

# Князев Алексей ИУ5-31Б Вариант №10

Файл *RK2\_refactoring.py*

```
from operator import itemgetter

class Brw:
    """Браузер"""
    def __init__(self, id, memory, name, dvp, cmp_id):
        self.id = id
        self.memory = memory #кол-во занимаемой памяти браузером(мб)
        self.name = name
        self.dvp = dvp #разработчик
        self.cmp_id = cmp_id

class Cmp:
    """Компьютер"""
    def __init__(self, id, name, type, os):
        self.id = id
        self.name = name
        self.type = type
        self.os = os

class CmpBrw:
    """
    'Браузеры компьютера' для реализации
    связи многие-ко-многим
    """
    def __init__(self, brw_id, cmp_id):
        self.cmp_id = cmp_id
        self.brw_id = brw_id

def get_one_to_many(brws, cmps):
    """Соединение данных один-ко-многим"""
    return [(b.name, b.memory, c.name)
            for b in brws
            for c in cmps
            if b.cmp_id == c.id]

def get_many_to_many(cmps, cmps_brws, brws):
    """Соединение данных многие-ко-многим"""
    many_to_many_temp = [(c.name, cb.cmp_id, cb.brw_id)
                        for c in cmps
                        for cb in cmps_brws
                        if c.id == cb.cmp_id]
    return [(b.name, b.memory, cmp_name)
            for cmp_name, cmp_id, brw_id in many_to_many_temp
            for b in brws if b.id == brw_id]

def calculate_total_memory(one_to_many_data, cmps):
    """Вычисляет суммарную память, используемую браузерами на каждом
    компьютере"""
    res_12_unsorted = []
    for c in cmps:
        c_brws = list(filter(lambda i: i[2] == c.name, one_to_many_data))
        if c_brws:
            c_memories = [memory for _, memory, _ in c_brws]
            c_memories_sum = sum(c_memories)
            res_12_unsorted.append((c.name, c_memories_sum))
    return sorted(res_12_unsorted, key=itemgetter(1), reverse=True)

def get_browsers_per_computer(many_to_many_data, cmps):
```

```

    """Возвращает словарь {компьютер: [список браузеров]} """
    res_13 = {}
    for c in cmps:
        c_brws = list(filter(lambda i: i[2] == c.name, many_to_many_data))
        c_brws_names = [x for x, _, _ in c_brws]
        res_13[c.name] = c_brws_names
    return res_13

# Данные (остаются без изменений)
cmps = [
    Cmp(1, 'Alexei', 'ноутбук', 'Windows 11'),
    Cmp(2, 'PC Artem', 'ультрабук', 'Windows 10'),
    Cmp(3, 'Dmitriy', 'настольный', 'Linux'),
    Cmp(4, 'Anton PC', 'ноутбук', 'macOS Catalina'),
    Cmp(5, 'Anonim', 'сервер', 'Windows 11')
]

brws = [
    Brw(1, 200, 'Yandex', 'Яндекс', 1),
    Brw(2, 350, 'Chrome', 'Google', 2),
    Brw(3, 300, 'Firefox', 'Mozilla Foundation', 4),
    Brw(4, 225, 'Opera', 'Opera Software', 5),
    Brw(5, 250, 'Safari', 'Apple', 3)
]

cmps_brws = [
    CmpBrw(1, 1),
    CmpBrw(1, 2),
    CmpBrw(3, 1),
    CmpBrw(3, 3),
    CmpBrw(3, 5),
    CmpBrw(2, 1),
    CmpBrw(2, 4),
    CmpBrw(5, 4),
    CmpBrw(4, 3),
    CmpBrw(5, 5),
]

def main():
    """Основная функция"""
    one_to_many_data = get_one_to_many(brws, cmps)
    many_to_many_data = get_many_to_many(cmps, cmps_brws, brws)

    print('Задание A1')
    res_11 = sorted(one_to_many_data, key=itemgetter(2))
    print(res_11)

    print('\nЗадание A2')
    res_12 = calculate_total_memory(one_to_many_data, cmps)
    print(res_12)

    print('\nЗадание A3')
    res_13 = get_browsers_per_computer(many_to_many_data, cmps)
    print(res_13)

if __name__ == '__main__':
    main()

```

Результаты:

Задание A1

[('Yandex', 200, 'Alexei'), ('Opera', 225, 'Anonim'), ('Firefox', 300, 'Anton PC'), ('Safari', 250, 'Dmitriy'), ('Chrome', 350, 'PC Artem')]

Задание A2

[('PC Artem', 350), ('Anton PC', 300), ('Dmitriy', 250), ('Anonim', 225), ('Alexei', 200)]

Задание A3

{'Alexei': ['Yandex', 'Firefox', 'Chrome'], 'PC Artem': ['Yandex'], 'Dmitriy': ['Firefox', 'Opera'], 'Anton PC': ['Chrome', 'Safari'], 'Anonim': ['Firefox', 'Safari']}

Файл *RK2\_test.py*

```
import unittest
from operator import itemgetter
from RK2_refactoring import ( # Замените your_module на имя вашего файла
    Brw, Cmp, CmpBrw, get_one_to_many,
    calculate_total_memory, get_browsers_per_computer
)

class TestDataProcessing(unittest.TestCase):

    def setUp(self):
        self.cmps = [
            Cmp(1, 'Alexei', 'ноутбук', 'Windows 11'),
            Cmp(2, 'PC Artem', 'ультрабук', 'Windows 10'),
            Cmp(3, 'Dmitriy', 'настольный', 'Linux')
        ]
        self.brws = [
            Brw(1, 200, 'Yandex', 'Яндекс', 1),
            Brw(2, 350, 'Chrome', 'Google', 2)
        ]
        self.cmps_brws = [CmpBrw(1, 1), CmpBrw(2, 2)]

    def test_get_one_to_many(self):
        expected = [('Yandex', 200, 'Alexei'), ('Chrome', 350, 'PC Artem')]
        result = get_one_to_many(self.brws, self.cmps)
        self.assertEqual(result, expected)

    def test_calculate_total_memory(self):
        one_to_many_data = [('Yandex', 200, 'Alexei'), ('Chrome', 350,
'Alexei')]
        expected = [('Alexei', 550)]
        result = calculate_total_memory(one_to_many_data, self.cmps)
        self.assertEqual(result, expected)
```

```
def test_get_browsers_per_computer(self):
    many_to_many_data = [('Yandex', 200, 'Alexei'), ('Chrome', 350,
'Alexei')]
    expected = {'Alexei': ['Yandex', 'Chrome'], 'Dmitriy': [], 'PC
Artem': []}
    result = get_browsers_per_computer(many_to_many_data, self.cmps)
    self.assertEqual(result, expected)

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

Результаты:

```
C:\Users\knyze\AppData\Local\Programs\
Testing started at 19:20 ...

Ran 3 tests in 0.001s

OK

Process finished with exit code 0
```