

Міністерство освіти і науки України
Національний Технічний Університет України
«Київський Політехнічний Інститут»
Навчально-науковий комплекс
«Інститут прикладного системного аналізу»
Кафедра системного проектування

Лабораторна робота №6
з дисципліни
«Проектування інформаційних систем»
Система автоматичного створення
довідника користувача та оформлення
коду за допомогою Coding Convention

Виконав:
студент групи ДА-72
Коваль С.С.

Мета роботи: за допомогою системи генерації довідника користувача створити документ у текстовому форматі для архітектурної програмної моделі..

Задача:

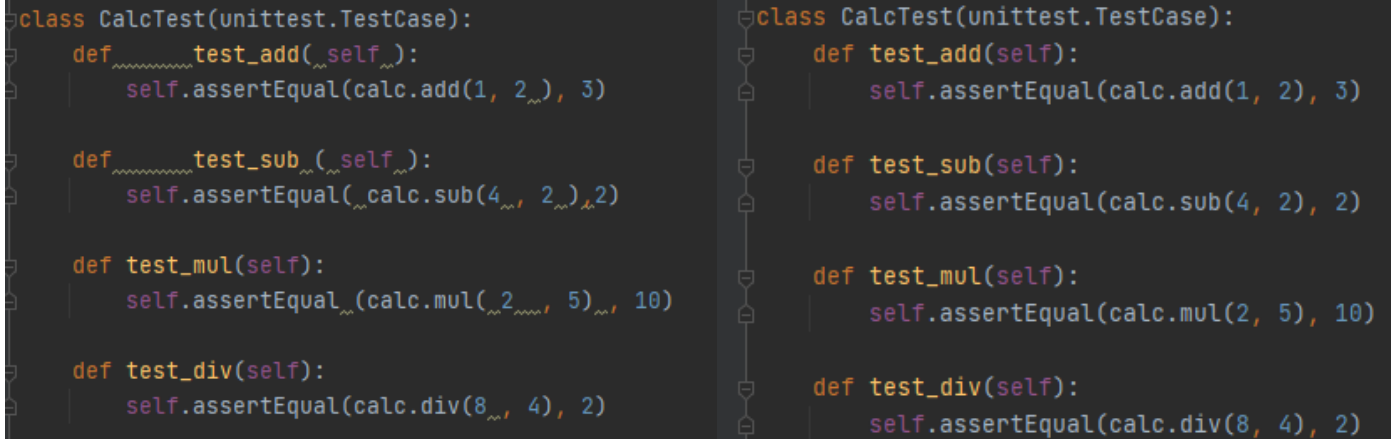
1. Вивчити теги системи генерації керівництва користувача.
2. Створити опис для всіх класів API з описом призначення кожного класу, методів класу і членів класу.
3. Згенерувати документацію.

Хід роботи

Наведемо приклади Coding Convention - PEP8:

Неправильно

Правильно



The image shows two side-by-side code snippets in a dark-themed editor. The left snippet, labeled 'Неправильно' (Incorrect), shows a Python class 'CalcTest' with methods 'test_add', 'test_sub', 'test_mul', and 'test_div'. The indentation is inconsistent, using a mix of tabs and spaces. The right snippet, labeled 'Правильно' (Correct), shows the same class and methods but with consistent 4-space indentation for all nested lines. The code in both snippets is as follows:

```
class CalcTest(unittest.TestCase):
    def test_add(self):
        self.assertEqual(calc.add(1, 2), 3)

    def test_sub(self):
        self.assertEqual(calc.sub(4, 2), 2)

    def test_mul(self):
        self.assertEqual(calc.mul(2, 5), 10)

    def test_div(self):
        self.assertEqual(calc.div(8, 4), 2)
```

Use 4 spaces per indentation level.

Class names in **CapWords**

Method, function and variables names

in `lowercase_with_underscores`

Private methods and properties start with `__double_underscore`

Always use 4 spaces for indentation (don't use tabs)

Max line-length: 80 characters (especially in comments)

Сгенеруємо автоматично документацію для декількох класів за допомогою pydoc.

```
(venv) C:\Users\Sergey\PycharmProjects\calcunit>python -m pydoc calc
Help on module calc:

NAME
    calc - The calc module

FUNCTIONS
    add(a, b)
        :param a: number a
        :param b: number b
        :return: a+b

    div(a, b)
        :param a: number a
        :param b: number b
        :return: a/b

    greaterThan(a, b)
        :param a: number a
        :param b: number b
        :return: a>b

    mul(a, b)
        :param a: number a
        :param b: number b
        :return: a*b

    sub(a, b)
        :param a: number a
        :param b: number b
        :return: a-b
```

Висновки

В даній лабораторній роботі було створено документацію за допомогою pyDoc. Документацію було створено автоматично за допомогою інструмента pyDoc. Також, було розглянуто Coding Convention PEP8 – що є сучасною домовленістю по коду для мови програмування Python.