Московский государственный технический университет им. Н.Э. Баумана

Факультет «Радиотехнический»
Кафедра ИУ5 «Системы обработки информации и управления»

Курс «Базовые компоненты интернет-технологий»

Отчёт по рубежному контролю №2
Вариант №1

Выполнил:
студент группы РТ5-31Б
Агеев Алексей
Подпись и дата:

Проверил:
преподаватель каф. ИУ5
Гапанюк Ю.Е.
Подпись и дата:

Москва, 2021 г

Описание задания

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).

Текст программы

tut.py

```
# используется для сортировки
from operator import itemgetter

class Group:
    """
    Группа - id, название.
    """
    def __init__(self, id, name):
        self.id = id
        self.name = name

class Student:
    """
    Студент - id, фамилия, id группы, кол-во статей.
    """
    def __init__(self, id, surname, group_id, article_num):
        self.id = id
        self.surname = surname
        self.group_id = group_id
        self.article_num = article_num

class StudentsOfGroup:
    """
    Студенты групы - id студента, id группы
    """
    def __init__(self, student_id, group_id):
        self.student_id = student_id
        self.group_id = group_id

#Создаем фамилии и названия групп
surname = ["Агеев", "Петров", "Иванов", "Гонт", "Гур", "Блур"]
group = ["Группа ИУ", "Группа РК", "РТ"]

#Создаем массивы объектов студент и групп
students = [Student(i, surname[i], (i + 1) % len(group), i % 4) for i in
range(len(surname))]
```

```
groups = [Group(i, group[i]) for i in range(len(group))]

#Создаём массив объектов типа студенты группы(многие ко многим)
Students_Of_Group = [StudentsOfGroup(students[i].id, groups[i%len(group)].id)
for i in range(len(surname))]

one_to_many = [(stud.surname, stud.article_num, grp.name)
               for grp in groups
               for stud in students
               if stud.group_id == grp.id]

many_to_many = [(stud.surname, stud.article_num, grp.name)
               for stofgr in Students_Of_Group
               for stud in students
               for grp in groups
               if(stud.id == stofgr.student_id and grp.id == stofgr.group_id)]

def task1():
    """
    1 задание
    """
    print("1 задание")
    res = []
    buff = []
    for _,_,grp in one_to_many:
        if grp.find("Группа") != -1 and grp not in buff:
            buff.append(grp)
            res.append((grp, list(name for name,_,grp_temp in one_to_many if grp_temp
== grp)))
    return res

def task2():
    """
    2 задание
    """
    print("2 задание")
    #массив среднего количества статей
    index = 0
    count = 0
    middle_num = 0
    answer2 = []
    gr = ""
    for name, num, grp in one_to_many:
        if (gr != grp):
            if (count != 0):
                answer2.append((gr, (round(middle_num/count, 2))))
            gr = grp
```

```
            middle_num = 0
            count = 0
        if (gr == grp):
            middle_num += num
            count += 1
    answer2.append((gr, (round(middle_num/count, 2)))) # так как последний append не будет произведен

    answer2 = sorted(answer2, key=itemgetter(1))

    return answer2


def task3():
    """
    3 задание
    """
    print("3 задание")
    answer3 = []
    for surname, num, grp in many_to_many:
        # если фамилия начинается с "А"
        if (surname[0] == "А"):
            answer3.append((surname, grp))
    return answer3
```

test_tut.py

```
import unittest
import tut
class Test_tut(unittest.TestCase):
    def test_tasks(self):
        self.assertEqual(tut.task1(), [('Группа ИУ', ['Иванов', 'Блур']), ('Группа РК', ['Агеев', 'Гонт'])])
        self.assertEqual(tut.task2(), [('РТ', 0.5), ('Группа ИУ', 1.5), ('Группа РК', 1.5)])
        self.assertEqual(tut.task3(),[('Агеев', 'Группа ИУ')])
if __name__ == "__main__":
    unittest.main()
```

Пример работы программы

```
1 задание
2 задание
3 задание
.
----------------------------------------------------------------------
Ran 1 test in 0.003s

OK
```