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-33   
Музика Денис

**Перевірив:**

кандидат наук   
 Пабирівський Віктор Володимирович

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(дата) (підпис викладача)

**Лабораторна робота №1**

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

**Необхідні CASE інструменти:** MS Visual Studio, MS SQL SERVER.

**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.

9. бібліотека AnalaizerClassLibrary метод RunEstimate

**Теоретичні відомості**

Модульне тестування є однією з ключових практик в розробці програмного забезпечення (ПЗ), спрямованою на підвищення якості коду і забезпечення його надійності. Основна ідея модульного тестування полягає в тому, щоб перевіряти окремі модулі або функції програми на коректність їх роботи. Важливо тестувати не окрему програму в цілому, а її окремі частини (модулі) незалежно одне від одного.

Основні принципи модульного тестування:

* Незалежність: Кожен модульний тест повинен бути незалежним від інших тестів. Результати одного тесту не повинні впливати на інші.
* Автоматизація: Тести повинні бути автоматизованими, щоб їх можна було легко і швидко виконувати.
* Ізоляція: Тести мають тестувати конкретний фрагмент коду (модуль) і не повинні залежати від інших модулів або зовнішніх факторів.
* Спрощення: Тести повинні бути простими та зрозумілими, щоб їх легко підтримувати.

Зворотній польський запис (ЗПЗ) - це математичний інфіксний спосіб запису арифметичних виразів, де операція розміщується між операндами (наприклад, 2 + 3). У ЗПЗ операція розміщується після операндів (наприклад, 2 3 +). Цей спосіб запису дозволяє уникнути використання дужок і пріоритетів операцій, що робить обчислення більш однозначним і ефективним для обчислювальних машин.

[TestMethod]

public void RunEstimate\_SimpleAddition\_ReturnsCorrectResult()

{

AnalaizerClassLibrary ent = new AnalaizerClassLibrary();

// Arrange

string expression = "2 3 +";

// Act

string result = ent.RunEstimate(expression);

// Assert

Assert.AreEqual("5", result);

}

[TestMethod]

public void RunEstimate\_DivideByZero\_ReturnsError()

{

AnalaizerClassLibrary ent = new AnalaizerClassLibrary();

// Arrange

string expression = "5 0 /";

// Act

string result = ent.RunEstimate(expression);

// Assert

Assert.IsTrue(result.StartsWith("&"));

}

[TestMethod]

public void RunEstimate\_InvalidExpression\_ReturnsError()

{

AnalaizerClassLibrary ent = new AnalaizerClassLibrary();

// Arrange

string expression = "2 + 3"; // Missing operands

// Act

string result = ent.RunEstimate(expression);

// Assert

Assert.IsTrue(result.StartsWith("&"));

}

[TestMethod]

public void RunEstimate\_ComplexExpression\_ReturnsCorrectResult()

{

AnalaizerClassLibrary ent = new AnalaizerClassLibrary();

// Arrange

string expression = "5 2 \* 8 + 4 /";

// Act

string result = ent.RunEstimate(expression);

// Assert

Assert.AreEqual("10", result);

}

Модульне тестування є важливою частиною розробки програмного забезпечення, оскільки воно допомагає виявляти та виправляти помилки на ранніх етапах розробки, зменшує ризик некоректної роботи програми після внесення змін, і підвищує загальну надійність ПЗ. Правильне впровадження модульного тестування сприяє створенню більш стабільних і підтримуваних програм.

Таким чином, модульне тестування на мові C# інтегрується в розробку програм для забезпечення їх надійності та якості.