

## Лабораторна робота №17

**Тема: Модульні тести.**

**Мета: Навчитися використовувати модульні тести.**

### Індивідуальне завдання

1. Для однієї з попередніх робіт, додати модульний тест.

### Хід роботи

1. Написання коду для модульного тесту.

```
#pragma once

//сумирование
int sum(int a, int b);

//деление
int division(int a, int b);

//отнимание
int minus(int a, int b);

//умножение
int multiplication(int a, int b);
```

Рисунок 1.1 - код у файлі (calc.h)

```
int sum(int a, int b)
{
    return a + b;
}
int division(int a, int b)
{
    return a / b;
}
int minus(int a, int b)
{
    return a - b;
}
int multiplication(int a, int b)
{
    return a * b;
}
```

Рисунок 1.2 - код у файлі (calc.cpp)

```
#include "pch.h"
#include "CppUnitTest.h"
#include "../Calcuator/calc.h"
#include "../Calcuator/calc.cpp"

using namespace Microsoft::VisualStudio::CppUnitTestFramework;

namespace calculatortests
{
    TEST_CLASS(calculatortests)
    {
    public:
        ✓ TEST_METHOD(Operation1)
        {
            int a = 15, b = 2;
            int answer = 17;
            Assert::AreEqual(sum(a,b ), answer);
        }
        ✓ TEST_METHOD(Operation2)
        {
            int a = 15, b = 2;
            int answer = 7.5;
            Assert::AreEqual(division(a, b), answer);
        }
        ✓ TEST_METHOD(Operation3)
        {
            int a = 15, b = 2;
            int answer = 13;
            Assert::AreEqual(minus(a, b), answer);
        }
        ✓ TEST_METHOD(Operation4)
        {
            int a = 15, b = 2;
            int answer = 30;
            Assert::AreEqual(multiplication(a, b), answer);
        }
    };
}
```

Рисунок 1.2 - код у файлі (calculator.tests.cpp)

## 2. Перевірка модульних тестів.

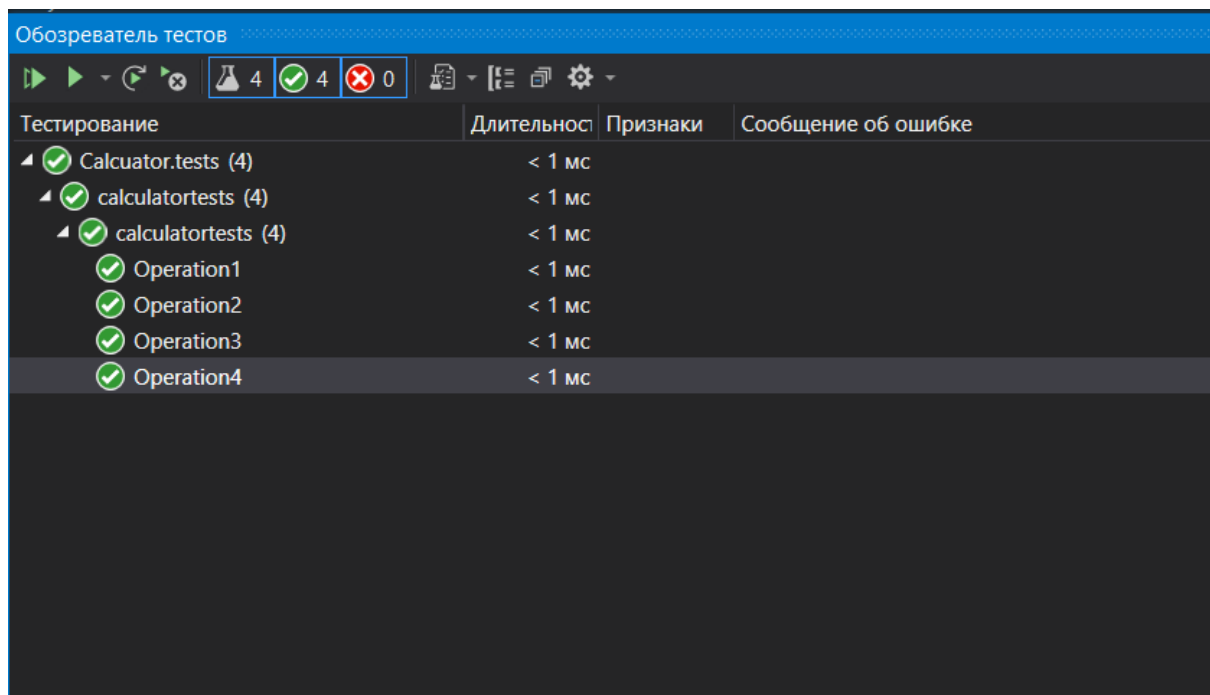


Рисунок 2 - результат перевірки тестів.

**Висновок:** Навчився використовувати модульні тести у власних програмах.