

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»

Кафедра інформаційних систем та мереж

Лабораторна робота №6
з дисципліни
СПЕЦІАЛІЗОВАНІ МОВИ ПРОГРАМУВАННЯ
на тему
Розробка та Unit тестування Python додатку

Виконав:
ст. гр. РІ-21сп
Рак В.П.
Прийняв:
Щербак С.С.

Мета лабораторної роботи: Створення юніт-тестів для додатка-калькулятора на основі класів

Завдання

Завдання 1: Тестування Додавання.

Напишіть юніт-тест, щоб перевірити, що операція додавання в вашому додатку-калькуляторі працює правильно. Надайте тестові випадки як для позитивних, так і для негативних чисел.

Завдання 2: Тестування Віднімання.

Створіть юніт-тести для переконання, що операція віднімання працює правильно. Тестуйте різні сценарії, включаючи випадки з від'ємними результатами.

Завдання 3: Тестування Множення.

Напишіть юніт-тести, щоб перевірити правильність операції множення в вашому калькуляторі. Включіть випадки з нулем, позитивними та від'ємними числами.

Завдання 4: Тестування Ділення.

Розробіть юніт-тести для підтвердження точності операції ділення. Тести повинні охоплювати ситуації, пов'язані з діленням на нуль та різними числовими значеннями.

Завдання 5: Тестування Обробки Помилки.

Створіть юніт-тести, щоб перевірити, як ваш додаток-калькулятор обробляє помилки. Включіть тести для ділення на нуль та інших потенційних сценаріїв помилок. Переконайтеся, що додаток відображає відповідні повідомлення про помилки.

Виконання роботи

Текст програмної реалізації:

""

Модуль тестів для калькулятора, який перевіряє коректність виконання математичних операцій: додавання, віднімання, множення, ділення, а також обробку помилок.

Завдання:

- Тестування основних математичних операцій калькулятора: додавання, віднімання, множення, ділення.
- Перевірка обробки помилок, таких як ділення на нуль і спроба обчислення квадратного кореня з від'ємного числа.

Клас:

- `TestCalculator`: клас тестів для калькулятора, що використовує бібліотеку `unittest` для проведення тестування.

Методи:

- `setUp(self)`: ініціалізує калькулятор перед кожним тестом.
- `test_addition(self)`: перевіряє правильність виконання операції додавання для різних варіантів чисел.
- `test_subtraction(self)`: перевіряє правильність виконання операції віднімання для різних варіантів чисел.
- `test_multiplication(self)`: перевіряє правильність виконання операції множення для різних варіантів чисел.
- `test_division(self)`: перевіряє правильність виконання операції ділення, зокрема перевірку ділення на нуль.
- `test_error_handling(self)`: перевіряє правильність обробки помилок при спробі виконати некоректні операції.

Програма використовує методи класу `unittest.TestCase` для виконання та перевірки результатів тестів. Тести перевіряють як коректні операції, так і ситуації, де можуть виникати помилки, гарантуючи правильну роботу калькулятора.

""

```
import unittest
```

```
from lab6.main import Calculator # Замінили на правильний шлях до main.py
```

```
class TestCalculator(unittest.TestCase):
```

```
def setUp(self):
```

```
    self.calculator = Calculator()
```

```
# Завдання 1: Тестування Додавання
```

```
def test_addition(self):
```

```
    self.calculator.num1 = 5
```

```
    self.calculator.num2 = 3
```

```
    self.calculator.operator = '+'
```

```
    self.calculator.calculate()
```

```
    self.assertEqual(self.calculator.result, 8)
```

```
    self.calculator.num1 = -5
```

```
    self.calculator.num2 = -3
```

```
    self.calculator.calculate()
```

```
    self.assertEqual(self.calculator.result, -8)
```

```
    self.calculator.num1 = -5
```

```
    self.calculator.num2 = 3
```

```
    self.calculator.calculate()
```

```
    self.assertEqual(self.calculator.result, -2)
```

```
# Завдання 2: Тестування Віднімання
```

```
def test_subtraction(self):
```

```
    self.calculator.num1 = 5
```

```
    self.calculator.num2 = 3
```

```
    self.calculator.operator = '-'
```

```
    self.calculator.calculate()
```

```
    self.assertEqual(self.calculator.result, 2)
```

```
    self.calculator.num1 = 3
```

```
    self.calculator.num2 = 5
```

```
    self.calculator.calculate()
```

```
    self.assertEqual(self.calculator.result, -2)
```

```
self.calculator.num1 = -5
self.calculator.num2 = -3
self.calculator.calculate()
self.assertEqual(self.calculator.result, -2)
```

Завдання 3: Тестування Множення

```
def test_multiplication(self):
    self.calculator.num1 = 5
    self.calculator.num2 = 3
    self.calculator.operator = '*'
    self.calculator.calculate()
    self.assertEqual(self.calculator.result, 15)
```

```
self.calculator.num1 = 0
self.calculator.num2 = 5
self.calculator.calculate()
self.assertEqual(self.calculator.result, 0)
```

```
self.calculator.num1 = -5
self.calculator.num2 = 3
self.calculator.calculate()
self.assertEqual(self.calculator.result, -15)
```

Завдання 4: Тестування Ділення

```
def test_division(self):
    self.calculator.num1 = 6
    self.calculator.num2 = 3
    self.calculator.operator = '/'
    self.calculator.calculate()
    self.assertEqual(self.calculator.result, 2)
```

```
self.calculator.num1 = 5
```

```

self.calculator.num2 = 0
self.calculator.calculate()
self.assertFalse(self.calculator.calculate()) # Перевіряємо, що повертається False

# Завдання 5: Тестування Обробки Помилки
def test_error_handling(self):
    self.calculator.num1 = -1
    self.calculator.operator = '√'
    self.assertFalse(self.calculator.calculate()) # Перевіряємо, що повертається False

    self.calculator.num1 = 5
    self.calculator.num2 = 0
    self.calculator.operator = '/'
    self.assertFalse(self.calculator.calculate()) # Перевіряємо, що повертається False

if __name__ == "__main__":
    unittest.main()

```

Результат роботи програми:

```

.Помилка: ділення на нуль.
Помилка: ділення на нуль.
.Неможливо обчислити квадратний корінь з від'ємного числа.
Помилка: ділення на нуль.
...
-----
Ran 5 tests in 0.001s

OK
Name                               Stmts  Miss  Cover
-----
lab6\main.py                        65     36    45%
lab6\test_calculator.py            67      1    99%
-----
TOTAL                               132     37    72%

```

Рис. 1 – Результат тестування

Висновок: У ході виконання ЛР я створив набір юніт-тестів, які перевіряють правильність основних арифметичних операцій у додатку-калькуляторі. Ці тести допомагають виявити та виправити будь-які проблеми або помилки, які можуть виникнути під час розробки чи обслуговування додатку, забезпечуючи його надійність і точність.