualStyles();

Application::SetCompatibleTextRenderingDefault(false);

Example\_1::MyForm form;

Application::Run(% form);

}

//Дание

int \*\*arr;

int size;

int max, min;

System::Void Example\_1::MyForm::вихідToolStripMenuItem\_Click(System::Object ^ sender, System::EventArgs ^ e) {

Application::Exit();

return System::Void();

}

System::Void Example\_1::MyForm::проПрограмуToolStripMenuItem\_Click(System::Object ^ sender, System::EventArgs ^ e) {

MessageBox::Show("Завдання:\nЯкщо максимум серед елементів, що розташовані вище бічної діагоналі квадратної матриці А, менше за мінімум серед елементів, що розташовані нижче головної діагоналі, то транспонуйте матрицю, інакше знайдіть А2.","Інформація про програму");

return System::Void();

}

System::Void Example\_1::MyForm::buttonRandom\_Click(System::Object ^ sender, System::EventArgs ^ e) {

if (size == 0) {

MessageBox::Show("Мітриця відсутня","Увага!");

return;

}

FillingArray(arr, size);

Show();

dataGridView1->AutoResizeRowHeadersWidth(DataGridViewRowHeadersWidthSizeMode::AutoSizeToAllHeaders);//ячейки

dataGridView1->AutoResizeColumns();

return System::Void();

}

System::Void Example\_1::MyForm::buttonCreate\_Click(System::Object ^ sender, System::EventArgs ^ e) {

size = Convert::ToInt32(numericUpDownSize->Value);

arr = new int\*[size];

for (int i = 0; i < size; i++)

arr[i] = new int[size];

for (int i = 0; i < size; i++)

for (int j = 0; j < size; j++)

arr[i][j] = 0;

//Создаем таблицу

dataGridView1->RowCount = size;

dataGridView1->ColumnCount = size;

Show();

dataGridView1->AutoResizeRowHeadersWidth(DataGridViewRowHeadersWidthSizeMode::AutoSizeToAllHeaders);//ячейки

dataGridView1->AutoResizeColumns();

return System::Void();

}

System::Void Example\_1::MyForm::buttonReading\_Click(System::Object ^ sender, System::EventArgs ^ e) {

if (textBoxReading->Text == "") {

MessageBox::Show("Введіть назву файла!","Увага!");

return;

}

//Зчитуемо назву файла

std::string filename;

Convert\_String\_to\_string(textBoxReading->Text, filename);

//Зчитуемо дані з файла

ReadingMatrix(arr, size, filename);

//Создаем таблицу

dataGridView1->RowCount = size;

dataGridView1->ColumnCount = size;

Show();

dataGridView1->AutoResizeRowHeadersWidth(DataGridViewRowHeadersWidthSizeMode::AutoSizeToAllHeaders);//ячейки

dataGridView1->AutoResizeColumns();

return System::Void();

}

System::Void Example\_1::MyForm::numericUpDownSize\_ValueChanged(System::Object ^ sender, System::EventArgs ^ e) {

return System::Void();

}

void Example\_1::MyForm::Show() {

for (int i = 0; i < size; i++) {

for (int j = 0; j < size; j++) {

//назавние таблицы в верхнем углду

dataGridView1->TopLeftHeaderCell->Value = "Матриці А";

//вывод номеров строк

dataGridView1->Rows[i]->HeaderCell->Value = Convert::ToString(i + 1);

//вывод номеров столбцов

dataGridView1->Columns[j]->HeaderCell->Value = Convert::ToString(j + 1);

//вывод значение матрицы

dataGridView1->Rows[i]->Cells[j]->Value = arr[i][j];

}

}

}

void Example\_1::MyForm::ShowResult(int \*\*res) {

for (int i = 0; i < size; i++) {

for (int j = 0; j < size; j++) {

//назавние таблицы в верхнем углду

dataGridViewResult->TopLeftHeaderCell->Value = "Матриці B";

//вывод номеров строк

dataGridViewResult->Rows[i]->HeaderCell->Value = Convert::ToString(i + 1);

//вывод номеров столбцов

dataGridViewResult->Columns[j]->HeaderCell->Value = Convert::ToString(j + 1);

//вывод значение матрицы

dataGridViewResult->Rows[i]->Cells[j]->Value = res[i][j];

}

}

}

System::Void Example\_1::MyForm::buttonActions\_Click(System::Object ^ sender, System::EventArgs ^ e) {

if (size == 0) {

MessageBox::Show("Мітриця відсутня", "Увага!");

return;

}

int \*\*result;

min = FindingMin(arr, size);//знаходження мінімального числа по завданю

max = FindingMax(arr, size);//знаходження максимального числа по завданю

if (max < min) {//Так як максимальне число меньше мінімального, то ми транспонуємо матрицю

MessageBox::Show("Так як максимальне число меньше мінімального, то ми транспонуємо матрицю","По завданю!");

result = TransposeArray(arr, size);

}

else {//Так як максимальне число більше мінімального, то ми матрицю підносимо у квадрат

MessageBox::Show("Так як максимальне число більше мінімального, то ми матрицю підносимо у квадрат", "По завданю!");

result = SquareArray(arr, size);

}

//Создаем таблицу

dataGridViewResult->RowCount = size;

dataGridViewResult->ColumnCount = size;

ShowResult(result);

dataGridViewResult->AutoResizeRowHeadersWidth(DataGridViewRowHeadersWidthSizeMode::AutoSizeToAllHeaders);//ячейки

dataGridViewResult->AutoResizeColumns();

return System::Void();

}

Файл matrix.h

#ifndef \_\_FUNCTION\_H\_\_

#define \_\_FUNCTION\_H\_\_

using namespace System;

int\*\* New2DArray(const int size);

void FillingArray(int \*\*arr, const int size);

int FindingMax(int \*\*arr, const int size);

int FindingMin(int \*\*arr, const int size);

int\*\* TransposeArray(int \*\*arr, const int size);

int\*\* SquareArray(int \*\*arr, const int size);

void ReadingMatrix(int \*\*(&arr), int &size, std::string filename);

std::string& Convert\_String\_to\_string(String^ s, std::string& os);//конвертируем System::string^ в std::string

String^ Convert\_string\_to\_String(std::string& os, String^ s);//конвертируем std::string в System::string^

#endif

Файл matrix.cpp

#include <iostream>

#include <string>

#include <fstream>

#include <random>

#include "matrix.h"

using namespace System;

using namespace System::Windows::Forms;

int\*\* New2DArray(const int size) {//створення масива

int\*\* Arr = new int\*[size];

for (int i = 0; i < size; ++i)

Arr[i] = new int[size];

return Arr;

}

void FillingArray(int \*\*arr, const int size) {//заповнення масива випадковими числами

std::random\_device rd;

std::mt19937 gen(rd());

std::uniform\_int\_distribution<int> dis(0, 10);

for (int i = 0; i < size; i++)

{

for (int j = 0; j < size; j++)

{

arr[i][j] = dis(gen);

}

}

}

int FindingMax(int \*\*arr, const int size) {//знаходження максимального числа

int max, m = 1;

max = arr[0][0];

for (int i = 0; i < size - 1; i++, m++)

{

for (int j = 0; j < size - m; j++)

{

if (arr[i][j] > max)

{

max = arr[i][j];

}

}

}

return max;

}

int FindingMin(int \*\*arr, const int size) {//знаходження мінімального числа

int min, m = 1;

min = arr[1][0];

for (int i = 1; i < size - 1; i++, m++)

{

for (int j = 0; j < m; j++)

{

if (arr[i][j] < min)

{

min = arr[i][j];

}

}

}

return min;

}

int\*\* TransposeArray(int \*\*arr, const int size) {//транспортування матриці

int\*\* res = New2DArray(size);

for (int i = 0; i < size; ++i)

for (int j = 0; j < size; ++j)

res[j][i] = arr[i][j];

return res;

}

int\*\* SquareArray(int \*\*arr, const int size) {//матриця в квадраті

int \*\*temp = New2DArray(size);

for (int i = 0; i < size; i++)

for (int j = 0; j < size; j++)

temp[i][j] = 0;

for (int i = 0; i < size; i++)

for (int j = 0; j < size; j++)

for (int k = 0; k < size; k++)

temp[i][j] += arr[i][k] \* arr[k][j];

return temp;

}

void ReadingMatrix(int \*\*(&arr), int &size, std::string filename) {//читання даних з файла

std::ifstream reading(filename);

if (reading) {

//зчитуємо розмірність матриці

reading >> size;

//Виділяємо память під матрицю

arr = new int\*[size];

for (int i = 0; i < size; i++)

arr[i] = new int[size];

//Хчитуємо матрицю

for (int i = 0; i < size; i++)

for (int j = 0; j < size; j++)

reading >> arr[i][j];

}

else

MessageBox::Show("Ошибка відкритя файла!","Ошибка!");

reading.close();

}

std::string& Convert\_String\_to\_string(String^ s, std::string &os) {

using namespace Runtime::InteropServices;

const char \*chars = (const char\*)(Marshal::StringToHGlobalAnsi(s)).ToPointer();

os = chars;

Marshal::FreeHGlobal(IntPtr((void\*)chars));

return os;

}

String^ Convert\_string\_to\_String(std::string &os, String^ s) {

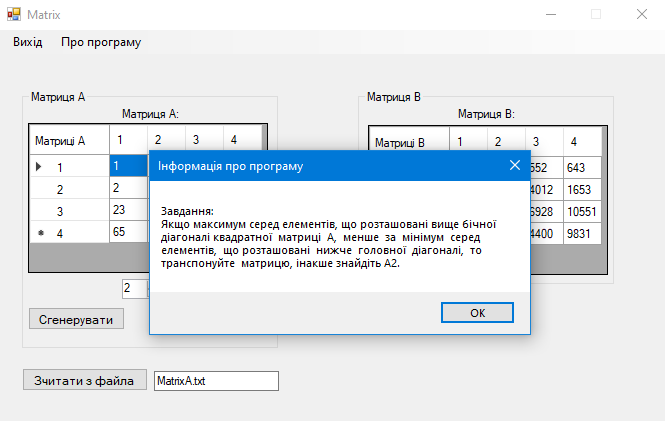
s = gcnew System::String(os.c\_str());

return s;

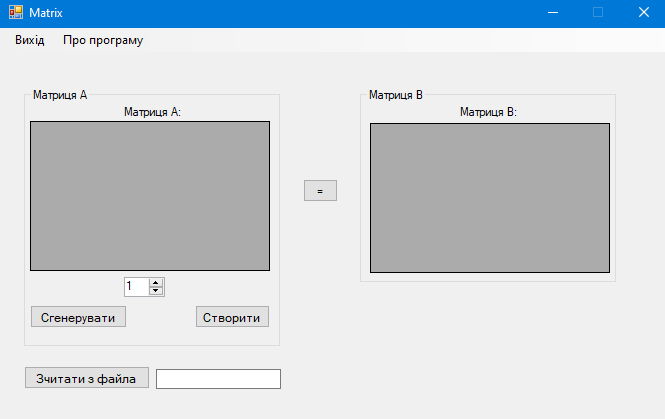
}

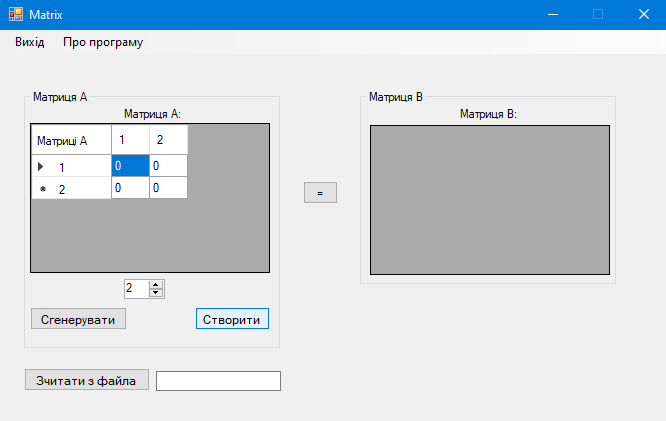
**Результати виповнення програми**

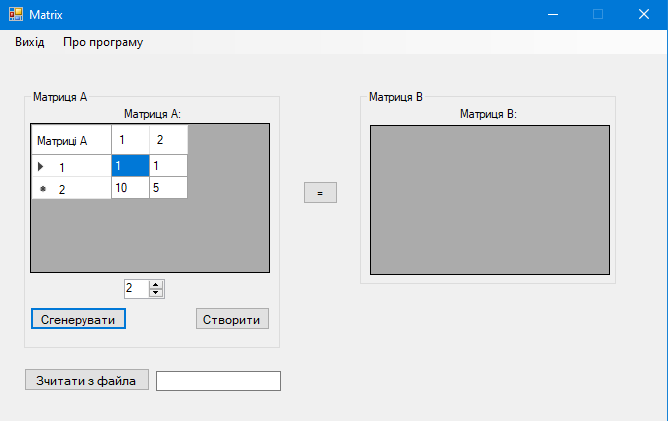
Інформація про програму:



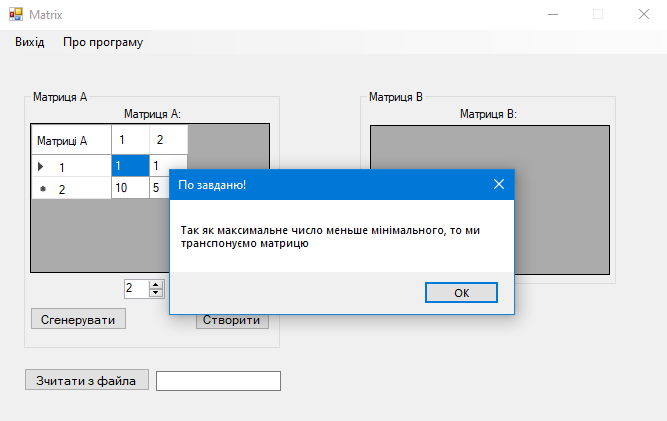
Створення та заповнення матриці:

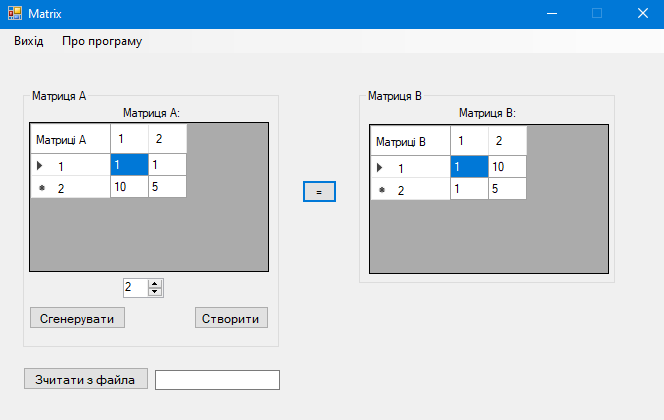




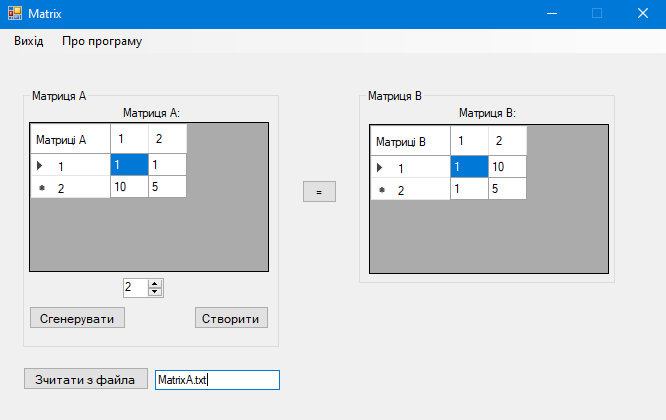


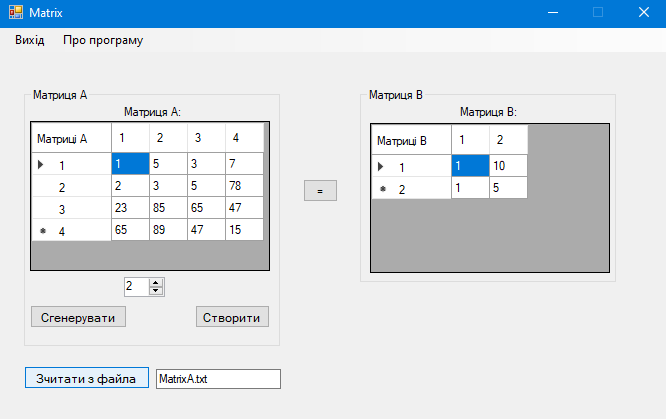
Виповнення завдання по натіску на знак ‘=’:

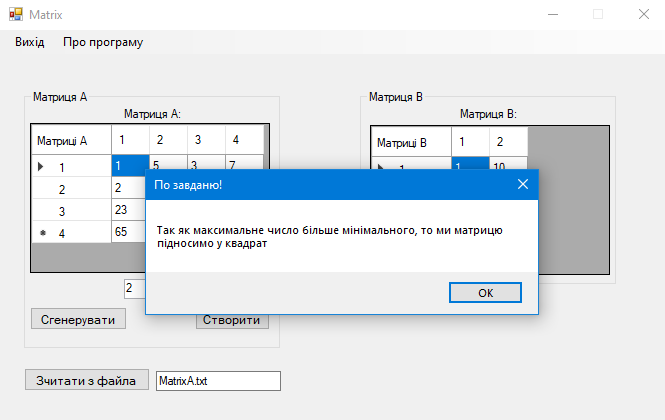


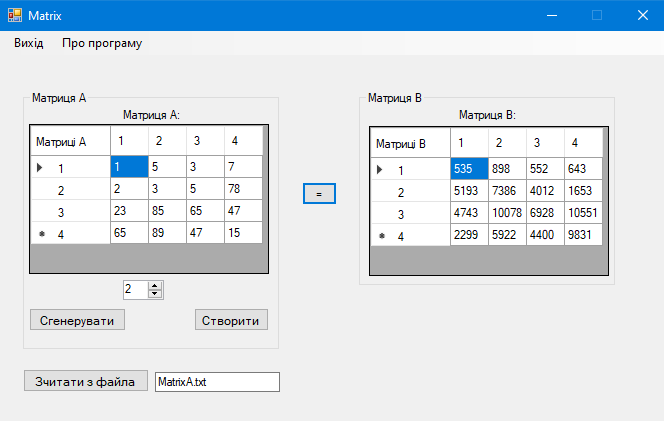


Зчитування з файла:









**Висновок**: **Графічний інтерфейс користувача (ГІП), графічний користувальницький інтерфейс (ДПІ) (англ. Graphical user interface, GUI)** - система засобів для взаємодії користувача з комп'ютером, заснована на представленні всіх доступних користувачеві системних об'єктів і функцій у вигляді графічних компонентів екрану (вікон, значків , меню, кнопок, списків і т. п.).

На відміну від інтерфейсу командного рядка, в GUI користувач має довільний доступ (за допомогою пристроїв введення - клавіатури, миші, джойстика і т. П.) До всіх видимих ​​екранних об'єктів (елементів інтерфейсу) і здійснює безпосереднє маніпулювання ними. Найчастіше елементи інтерфейсу в GUI реалізовані на основі метафор і відображають їх призначення і властивості, що полегшує розуміння і освоєння програм непідготовленими користувачами.

Графічний інтерфейс користувача є частиною призначеного для користувача інтерфейсу і визначає взаємодію з користувачем на рівні візуалізованою інформації.