Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

Факультет «Информатика и системы управления»

Кафедра «Системы обработки информации и управления»

Рубежный контроль №2
по дисциплине «Базовые компоненты интернет-технологий»

Выполнил:
студент группы ИУ5-35Б Тазенков И. Д.

г. 2021

Текст программы:

Ds.py

```
class Cd:
    """CD-диск"""

    def __init__(self, id, song, author, price, lib_id):
        self.id = id
        self.song = song
        self.author = author
        self.price = price
        self.lib_id = lib_id


class Lib:
    """Библиотека"""

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


class LibCd:
    """
    'Диск из библиотеки' для реализации
    связи многие-ко-многим
    """

    def __init__(self, cd_id, lib_id):
        self.cd_id = cd_id
        self.lib_id = lib_id

cdc = [
    Cd(1, 'Группа крови', 'Цой', 200, 1),
    Cd(2, 'Туман', 'Сектор газа', 300, 1),
    Cd(3, 'Кукла колдуна', 'Сектор газа', 442, 1),
    Cd(4, 'Земля у дома', 'Земляне', 126, 1),
    Cd(5, 'Искала', 'Земфира', 672, 2),
    Cd(6, 'Хочешь?', 'Земфира', 456, 2),
    Cd(7, 'Ромашки', 'Земфира', 300, 2),
    Cd(9, 'Гранитный камушек', 'Божья коровка', 228, 2),
    Cd(11, 'Shake it off', 'Taylor Swift', 546, 3),
    Cd(14, 'Complicated', 'Avril Lavigne', 219, 3),
    Cd(10, 'Poker Face', 'Lady Gaga', 341, 3),
    Cd(8, 'Мы Ранетки', 'Ранетки', 145, 4),
    Cd(12, 'Зима', 'Ранетки', 765, 4),
    Cd(13, 'Ангелы', 'Ранетки', 99, 4)

]

# Сотрудники
libs = [
    Lib(1, 'Диски 80-ых'),
    Lib(2, 'Все о Земфире'),
    Lib(3, 'Иностранная попса'),
    Lib(4, 'Диски для девочек')

]

libs_cdc = [
    LibCd(1, 1),
    LibCd(2, 1),
    LibCd(3, 1),
    LibCd(4, 1),
    LibCd(5, 2),
    LibCd(6, 2),
```

```
    LibCd(7, 2),
    LibCd(8, 4),
    LibCd(9, 2),
    LibCd(10, 3),
    LibCd(11, 3),
    LibCd(12, 4),
    LibCd(13, 4),
    LibCd(14, 3),

]
```

RK1.py

```
# -*- coding: utf-8 -*-
# используется для сортировки
from operator import itemgetter

from ds import cdc, libs, libs_cdc


class RK1:
    def __init__(self, cdc, libs, libs_cdc):
        self.cdc = cdc
        self.libs = libs
        self.libs_cdc = libs_cdc
        self.one_to_many = [(c.song, c.author, c.price, l.name)
                            for l in libs
                            for c in cdc
                            if c.lib_id == l.id]

        # Соединение данных многие-ко-многим
        self.many_to_many_temp = [(l.name, lc.lib_id, lc.cd_id)
                                  for l in libs
                                  for lc in libs_cdc
                                  if l.id == lc.lib_id]

        self.many_to_many = [(c.song, c.author, c.price, lib_name)
                             for lib_name, lib_id, cd_id in self.many_to_many_temp
                             for c in cdc if c.id == cd_id]

    def N1(self):
        print('Задание Д1')
        res_11 = [(cd[1], cd[3]) for cd in self.one_to_many if cd[1][-1] == 'a' or
cd[1][-1] == 'a']
        for i in res_11:
            print(*i, sep=' --- ')
        return res_11

    def N2(self):
        print('\nЗадание Д2')
        res_12_unsorted = []
        # Перебираем все библиотеки
        for l in self.libs:
            # Список дисков библиотеки
            l_cds = list(filter(lambda i: i[3] == l.name, self.one_to_many))
            # Если библиотека не пустая
            if len(l_cds) > 0:
                # Цены дисков библиотеки
                l_prices = [price for _, _, price, _ in l_cds]
                # Средняя цена дисков библиотеки
                l_prices_av = sum(l_prices) / len(l_prices)
                res_12_unsorted.append((l.name, l_prices_av))

        # Сортировка по средней цене
        res_12 = sorted(res_12_unsorted, key=itemgetter(1), reverse=True)
        for i in res_12:
            print(*i, sep=' --- ')
```

```
        return res_12


    def N3(self):
        print('\nЗадание Д3')
        res_13 = {}
        # Перебираем все библиотеки
        for l in self.libs:
            if l.name[0] == 'Д':
                # Список дисков библиотеки
                l_cds = list(filter(lambda i: i[3] == l.name, self.many_to_many))
                # Только название песни
                l_cdc_names = [x for x, _, _, _ in l_cds]
                # Добавляем результат в словарь
                # ключ - библиотека, значение - список дисков
                res_13[l.name] = l_cdc_names

        for key, value in res_13.items():
            print(key, end=':\n')
            for v in value:
                print('\t', v)
        return res_13


if __name__ == '__main__':
    rk = RK1(cdc, libs, libs_cdc)
    rk.N1()
    rk.N2()
    rk.N3()
```

testRK1.py

```
import unittest

from RK1 import RK1
from ds import cdc, libs, libs_cdc



res_1 = [('Сектор газа', 'Диски 80-ых'),
         ('Сектор газа', 'Диски 80-ых'),
         ('Земфира', 'Все о Земфире'),
         ('Земфира', 'Все о Земфире'),
         ('Земфира', 'Все о Земфире'),
         ('Божья коровка', 'Все о Земфире'),
         ('Lady Gaga', 'Иностранная попса')
         ]
res_2 = [
    ('Все о Земфире', 414.0),
    ('Иностранная попса', 368.6666666666667),
    ('Диски для девочек', 336.3333333333333),
    ('Диски 80-ых', 267.0)
]

res_3 = {
    'Диски 80-ых':
        [
            'Группа крови',
            'Туман',
            'Кукла колдуна',
            'Земля у дома'
        ],
    'Диски для девочек':
        [
            'Мы Ранетки',
            'Зима',
            'Ангелы'
```

```
        ]
}

class MyTestCase(unittest.TestCase):
    def test_n1(self):
        rk = RK1(cdc, libs, libs_cdc)
        self.assertEqual(res_1, rk.N1())

    def test_n2(self):
        rk = RK1(cdc, libs, libs_cdc)
        self.assertEqual(res_2, rk.N2())


    def test_n3(self):
        rk = RK1(cdc, libs, libs_cdc)
        self.assertEqual(res_3, rk.N3())



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

Результат выполнения

```
Testing started at 15:34 ...
Launching pytest with arguments C:/Users/vtaze/iCloudDrive/Documents/GitHub/Lab_BKIT/RK2/testRK1.

============================= test session starts ==============================
collecting ... collected 3 items

testRK1.py::MyTestCase::test_n1 PASSED                                   [ 33%]Задание Д1
Сектор газа --- Диски 80-ых
Сектор газа --- Диски 80-ых
Земфира --- Все о Земфире
Земфира --- Все о Земфире
Земфира --- Все о Земфире
Божья коровка --- Все о Земфире
Lady Gaga --- Иностранная попса

testRK1.py::MyTestCase::test_n2 PASSED                                   [ 66%]
Задание Д2
Все о Земфире --- 414.0
Иностранная попса --- 368.6666666666667
Диски для девочек --- 336.3333333333333
Диски 80-ых --- 267.0

testRK1.py::MyTestCase::test_n3 PASSED                                   [100%]
Задание Д3
Диски 80-ых:
     Группа крови
     Туман
     Кукла колдуна
     Земля у дома
Диски для девочек:
     Мы Ранетки
     Зима
     Ангелы


============================== 3 passed in 0.09s ===============================

Process finished with exit code 0
```