Міністерство освіти і науки України

Національний університет «Львівська політехніка»

Інститут прикладної математики та фундаментальних наук

Кафедра прикладної математики

**Звіт**

про виконання лабораторної роботи №1

з курсу «Автоматизовані системи тестування програмного продукту»

**Виконав:**

студент групи ПМ-32

Янів П.В.

**Перевірив:**

Любінський Б. Б.

Львів-2024

**Варіант: 24**

**Завдання:** 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.

24.бібліотека AnalaizerClassLibrary методи IsDelimeter IsOperator

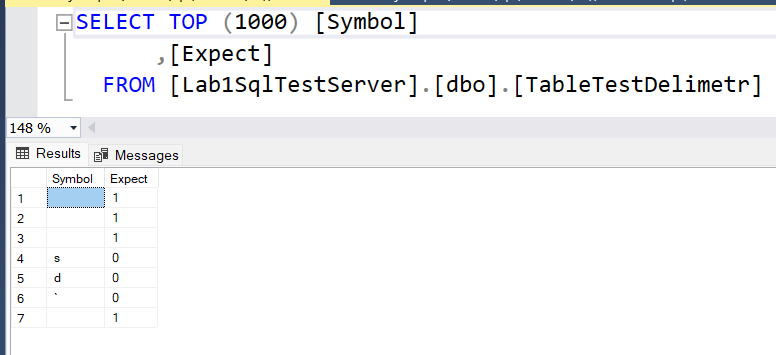
**Хід роботи:**

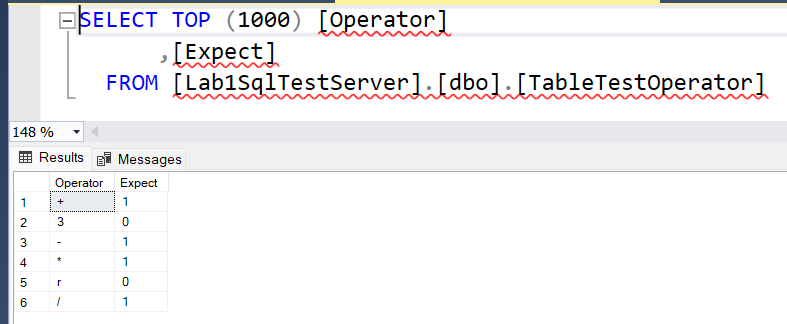
Для виконання роботи створив тест у Visual Studio.

В Solution Explorer додав reference для бібліотеки AnalaizerClassLibrary

Далі в Sql server management studio під’єднався до sql-сервера який по-замовчуванню стоїть у Visual Studio, (LocalDB)\MSSQLLoacalDB:

На цьому сервері створив власну базу даних та в цій базі створив дві таблиці з наборами даних для тестування методів(TableTestDelimetr, TableTestOperator):



****

Тепер щоб отримати доступ до бази даних з тесту я додав у код шлях до створеної бази та назву таблиці.

Метод IsOperator перевіряє тип даних string і коми символ є оператором то він повертає значення true, в протилежному випадку false. Щоб перевірити метод я передаю у цей метод оператор і очікую що він поверне значення true і навпаки.

Метод IsDelimetr перевіряє тип даних char і коми символ є роздилителем то він повертає значення true, в протилежному випадку false. Щоб перевірити метод я передаю у цей метод пробіл і очікую що він поверне значення true і навпаки.

**Код:**

using System;

using AnalaizerClassLibrary;

using Microsoft.VisualStudio.TestTools.UnitTesting;

using System.Data;

using System.Reflection;

using System.Diagnostics.Eventing.Reader;

namespace UnitTest1

{

[TestClass]

public class UnitTest1

{

public Type ClonClass = typeof(AnalaizerClass);

public TestContext TestContext { get; set; }

[DataSource("System.Data.SqlClient", @"Data Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=C:\Users\Admin\Lab1SqlTestServer.mdf;Integrated Security=True;Connect Timeout=30", "TableTestDelimetr", DataAccessMethod.Sequential)]

[TestMethod]

public void Test\_IsDelimetr\_DB()

{

//Arrange

char c = (char)Convert.ToChar(TestContext.DataRow["Symbol"]);

//Expected

bool expected = (bool)TestContext.DataRow["Expect"];

//Actual

var PrMet = Activator.CreateInstance(typeof(AnalaizerClass));

MethodInfo method = ClonClass.GetMethod("IsDelimetr",

BindingFlags.NonPublic | BindingFlags.Instance);

if (PrMet == null)

{

object actual = new object[] { Convert.ToBoolean(method.Invoke(PrMet, new object[] { c })) };

Assert.AreEqual(expected, actual);

}

else

{

}

}

[DataSource("System.Data.SqlClient", @"Data Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=C:\Users\Admin\Lab1SqlTestServer.mdf;Integrated Security=True;Connect Timeout=30", "TableTestOperator", DataAccessMethod.Sequential)]

[TestMethod]

public void Test\_IsOperator\_DB()

{

//Arrange

string d = (string)TestContext.DataRow["Operator"];

//Expected

bool expected = (bool)TestContext.DataRow["Expect"];

//Actual

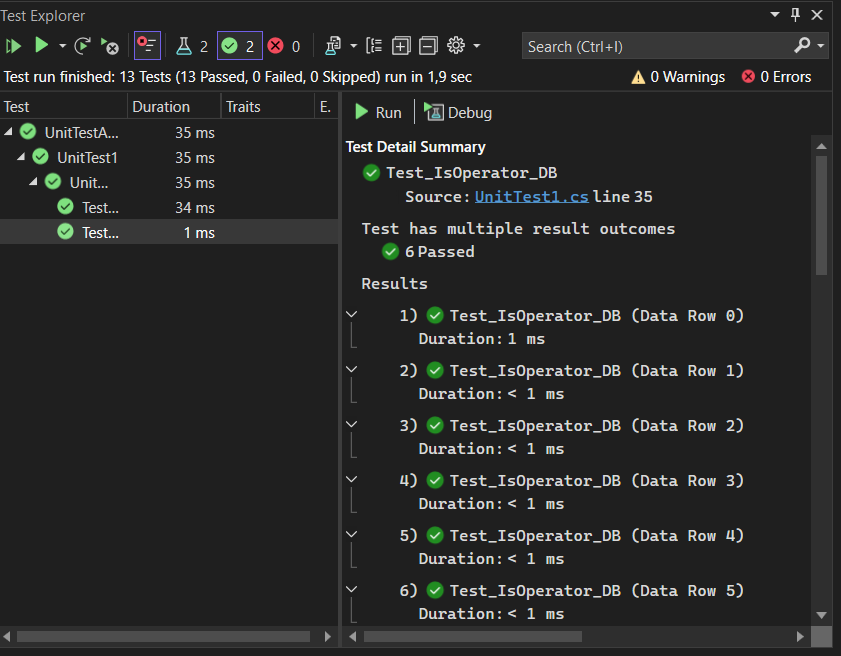
bool actual = AnalaizerClass.IsOperator(d);

Assert.AreEqual(expected, actual);

}

}

}**Результат виконання:**

****

**GitHub репозиторій:** **https://github.com/PetroYaniv/TestSystem.git**

**Висновок:** Для виконання лабораторної роботи написав модульний тест для перевірки двох методів бібліотеки. В результаті написав модульний тест який бере значення для перевірки з бази даних яка під’єднана до тесту та успішно перевіряє ці методи.