Міністерство освіти і науки України   
Національний університет «Львівська політехніка»

Інститут прикладної математики та фундаментальних наук (ІМФН)   
Кафедра прикладної математики

**ЗВІТ**

Про виконання лабораторної роботи № 1

«Модульне тестування ПЗ»

Виконала:

Студентка групи ПМ-32

Лопата З.О.

Прийняв:

Любінський Б.Б.

Львів-2024

**Мета:** Закріпити на практиці написання модульних тестів на мові C#.

**Постановка задачі:**

Задано специфікацію(файл SpecCalc.PDF), на основі якої побудовано

додаток "Калькулятор"(файл Calculator.zip), в якому є дві бібліотеки

AnalaizerClassLibrary та CalcClassBr. Необхідно написати набір модульних

тестів з максимальним покриттям коду для заданих методів згідно варіанту.

Набір даних на якому будуть працювати модульні тести містяться в базі

даних. Кількість даних та структуру бази даних необхідно спроектувати під

варіант завдання.

**Варіант 15:** бібліотека CalcClassBr метод Mod.

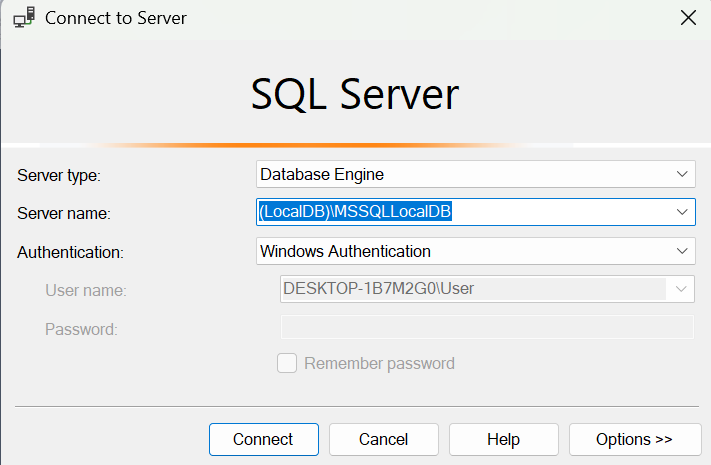
**Хід виконання**

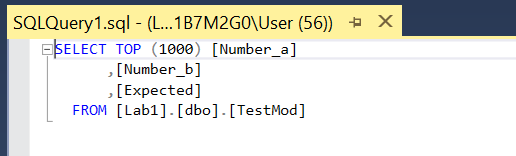
Для виконання роботи створила тест у Visual Studio.

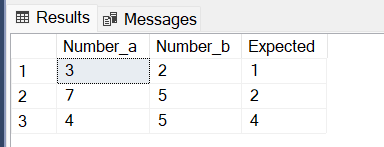
В Solution Explorer додала reference для бібліотеки CalcClassBr.

Далі в SQL Server Management Studio під’єдналася до sql-сервера який по замовчуванню стоїть у Visual Studio, (LocalDB)\MSSQLLocalDB:

На цьому сервері створила власну базу даних та в цій базі створила дві таблиці з наборами даних для тестування методу Mod:







Щоб отримати доступ до бази даних, додаємо в код шлях до бази даних:

[DataSource("System.Data.SqlClient", @"Data Source=(LocalDB)\MSSQLLocalDB;Initial Catalog=Lab1;Integrated Security=True", "TestMod",

DataAccessMethod.Sequential)]

Метод Mod перевіряє ділення по модулю.

**Код**:

using ErrorLibrary;

using Microsoft.VisualStudio.TestTools.UnitTesting;

using System;

using System.Data;

namespace Mod\_Test

{

[TestClass]

public class UnitTest1

{

public TestContext TestContext { get; set; }

//Функція має викликати виняток, коли значення a є більше за межі типу int,

//а значення результату операції може виходити за ці межі

[TestMethod]

[ExpectedException(typeof(ArgumentOutOfRangeException))]

public void TestMod()

{

//arrange

long a = long.MaxValue; long b = 2;

CalcClassBr.CalcClass.Mod(a, b);

}

[DataSource("System.Data.SqlClient", @"Data Source=(LocalDB)\MSSQLLocalDB;Initial Catalog=Lab1;Integrated Security=True", "TestMod",

DataAccessMethod.Sequential)]

[TestMethod]

public void TestMod1()

{

//Arrange

int a = (int)TestContext.DataRow["Number\_a"];

int b = (int)TestContext.DataRow["Number\_b"];

//Expected

int expected = (int)TestContext.DataRow["Expected"];

//Actual

int actual = CalcClassBr.CalcClass.Mod(a, b);

Assert.AreEqual(expected, actual);

}

//ділення на нуль

[TestMethod]

[ExpectedException(typeof(DivideByZeroException))]

public void TestMod2()

{

// arrange

long a = 3; long b = 0;

// act

CalcClassBr.CalcClass.Mod(a, b);

}

//Перевірка повідомлення про помилку при ділені на нуль

[TestMethod]

public void TestMod\_DivideByZeroErrorMessage()

{

// arrange

long a = 3; long b = 0;

try

{

// act

CalcClassBr.CalcClass.Mod(a, b);

}

catch (DivideByZeroException x)

{

// assert

Assert.AreEqual(ErrorsExpression.ERROR\_09, x.Message);

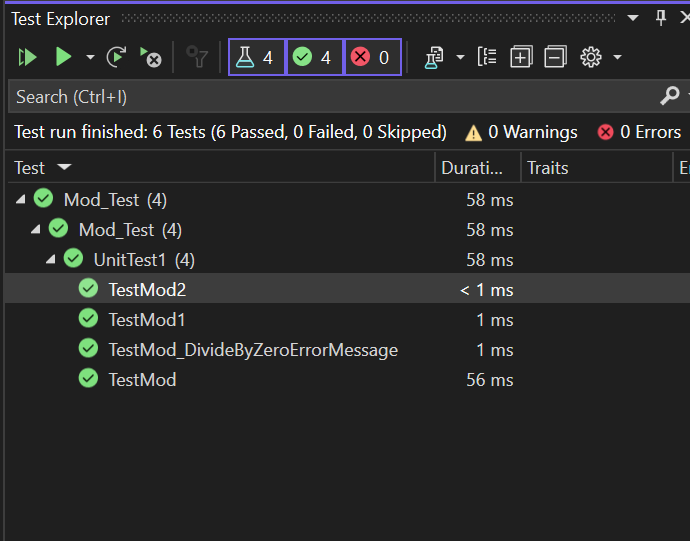
}

}

}

}

**Результати виконання:**



**GitHub:** <https://github.com/Zlata-Lopata/Test.git>

**Висновок:** Під час виконання лабораторної роботи написала модульний тест для перевірки методу Mod бібліотеки CalcClassBr . В результаті написала 4 модульні тести, один з яких бере значення для перевірки з бази даних, яка під’єднана до тесту, та успішно перевіряє ці методи. В інших тестах перевіряється ділення з найбільшим значенням int, ділення на нуль та перевірка виводу повідомлення.