МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

**Український державний університет науки і технологій**

Кафедра «Комп’ютерні інформаційні технології»

**Лабораторна робота №4**

**з дисципліни «Основи програмної інженерії»**

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

Виконав: студент гр. ПЗ2112

Субботін С.П.

Прийняла:   
доц. Куроп’ятник О. С.

Дніпро, 2022

Тема. Розробка програм із графічним інтерфейсом користувача.

Мета роботи. Отримати практичні навички розробки програм із графічним інтерфейсом користувача на основі механізму Windows Forms.

Постановка завдання:

Знайти середнє арифметичне сум рядків, в яких кількість додатних елементів більше кількості від’ємних.

Текст програми:

1)

#pragma once

#include "Functions1.h"

#include "Func.h"

namespace Project1 {

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>

/// Summary for MyForm

/// /summary>

public

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

public:

MyForm(void) {

InitializeComponent();

//

// TODO: Add the constructor code here

//

}

protected:

/// <summary>

/// Clean up any resources being used.

/// /summary>

~MyForm() {

if (matrix -> data != nullptr) deleting\_matrix(matrix);

if (components) {

delete components;

}

}

private:

Matrix \* matrix = nullptr;

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

System::Windows::Forms::NumericUpDown ^ Rows;

System::Windows::Forms::NumericUpDown ^ Cols;

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

System::Windows::Forms::Button ^ CreateMatrix;

System::Windows::Forms::ComboBox ^ FillMatrix;

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

System::Windows::Forms::Button ^ ResultMatrix;

/// <summary>

/// Required designer variable.

/// /summary>

System::ComponentModel::Container ^ components;

#pragma region Windows Form Designer generated code

/// <summary>

/// Required method for Designer support - do not modify

/// the contents of this method with the code editor.

/// /summary>

void InitializeComponent(void) {

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

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

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

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

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

this -> FillMatrix = (gcnew System::Windows::Forms::ComboBox());

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

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

(cli::safe\_cast<System::ComponentModel::ISupportInitialize^>(

this -> dataGridView1))-> BeginInit();

(cli::safe\_cast<System::ComponentModel::ISupportInitialize^>(

this-> Rows))-> BeginInit();

(cli::safe\_cast<System::ComponentModel::ISupportInitialize^>(

this -> Cols))-> BeginInit();

this -> groupBox1 -> SuspendLayout();

this -> SuspendLayout();

//

// dataGridView1

//

this-> dataGridView1 -> AllowUserToAddRows = false;

this -> dataGridView1 -> AllowUserToDeleteRows = false;

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

this -> dataGridView1 -> EditMode =System::Windows::Forms::DataGridViewEditMode::EditProgrammatically;

this -> dataGridView1 -> Location = System::Drawing::Point(40, 150);

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

this -> dataGridView1 -> Size = System::Drawing::Size(350, 260);

this -> dataGridView1 -> TabIndex = 1;

this->ForeColor = System::Drawing::Color::Gray;

// numericUpDownRows

//

this -> Rows -> Font =(gcnew System::Drawing::Font(L"Microsoft Sans Serif", 11));

this -> Rows -> Location = System::Drawing::Point(21, 25);

this -> Rows -> Maximum =System::Decimal(gcnew cli::array<System::Int32>(4) { 10, 0, 0, 0 });

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

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

this -> Rows -> Size = System::Drawing::Size(44, 24);

this -> Rows -> TabIndex = 2;

this -> Rows -> Value =

System::Decimal(gcnew cli::array<System::Int32>(4) { 5, 0, 0, 0 });

//

// numericUpDownCols

//

this -> Cols -> Font =(gcnew System::Drawing::Font(L"Microsoft Sans Serif", 11));

this -> Cols -> Location = System::Drawing::Point(81, 25);

this -> Cols -> Maximum =System::Decimal(gcnew cli::array<System::Int32>(4) { 10, 0, 0, 0 });

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

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

this -> Cols -> Size = System::Drawing::Size(44, 24);

this -> Cols -> TabIndex = 3;

this-> Cols -> Value =System::Decimal(gcnew cli::array<System::Int32>(4) { 5, 0, 0, 0 });

//

// groupBox1

//

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

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

this -> groupBox1 -> Font =

(gcnew System::Drawing::Font(L"Microsoft Sans Serif", 12));

this -> groupBox1 -> Location = System::Drawing::Point(250, 30);

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

this -> groupBox1 -> Size = System::Drawing::Size(140, 60);

this -> groupBox1 -> TabIndex = 4;

this -> groupBox1 -> TabStop = false;

this -> groupBox1 -> Text = L"Розмірність";

//

// GenerateBtn

//

this -> CreateMatrix -> Font =

(gcnew System::Drawing::Font(L"Microsoft Sans Serif", 12));

this -> CreateMatrix -> Location = System::Drawing::Point(40, 90);

this -> CreateMatrix -> Name = L"Create";

this -> CreateMatrix -> Size = System::Drawing::Size(180, 40);

this -> CreateMatrix -> TabIndex = 5;

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

this -> CreateMatrix -> UseVisualStyleBackColor = true;

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

//

// FillMethodBox

//

this -> FillMatrix -> DropDownStyle =

System::Windows::Forms::ComboBoxStyle::DropDownList;

this -> FillMatrix -> Font =

(gcnew System::Drawing::Font(L"Microsoft Sans Serif", 12));

this -> FillMatrix -> FormattingEnabled = true;

this -> FillMatrix -> Items -> AddRange(

gcnew cli::array<System::Object^>(2) { L"випадковими числами", L"власноруч" });

this -> FillMatrix -> Location = System::Drawing::Point(100, 38);

this -> FillMatrix -> MaxDropDownItems = 2;

this -> FillMatrix -> Name = L"Заповнити";

this -> FillMatrix -> Size = System::Drawing::Size(120, 28);

this -> FillMatrix -> TabIndex = 8;

//

// label1

//

this -> label1 -> AutoSize = true;

this -> label1 -> Font =

(gcnew System::Drawing::Font(L"Microsoft Sans Serif", 13));

this -> label1 -> Location = System::Drawing::Point(36, 40);

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

this -> label1 -> Size = System::Drawing::Size(33, 22);

this -> label1 -> TabIndex = 9;

this -> label1 -> Text = L"Метод";

//

// ResultBtn

//

this-> ResultMatrix -> Font =

(gcnew System::Drawing::Font(L"Microsoft Sans Serif", 10));

this -> ResultMatrix -> Location = System::Drawing::Point(40, 432);

this -> ResultMatrix -> Name = L"Result";

this -> ResultMatrix -> Size = System::Drawing::Size(350, 77);

this-> ResultMatrix -> TabIndex = 10;

this-> ResultMatrix -> Text =

L" Середнє арифметичне сум рядків, "

L"в яких кількість додатних елементів"

L"більше кількості від’ємних";

this -> ResultMatrix -> UseVisualStyleBackColor = true;

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

//

// MyForm

//

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

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

this -> ClientSize = System::Drawing::Size(434, 531);

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

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

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

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

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

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

this -> MaximizeBox = false;

this -> MaximumSize = System::Drawing::Size(450, 570);

this -> MinimumSize = System::Drawing::Size(450, 570);

this -> Name = L"MyForm";

this -> Text = L"MyForm";

this->BackColor = System::Drawing::Color::LightYellow;

(cli::safe\_cast<System::ComponentModel::ISupportInitialize^>(

this -> dataGridView1))-> EndInit();

(cli::safe\_cast<System::ComponentModel::ISupportInitialize^>(

this -> Rows)) -> EndInit();

(cli::safe\_cast<System::ComponentModel::ISupportInitialize^>(

this -> Cols))-> EndInit();

this -> groupBox1 -> ResumeLayout(false);

this -> ResumeLayout(false);

this -> PerformLayout();

}

#pragma endregion

System::Void GenerateBtn\_Click(System::Object ^ sender,

System::EventArgs ^ e);

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

};

} // namespace bope4

2)

#include "Form.h"

using namespace System;

using namespace System::Windows::Forms;

[STAThreadAttribute] int main(array<String^> ^ args) {

Application::EnableVisualStyles();

Application::SetCompatibleTextRenderingDefault(false);

Project1::MyForm form;

Application::Run(% form);

return 0;

}

System::Void Project1::MyForm::GenerateBtn\_Click(System::Object ^ sender,

System::EventArgs ^ e) {

if (matrix != nullptr) {

deleting\_matrix(matrix);

matrix = nullptr;

}

matrix = creating\_matrix(Convert::ToInt32(Rows -> Value),

Convert::ToInt32(Cols -> Value));

if (FillMatrix -> SelectedItem == "випадковими числами") {

dataGridView1 -> EditMode = DataGridViewEditMode::EditProgrammatically;

randomization(matrix);

}else if (FillMatrix -> SelectedItem == "власноруч") {

dataGridView1 -> EditMode = DataGridViewEditMode::EditOnEnter;

}

else {

MessageBox::Show("Оберіть метод заповнення матриці!", "Помилка!", MessageBoxButtons::OK,

MessageBoxIcon::Exclamation);

return System::Void();

}

dataGridView1 -> RowCount = matrix -> rows;

dataGridView1 -> ColumnCount = matrix -> cols;

for (int i = 0; i < matrix ->rows; i ++ ) {

for (int j = 0; j < matrix ->cols; j ++ ) {

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

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

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

}

}

dataGridView1 -> AutoResizeRowHeadersWidth(

DataGridViewRowHeadersWidthSizeMode::AutoSizeToAllHeaders);

dataGridView1 -> AutoResizeColumns();

return System::Void();

}

System::Void Project1::MyForm::ResultBtn\_Click(System::Object ^ sender,

System::EventArgs ^ e) {

if (dataGridView1 -> EditMode == DataGridViewEditMode::EditOnEnter)

for (int i = 0; i < matrix ->rows; i ++ )

for (int j = 0; j < matrix ->cols; j ++ )

matrix -> data[i][j] =

Convert::ToInt32(dataGridView1 -> Rows[i] -> Cells[j] -> Value);

MessageBox::Show(Convert::ToString(processing(matrix)), "Результат обчислення.",

MessageBoxButtons::OK, MessageBoxIcon::Information);

return System::Void();

}

**3)**

#pragma once

#ifndef FUNC\_H

#define FUNC\_H

#include <cstdlib>

#include <ctime>

struct Matrix {

size\_t rows = 0;

size\_t cols = 0;

int\*\* data = nullptr;

};

Matrix \* creating\_matrix(size\_t rows, size\_t cols);

void randomization(Matrix\*);

void deleting\_matrix(Matrix\*);

#endif

**4)**

#pragma once

#ifndef FUNCTIONS1\_H

#define FUNCTIONS1\_H

#include "Func.h"

bool checking(const int\*, const size\_t);

float processing(Matrix\*);

#endif

**5)**

#include "Func.h"

int random\_numbers\_area(int min, int max) {

static const double fraction = 1.0 / (static\_cast<double>(RAND\_MAX) + 1.0);

return static\_cast<int>(rand() \* fraction \* (max - min + 1) + min);

}

Matrix \* creating\_matrix(size\_t rows, size\_t cols) {

Matrix \* M = new Matrix;

M -> rows = rows;

M -> cols = cols;

M -> data = new int\* [M -> rows];

for (size\_t i = 0; i < M ->rows; i ++ ) M -> data[i] = new int[M -> cols]{};

return M;

}

void randomization(Matrix\* M) {

srand(static\_cast<unsigned int>(time(0)));

for (size\_t i = 0; i < M->rows; i++ )

{

for (size\_t j = 0; j < M->cols; j ++ )

{

M->data[i][j] = random\_numbers\_area(-20, 20);

}

}

}

void deleting\_matrix(Matrix\* M) {

for (size\_t i = 0; i < M->rows; i++) {

delete[] M->data[i];

}

delete[] M -> data;

M -> rows = 0;

M -> cols = 0;

M -> data = nullptr;

}

**6)**

#include "Functions1.h"

bool checking(const int\* row, const size\_t size) {

unsigned short Positive = 0;

unsigned short Negative = 0;

for (size\_t i = 0; i < size; i ++ ) {

if (row[i] > 0)

Positive ++ ;

else if (row[i] < 0)

Negative ++ ;

}

return Positive > Negative;

}

float processing(Matrix \* M) {

float Rows = 0;

float Sum = 0;

for (size\_t i = 0; i < (M ->rows); i ++ ) {

if (checking(M -> data[i], M -> cols)) {

int rowSum = 0;

for (size\_t j = 0; j < M ->cols; j ++ ) {

rowSum += M -> data[i][j];

}

Rows ++ ;

Sum += rowSum;

}

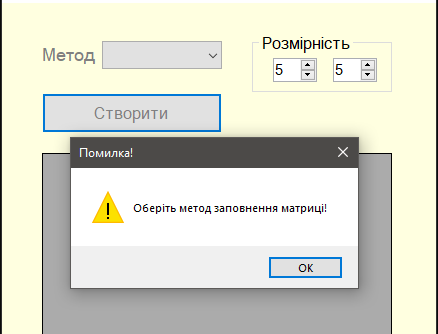
}

return Sum / Rows;

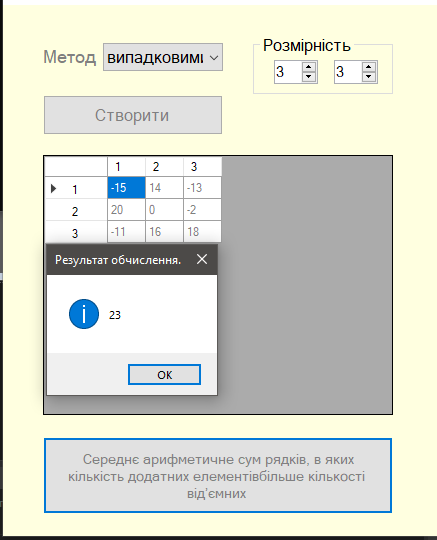
}

Скріншоти тестів:

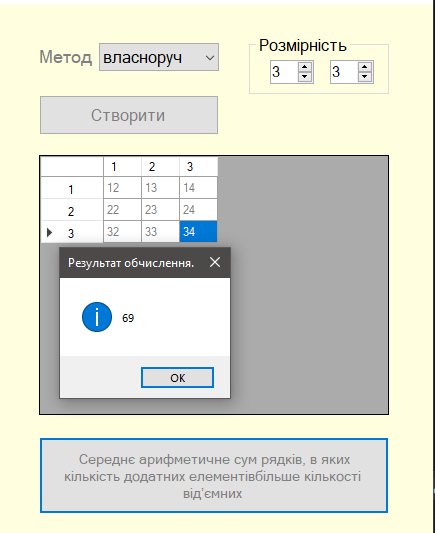
При спробі обчислити матрицю, не ввівши ніякі значення, буде помилка.

****

Введення значення випадковими числами.



Власноруч введення.



**Висновок:** У Windows Forms є безліч інших можливостей, які спрощують та прискорюють реалізацію спільних завдань, таких як створення діалогових вікон, друк, додавання довідки та документації, а також локалізація програм різними мовами. Крім того, у Windows Forms застосовується ефективна система безпеки .NET Framework. Завдяки ній можна створювати більш надійні програми.