

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 №7
на тему
«Тестування методом чорного ящика»

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

Завдання :

Відповідно до варіанту провести тестування додатку за методом “чорного ящика”. Результати тестування внести в таблицю

Хід роботи :

Метод тестування "чорний ящик" - це техніка тестування програмного забезпечення, при якій програма розглядається як "чорний ящик", тобто як система, про яку відомо тільки вхідні та вихідні дані, але не відомо, як саме вона їх обробляє. У цьому випадку тестування проводиться шляхом введення різних вхідних даних і аналізу відповідних результатів, щоб перевірити, чи працює програма правильно.

У методі "чорний ящик" не потрібно знати внутрішні деталі роботи програми, що робить його особливо корисним, коли тестування проводиться на програмах зі складними алгоритмами, або на програмах, розроблених сторонніми розробниками, для яких внутрішня структура програми не доступна.

Одним з переваг методу "чорний ящик" є те, що він дозволяє виявити баги, що можуть бути пропущені при інших методах тестування. Наприклад, при тестуванні методом "білий ящик" можуть бути пропущені помилки, пов'язані зі змінними, що не використовуються, але при тестуванні методом "чорний ящик" такі помилки будуть виявлені, оскільки програмі немає доступу до внутрішніх змінних.

Однак, недоліком методу "чорний ящик" є те, що він не дозволяє точно визначити місце виникнення помилки. Тому, якщо програма не працює правильно, може знадобитися використання інших методів тестування для знаходження конкретної проблеми.

Код застосунку :

```
import cmath
def calculate(a, b, c):
    discriminant = b ** 2 - 4 * a * c
    x1 = (-b + cmath.sqrt(discriminant)) / (2 * a)
    x2 = (-b - cmath.sqrt(discriminant)) / (2 * a)
    if discriminant == 0 :
        print(f'x = {x1.real:.2f}')
    else:
        print(f'x1 = {x1.real:.2f}')
        print(f'x2 = {x2.real:.2f}')
    print(f'Discriminant = {discriminant.real:.2f}')
```

```

if __name__ == "__main__":
    a = float(input("Input a value: "))
    b = float(input("Input b value: "))
    c = float(input("Input c value: "))
    calculate(a, b, c)

```

Тестування методом “чорного ящика” :

Номер тесту	Вхідні дані	Очікуваний результат	Мета тесту	Реакція програми
1	a=1, b=-3, c=2	x1=2.0, x2=1.0	Перевірити, чи правильно вирішується квадратне рівняння з додатнім дискримінантом	x1 = 2.00 x2 = 1.00 Discriminant = 1.00
2	a=2, b=-4, c=2	x=-1.0, Discriminant=0.0	Перевірити, чи правильно вирішується квадратне рівняння з нульовим дискримінантом	x = 1.00 Discriminant = 0.00
4	a=1, b=2, c=3	x1=-1+1.41421356j, x2=-1-1.41421356j, Discriminant=-8.0	Перевірити, чи правильно вирішується квадратне рівняння з від'ємним дискримінантом	x1 = -1.00 x2 = -1.00 Discriminant = -8.00
5	a=2, b=4, c=2	x=-1.0, Discriminant=0.0	Перевірити, чи правильно вирішується квадратне рівняння, коли всі коефіцієнти множаться на ту саму ненульову константу	x = -1.00 Discriminant = 0.00

6	a=1, b=1, c=1	$x_1 = -0.5 + 0.8660254j$, $x_2 = -0.5 - 0.8660254j$, Discriminant = -3.0	Перевірити, чи правильно вирішується квадратне рівняння з від'ємним дискримінантом та комплексними коренями	$x_1 = -0.50$ $x_2 = -0.50$ Discriminant = -3.00
---	---------------	---	---	--

Висновок :

На лабораторній роботі було побудовано та протестовано додаток для обчислення квадратного рівняння . Тестування відбувалось методом “чорного ящика” . За результатами тестування можна зробити наступний висновок : на базовій рівні додаток працює коректно . Для обрахування комплексних чисел , числа 0 програма потребує покращення . Алгоритм працює на розділення варіантів вирішення рівняння .