



**Министерство науки и высшего образования
Российской Федерации Федеральное государственное
бюджетное образовательное учреждение высшего
образования
«Московский государственный технический
университет имени Н.Э. Баумана
(национальный исследовательский университет МГТУ
им. Н.Э. Баумана)**

**Факультет «Информатика и системы управления»
Кафедра «Системы обработки информации и
управления»**

Рубежный контроль №2.

по предмету

«Базовые компоненты интернет-технологий»

Выполнил:

студент группы № ИУ5-33Б

Пермяков Дмитрий

Проверил:

Преподаватель кафедры ИУ-5

Гапанюк Юрий

2022 г.

Постановка задачи:

- 1) Проведите рефакторинг текста программы рубежного контроля №1 таким образом, чтобы он был пригоден для модульного тестирования.
- 2) Для текста программы рубежного контроля №1 создайте модульные тесты с применением TDD - фреймворка (3 теста).

Код программы:

Код программы к РК1: [тут](#)

```
# Copyright © 2022 mightyKingRichard <dimapermyakov55@gmail.com>

import unittest
from main import *

class test_of_module_test(unittest.TestCase):

    def setUp(self):
        self.one_to_many = [
            (ide.name, ide.count_users,
             [language.name for ell in ide.languageID for language in
              LANGUAGES if language.ID == ell])
            for ide in IDE
        ]
        self.many_to_many_temp = [
            (language.name, language.ID, ide_language.IDE_ID)
            for language in LANGUAGES
            for ide_language in IDE_LANGUAGES
            if language.ID == ide_language.language_ID
        ]
        self.many_to_many = [
            (ide_temp.name, ide_temp.count_users,
             [lang.name for el in ide_temp.languageID for lang in LANGUAGES
              if lang.ID == el])
            for name_language, languageID, ideID in self.many_to_many_temp
            for ide_temp in IDE
            if ide_temp.ID == ideID
        ]

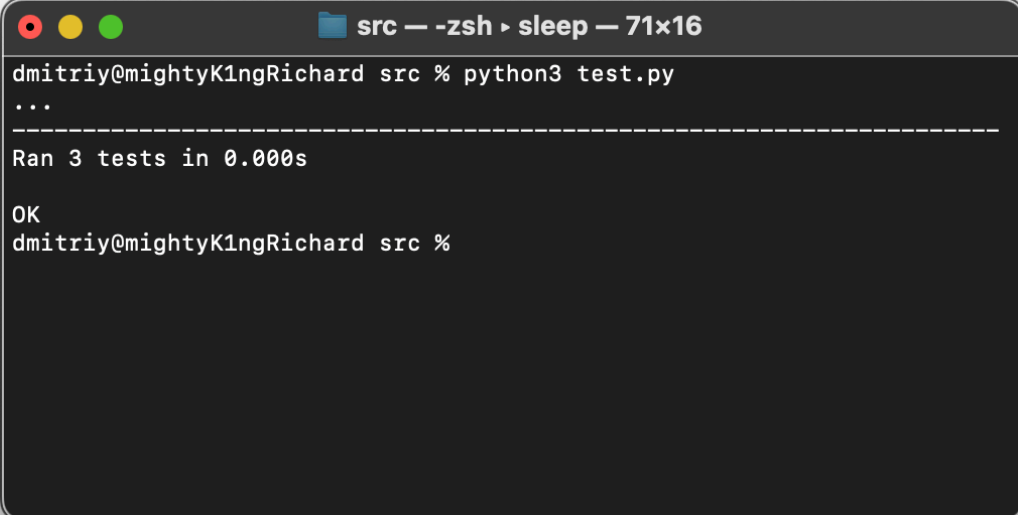
    def test_task01(self):
        expected_result = [
            ('Visual Studio', 343304, ['C#', 'C++', 'HTML', 'Python',
            'Rust']),
            ('VScode', 908311, ['C#', 'C++', 'CSS', 'HTML', 'JavaScript',
            'Python', 'Rust'])
        ]
        result = task_01(self.one_to_many)
        self.assertEqual(result, expected_result)

    def test_task02(self):
        expected_result = [
            ('VScode', 7), ('Visual Studio', 5), ('PyCharm', 3), ('XCode',
            3),
            ('Atom', 1), ('LITE', 1),
            ('IntelliJ IDEA', 1), ('Rider', 1)
        ]
        result = task_02(self.one_to_many)
        self.assertEqual(result, expected_result)
```

```
def test_task03(self):
    expected_result = [
        ('Atom', ['CSS']), ('IntelliJ IDEA', ['JavaScript']), ('PyCharm',
        ['Python', 'HTML', 'CSS']),
        ('Rider', ['C#']), ('Visual Studio', ['C#', 'C++', 'Python',
        'HTML', 'Rust']), ('XCode', ['Swift', 'HTML', 'CSS'])
    ]
    result = task_03(self.many_to_many)
    self.assertEqual(result, expected_result)

if __name__ == '__main__':
    unittest.main()
```

Результат работы программы:

A terminal window with a dark background and light text. The title bar shows a folder icon, the text 'src', and a prompt '-zsh' followed by a right arrow and 'sleep' with a window size of '71x16'. The terminal content shows the command 'python3 test.py' being executed, followed by three dots, a dashed line separator, the output 'Ran 3 tests in 0.000s', the word 'OK', and the prompt 'dmitriy@mightyKingRichard src %'.

```
src — -zsh ▶ sleep — 71x16
dmitriy@mightyKingRichard src % python3 test.py
...
-----
Ran 3 tests in 0.000s
OK
dmitriy@mightyKingRichard src %
```