

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.

## Варіант: 24

**Завдання:** Задано специфікацію(файл SpecCalc.PDF), на основі якої побудовано додаток "Калькулятор"(файл Calculator.zip), в якому є дві бібліотеки AnalizerClassLibrary 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 AnalizerClassLibrary методи IsDelimiter IsOperator

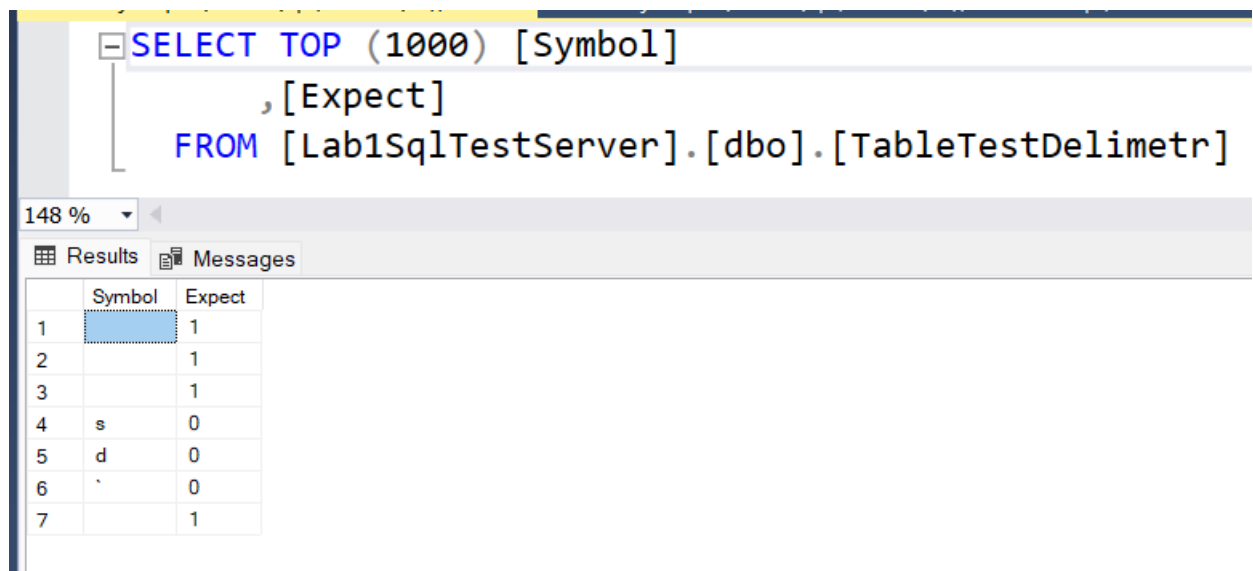
### Хід роботи:

Для виконання роботи створив тест у Visual Studio.

В Solution Explorer додав reference для бібліотеки AnalizerClassLibrary

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

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

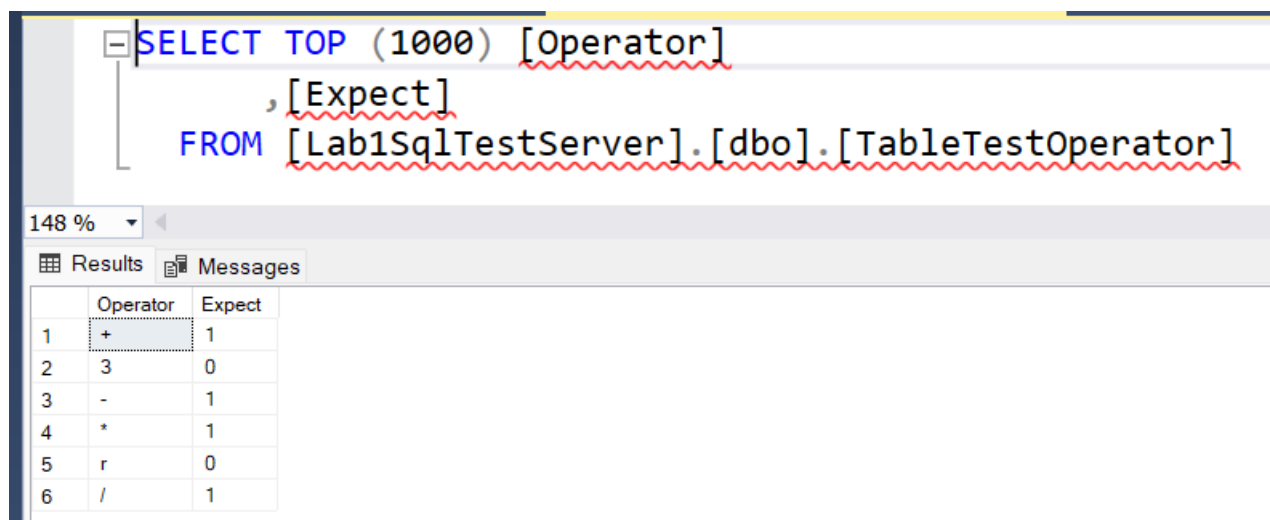


The screenshot shows a SQL query window with the following text:

```
SELECT TOP (1000) [Symbol]
, [Expect]
FROM [Lab1SqlTestServer].[dbo].[TableTestDelimetr]
```

Below the query window, the 'Results' tab is active, displaying a table with two columns: 'Symbol' and 'Expect'. The table contains 7 rows of data.

	Symbol	Expect
1		1
2		1
3		1
4	s	0
5	d	0
6	.	0
7		1



The screenshot shows a SQL query window with the following text:

```
SELECT TOP (1000) [Operator]
, [Expect]
FROM [Lab1SqlTestServer].[dbo].[TableTestOperator]
```

Below the query window, the 'Results' tab is active, displaying a table with two columns: 'Operator' and 'Expect'. The table contains 6 rows of data.

	Operator	Expect
1	+	1
2	3	0
3	-	1
4	*	1
5	r	0
6	/	1

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 роздільником 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;In
tegrated 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;In
tegrated 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 = AnalizerClass.IsOperator(d);
    Assert.AreEqual(expected, actual);
}
```

```
}
```

**Результат виконання:**

The screenshot shows the Test Explorer window with a toolbar at the top. The status bar indicates 'Test run finished: 13 Tests (13 Passed, 0 Failed, 0 Skipped) run in 1,9 sec' with 0 warnings and 0 errors. The test list on the left shows a hierarchy of tests, with 'Test\_IsOperator\_DB' selected. The right pane displays the 'Test Detail Summary' for this test, showing it passed 6 times. Below the summary, the 'Results' section lists six individual test outcomes, each with a green checkmark and a duration of 1 ms or less.

Test	Duration	Traits	E.
UnitTestA...	35 ms		
UnitTest1	35 ms		
Unit...	35 ms		
Test...	34 ms		
Test...	1 ms		

**Test Detail Summary**

- Test\_IsOperator\_DB
  - Source: [UnitTest1.cs](#) line 35

Test has multiple result outcomes

- 6 Passed

**Results**

- Test\_IsOperator\_DB (Data Row 0)
  - Duration: 1 ms
- Test\_IsOperator\_DB (Data Row 1)
  - Duration: < 1 ms
- Test\_IsOperator\_DB (Data Row 2)
  - Duration: < 1 ms
- Test\_IsOperator\_DB (Data Row 3)
  - Duration: < 1 ms
- Test\_IsOperator\_DB (Data Row 4)
  - Duration: < 1 ms
- Test\_IsOperator\_DB (Data Row 5)
  - Duration: < 1 ms

**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.