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»

Рубежный контроль №2
Вариант 4

Выполнил:
студент группы ИУ5-34Б
Гордеев Никита
Подпись и дата:

Проверил:
Гапанюк Ю. Е.
Подпись и дата:

Москва, 2021 г.

Постановка задачи

Вариант Г.

1. «Отдел» и «Сотрудник» связаны соотношением один-ко-многим. Выведите список всех отделов, у которых название начинается с буквы «А», и список работающих в них сотрудников.
2. «Отдел» и «Сотрудник» связаны соотношением один-ко-многим. Выведите список отделов с максимальной зарплатой сотрудников в каждом отделе, отсортированный по максимальной зарплате.
3. «Отдел» и «Сотрудник» связаны соотношением многие-ко-многим. Выведите список всех связанных сотрудников и отделов, отсортированный по отделам, сортировка по сотрудникам произвольная.

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).

4	Компьютер	Дисплейный класс

Текст программы

Файл: RK_2.py

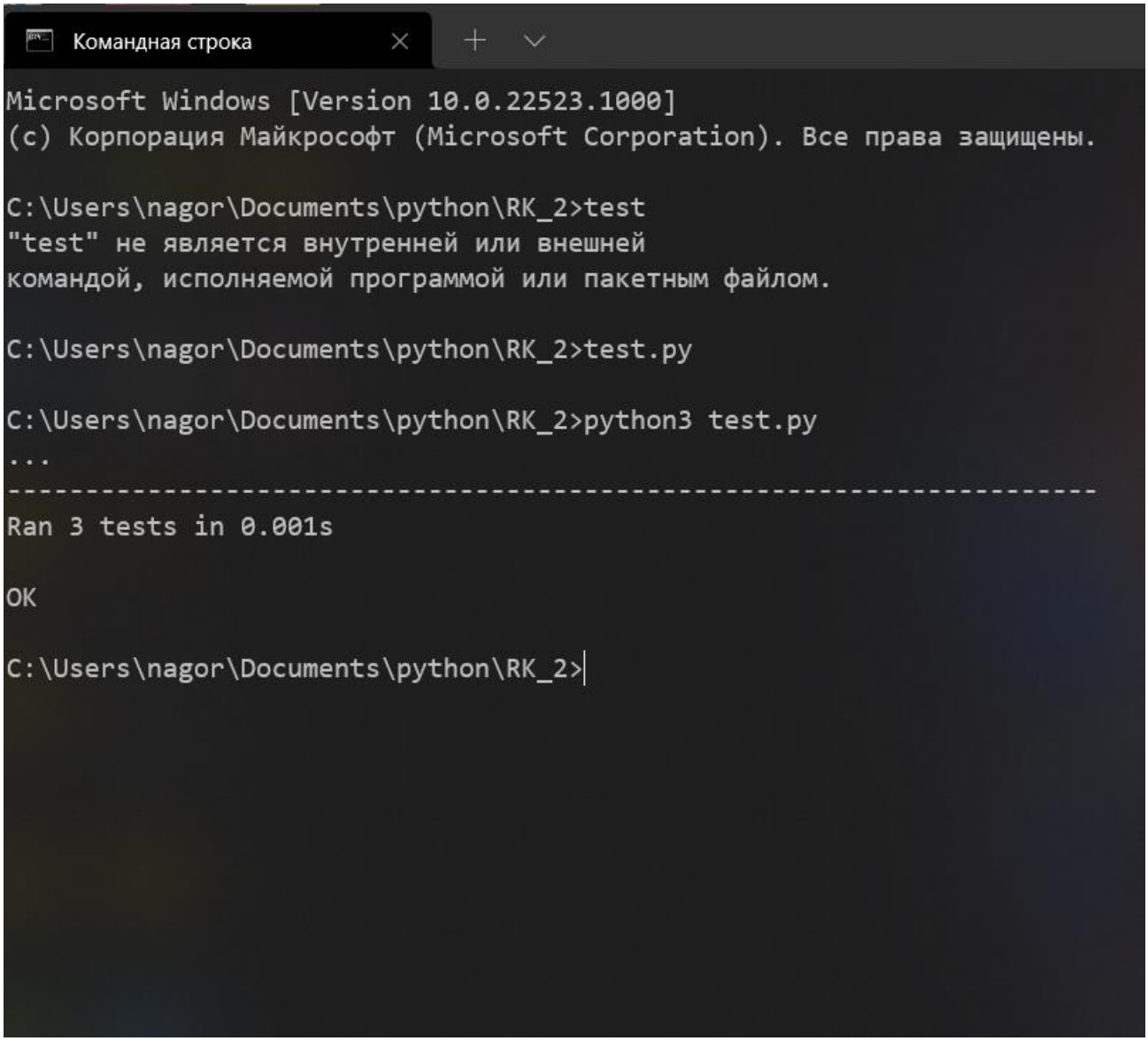
```python
# используется для сортировки
from operator import itemgetter

class Comp:
    def __init__(self, id, fio, sal, disp_class_id):
        self.id = id
        self.fio = fio
        self.sal = sal
        self.disp_class_id = disp_class_id

class Disp_class:
    def __init__(self, id, name):
        self.id = id
        self.name = name

class CompDisp_class:
    def __init__(self, disp_class_id, comp_id):
        self.disp_class_id = disp_class_id
        self.comp_id = comp_id

disp_classs = [Disp_class(1, 'А-класс'),
    Disp_class(2, 'Б-класс'),
    Disp_class(3, 'В-класс'),

    Disp_class(11, 'Г-класс'),
    Disp_class(22, 'Д-класс'),
    Disp_class(33, 'Е-класс'),]

comps = [Comp(1, 'Asus', 134200, 2),
    Comp(2, 'HP', 87123, 3),
    Comp(3, 'Dell', 204500, 2),
    Comp(4, 'ThinkPad', 64990, 1),
    Comp(5, 'Acer', 109990, 3),]

comps_disp_classs = [CompDisp_class(1,1),
    CompDisp_class(3,2),
    CompDisp_class(3,3),
    CompDisp_class(2,4),
    CompDisp_class(1,5),

    CompDisp_class(33,1),
    CompDisp_class(33,2),
    CompDisp_class(11,3),
    CompDisp_class(11,4),
    CompDisp_class(22,5),]


def part_1(one_to_many):
    res_11 = [(o.name,list(fio for fio,_,name in one_to_many if name == o.name)) for o in disp_classs if o.name[0] == 'A']
    return res_11
```

```python
def part_2(one_to_many):
    res_12_unsorted = []
    # Перебираем все дисплейные классы
    for o in disp_classs:
        # Список компьютеров дисплейного класса
        o_comps = list(filter(lambda x: x[2] == o.name, one_to_many))
        # Если дисплейный класс не пустой
        if len(o_comps) > 0:
            res_12_unsorted.append((o.name, max(o_comps, key=lambda x: x[1])[1]))

    # Сортировка по максимальной стоимости
    res_12 = sorted(res_12_unsorted, key=itemgetter(1), reverse=True)
    return res_12



def part_3(many_to_many):
    res_13 = []
    # Перебираем все дисплейные классы
    for comp,_,disp_class in many_to_many:
        res_13.append((comp, disp_class))
    res_13 = sorted(res_13, key=itemgetter(1))
    return res_13



def main():
    one_to_many = [(m.fio, m.sal, o.name)
        for o in disp_classs
        for m in comps
        if m.disp_class_id == o.id]

    many_to_many_temp = [(o.name, mo.disp_class_id, mo.comp_id)
        for o in disp_classs
        for mo in comps_disp_classs
        if o.id == mo.disp_class_id]

    many_to_many = [(m.fio, m.sal, disp_class_name)
        for disp_class_name, disp_class_id, comp_id in many_to_many_temp
        for m in comps if m.id == comp_id]

    print(one_to_many)

    print(many_to_many)

    print('Задание Г1')
    print(part_1(one_to_many))
    print('\nЗадание Г2')
    print(part_2(one_to_many))
    print('\nЗадание Г3')
    print(part_3(many_to_many))

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

Файл: test.py

```
from RK_2 import *
import unittest

class Test1(unittest.TestCase):
    def setUp(self):
        self.part_1 = part_1
    def test_roots(self):
        self.assertEqual(self.part_1([('Asus', 134200, 'А-класс'), ('Acer', 109990, 'А-класс'), ('ThinkPad', 64990, 'Б
        self.assertEqual(self.part_1([('Asus', 134200, 'А-класс'), ('Dell', 204500, 'Б-класс'), ('ThinkPad', 64990, 'Б
        self.assertEqual(self.part_1([('ThinkPad', 64990, 'А-класс'), ('Asus', 134200, 'Б-класс'), ('Dell', 204500, 'Б

class Test2(unittest.TestCase):
    def setUp(self):
        self.part_2 = part_2
    def test_roots(self):
        self.assertEqual(self.part_2([('Asus', 134200, 'А-класс'), ('Acer', 109990, 'А-класс'), ('ThinkPad', 64990, 'Б
        self.assertEqual(self.part_2([('Asus', 134200, 'А-класс'), ('Dell', 204500, 'Б-класс'), ('ThinkPad', 64990, 'Б
        self.assertEqual(self.part_2([('ThinkPad', 64990, 'А-класс'), ('Asus', 134200, 'Б-класс'), ('Dell', 204500, 'Б

class Test3(unittest.TestCase):
    def setUp(self):
        self.part_3 = part_3
    def test_roots(self):
        self.assertEqual(self.part_3([('Asus', 134200, 'А-класс'), ('Acer', 109990, 'А-класс'), ('ThinkPad', 64990, 'Б
        self.assertEqual(self.part_3([('Asus', 134200, 'А-класс'), ('Acer', 109990, 'А-класс'), ('ThinkPad', 64990, 'Б
        self.assertEqual(self.part_3([('Asus', 134200, 'А-класс'), ('Acer', 109990, 'А-класс'), ('ThinkPad', 64990, 'Б

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

Результат выполнения


Командная строка
Microsoft Windows [Version 10.0.22523.1000]
(c) Корпорация Майкрософт (Microsoft Corporation). Все права защищены.
C:\Users\nagor\Documents\python\RK_2>test
"test" не является внутренней или внешней
командой, исполняемой программой или пакетным файлом.
C:\Users\nagor\Documents\python\RK_2>test.py
C:\Users\nagor\Documents\python\RK_2>python3 test.py
...
Ran 3 tests in 0.001s
OK
C:\Users\nagor\Documents\python\RK_2>