Unit-тестирование - это процесс тестирования отдельных модулей или компонентов программного обеспечения, чтобы убедиться в их правильной работе. В Python для написания unit-тестов используется стандартная библиотека unittest, которая предоставляет набор инструментов для написания и запуска тестов.

Пример написания теста с использованием unittest:

import unittest

def add\_numbers(x, y):

return x + y

class TestAddNumbers(unittest.TestCase):

def test\_add\_numbers(self):

self.assertEqual(add\_numbers(1, 2), 3)

self.assertEqual(add\_numbers(-1, 1), 0)

self.assertEqual(add\_numbers(0, 0), 0)

if \_\_name\_\_ == '\_\_main\_\_':

unittest.main()

В этом примере мы определяем функцию add\_numbers, которая складывает два числа, и класс TestAddNumbers, который наследуется от unittest.TestCase. В классе TestAddNumbers мы определяем метод test\_add\_numbers, который проверяет правильность работы функции add\_numbers с помощью метода assertEqual.

Метод assertEqual сравнивает два значения и выдает ошибку, если они не равны. Если все проверки пройдены успешно, то тест считается пройденным.

Чтобы запустить тесты, нужно запустить этот файл как обычную программу. Если все тесты пройдены успешно, то вы увидите сообщение об успешном прохождении тестов.

unittest позволяет также использовать другие методы для проверки результатов, например, assertTrue, assertIn и другие. Также можно использовать декораторы setUp и tearDown, чтобы настроить и очистить состояние перед и после каждого теста.