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

Курс «Базовые компоненты интернет-технологий»
Рубежный контроль №2
Вариант №4 (Г)

Выполнил:
студент группы ИУ5-34Б:
Даниелян Алла Арамановна
Подпись и дата:

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

Москва, 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).

Вариант Г.
«Отдел» и «Сотрудник» связаны соотношением один-ко-многим.
Выведите список всех отделов (Rooms) у которых название (Owner_name) начинается с буквы «А»,
и список работающих в них (Computer.User_name) сотрудников.
«Отдел» и «Сотрудник» связаны соотношением один-ко-многим.
Выведите список отделов с максимальной (Computer.Price) зарплатой сотрудников в каждом (Room) отделе,
отсортированный по максимальной зарплате.
«Отдел» и «Сотрудник» связаны соотношением многие-ко-многим.
Выведите список всех связанных сотрудников (Computer.User_name)) и отделов (Room), отсортированный по отделам,
сортировка по сотрудникам произвольная.

Текст программы после рефакторинга:
```
from operator import itemgetter
from Room import Room
from Computer import Computer
from Computers_in_room import Computers_in_room
room = [
    Room (1, "Иванов В.П."),
    Room (2,"Арпов А.А."),
    Room (3, "Игнатьев И.И."),
    Room (4, "Антошкин А.К."),
    Room (5, "Репкин В.В.")
]

# COMPUTERS
comp = [
    Computer (1,"Васькин", 50000,1),
    Computer (2,"Пупкин", 52000,2),
    Computer (3,"Анюткин", 60000,1),
    Computer (4,"Гринев", 30000,3),
    Computer (5,"Одинцова", 31000,4),
    Computer (10, "Десятцова", 100000,4),
    Computer (6,"Базаров", 20000,3),
    Computer (7,"Твист", 100000,5),
    Computer (8,"Дван", 760000,5),
    Computer (9,"Кран", 50000,5)
]

comp_room = [
    Computers_in_room (1,1),
    Computers_in_room (2,2),
    Computers_in_room (3,3),
```

```
    Computers_in_room (4,4),
    Computers_in_room (5,5),

    Computers_in_room (1,6),
    Computers_in_room (2,7),
    Computers_in_room (3,8),
    Computers_in_room (4,9),
    Computers_in_room (5,1)

]
 # Соединение данных один-ко-многим
def OTM(rm,cmp):
    return [ (e.User_name, e.Price, d.ID_room, d.Owner_name)
        for d in rm
        for e in cmp
        if e.ID_room==d.ID_room
        ]

# Соединение данных многие-ко-многим
def MTM(rm,cmpr,cmp):
    temp = [(d.Owner_name, ed.CompID, ed.RoomID)
        for d in rm
        for ed in cmpr
        if d.ID_room==ed.RoomID]
    return [(e.User_name, e.Price, own_name, room_id)
        for own_name, comp_id, room_id in temp
        for e in cmp
        if e.ID_computer==comp_id ]

def G1 (a, otm):
    r=sorted(otm,key = itemgetter(3))
    cache = ""
    result = []
    count = -1
    for x in r:
        if cache != x[3] and x[3][0] == a:
            cache = x[3]
            result.append(f"Комната №{x[2]}, Ответственный: {x[3]}, Список пользователей:")
            count+=1
            result[count] +=x[0]+' '
        elif x[3][0] == 'A':
            result[count] +=x[0]+' '
        else:
            break
    return result

def G2(rm,otm):
    # Перебираем все ROOMS
    res2={}
    for d in rm:
        # Список компьютеров комнаты
```

```
        room_list = list(filter(lambda i: i[2]==d.ID_room, otm))
        mx = float(0)
        #Собираем списки ключей и значений
        if len(room_list)>0:
            for r in room_list:
                if float(r[1])>mx:
                    mx = float(r[1])
            res2["Комната №{}".format(room_list[0][2])] = mx
    #Сортируем словарь по возрастанию значений
    res_tuples = sorted(res2.items(), key = itemgetter(1))
    return res_tuples
def G3(rm,mtm):
    res3 = {}
# Перебираем все ROOMS
    for d in rm:
        # Список ROOMS
        rm_comps = list(filter(lambda i: i[3]==d.ID_room, mtm))
        # Только UserName
        rm_comps_unm = [x for x,_,_,_ in rm_comps]
        res3["Комната №{}".format(d.ID_room)] = rm_comps_unm
    return res3
def main():
    """Основная функция"""

    # Соединение данных один-ко-многим
    one_to_many = OTM(room, comp)
    # Соединение данных многие-ко-многим
    #many_to_many_temp = MTM(room,comp_room)
    many_to_many = MTM(room,comp_room, comp)
    print('Задание Г1')
    print (G1('A',one_to_many))
    print('\nЗадание Г2')
    print (G2(room,one_to_many))
    print('\nЗадание Г3')
    print (G3(room, many_to_many))

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

Файл test_tdd_rk.py

```
import unittest
from main import OTM,MTM,G1,G2,G3
from Room import Room
from Computer import Computer
from Computers_in_room import Computers_in_room
room = [
    Room (1, "Иванов В.П."),
    Room (2,"Арпов А.А."),
    Room (3, "Игнатьев И.И."),
    Room (4, "Антошкин А.К."),
    Room (5, "Репкин В.В.")
]
```

```
# COMPUTERS
comp = [
    Computer (1,"Васькин", 50000,1),
    Computer (2,"Пупкин", 52000,2),
    Computer (3,"Анюткин", 60000,1),
    Computer (4,"Гринев", 30000,3),
    Computer (5,"Одинцова", 31000,4),
    Computer (10, "Десятцова", 100000,4),
    Computer (6,"Базаров", 20000,3),
    Computer (7,"Твист", 100000,5),
    Computer (8,"Дван", 760000,5),
    Computer (9,"Кран", 50000,5)
]

comp_room = [
    Computers_in_room (1,1),
    Computers_in_room (2,2),
    Computers_in_room (3,3),
    Computers_in_room (4,4),
    Computers_in_room (5,5),

    Computers_in_room (1,6),
    Computers_in_room (2,7),
    Computers_in_room (3,8),
    Computers_in_room (4,9),
    Computers_in_room (5,1)

]
otm = OTM(room,comp)
mtm = MTM(room,comp_room, comp)
class TestLab1_get_sqr_root (unittest.TestCase):
    def test_one (self):
        self.assertEqual (G1('A', otm),['Комната №4, Ответственный: Антошкин А.К., Список пользователей:Одинцова Десятцова ', 'Комната №2, Ответственный: Арпов А.А., Список пользователей:Пупкин '])
    def test_two(self):
        self.assertEqual (G2(room,otm),[('Комната №3', 30000.0), ('Комната №2', 52000.0), ('Комната №1', 60000.0), ('Комната №4', 100000.0), ('Комната №5', 760000.0)])
    def test_three (self):
        self.assertEqual(G3(room, mtm),{'Комната №1': ['Васькин', 'Одинцова'], 'Комната №2': ['Пупкин'], 'Комната №3': ['Анюткин'], 'Комната №4': ['Гринев'], 'Комната №5': ['Одинцова']})
```

Результат выполнения основной программы:

```
(env_rk) PS C:\Users\allad\OneDrive\Документы\rk1_bkit> & c:/Users/allad/OneDrive/Документы/rk1_bkit/env_rk/Script
s/python.exe c:/Users/allad/OneDrive/Документы/rk1_bkit/main.py
Задание Г1
['Комната №4, Ответственный: Антошкин А.К., Список пользователей:Одинцова Десятцова ', 'Комната №2, Ответственный:
 Арпов А.А., Список пользователей:Пупкин ']

Задание Г2
[('Комната №3', 30000.0), ('Комната №2', 52000.0), ('Комната №1', 60000.0), ('Комната №4', 100000.0), ('Комната №5
', 760000.0)]

Задание Г3
{'Комната №1': ['Васькин', 'Одинцова'], 'Комната №2': ['Пупкин'], 'Комната №3': ['Анюткин'], 'Комната №4': ['Грине
в'], 'Комната №5': ['Одинцова']}
(env_rk) PS C:\Users\allad\OneDrive\Документы\rk1_bkit>
```

Результат выполнения тестов:

```
(env_rk) PS C:\Users\allad\OneDrive\Документы\rk1_bkit> python -m unittest .\test_tdd_rk.py
...
----------------------------------------------------------------------
Ran 3 tests in 0.001s

OK
(env_rk) PS C:\Users\allad\OneDrive\Документы\rk1_bkit>
```