

**Московский государственный технический
университет им. Н.Э. Баумана.**

Факультет «Информатика и управление»

Кафедра ИУ5 «Системы обработки информации и управления»

Рубежный контроль №2

по дисциплине «Базовые компоненты интернет-технологий»

Выполнил:

студент группы ИУ5-33

Ефременко Д.С.

Проверил:

преподаватель каф.
ИУ5

Гапанюк Ю.Е.

Задание:

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).

Текст программы:

main.py

```
# используется для сортировки
from operator import itemgetter

class Driver:
    """Водитель"""

    def __init__(self, id, fio, sal, exp_id):
        self.id = id
        self.fio = fio
        self.sal = sal
        self.exp_id = exp_id #водительский стаж

class Auto:
    """Автопарк"""

    def __init__(self, id, mode):
        self.id = id
        self.mode = mode #виды автобусов

class AutoDriver:
    """
    'Водитель автопарка' для реализации
    связи многие-ко-многим
    """

    def __init__(self, driv_id, auto_id):
        self.driv_id = driv_id
        self.auto_id = auto_id

# Автопарк
auto = [
    Auto(1, 'туристический'),
    Auto(2, 'городской'),
    Auto(3, 'школьный'),
    Auto(4, 'экскурсионный'),
    Auto(5, 'перонный'),
    Auto(6, 'пригородный'),
]

# Водители
driv = [
    Driver(1, 'Ефременко', 50000, 5),
    Driver(2, 'Семенов', 30000, 3),
    Driver(3, 'Стебунов', 45000, 4),
    Driver(4, 'Носкин', 44000, 6),
    Driver(5, 'Бегларов', 45000, 2),
    Driver(6, 'Алешин', 23000, 4),
]
```

```

Driver(7, 'Ахтамбаев', 40000, 2),
Driver(8, 'Андреев', 45000, 3)
]

auto_drivers = [
    AutoDriver(1, 1),
    AutoDriver(3, 2),
    AutoDriver(3, 4),
    AutoDriver(5, 1),
    AutoDriver(6, 3),
    AutoDriver(2, 1),
    AutoDriver(4, 2),
    AutoDriver(3, 3),
    AutoDriver(4, 5),
    AutoDriver(6, 2),
]

def Task1(one_to_many):
    print('Задание В1')
    res_11 = []
    for fio, sal, auto_name in one_to_many:
        if 'А' in fio[0]:
            res_11.append((fio, auto_name))
    return res_11

def Task2(one_to_many):
    print('Задание В2')
    buff = []
    for a in auto:
        # список видов транспорта
        a_modes = list(filter(lambda i: i[2] == a.mode, one_to_many))
        if len(a_modes) > 0:
            a_sal = [sal for _, sal, _ in a_modes]
            min_sal = min(a_sal)
            buff.append((a.mode, min_sal))
    res_12 = sorted(buff, key=itemgetter(1))
    return res_12

def Task3(many_to_many):
    print('Задание В3')
    buff = []
    for fio, sal, auto_name in many_to_many:
        buff.append((fio, auto_name))
    res_13 = list(sorted(buff, key=itemgetter(0)))
    return res_13

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

    # Соединение данных один-ко-многим
    one_to_many = [(d.fio, d.sal, a.mode)
                   for a in auto
                   for d in driv
                   if d.exp_id == a.id]

    # Соединение данных многие-ко-многим
    many_to_many_temp = [(a.mode, ad.auto_id, ad.driv_id)
                         for a in auto
                         for ad in auto_drivers
                         if a.id == ad.auto_id]

    many_to_many = [(d.fio, d.sal, auto_name)
                    for auto_name, auto_id, driv_id in many_to_many_temp
                    for d in driv if d.id == driv_id]

```

test.py

[illegible]

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

Результат выполнения программы:

```
Задание B1  
Задание B2  
Задание B3  
...  
-----  
Ran 3 tests in 0.000s  
  
OK  
  
Process finished with exit code 0
```