Московский государственный технический
Университет им. Н.Э. Баумана

Факультет «Информатика и системы управления»
Кафедра ИУ5 «Системы обработки информации и управления»

Курс «Базовые компоненты интернет-технологий»

Отчет по рубежному контролю №2

Выполнил:
студент группы ИУ5-51Б
Афанасьев Д. М.

Проверил:
Гапанюк Е.Ю.

2022

Условия рубежного контроля №2 по курсу БКИТ

Рубежный контроль представляет собой разработку тестов на языке Python.

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).

Измененный код РК1:

```
 from operator import itemgetter
class Teacher:
    def __init__(self, id, name, wage, id_course):
        self.id = id
        self.name = name
        self.wage = wage
        self.id_course = id_course
class Course:
    def __init__(self, id, name):
        self.id = id
        self.name = name
class Teacher_and_course:
    def __init__(self, id_teacher, id_course):
        self.id_teacher = id_teacher
        self.id_course = id_course
teachers = [
    Teacher(1, "Александров", 1000, 1),
    Teacher(2, "Иванов", 1500, 3),
    Teacher(3, "Барановский", 2000, 2),
    Teacher(4, "Егоров", 1600, 4),
    Teacher(5, "Ченский", 1800, 2),
    Teacher(6, "Соколов", 1300, 4),
    Teacher(7, "Сороковиков", 1900, 1),
    Teacher(8, "Смирнов", 1600, 3),
]
courses = [
    Course(1, "физика"),
    Course(2, "математика"),
    Course(3, "химия"),
    Course(4, "информатика")
    ]
teacherofcourse = [
    Teacher_and_course(1, 1),
    Teacher_and_course(2, 3),
    Teacher_and_course(3, 2),
    Teacher_and_course(4, 4),
    Teacher_and_course(5, 2),
    Teacher_and_course(6, 4),
    Teacher_and_course(7, 3),
    Teacher_and_course(8, 1),
    ]
def one_to_many(teachers, courses):
    return [(t.name, t.wage, c.name)
            for t in teachers
```

```
            for c in courses
            if t.id_course == c.id]
def many_to_many(teachers, courses):
    many_to_many_tmp = [(c.name, tc.id_course, tc.id_teacher)
                        for c in courses
                        for tc in teacherofcourse
                        if c.id == tc.id_course]
    return [(t.id, id_course)
            for name, id_course, id_teacher in many_to_many_tmp
            for t in teachers
            if t.id == id_teacher]

def many_to_many(teachers, courses):
        many_to_many_temp = [(c.name, tc.id_course, tc.id_teacher)
                             for c in courses
                             for tc in teacherofcourse
                             if c.id == tc.id_course]
        return [(t.name, t.wage, course_name)
                for course_name, id_course, id_teacher in many_to_many_temp
                for t in teachers
                if t.id == id_teacher]

def A1(courses, teachers) -> list:
    res1 = sorted(one_to_many(teachers, courses), key=itemgetter(2))
    return list(res1)
def A2(courses, teachers) -> list:
    res2_unsorted = []
    for c in courses:
        c_courses = list(
            filter(lambda i: i[2] == c.name, one_to_many(teachers, courses)))
        if len(c_courses) > 0:
            c_sizes = [sal for _, sal, _ in c_courses]
            c_sizes_sum = sum(c_sizes) / 2
            res2_unsorted.append((c.name, c_sizes_sum))
    return sorted(res2_unsorted, key=itemgetter(1), reverse=True)

def A3(courses, teachers):
    res3 = {}
    for c in courses:
        if 'физика' in c.name:
            c_courses = list(
                filter(lambda i: i[2] == c.name, many_to_many(teachers,
courses)))
            c_courses_names = [x for x, _, _ in c_courses]
            res3[c.name] = c_courses_names
    return res3

if __name__ == '__main__':
    print('Task A1')
    print(A1(courses, teachers))
    print('Task A2')
    print(A2(courses, teachers))
    print('Task A3')
    print(A3(courses, teachers))
```

Тестирование:

```python
import unittest
from changed_rk1 import *

class rk_test(unittest.TestCase):
    def setUp(self):
        self.courses = [
            Course(1, "физика"),
            Course(2, "математика"),
            Course(3, "химия"),
            Course(4, "информатика")
        ]
        self.teachers = [
            Teacher(1, "Александров", 1000, 1),
            Teacher(2, "Иванов", 1500, 3),
            Teacher(3, "Барановский", 2000, 2),
            Teacher(4, "Егоров", 1600, 4),
            Teacher(5, "Ченский", 1800, 2),
            Teacher(6, "Соколов", 1300, 4),
            Teacher(7, "Сороковиков", 1900, 1),
            Teacher(8, "Смирнов", 1600, 3),
        ]
        self.teachers_and_courses = [
            Teacher_and_course(1, 1),
            Teacher_and_course(2, 3),
            Teacher_and_course(3, 2),
            Teacher_and_course(4, 4),
            Teacher_and_course(5, 2),
            Teacher_and_course(6, 4),
            Teacher_and_course(7, 3),
            Teacher_and_course(8, 1),
        ]
    def test_A1(self):
        expected_result = [
        ('Егоров', 1600, 'информатика'),
        ('Соколов', 1300, 'информатика'),
        ('Барановский', 2000, 'математика'),
        ('Ченский', 1800, 'математика'),
        ('Александров', 1000, 'физика'),
        ('Сороковиков', 1900, 'физика'),
        ('Иванов', 1500, 'химия'),
        ('Смирнов', 1600, 'химия'),
        ]
        result = A1(self.courses, self.teachers)
        self.assertEqual(result, expected_result)
    def test_A2(self):
        expected_result = [
            ('математика', 1900.0),
            ('химия', 1550.0),
            ('физика', 1450.0),
            ('информатика', 1450.0)
        ]
        result = A2(self.courses, self.teachers)
        self.assertEqual(result, expected_result)
    def test_A3(self):
        expected_result = {
            'физика': ['Александров', 'Смирнов'],
        }
        result = A3(self.courses, self.teachers)
        self.assertEqual(result, expected_result)
if __name__ == '__main__':
    unittest.main()
```

Результат тестирования:

```
C:\Users\Дани\AppData\Local\Programs\Python\Python311\python.exe "C:/Microsoft/Python/PyCharm Community Edition 2022.2
 .3/plugins/python-ce/helpers/pycharm/_jb_pytest_runner.py" --path C:\Users\Дани\PycharmProjects\RK2\test_file.py
Testing started at 14:13 ...
Launching pytest with arguments C:\Users\Дани\PycharmProjects\RK2\test_file.py --no-header --no-summary -q in
 C:\Users\Дани\PycharmProjects\RK2

============================= test session starts =============================
collecting ... collected 3 items
```

```
test_file.py::rk_test::test_A1 PASSED                                    [ 33%]
test_file.py::rk_test::test_A2 PASSED                                    [ 66%]
test_file.py::rk_test::test_A3 PASSED                                    [100%]

============================== 3 passed in 0.02s ==============================

Process finished with exit code 0
```