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

Выполнил:		Проверил:
студент группы ИУ5-32Б		преподаватель каф. ИУ5
Панов Г. Д.		Гапанюк Ю. Е.
Подпись и дата		Подпись и дата

Москва, 2021 г.

Задание:

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).

Листинг программы:

1.

```
# используется для сортировки
from operator import itemgetter
# используется для разбиения названия диска по словам
import re

class Emp:
    """CD-диск"""

    def __init__(self, id, fio, sal, dep_id):
        self.id = id                            # id записи о диске
        self.fio = fio                          # название диска
        self.sal = sal                          # цена диска
        self.dep_id = dep_id                    # id записи о библиотеке


class Dep:
    """Библиотека CD-дисков"""

    def __init__(self, id, name):
        self.id = id                            # id записи о библиотеке
        self.name = name                        # наименование библиотеки


class EmpDep:
    """
    'Диски в библиотеке' для реализации
    связи многие-ко-многим
    """

    def __init__(self, dep_id, emp_id):
        self.dep_id = dep_id                    # id записи о библиотеке
        self.emp_id = emp_id                    # id записи о диске



# Библиотеки CD-дисков
deps = [
    Dep(1, 'библиотека фильмов'),
    Dep(2, 'архивная библиотека документов'),
    Dep(3, 'библиотека музыки'),
```

```
    Dep(11, 'библиотека (другая) фильмов'),
    Dep(22, 'архивная (другая) библиотека документов'),
    Dep(33, 'библиотека (другая) музыки'),
]

# CD-диски
emps = [
    Emp(1, 'Земля вампиров', 2000, 1),
    Emp(2, 'Виндоус 7', 6000, 2),
    Emp(3, 'Пакет Office', 7000, 2),
    Emp(4, 'Альбом AC/DC', 2500, 3),
    Emp(5, 'Самоучитель по ловле кальмаров', 300, 2),
]

# CD-диски в библиотеках (связь многие-ко-многим)
emps_deps = [
    EmpDep(1, 1),
    EmpDep(2, 2),
    EmpDep(2, 3),
    EmpDep(3, 4),
    EmpDep(2, 5),

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

# Соединение данных один-ко-многим
one_to_many = [(e.fio, e.sal, d.name)
               for d in deps
               for e in emps
               if e.dep_id == d.id]

# Соединение данных многие-ко-многим
many_to_many_temp = [(d.name, ed.dep_id, ed.emp_id)
                     for d in deps
                     for ed in emps_deps
                     if d.id == ed.dep_id]

many_to_many = [(e.fio, e.sal, dep_name)
               for dep_name, dep_id, emp_id in many_to_many_temp
               for e in emps if e.id == emp_id]

# Задание Б1
def first_task():
    res_11 = sorted(one_to_many, key=itemgetter(0))
    # print(res_11)
    return res_11

# Задание Б2
def second_task():
    res_12_unsorted = []
    # Перебираем все библиотеки
    for d in deps:
        # Список библиотек
        d_emps = list(filter(lambda i: i[2] == d.name, one_to_many))
        count = sum(isinstance(x, tuple) for x in d_emps)
        # Если библиотека не пустая
        if len(d_emps) > 0:
```

```
            res_12_unsorted.append((d.name, count))

    # Сортировка по количеству дисков
    res_12 = sorted(res_12_unsorted, key=itemgetter(1), reverse=True)
    # print(res_12)
    return res_12

# Задание Б3
def third_task():
    res_13 = {}
    # Перебираем все диски
    for e in emps:
        # проверяем оканчивается ли CD-диск на 'ов'
        if 'ов' in str(re.split('; |, |: |-', e.fio)):
            # Список библиотек данного диска
            d_emps = list(filter(lambda i: i[0] == e.fio, many_to_many))
            # Только названия библиотек
            demps_names = [x[2] for x in d_emps]
            # Добавляем результат в словарь
            # ключ - CD-диск, значение - список библиотек, в которых есть
этот CD-диск
            res_13[e.fio] = demps_names
    # print(res_13)
    return res_13


def main():
    """Основная функция"""

    print('Задание Б1')
    print(first_task())

    print('\nЗадание Б2')
    print(second_task())

    print('\nЗадание Б3')
    print(third_task())
    # print(list(third_task())[0])


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

2.

```
import unittest
from main import first_task
from main import second_task
from main import third_task


class TestFirstTask(unittest.TestCase):
    def test_value_count(self):
        expected_result = 5
        actual_result = len(first_task())
        self.assertEqual(expected_result, actual_result, "The expected
number of employees differs from the actual one")
```

```
class TestSecondTask(unittest.TestCase):
    def test_sort_by_max(self):
        expected_result = '311'
        actual_result = str(second_task()[0][1]) + str(second_task()[1][1])
+ str(second_task()[1][1])
        self.assertEqual(expected_result, actual_result, "Something went
wrong with the sorting")


class TestThirdTask(unittest.TestCase):
    def test_name_end(self):
        count = 0
        for el in list(third_task()):
            if str(el).endswith('ов'):
                count += 1
        actual_result = count
        expected_result = 2
        self.assertEqual(expected_result, actual_result, "Check the endings
of words")


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

Пример работы: