****

**Тема: Интеграционное тестирование.**

def cot(x):

    return (1/math.tan(x))

def sec(x):

    return 1/math.cos(x)

def csc(x):

    return 1/math.sin(x)

def lab(x):

    if x<=0:

        y=55         #(((((sec(x)-cot(x))+math.cos(x))\*math.tan(x))+(csc(x)/csc(x)/cot(x)))\*\*2)

    elif x>0:

        y=math.cos(x)

        return y

    #pytest ods.py

def test\_answer():

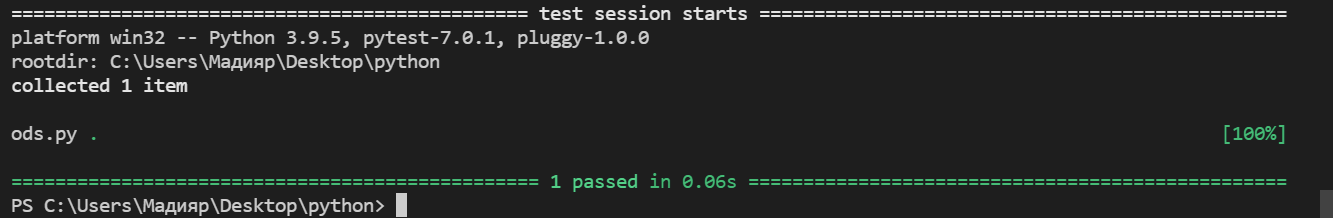
    assert lab(1) == math.cos(1)

    assert cot(1) == (1/math.tan(1))

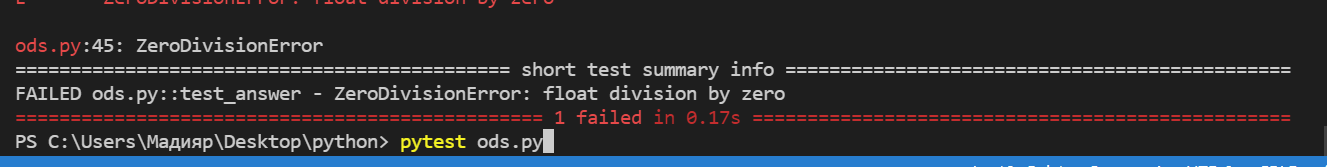
    assert sec(1)==1/math.cos(1)

    assert csc(1)==1/math.sin(1)

При правильном коде, выдает положительный результат:



При испорченном коде выдаёт отрицательный результат:



**Заключение:** интеграционное тестирование — одна из фаз тестирования программного обеспечения, при которой отдельные программные модули объединяются и тестируются в группе. Обычно интеграционное тестирование проводится после модульного тестирования и предшествует системному тестированию. Интеграционное тестирование в качестве входных данных использует модули, над которыми было проведено модульное тестирование, группирует их в более крупные множества, выполняет тесты, определённые в плане тестирования для этих множеств, и представляет их в качестве выходных данных и входных для последующего системного тестирования.

Т.е. здесь проверяется взаимодействие интерфейсов друг с другом. Главным образом, нужно убедиться, что при взаимодействии части системы отрабатывают как задумано.