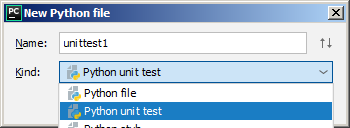
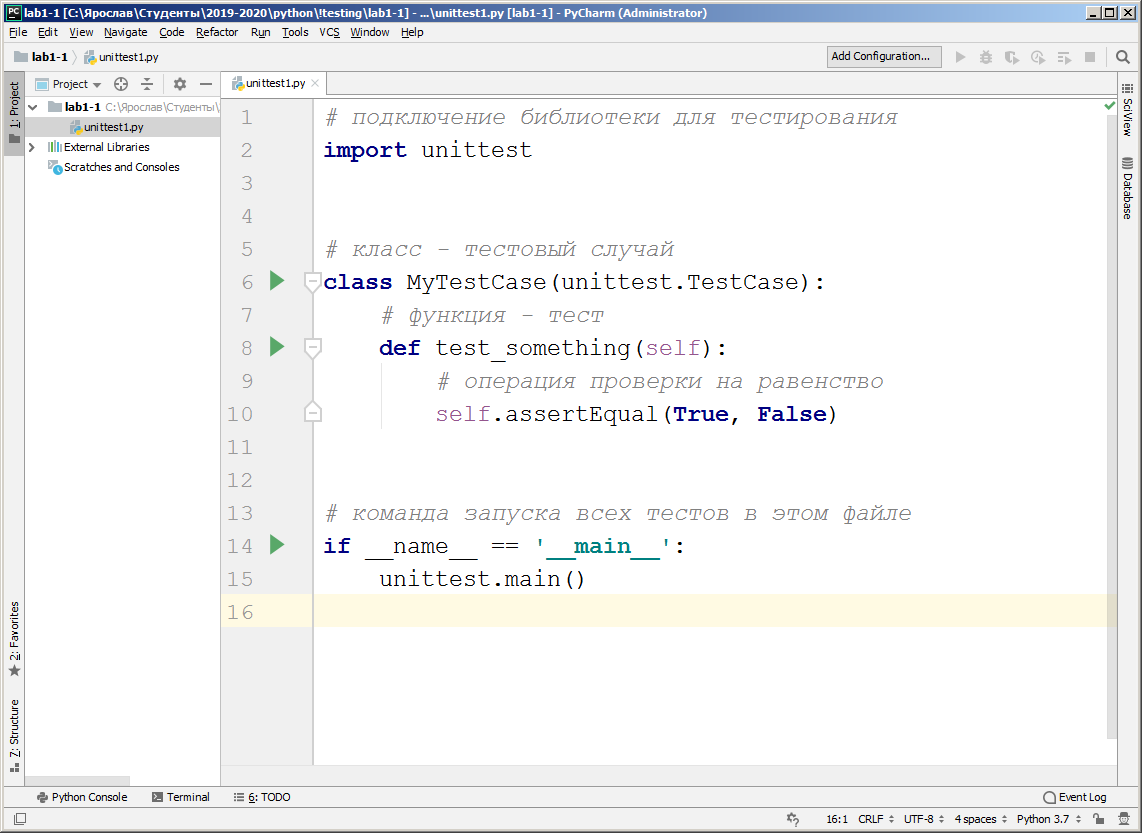
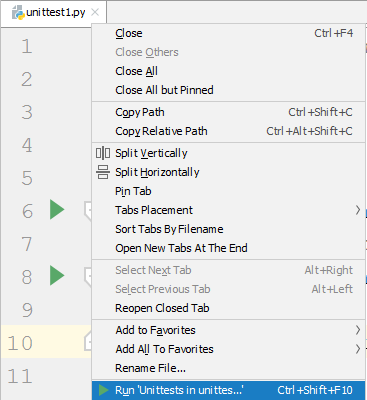
1. Создайте новый проект типа «Pure Python» и в нем – новый Python file типа «Python unit test»:



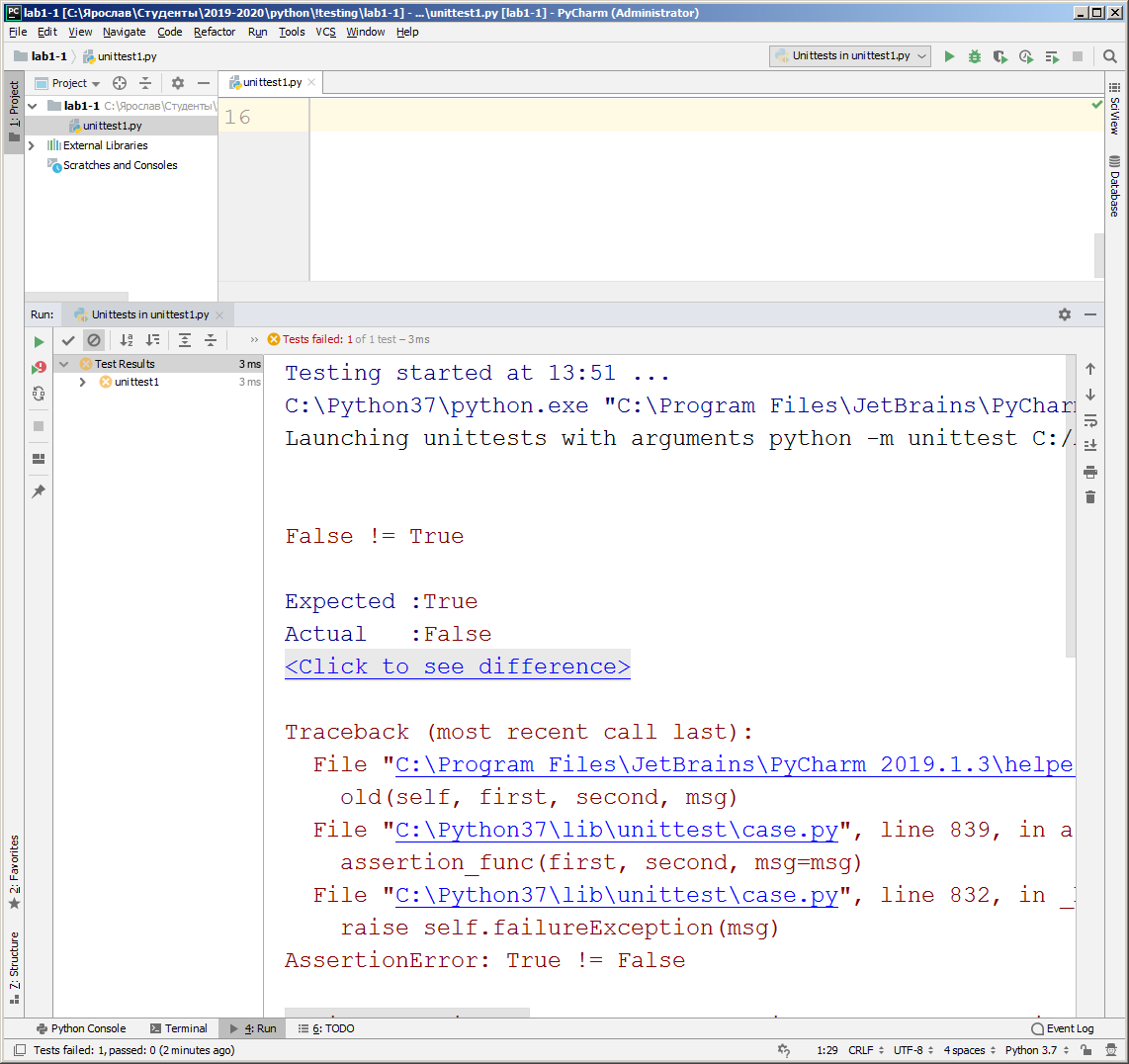
2. Рассмотрите содержание созданного по умолчанию файла с тестом:

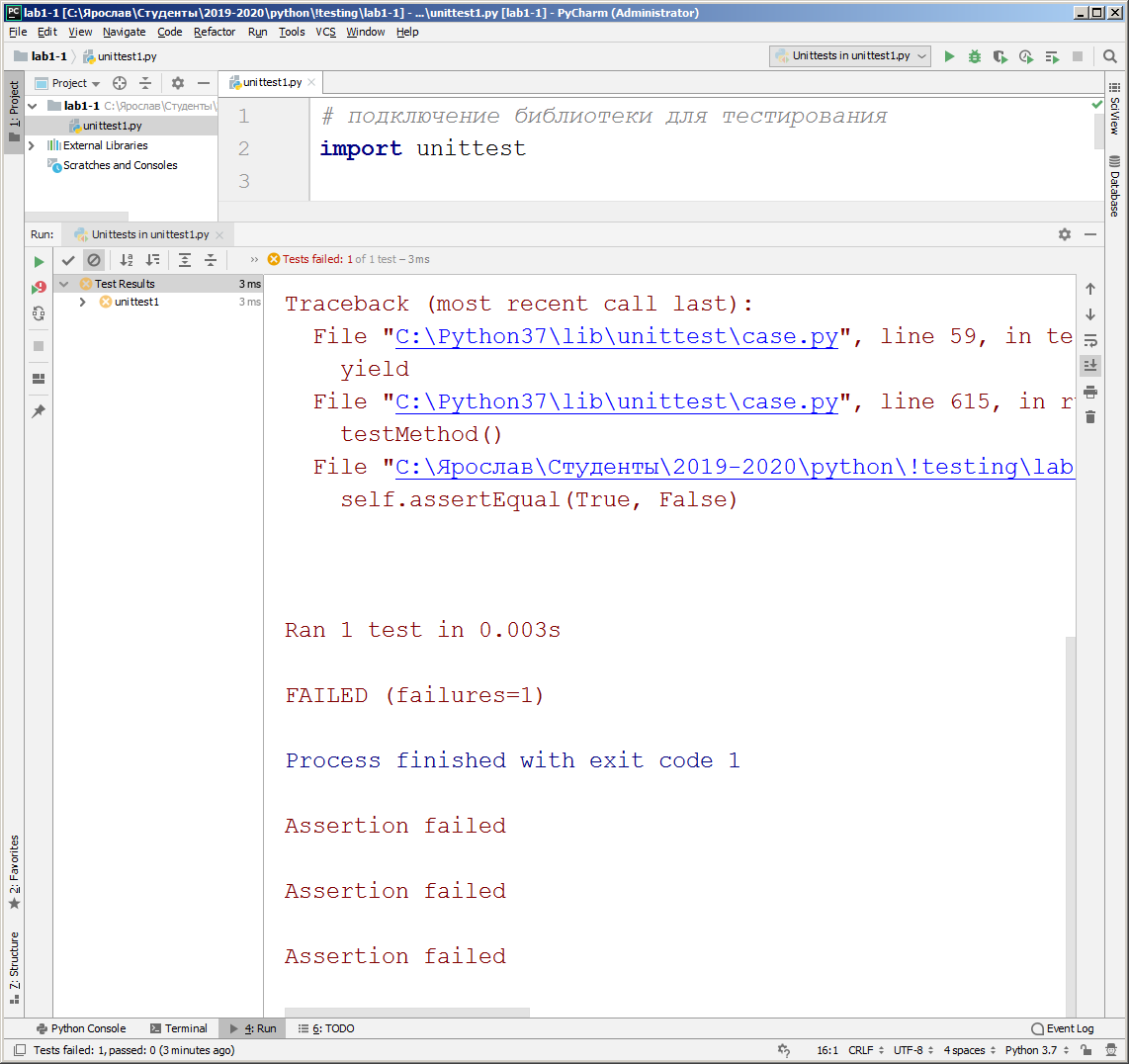


3. Запустите тест:

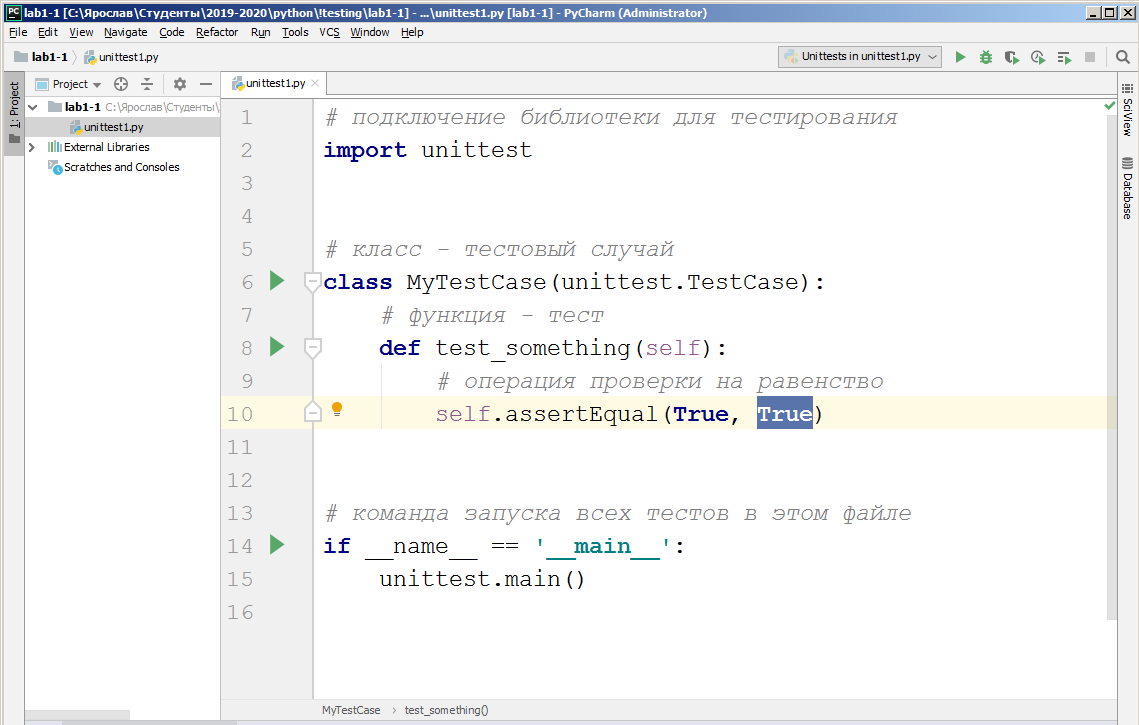


4. Ознакомьтесь с отчетом об ошибке:

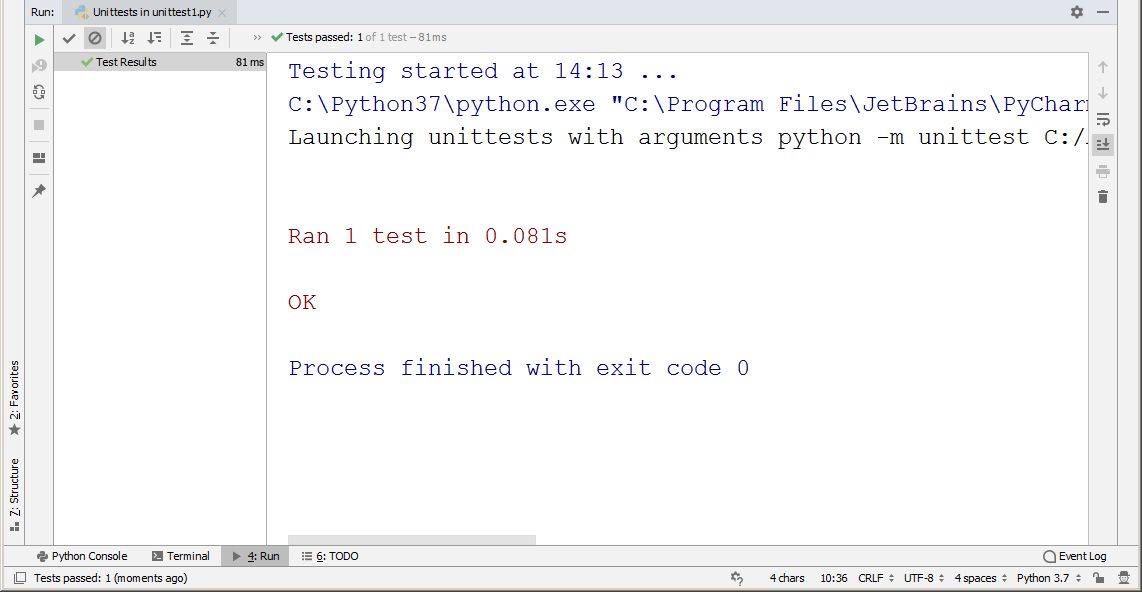




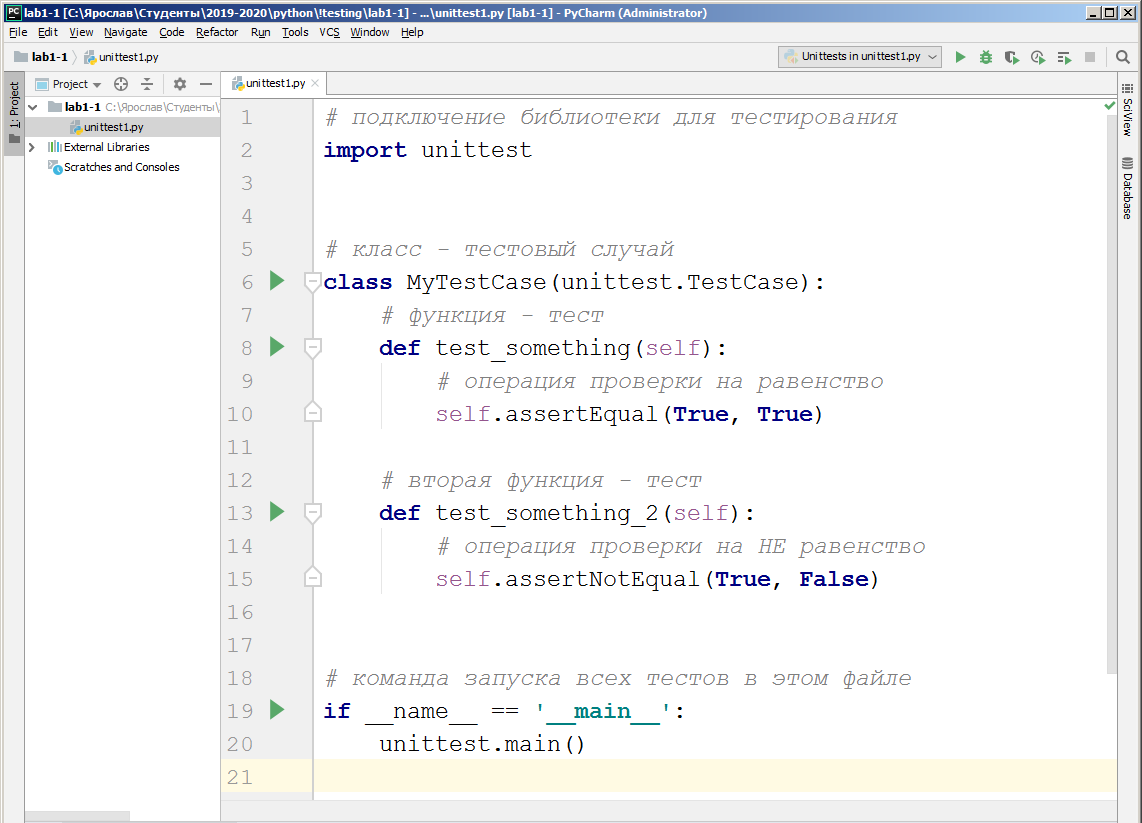
5. Исправим тест, чтобы он проходил:



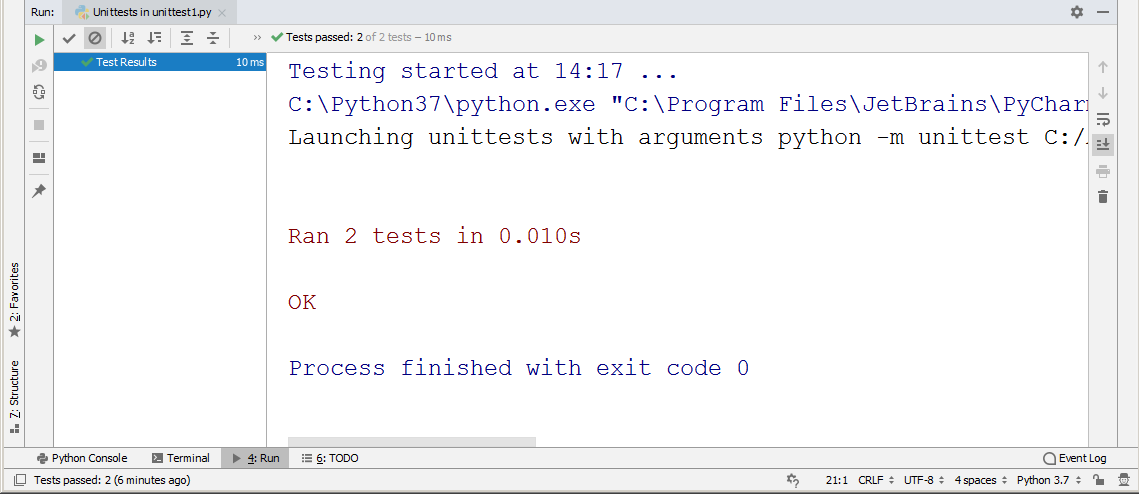
6. Запустим еще раз, рассмотрим отчет:



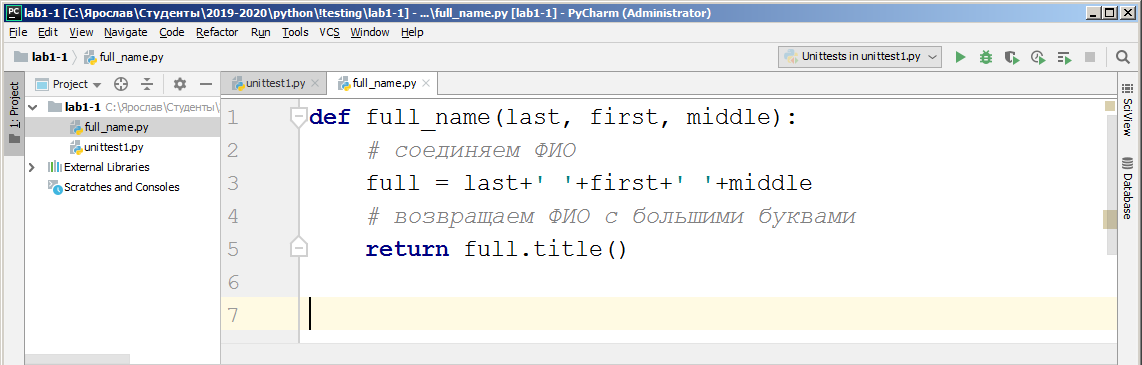
7. Сделайте копию теста и исправьте его по-другому – используйте операцию проверки на неравенство «assertNotEqual»:



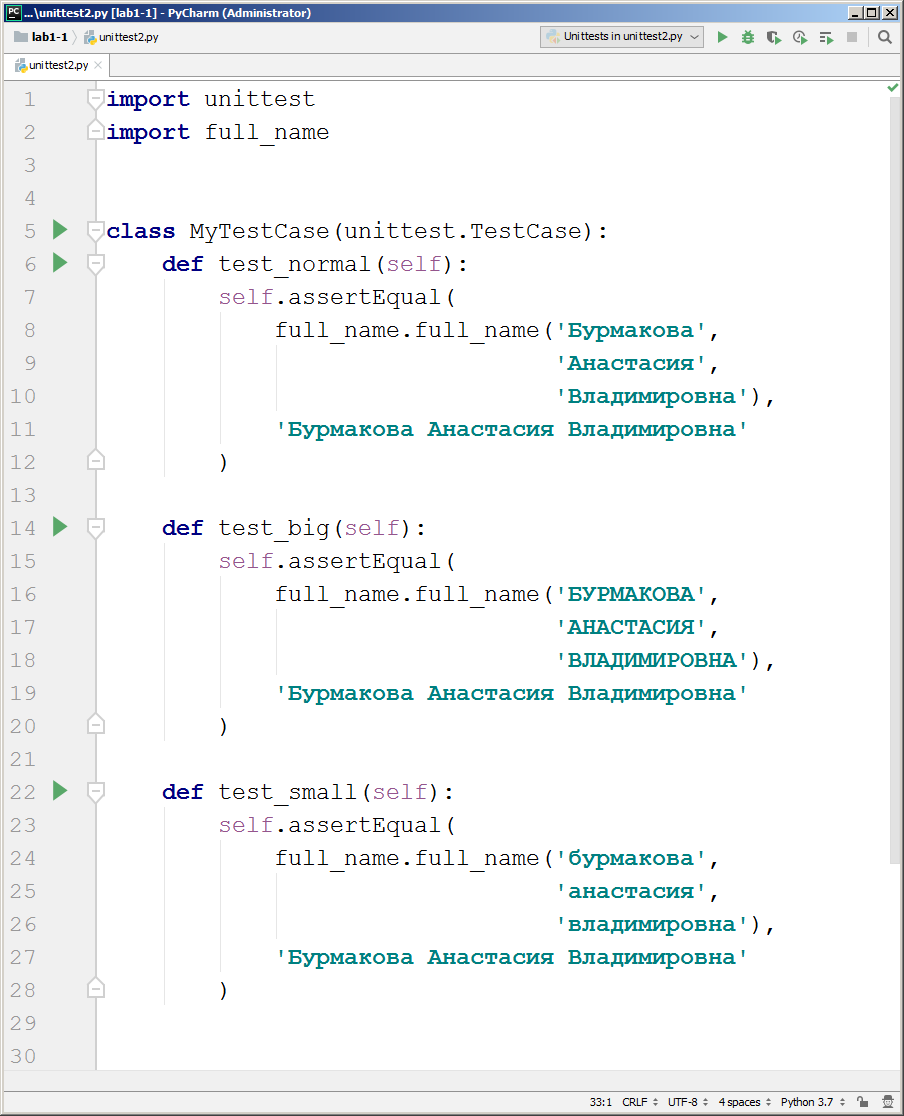
8. Убедитесь в отчете в выполнении двух тестов:



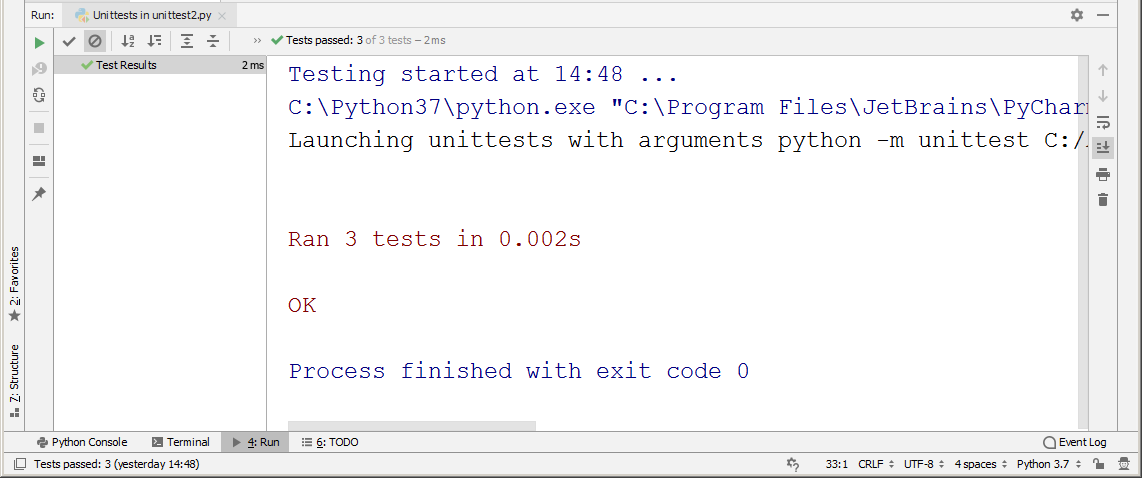
9. В том же проекте создайте новый Python file типа «Python file». В нем напишите функцию объединения фамилии, имени и отчества, переданных в качестве отдельных параметров, в одну строку. Позже мы эту функцию будем тестировать.



10. В том же проекте создайте новый Python file типа «Python unit test». В нем напишите три функции-теста для тестирования функции full\_name со своими ФИО, написанными с первых заглавных букв, целиком заглавными буквами и целиком прописными буквами. Ожидаемый результат функции – ФИО одной строкой, каждое слово с первой заглавной буквы.



11. Рассмотрите отчет об успешном прохождении тестов:



12. Создайте новый Python file типа «Python file» с названием **survey.py**.   
В нем напишите класс для сбора ответов на открытый (без вариантов ответа) анонимный опроса.

class AnonymousSurvey:

"""Сбор анонимных ответов на опросы. """

def \_\_init\_\_(self, question):

"""Сохраняет вопрос и готовится к сохранению вариантов ответов."""

self.question = question

self.responses = []

def show\_question(self):

"""Выводит вопрос"""

print(self.question)

def store\_response(self, new\_response):

"""Сохранение одного ответа"""

self.responses.append(new\_response)

def show\_results(self):

"""Вывод сохраненных ответов"""

print("Ответы:")

for response in self.responses:

print("- " + response)

13. Создайте новый Python file типа «Python file» с названием **survey\_demo.py**. В нем напишите программу для сбора ответов от пользователя с помощью созданного ранее класса.

from survey import AnonymousSurvey

question = "Какой язык программирования для вас наиболее интересен?"

my\_survey = AnonymousSurvey(question)

my\_survey.show\_question()

print("Нажмите 'q' для выхода с опроса. \n")

while True:

response = input("Language: ")

if response == 'q':

break

my\_survey.store\_response(response)

print("\n Спасибо за ответ")

my\_survey.show\_results()

14. Создайте новый Python file типа «Python unit test» с названием **survey\_test.py**. В нем напишите тест для функции добавления ответов пользователей.

import unittest

from survey import AnonymousSurvey

class TestAnonymousSurvey(unittest.TestCase):

"""Тесты для класса AnonymousSurvey"""

def test\_store\_single\_response(self):

"""Проверяет, что один ответ сохранен правильно."""

question = "Какой язык программирования вам по душе?"

my\_survey = AnonymousSurvey(question)

my\_survey.store\_response('Java')

self.assertIn('Java', my\_survey.responses)

def test\_fife\_responses(self):

"""Проверяет, что пять ответов были сохранены"""

question = "Какой язык программирования вам по душе?"

my\_survey = AnonymousSurvey(question)

responses = ['Java', 'Python', 'C#', 'Go', 'JavaScript']

for response in responses:

my\_survey.store\_response(response)

for response in responses:

self.assertIn(response, my\_survey.responses)

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

unittest.main()