**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»**

Інститут **ІКНІ**

Кафедра **ПЗ**



**ЗВІТ**

**Про виконання лабораторної роботи №6**

*“Перевантаження функцій і операцій, дружні функції”*

**З дисципліни:** “Об`єктно-орієнтоване програмування”

**Лектор:**

Доцент каф. ПЗ

Коротєєва Т.О.

**Виконав:**

ст. гр. ПЗ-11

Мартинюк Н.В.

**Прийняла:**

Доцент. каф. ПЗ

Коротєєва Т.О.

«\_\_\_» \_\_\_\_\_\_ 2020 р.

∑ = \_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_

Львів – 2020

**Тема:** Перевантаження функцій і операцій, дружні функції.

**Мета:** Навчитись використовувати механізм перевантаження функцій та операцій. Навчитися створювати та використовувати дружні функції. Ознайомитися зі статичними полями і методами, та навчитись їх використовувати.

**Завдання для лабораторної роботи:**

На основі класу з попередньої лабораторної:

1. Перевантажити як мінімум три функції-члени з попереднього завдання.
2. Перевантажити операції згідно з варіантом. Для операцій, для яких не вказані символи, вибрати символи самостійно.
3. Створити дружні функції згідно з варіантом.
4. Стоврити статичні поля та статичні методи згідно з варіантом.
5. Продемонструвати розроблені можливості класу завдяки створеному віконному застосуванню.
6. Оформити звіт до лабораторної роботи.

**Індивідуальне завдання:**

**Варіант 7.**

Клас Complex – комплексне число. Клас повинен містити функції-члени, які реалізовують: а) Додавання; б) Віднімання; в) Множення; г) Піднесення до n-го степеня; ґ) Знаходження модуля комплексного числа; д) Задання значень полів; е) Зчитування(отримання значень полів); є) Представлення в тригонометричній формі; ж) Введення компелксного числа з форми; з) Виведення комплексного числа на форму.

**Перевантажити операції, як функції-члени:**

Додавання

Віднімання

Множення

Знаходження модуля

**Перевантажити операції, як дружні функції:**

Введення комплексного числа з форми (“<<”)

Виведення комплексного числа на форму (“>>”)

Більше (“>”)

Менше (“<”)

Рівне (“==”) (При порівнянні порівнювати модулі комплексних чисел)

**Створити статичне поле, в якому б містилась інформація про кі-сть створених об’єктів, а також статичні ф-ції для роботи з цим полем.**

**Код програми:**

**MyForm.cpp**

#include "MyForm.h"

#include <Windows.h>

using namespace lab06;

int WINAPI WinMain(HINSTANCE, HINSTANCE, LPSTR, int)

{

Application::EnableVisualStyles();

Application::SetCompatibleTextRenderingDefault(false);

Application::Run(gcnew MyForm);

return 0;

}

**MyForm.h**

#include "Complex.h"

#include <string.h>

#include <string>

namespace lab06 {

int counter = 0;

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::TextBox^ textBox1;

protected:

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

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

private: System::Windows::Forms::TextBox^ textBox2;

private: System::Windows::Forms::Label^ label3;

private: System::Windows::Forms::Label^ label4;

private: System::Windows::Forms::TextBox^ textBox3;

private: System::Windows::Forms::Label^ label5;

private: System::Windows::Forms::Label^ label6;

private: System::Windows::Forms::TextBox^ textBox4;

private: System::Windows::Forms::TextBox^ textBox5;

private: System::Windows::Forms::Button^ button1;

private: System::Windows::Forms::Button^ button2;

private: System::Windows::Forms::TextBox^ textBox6;

private: System::Windows::Forms::Button^ button3;

private: System::Windows::Forms::Button^ button4;

private: System::Windows::Forms::Button^ button5;

private: System::Windows::Forms::Button^ button6;

private: System::Windows::Forms::Button^ button7;

private: System::Windows::Forms::Button^ button8;

private: System::Windows::Forms::Button^ button9;

private: System::Windows::Forms::Button^ button10;

private: System::Windows::Forms::Button^ button11;

private: System::Windows::Forms::Button^ button12;

private: System::Windows::Forms::Button^ button13;

protected:

private:

/// <summary>

/// Обязательная переменная конструктора.

/// </summary>

System::ComponentModel::Container^ components;

#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->textBox1 = (gcnew System::Windows::Forms::TextBox());

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

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

this->textBox2 = (gcnew System::Windows::Forms::TextBox());

this->label3 = (gcnew System::Windows::Forms::Label());

this->label4 = (gcnew System::Windows::Forms::Label());

this->textBox3 = (gcnew System::Windows::Forms::TextBox());

this->label5 = (gcnew System::Windows::Forms::Label());

this->label6 = (gcnew System::Windows::Forms::Label());

this->textBox4 = (gcnew System::Windows::Forms::TextBox());

this->textBox5 = (gcnew System::Windows::Forms::TextBox());

this->button1 = (gcnew System::Windows::Forms::Button());

this->button2 = (gcnew System::Windows::Forms::Button());

this->textBox6 = (gcnew System::Windows::Forms::TextBox());

this->button3 = (gcnew System::Windows::Forms::Button());

this->button4 = (gcnew System::Windows::Forms::Button());

this->button5 = (gcnew System::Windows::Forms::Button());

this->button6 = (gcnew System::Windows::Forms::Button());

this->button7 = (gcnew System::Windows::Forms::Button());

this->button8 = (gcnew System::Windows::Forms::Button());

this->button9 = (gcnew System::Windows::Forms::Button());

this->button10 = (gcnew System::Windows::Forms::Button());

this->button11 = (gcnew System::Windows::Forms::Button());

this->button12 = (gcnew System::Windows::Forms::Button());

this->button13 = (gcnew System::Windows::Forms::Button());

this->SuspendLayout();

//

// textBox1

//

this->textBox1->Location = System::Drawing::Point(245, 47);

this->textBox1->Multiline = true;

this->textBox1->Name = L"textBox1";

this->textBox1->Size = System::Drawing::Size(113, 48);

this->textBox1->TabIndex = 0;

//

// label1

//

this->label1->AutoSize = true;

this->label1->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif", 16));

this->label1->Location = System::Drawing::Point(12, 47);

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

this->label1->Size = System::Drawing::Size(196, 31);

this->label1->TabIndex = 1;

this->label1->Text = L"Перше число: ";

//

// label2

//

this->label2->AutoSize = true;

this->label2->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif", 16));

this->label2->Location = System::Drawing::Point(364, 59);

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

this->label2->Size = System::Drawing::Size(30, 31);

this->label2->TabIndex = 2;

this->label2->Text = L"+";

//

// textBox2

//

this->textBox2->Location = System::Drawing::Point(400, 47);

this->textBox2->Multiline = true;

this->textBox2->Name = L"textBox2";

this->textBox2->Size = System::Drawing::Size(113, 48);

this->textBox2->TabIndex = 3;

//

// label3

//

this->label3->AutoSize = true;

this->label3->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif", 16));

this->label3->Location = System::Drawing::Point(519, 59);

this->label3->Name = L"label3";

this->label3->Size = System::Drawing::Size(20, 31);

this->label3->TabIndex = 4;

this->label3->Text = L"i";

//

// label4

//

this->label4->AutoSize = true;

this->label4->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif", 16));

this->label4->Location = System::Drawing::Point(519, 123);

this->label4->Name = L"label4";

this->label4->Size = System::Drawing::Size(20, 31);

this->label4->TabIndex = 9;

this->label4->Text = L"i";

//

// textBox3

//

this->textBox3->Location = System::Drawing::Point(400, 111);

this->textBox3->Multiline = true;

this->textBox3->Name = L"textBox3";

this->textBox3->Size = System::Drawing::Size(113, 48);

this->textBox3->TabIndex = 8;

//

// label5

//

this->label5->AutoSize = true;

this->label5->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif", 16));

this->label5->Location = System::Drawing::Point(364, 123);

this->label5->Name = L"label5";

this->label5->Size = System::Drawing::Size(30, 31);

this->label5->TabIndex = 7;

this->label5->Text = L"+";

//

// label6

//

this->label6->AutoSize = true;

this->label6->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif", 16));

this->label6->Location = System::Drawing::Point(12, 111);

this->label6->Name = L"label6";

this->label6->Size = System::Drawing::Size(184, 31);

this->label6->TabIndex = 6;

this->label6->Text = L"Друге число: ";

//

// textBox4

//

this->textBox4->Location = System::Drawing::Point(245, 111);

this->textBox4->Multiline = true;

this->textBox4->Name = L"textBox4";

this->textBox4->Size = System::Drawing::Size(113, 48);

this->textBox4->TabIndex = 5;

//

// textBox5

//

this->textBox5->Location = System::Drawing::Point(18, 203);

this->textBox5->Multiline = true;

this->textBox5->Name = L"textBox5";

this->textBox5->ReadOnly = true;

this->textBox5->Size = System::Drawing::Size(680, 68);

this->textBox5->TabIndex = 10;

//

// button1

//

this->button1->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif", 16));

this->button1->Location = System::Drawing::Point(18, 368);

this->button1->Name = L"button1";

this->button1->Size = System::Drawing::Size(110, 47);

this->button1->TabIndex = 11;

this->button1->Text = L"+";

this->button1->UseVisualStyleBackColor = true;

this->button1->Click += gcnew System::EventHandler(this, &MyForm::button1\_Click);

//

// button2

//

this->button2->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif", 16));

this->button2->Location = System::Drawing::Point(153, 368);

this->button2->Name = L"button2";

this->button2->Size = System::Drawing::Size(110, 47);

this->button2->TabIndex = 12;

this->button2->Text = L"-";

this->button2->UseVisualStyleBackColor = true;

this->button2->Click += gcnew System::EventHandler(this, &MyForm::button2\_Click);

//

// textBox6

//

this->textBox6->Location = System::Drawing::Point(18, 295);

this->textBox6->Multiline = true;

this->textBox6->Name = L"textBox6";

this->textBox6->Size = System::Drawing::Size(113, 48);

this->textBox6->TabIndex = 13;

//

// button3

//

this->button3->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif", 16));

this->button3->Location = System::Drawing::Point(153, 296);

this->button3->Name = L"button3";

this->button3->Size = System::Drawing::Size(241, 47);

this->button3->TabIndex = 14;

this->button3->Text = L"Степінь 1-го числа";

this->button3->UseVisualStyleBackColor = true;

this->button3->Click += gcnew System::EventHandler(this, &MyForm::button3\_Click);

//

// button4

//

this->button4->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif", 16));

this->button4->Location = System::Drawing::Point(284, 368);

this->button4->Name = L"button4";

this->button4->Size = System::Drawing::Size(110, 47);

this->button4->TabIndex = 15;

this->button4->Text = L"\*";

this->button4->UseVisualStyleBackColor = true;

this->button4->Click += gcnew System::EventHandler(this, &MyForm::button4\_Click);

//

// button5

//

this->button5->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif", 16));

this->button5->Location = System::Drawing::Point(418, 368);

this->button5->Name = L"button5";

this->button5->Size = System::Drawing::Size(280, 47);

this->button5->TabIndex = 16;

this->button5->Text = L"Модуль 1-го числа";

this->button5->UseVisualStyleBackColor = true;

this->button5->Click += gcnew System::EventHandler(this, &MyForm::button5\_Click);

//

// button6

//

this->button6->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif", 16));

this->button6->Location = System::Drawing::Point(418, 421);

this->button6->Name = L"button6";

this->button6->Size = System::Drawing::Size(280, 47);

this->button6->TabIndex = 17;

this->button6->Text = L"Модуль 2-го числа";

this->button6->UseVisualStyleBackColor = true;

this->button6->Click += gcnew System::EventHandler(this, &MyForm::button6\_Click);

//

// button7

//

this->button7->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif", 16));

this->button7->Location = System::Drawing::Point(418, 296);

this->button7->Name = L"button7";

this->button7->Size = System::Drawing::Size(280, 47);

this->button7->TabIndex = 18;

this->button7->Text = L"Степінь 2-го числа";

this->button7->UseVisualStyleBackColor = true;

this->button7->Click += gcnew System::EventHandler(this, &MyForm::button7\_Click);

//

// button8

//

this->button8->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif", 16));

this->button8->Location = System::Drawing::Point(18, 421);

this->button8->Name = L"button8";

this->button8->Size = System::Drawing::Size(110, 47);

this->button8->TabIndex = 19;

this->button8->Text = L"1<2";

this->button8->UseVisualStyleBackColor = true;

this->button8->Click += gcnew System::EventHandler(this, &MyForm::button8\_Click);

//

// button9

//

this->button9->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif", 16));

this->button9->Location = System::Drawing::Point(153, 421);

this->button9->Name = L"button9";

this->button9->Size = System::Drawing::Size(110, 47);

this->button9->TabIndex = 20;

this->button9->Text = L"1>2";

this->button9->UseVisualStyleBackColor = true;

this->button9->Click += gcnew System::EventHandler(this, &MyForm::button9\_Click);

//

// button10

//

this->button10->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif", 16));

this->button10->Location = System::Drawing::Point(284, 421);

this->button10->Name = L"button10";

this->button10->Size = System::Drawing::Size(110, 47);

this->button10->TabIndex = 21;

this->button10->Text = L"==";

this->button10->UseVisualStyleBackColor = true;

this->button10->Click += gcnew System::EventHandler(this, &MyForm::button10\_Click);

//

// button11

//

this->button11->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif", 16));

this->button11->Location = System::Drawing::Point(18, 474);

this->button11->Name = L"button11";

this->button11->Size = System::Drawing::Size(680, 47);

this->button11->TabIndex = 22;

this->button11->Text = L"Тригонометрична форма 1-го числа";

this->button11->UseVisualStyleBackColor = true;

this->button11->Click += gcnew System::EventHandler(this, &MyForm::button11\_Click);

//

// button12

//

this->button12->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif", 16));

this->button12->Location = System::Drawing::Point(18, 527);

this->button12->Name = L"button12";

this->button12->Size = System::Drawing::Size(680, 47);

this->button12->TabIndex = 23;

this->button12->Text = L"Тригонометрична форма 2-го числа";

this->button12->UseVisualStyleBackColor = true;

this->button12->Click += gcnew System::EventHandler(this, &MyForm::button12\_Click);

//

// button13

//

this->button13->Font = (gcnew System::Drawing::Font(L"Microsoft Sans Serif", 16));

this->button13->Location = System::Drawing::Point(18, 580);

this->button13->Name = L"button13";

this->button13->Size = System::Drawing::Size(680, 47);

this->button13->TabIndex = 24;

this->button13->Text = L"Кіл-сть об\'єктів";

this->button13->UseVisualStyleBackColor = true;

this->button13->Click += gcnew System::EventHandler(this, &MyForm::button13\_Click);

//

// MyForm

//

this->AutoScaleDimensions = System::Drawing::SizeF(8, 16);

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

this->ClientSize = System::Drawing::Size(716, 648);

this->Controls->Add(this->button13);

this->Controls->Add(this->button12);

this->Controls->Add(this->button11);

this->Controls->Add(this->button10);

this->Controls->Add(this->button9);

this->Controls->Add(this->button8);

this->Controls->Add(this->button7);

this->Controls->Add(this->button6);

this->Controls->Add(this->button5);

this->Controls->Add(this->button4);

this->Controls->Add(this->button3);

this->Controls->Add(this->textBox6);

this->Controls->Add(this->button2);

this->Controls->Add(this->button1);

this->Controls->Add(this->textBox5);

this->Controls->Add(this->label4);

this->Controls->Add(this->textBox3);

this->Controls->Add(this->label5);

this->Controls->Add(this->label6);

this->Controls->Add(this->textBox4);

this->Controls->Add(this->label3);

this->Controls->Add(this->textBox2);

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

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

this->Controls->Add(this->textBox1);

this->Name = L"MyForm";

this->Text = L"ComplexHelper";

this->ResumeLayout(false);

this->PerformLayout();

}

private: System::Void button1\_Click(System::Object^ sender, System::EventArgs^ e)

{

if (textBox1->Text != " " || textBox2->Text != " " || textBox3->Text != " " || textBox4->Text != " ")

{

Complex elem1(Convert::ToDouble(textBox1->Text), Convert::ToDouble(textBox2->Text));

Complex elem2(Convert::ToDouble(textBox4->Text), Convert::ToDouble(textBox3->Text));

Complex sum;

sum = elem1 + elem2;

textBox5->Text = textBox5->Text << sum;

}

}

private: System::Void button2\_Click(System::Object^ sender, System::EventArgs^ e)

{

if (textBox1->Text != " " || textBox2->Text != " " || textBox3->Text != " " || textBox4->Text != " ")

{

Complex elem1(Convert::ToDouble(textBox1->Text), Convert::ToDouble(textBox2->Text));

Complex elem2(Convert::ToDouble(textBox4->Text), Convert::ToDouble(textBox3->Text));

Complex substraction;

substraction = elem1 - elem2;

textBox5->Text = textBox5->Text << substraction;

}

}

private: System::Void button4\_Click(System::Object^ sender, System::EventArgs^ e)

{

if (textBox1->Text != " " || textBox2->Text != " " || textBox3->Text != " " || textBox4->Text != " ")

{

Complex elem1(Convert::ToDouble(textBox1->Text), Convert::ToDouble(textBox2->Text));

Complex elem2(Convert::ToDouble(textBox4->Text), Convert::ToDouble(textBox3->Text));

Complex dobutok;

dobutok = elem1 \* elem2;

textBox5->Text = textBox5->Text << dobutok;

}

}

private: System::Void button8\_Click(System::Object^ sender, System::EventArgs^ e)

{

if (textBox1->Text != " " || textBox2->Text != " " || textBox3->Text != " " || textBox4->Text != " ")

{

Complex elem1(Convert::ToDouble(textBox1->Text), Convert::ToDouble(textBox2->Text));

Complex elem2(Convert::ToDouble(textBox4->Text), Convert::ToDouble(textBox3->Text));

if (elem1 < elem2)

{

textBox5->Text = "True";

}

else

{

textBox5->Text = "False";

}

}

}

private: System::Void button9\_Click(System::Object^ sender, System::EventArgs^ e)

{

if (textBox1->Text != " " || textBox2->Text != " " || textBox3->Text != " " || textBox4->Text != " ")

{

Complex elem1(Convert::ToDouble(textBox1->Text), Convert::ToDouble(textBox2->Text));

Complex elem2(Convert::ToDouble(textBox4->Text), Convert::ToDouble(textBox3->Text));

if (elem1 > elem2)

{

textBox5->Text = "True";

}

else

{

textBox5->Text = "False";

}

}

}

private: System::Void button10\_Click(System::Object^ sender, System::EventArgs^ e)

{

if (textBox1->Text != " " || textBox2->Text != " " || textBox3->Text != " " || textBox4->Text != " ")

{

Complex elem1(Convert::ToDouble(textBox1->Text), Convert::ToDouble(textBox2->Text));

Complex elem2(Convert::ToDouble(textBox4->Text), Convert::ToDouble(textBox3->Text));

if (elem1 == elem2)

{

textBox5->Text = "True";

}

else

{

textBox5->Text = "False";

}

}

}

private: System::Void button3\_Click(System::Object^ sender, System::EventArgs^ e)

{

if (textBox1->Text != " " || textBox2->Text != " " || textBox6->Text != " ")

{

Complex elem(Convert::ToDouble(textBox1->Text), Convert::ToDouble(textBox2->Text), Convert::ToInt16(textBox6->Text));

textBox5->Text = Convert::ToString(elem.PowerR(elem.real, elem.imag, elem.power)) + "+i" + Convert::ToString(elem.PowerI(elem.real, elem.imag, elem.imag));

}

else

textBox5->Text = "Error";

}

private: System::Void button7\_Click(System::Object^ sender, System::EventArgs^ e)

{

if (textBox3->Text != " " || textBox4->Text != " " || textBox6->Text != " ")

{

Complex elem(Convert::ToDouble(textBox4->Text), Convert::ToDouble(textBox3->Text), Convert::ToInt16(textBox6->Text));

textBox5->Text = Convert::ToString(elem.PowerR(elem.real, elem.imag, elem.power)) + "+i" + Convert::ToString(elem.PowerI(elem.real, elem.imag, elem.imag));

}

else

textBox5->Text = "Error";

}

private: System::Void button5\_Click(System::Object^ sender, System::EventArgs^ e)

{

if (textBox1->Text != " " || textBox2->Text != " ")

{

Complex c1(Convert::ToDouble(textBox1->Text), Convert::ToDouble(textBox2->Text));

double rez;

rez = c1.Module(c1.real, c1.imag);

textBox5->Text = Convert::ToString(rez);

}

else

textBox5->Text = "Error";

}

private: System::Void button6\_Click(System::Object^ sender, System::EventArgs^ e)

{

if (textBox3->Text != " " || textBox4->Text != " ")

{

Complex c1(Convert::ToDouble(textBox4->Text), Convert::ToDouble(textBox3->Text));

double rez;

rez = c1.Module(c1.real, c1.imag);

textBox5->Text = Convert::ToString(rez);

}

else

textBox5->Text = "Error";

}

private: System::Void button11\_Click(System::Object^ sender, System::EventArgs^ e)

{

if (textBox1->Text != " " || textBox2->Text != " ")

{

Complex elem1(Convert::ToDouble(textBox1->Text), Convert::ToDouble(textBox2->Text));

double r = elem1.Module(elem1.real, elem1.imag);

double cosinus = elem1.TrigonometricFormCosinus(elem1.real, elem1.imag);

double sinus = elem1.TrigonometricFormSinus(elem1.real, elem1.imag);

textBox5->Text = Convert::ToString(r) + "(" + Convert::ToString(cosinus) + "+i" + Convert::ToString(sinus); \

}

else

textBox5->Text = "Error";

}

private: System::Void button12\_Click(System::Object^ sender, System::EventArgs^ e)

{

if (textBox3->Text != " " || textBox4->Text != " ")

{

Complex elem1(Convert::ToDouble(textBox4->Text), Convert::ToDouble(textBox3->Text));

double r = elem1.Module(elem1.real, elem1.imag);

double cosinus = elem1.TrigonometricFormCosinus(elem1.real, elem1.imag);

double sinus = elem1.TrigonometricFormSinus(elem1.real, elem1.imag);

textBox5->Text = Convert::ToString(r) + "(" + Convert::ToString(cosinus) + "+i" + Convert::ToString(sinus);

}

else

textBox5->Text = "Error";

}

private: System::Void button13\_Click(System::Object^ sender, System::EventArgs^ e)

{

textBox5->Text = Convert::ToString(Complex::Get\_HowManyObjCreated());

}

};

}

**Complex.h**

#pragma once

#include <iostream>

class Complex

{

static int HowManyObjects;

public:

double real, imag;

int power;

Complex();

Complex(double realN, double imagN);

Complex(double realN);

Complex(double realN, double imagN, int pow);

double Module(double real, double imag);

double TrigonometricFormCosinus(double real, double imag);

double TrigonometricFormSinus(double real, double imag);

Complex operator+(Complex elem1)

{

return Complex(real + elem1.real, imag + elem1.imag);

}

Complex operator-(Complex elem1)

{

return Complex(real - elem1.real, imag - elem1.imag);

}

Complex operator\*(Complex elem1)

{

Complex rez;

rez.real = (real \* elem1.real) - (imag \* elem1.imag);

rez.imag = (real \* elem1.imag) + (imag \* elem1.real);

return rez;

}

Complex operator^(Complex elem) //перевантаження оператора модуля

{

Complex rez = sqrt(pow(elem.real, 2) + pow(elem.imag, 2));

return rez;

}

friend System::String^ operator<<(System::String^ out, Complex point)

{

out = point.real + "+i" + point.imag;

return out;

}

friend std::istream& operator>>(std::istream& in, Complex point)

{

in >> point.real;

in >> point.imag;

return in;

}

friend bool operator<(Complex elem1, Complex elem2)

{

if ((sqrt(pow(elem1.real, 2) + pow(elem1.imag, 2))) < (sqrt(pow(elem2.real, 2) + pow(elem2.imag, 2))))

{

return true;

}

else

{

return false;

}

}

friend bool operator>(Complex elem1, Complex elem2)

{

if ((sqrt(pow(elem1.real, 2) + pow(elem1.imag, 2))) > (sqrt(pow(elem2.real, 2) + pow(elem2.imag, 2))))

{

return true;

}

else

{

return false;

}

}

friend bool operator==(Complex elem1, Complex elem2)

{

if ((sqrt(pow(elem1.real, 2) + pow(elem1.imag, 2))) == (sqrt(pow(elem2.real, 2) + pow(elem2.imag, 2))))

{

return true;

}

else

{

return false;

}

}

double PowerR(double real, double imag, int power);

double PowerI(double real, double imag, int power);

static void Set\_HowManyObjCreated(int num);

static int Get\_HowManyObjCreated();

};

**Complex.cpp**

#include "Complex.h"

int Complex::HowManyObjects = 0;

Complex::Complex()

{

HowManyObjects++;

}

Complex::Complex(double realN, double imagN)

{

HowManyObjects++;

real = realN;

imag = imagN;

}

Complex::Complex(double realN)

{

HowManyObjects++;

real = realN;

}

Complex::Complex(double realN, double imagN, int pow)

{

HowManyObjects++;

real = realN;

imag = imagN;

power = pow;

}

double Complex::Module(double real, double imag)

{

return sqrt(pow(real, 2) + pow(imag, 2));

}

double Complex::TrigonometricFormSinus(double real, double imag)

{

double sinus;

sinus = imag / (Module(real, imag));

return sinus;

}

double Complex::TrigonometricFormCosinus(double real, double imag)

{

double cosinus;

cosinus = real / (Module(real, imag));

return cosinus;

}

double Complex::PowerR(double real, double imag, int power)

{

double result = 1;

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

{

result \*= ((real \* real) - (imag \* imag));

}

return result;

}

double Complex::PowerI(double real, double imag, int power)

{

double result = 1;

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

{

result \*= ((real \* imag) + (imag \* real));

}

return result;

}

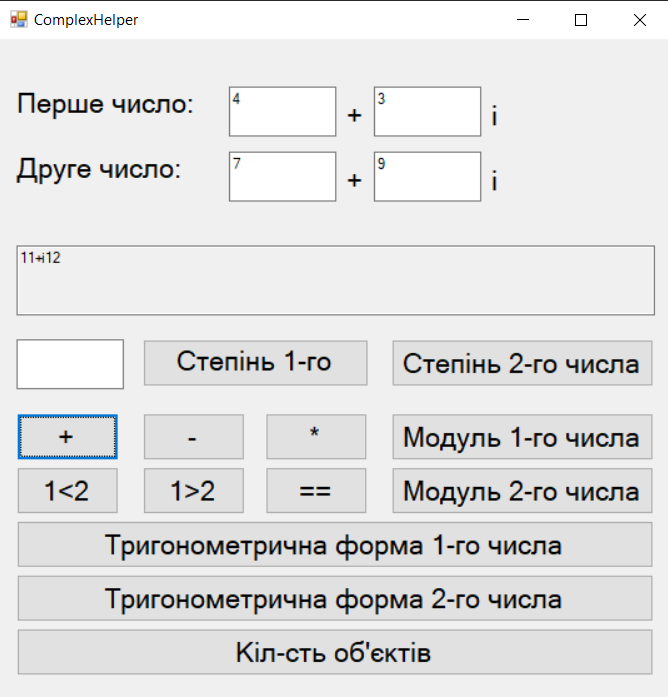
int Complex::Get\_HowManyObjCreated()

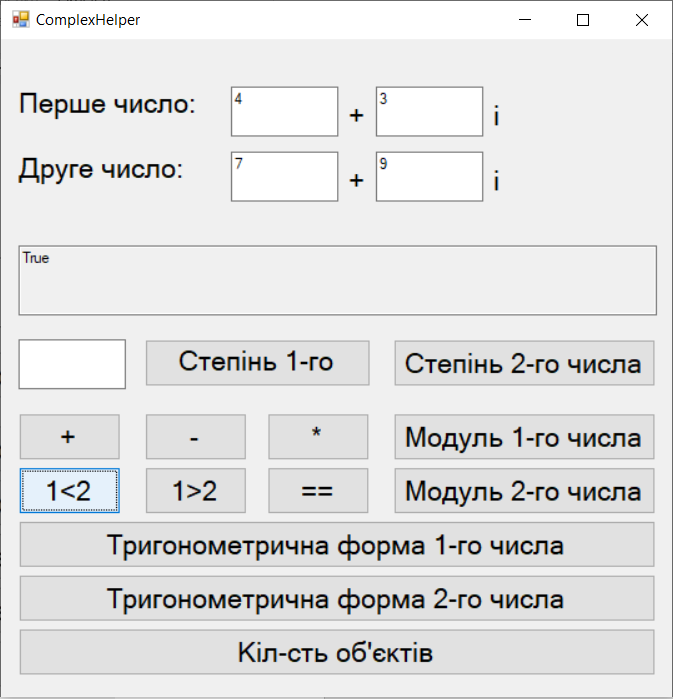
{

return HowManyObjects;

}

**Результат виконання:**

****

****

**Висновок:**

У цій лабораторній, я навчився створювати дружні функції та

використовувати їх, перевантажувати та перевизначати оператори,

використовуючи їх при обчисленнях та виводу. Ознайомився із статичними

полями та методами.