Московский государственный технический университет имени Н.Э.Баумана

Факультет Радиотехнический

Кафедра ИУ5 Системы обработки информации и управления

Отчет по рубежному контролю № 2 по курсу Базовые компоненты

Исполнитель
Студент группы РТ5-31Б _____ Татаев С.А.
«__»_____ 2022 г.

Проверил
Доцент кафедры ИУ5 _____ Гапанюк Ю.Е.
«__»_____ 2022 г.

г. Москва – 2022

Задание РК2

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).

Задание РК1

Предметная область E, вариант 27. Классы: Преподаватель, Учебный курс.

Задания:

1. «Преподаватель» и «Учебный курс» связаны соотношением один-ко-многим. Выведите список всех языков, у которых в названии присутствует буква «С», и список его синтаксических конструкций.
2. «Преподаватель» и «Учебный курс» связаны соотношением один-ко-многим. Выведите список языков со средним кол-вом букв в названии синтаксических конструкций, отсортированный по среднему кол-ву букв. Среднее кол-во букв в названии должно быть округлено до 2 знаков после запятой.
3. «Преподаватель» и «Учебный курс» связаны соотношением многие-ко-многим. Выведите список всех синтаксических конструкций, у которых название начинается с буквы «е», и названия их языков программирования.

Листинг программы, в которой выполняются задания и для которой был проведён рефакторинг (RK1.py)

```
'''Вариант - E, вариант предметной области - 27
("Учебный курс - Преподаватель")'''
from operator import itemgetter


class Teacher:
    def __init__(self, id, fio, sal, course_id):
        self.id = id
        self.fio = fio
        self.sal = sal
        self.course_id = course_id


class Course:
```

```
    def __init__(self, id, name):
        self.id = id
        self.name = name

class TC:
    def __init__(self, course_id, teacher_id):
        self.course_id = course_id
        self.teacher_id = teacher_id

courses = [Course(1, "Мат. анализ"),
           Course(2, "Ораторское мастерство"),
           Course(3, "Линейная алгебра"),
           Course(4, "Прикладная ритуалистика и оккультные технологии")]

teachers = [Teacher(1, "Гжегож Бженчишчикевич", 30000, 2),
            Teacher(2, "Жак Ле-Вак", 15000, 1),
            Teacher(3, "GORUDA SUMITH", 999999, 4)]

tc = [TC(1, 2),
      TC(2, 1),
      TC(3, 2),
      TC(4, 3)]

otm = [(t.fio, t.sal, c.name)
       for t in teachers
       for c in courses
       if t.course_id==c.id]

mtm_temp = [(c.name, _.course_id, _.teacher_id)
       for c in courses
       for _ in tc
       if c.id==_.course_id]
```

```
mtm = [(t.fio, t.sal, course_name)
        for course_name, course_id, teacher_id in mtm_temp
        for t in teachers if t.id==teacher_id]


def task1(otm):
    word = 'н'
    result1 = [c.name for c in courses if word in c.name]
    result2 = [t[0] for t in otm if t[2] in result1]
    return result1, result2


def task2(otm):
    result = []
    for c in courses:
        sals = [s[1] for s in otm if s[2] == c.name]
        if sals:
            avsal = round(sum(sals)/len(sals), 2)
        else:
            avsal = 0
        result.append((c.name, avsal))
    return sorted(result, key=itemgetter(1), reverse=True)


def task3(mtm):
    char3 = "Ж"
    return [(t.fio,[m[2] for m in mtm if m[0]==t.fio]) for t in teachers if t.fio[0] == char3]


if __name__ == '__main__':
    task1(otm)
    task2(otm)
    task3(mtm)
```

Листинг программы, в которой проводятся тесты (RK2 Tataev RT5-31B.py)

```
import unittest
import RK1


class testRK1(unittest.TestCase):
    def setUp(self):
        self.test1 = (['Мат. анализ', 'Линейная алгебра', 'Прикладная ритуалистика и оккультные технологии'], ['Жак Ле-Вак', 'GORUDA SUMITH'])
        self.test2 = [('Прикладная ритуалистика и оккультные технологии', 999999.0), ('Ораторское мастерство', 30000.0), ('Мат. анализ', 15000.0), ('Линейная алгебра', 0)]
        self.test3 = [('Жак Ле-Вак', ['Мат. анализ', 'Линейная алгебра'])]


    def test1_rk(self):
        self.assertEqual(RK1.task1(RK1.otm), self.test1)


    def test2_rk(self):
        self.assertEqual(RK1.task2(RK1.otm), self.test2)


    def test3_rk(self):
        self.assertEqual(RK1.task3(RK1.mtm), self.test3)


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

Результаты работы программы

```
...
----------------------------------------------------------------------
Ran 3 tests in 0.001s
```