

Московский государственный технический университет им. Н.Э. Баумана

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

Отчет по РК1 по курсу **Базовые компоненты интернет технологий**

Вариант 14

Подготовил:

Студент группы РТ5-31Б

Мамаев Т.Э.

Проверил:

Доцент кафедры ИУ5

Гапанюк Ю.Е.

24 Октябрь 2021г

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).

Вариант Е.

1. «Отдел» и «Сотрудник» связаны соотношением один-ко-многим. Выведите список всех отделов, у которых в названии присутствует слово «отдел», и список работающих в них сотрудников.
2. «Отдел» и «Сотрудник» связаны соотношением один-ко-многим. Выведите список отделов со средней зарплатой сотрудников в каждом отделе, отсортированный по средней зарплате. Средняя зарплата должна быть округлена до 2 знака после запятой (*отдельной функции вычисления среднего значения в Python нет, нужно использовать*

комбинацию функций вычисления суммы и количества значений; для округления необходимо использовать функцию <https://docs.python.org/3/library/functions.html#round>).

3. «Отдел» и «Сотрудник» связаны соотношением многие-ко-многим. Выведите список всех сотрудников, у которых фамилия начинается с буквы «А», и названия их отделов.

Текст программы

RK2_Mamaev.py

```
from operator import itemgetter
```

```
"""CD-диск"""
```

```
class CD:
```

```
    def __init__(self, id, name_cd, cap, lib_id):
```

```
        self.id = id
```

```
        self.name_cd = name_cd # имя
```

```
        self.cap = cap # вместимость диска (Мегабайт)
```

```
        self.lib_id = lib_id # в какой библиотеке
```

```
"""Библиотека CD-дисков"""
```

```
class lib:
```

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

```
        self.id = id
```

```
        self.name = name
```

```
class CdLib:
```

```
    """связь многие ко многим"""
```

```
    def __init__(self, lib_id, cd_id):
```

```
        self.lib_id = lib_id
```

```
        self.cd_id = cd_id
```

```
# библиотеки
```

```
libs = [  
    lib(1, 'Первая Библиотека'),  
    lib(2, 'Вторая Библиотека'),  
    lib(3, 'Третья Библиотека'),  
    lib(4, 'Четвертая Библиотека'),  
    lib(5, 'Пятая Библиотека'),  
    lib(6, 'Шестая Библиотека'),  
]
```

```
# CD-диски
```

```
cds = [  
    CD(1, 'Amphire', 5120, 2),  
    CD(2, 'kodack', 2048, 1),  
    CD(3, 'Samsung', 1024, 3),  
    CD(4, 'Sharp', 3072, 4),  
    CD(5, 'Sony', 4096, 5),  
    CD(6, 'Piratsky', 2048, 1),  
    CD(7, 'Palm', 512, 6),  
    CD(8, 'IBM', 4096, 4),  
    CD(9, 'Toshiba', 4096, 5)  
]
```

```
cds_libs = [  
    CdLib(1,1),  
    CdLib(1,2),  
    CdLib(1,4),  
    CdLib(1,8),  
    CdLib(2,4),
```

```

CdLib(2,5),
CdLib(2,6),
CdLib(2,1),
CdLib(3,3),
CdLib(4,1),
CdLib(4,5),
CdLib(5,4),
CdLib(5,2),
CdLib(6,7),
]

def main():

    """Основная функция"""

    # Соединение данных один-ко-многим
one_to_many = [(h.name_cd, h.cap, p.name)

    for p in libs

    for h in cds

    if h.lib_id==p.id]

    # Соединение данных многие-ко-многим
many_to_many_temp = [(p.name, ph.lib_id, ph.cd_id)

    for p in libs

    for ph in cds_libs if p.id==ph.lib_id]
many_to_many = [(h.name_cd, h.cap, lib_name)

    for lib_name, lib_id, cd_id in many_to_many_temp

    for h in cds if h.id==cd_id]

def E1():

    print('Задание E1')

    res_E1 = []

```

```

for name_cd, cap, name in one_to_many:

    if 'Первая' in name: # Ищем библиотеки с ключевым словом "Вторая"

        res_E1.append((name, name_cd))

return res_E1

```

```

def E2():

    print("\nЗадание E2') # находим среднюю вместимость дисков

    res_E2_unsorted = [] # Перебираем все библиотеки

    for p in libs:

        # Список дисков библиотеки

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

        # Если в библиотеке есть диск

        if len(list_cd) > 0: # вместимость CD

            list_cap = [cap for _, cap, _ in list_cd]

            # средняя вместимость

            avg_sum = sum(list_cap)/len(list_cap)

            res_E2_unsorted.append((p.name, avg_sum))

    res_E2 = sorted(res_E2_unsorted, key=itemgetter(1))

    return res_E2

```

```

def E3():

    print("\nЗадание E3')

    # находим диски, начинающиеся с "P" и выводим их библиотеки

    res_E3 = []

    for name_cd, cap, name in many_to_many:

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

            res_E3.append((name_cd, name))

    return res_E3

```

Test.py

```
import unittest
```

```
import RK2_Mamaev
```

```
class RK_Test(unittest.TestCase):
```

```
    def test_E1(self):
```

```
        self.assertEqual(RK2_Mamaev.E1(), [('Первая Библиотека', 'kodack'), ('Первая Библиотека', 'Piratsky')])
```

```
    def test_E2(self):
```

```
        self.assertEqual(RK2_Mamaev.E2(), [('Шестая Библиотека', 512.0), ('Третья Библиотека', 1024.0), ('Первая Библиотека', 2048.0), ('Четвертая Библиотека', 3584.0), ('Пятая Библиотека', 4096.0), ('Вторая Библиотека', 5120.0)])
```

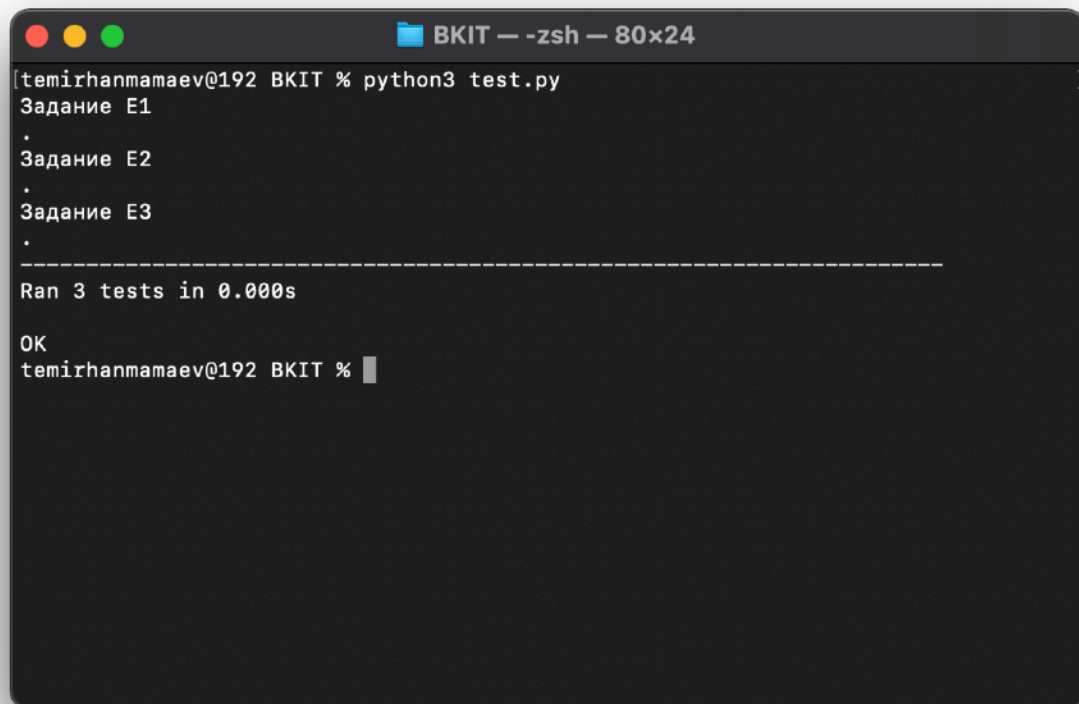
```
    def test_E3(self):
```

```
        self.assertEqual(RK2_Mamaev.E3(), [('Piratsky', 'Вторая Библиотека'), ('Palm', 'Шестая Библиотека')])
```

```
if __name__ == '__main__':
```

```
    unittest.main()
```

Результат выполнения программы

A terminal window with a dark background and light gray text. The title bar at the top shows three colored window control buttons (red, yellow, green) on the left, followed by a blue folder icon and the text "BKIT — -zsh — 80x24". The terminal content shows a user running a Python script. The output consists of three lines, each starting with "Задание E" followed by a period. After a dashed separator line, it shows the execution time. The prompt "OK" is shown on a new line, followed by the terminal prompt and a cursor.

```
[temirhanmamaev@192 BKIT % python3 test.py  
Задание E1  
.  
Задание E2  
.  
Задание E3  
.  
-----  
Ran 3 tests in 0.000s  
  
OK  
temirhanmamaev@192 BKIT %
```