

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ імені Тараса Шевченка  
ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ  
Кафедра програмних систем і технологій  
Дисципліна  
«Якість програмного забезпечення та тестування»  
Лабораторна робота №5  
на тему  
«Створення тестового плану»

Виконав:	Мішак Максим		Перевірила:	Карнаух Тетяна Олександрівна
Група	ІПЗ-33		Дата перевірки	
Форма навчання	денна		Оцінка	
Спеціальність	121			
2023				

### **Завдання :**

У відповідності до отриманого завдання, скласти програму мовою C#. Номер завдання має відповідати номеру в списку академічної групи. На основі готового програмного коду побудувати пласку модель керуючого графу програми. В програмному коді необхідно відмітити кожний вузол – кожну активну дію.

На основі розробленої програми виділити необхідну кількість шляхів таким чином, щоб загальна ступінь протестованості додатку склала 100%. Для кожного шляху сформувати тестовий набір (перелік вхідних параметрів), при якому виконання програми відбудеться саме цим шляхом.

Кожен тест необхідно супроводити відповідним скрін-шотом.

### **Хід роботи :**

Згідно методичним вказанням формула для мого варіанта виглядає так :

$$7. \quad c = \frac{(2a - b) \cdot (2b + a)}{a - 2b}$$

Для обрахунку було написано програму на C# .

### **Код застосунку :**

```
using System;

namespace sqt5
{
    public class Calculator
    {
        public Calculator()
        {
```

```

    }

    public void Validation()
    {
        Console.Write("a = ");

        double a = Convert.ToDouble(Console.ReadLine());

        Console.Write("b = ");

        double b = Convert.ToDouble(Console.ReadLine());

        if ((a - (2 * b)) != 0)
        {
            double[] values = new double[2];

            values[0] = a;

            values[1] = b;

            Calculate(values);
        }

        else Console.WriteLine("this values a wrong");
    }

    protected double Calculate(double[] values)
    {
        double Result = (((2 * values[0]) - values[1]) * ((2 * values[1]) +
values[0])) / (values[0] - (2 * values[1]));

        DisplayDaya(Result);

        return Result;
    }

```

```

    }

    public void DisplayDaya(double Result)
    {
        Console.WriteLine(Result);
    }
}

using System;

namespace sqt5
{
    class Program
    {
        static void Main(string[] args)
        {
            Calculator calculator = new Calculator();
            calculator.Validation();

        }
    }
}

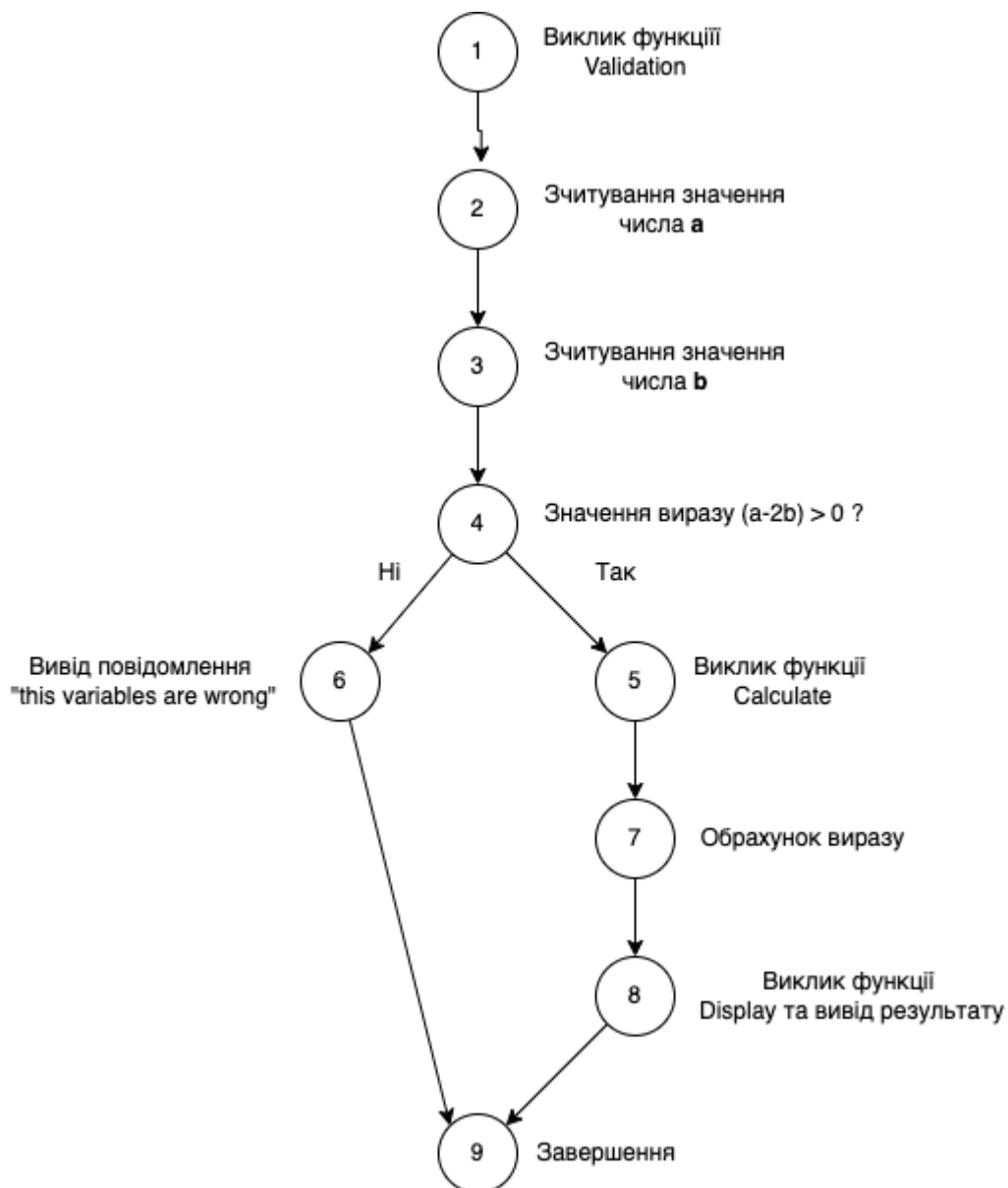
```

Для слідування базового правила математики - ділення на 0 заборонено ,  
було додано перевірку на 0 значення знаменника . Перевірку реалізовано за

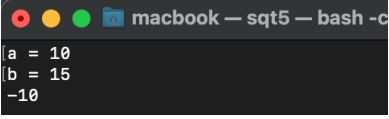
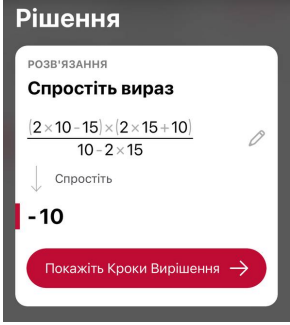
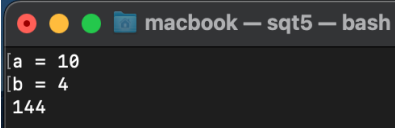
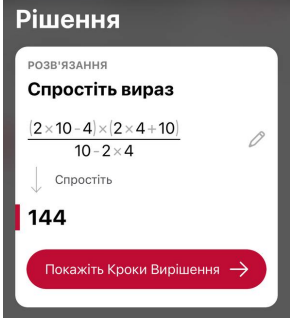
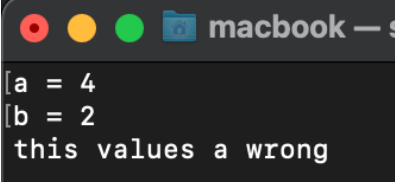
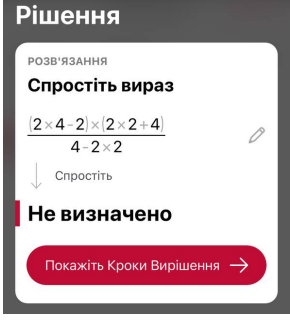
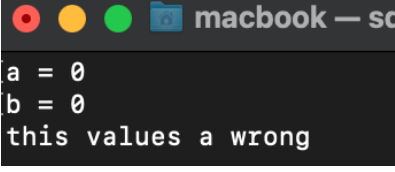
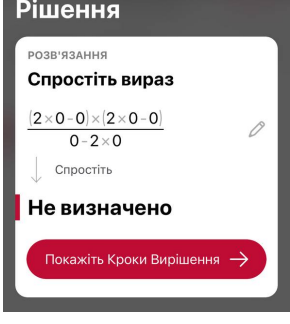
допомогою логічного оператора  $\neq$  (не дорівнює) в наступному фрагменті коду :

```
if ((a - (2 * b)) != 0)
```

На основі програмного коду було побудовано пласку модель керуючого графу програми :



Розроблену програму було протестовано в різних критичних випадках , результати тестування наведені нижче :

Номер тесту	Набір даних	Шлях	Результат	Перевірка
1	a = 10 b = 15	1,2,3,4,5 ,6,7,8,9		
2	a = 10 b = 4	1,2,3,4,5 ,6,7,8,9		
3	a = 4 b = 2	1,2,3,4,6 ,9		
4	a = 0 b = 0	1,2,3,4,6 ,9		

## **Висновок :**

Було написано програму , котра рахує відповідну формулу та протестовано всі критичні моменти . Таким чином маємо додаток , який може фільтрувати значення , рахувати значення та виводити його значення . В додатку збережено правило інкапсуляції та розподілення обов'язків . Для цього було створено окремий клас та методи , які викликались з методу Main . Було протестовано програму на значеннях , які не повинні видавати результат , а саме : 0,0 та 4,2 . Для відбору значень використовується метод Validation . Також для перевірки використовується додаток Photomath , за допомогою якого було протестовано відповідні значення для зазначеної формули .