

Московский государственный технический университет им.
Н.Э. Баумана

Факультет “Радиотехнический”
Кафедра ИУ5 “Системы обработки информации и управления”

Отчет по РК2 по курсу
Базовые компоненты интернет технологий

Вариант 11

Выполнил:
Студент группы РТ5-31Б
Корсаков Н.А.
Проверил:
Доцент кафедры ИУ5
Гапанюк Ю.Е.

2021г.

1. Описание задания

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

2. Текст программы

RK2.py

```
from operator import itemgetter
```

```
class PR:
```

```
    def __init__(self, id, name_pr, v, pc_id):
```

```
        self.id = id
```

```
        self.name_pr = name_pr
```

```
        self.v = v
```

```
        self.pc_id = pc_id
```

```
class PC:
```

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

```
        self.id = id
```

```
        self.name = name
```

```
class PR_Pc:
```

```
    """
```

```
    СВЯЗЬ МНОГИЕ КО МНОГИМ
```

```
    """
```

```
    def __init__(self, pc_id, pr_id):
```

```
        self.pc_id = pc_id
```

```
        self.pr_id = pr_id
```

```
Pcs = [
```

```
    PC(1, 'компьютер Acer'),
```

```
    PC(2, 'компьютер Samsung'),
```

```
    PC(3, 'компьютер Lenovo'),
```

```
    PC(4, 'Imac'),
```

```
]
```

```
prs = [
```

```
    PR(1, 'PyCharm', 224, 3),
```

```
    PR(2, 'Photoshop', 2048, 2),
```

```
PR(3, 'Zoom', 289, 1),
PR(4, 'Telegram', 129, 4),
]
```

```
prs_pcs = [
    PR_Pc(1,1),
    PR_Pc(1,4),
    PR_Pc(2,2),
    PR_Pc(2,3),
    PR_Pc(3,4),
    PR_Pc(3,3),
    PR_Pc(4,4),
    PR_Pc(4,1),
]
```

```
# Соединение данных один-ко-многим
one_to_many = [(pr.name_pr, pr.v, p.name)
for p in Pcs
for pr in prs
if pr.pc_id==p.id]

# Соединение данных многие-ко-многим
many_to_many_temp = [(p.name, pp.pc_id, pp.pr_id)
for p in Pcs
for pp in prs_pcs
if p.id==pp.pc_id]

many_to_many = [(pr.name_pr, pr.v, pc_name)
for pc_name, pc_id, pr_id in many_to_many_temp
for pr in prs if pr.id==pr_id]

def e1():
    print("Задание E1")
    res1 = []
    for name_pr, v, name in one_to_many:
        if 'компьютер' in name:
            res1.append((name, name_pr))
    return res1

def e2():
    print("\nЗадание E2")
    res2_unsorted = []
```

```

for p in Pcs:

    prss = list(filter(lambda i: i[2]==p.name, one_to_many))

    if len(prss) > 0:

        v_ob = [v for _,v,_ in prss]

        sum_a = sum(v_ob)/len(v_ob)

        res2_unsorted.append((p.name, sum_a))

res2 = sorted(res2_unsorted, key=itemgetter(1))

return res2

```

```

def e3():

    print("\nЗадание E3")

    res3 = []

    for name_pr, v, name in many_to_many:

        if name_pr.find("P") == 0:

            res3.append((name_pr, name))

    return res3

```

```

def main():

    e1()

    e2()

    e3()

```

```

if __name__ == '__main__':

    main()

```

testRK2.py

```

import unittest

import RK2

```

```

class RK_Test(unittest.TestCase):

    def test1(self):

        self.assertEqual(RK2.e1(), [('компьютер Acer', 'Zoom'), ('компьютер Samsung', 'Photoshop')])

    def test2(self):

        self.assertEqual(RK2.e2(), [('Imac', 129.0), ('компьютер Lenovo', 224.0), ('компьютер Acer', 289.0), ('компьютер Samsung', 2048.0)])

    def test3(self):

        self.assertEqual(RK2.e3(), [('PyCharm', 'компьютер Acer'), ('Photoshop', 'компьютер Samsung'), ('PyCharm', 'Imac')])

if __name__ == '__main__':

    unittest.main()

```

3. Результат работы программы

```
Ran 3 tests in 0.003s
```

```
OK
```