

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

**Факультет «Информатика и системы управления»
Кафедра ИУ5 «Системы обработки информации и управления»**

Курс «Парадигмы и конструкции языков программирования»

**Отчет по РК2
Вариант № 16-Г**

Выполнил:
студент группы ИУ5-34Б:
Суслов Дмитрий Сергеевич
Подпись и дата:

Проверил:
преподаватель каф. ИУ5
Гапанюк Ю. Е.
Подпись и дата:

Москва, 2023 г.

RK1.py

```
class Book:
    def __init__(self, id, title, price, shop_id):
        self.id = id
        self.title = title
        self.price = price
        self.shop_id = shop_id

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

class BookShop:
    def __init__(self, book_id, shop_id):
        self.book_id = book_id
        self.shop_id = shop_id

shops = [
    Shop(1, "Books and Humans"),
    Shop(2, "Book World"),
    Shop(3, "Read and enjoy"),
    Shop(4, "The reading owl")
]

books = [
    Book(1, "Пикник на обочине", 950, 1),
    Book(2, "Три товарища", 1100, 1),
    Book(3, "Зеленая миля", 1200, 2),
    Book(4, "Десять негрят", 850, 3),
    Book(5, "Турецкий гамбит", 1050, 4)
]

books_shops = [
    BookShop(1, 1),
    BookShop(1, 2),
    BookShop(2, 3),
    BookShop(2, 2),
    BookShop(2, 1),
    BookShop(3, 4),
    BookShop(3, 1),
    BookShop(4, 2),
    BookShop(4, 4),
    BookShop(5, 3),
]

class RK1:
    def __init__(self):
```

```

        self.one_to_many = [(b.title, b.price, s.name)
                             for b in books
                             for s in shops
                             if b.shop_id == s.id]

        # Соединение данных многие-ко-многим
        self.many_to_many_temp = [(s.name, sh.shop_id,
sh.book_id)

                                for s in shops
                                for sh in books_shops
                                if s.id == sh.shop_id]

        self.many_to_many = [(b.title, b.price, shop_name)
                              for shop_name, shop_id, book_id in
self.many_to_many_temp
                              for b in books if b.id == book_id]

    def first(self):
        correct_shop_names = list(filter(lambda x: 'book' in
x.lower(), [sh.name for sh in shops]))
        res1 = {name: [(a[0], a[1]) for a in self.one_to_many if
a[2] == name] for name in correct_shop_names}
        return res1

    def second(self):
        shop_names = [sh.name for sh in shops]
        res2 = sorted([(name, max([a[1] for a in
self.one_to_many if a[2] == name])) for name in shop_names],
                      key=lambda x: -x[1])
        return res2

    def third(self):
        s = ''
        shop_names = [sh.name for sh in shops]
        for name in sorted(shop_names):
            s += '\n' + name + '\n'
            s += '\n'.join([(f'\t{a[0]} {a[1]}') for a in
self.many_to_many if a[2] == name])
        return s

def main():
    """Основная функция"""
    rk = RK1()

    print('Задание Г1:\n"Магазин" и "Книга" связаны отношением
    один-ко-многим. Выведите список всех магазинов, '
          'в названии которых присутствует слово "book", и
    список книг в этих магазинах\n')

    print(rk.first())

    print('\nЗадание Г2:\n"Магазин" и "Книга" связаны отношением

```

```

один-ко-многим. Выведите список магазинов с '
    'максимальной ценой одной книги в каждом магазине,
отсортированный по максимальной цене')

print(rk.second())

print('\nЗадание ГЗ:\n"Магазин" и "Книга" связаны отношением
многие-ко-многим. Выведите список всех '
    'связанных книг и магазинов, отсортированный по
магазину, сортировка по книгам произвольная')

print(rk.third())

if __name__ == '__main__':
    main()

```

tests.py

```

import unittest
from RK1 import RK1

class TestCalculator(unittest.TestCase):
    def setUp(self):
        self.tr1 = {'Books and Humans': [('Пикник на обочине',
950), ('Три товарища', 1100)], 'Book World': [('Зеленая миля',
1200)]}

        self.tr2 = [('Book World', 1200), ('Books and Humans',
1100), ('The reading owl', 1050), ('Read and enjoy', 850)]

        self.tr3 = ('\n'
            'Book World\n'\tПикник на обочине 950\n'\tТри
товарища 1100\n'\tДесять негрят 850\n'
            'Books and Humans\n'
            '\tПикник на обочине 950\n'
            '\tТри товарища 1100\n'
            '\tЗеленая миля 1200\n'
            'Read and enjoy\n'
            '\tТри товарища 1100\n'
            '\tТурецкий гамбит 1050\n'
            'The reading owl\n'
            '\tЗеленая миля 1200\n'
            '\tДесять негрят 850')

        self.rk = RK1()

    def test_first(self):
        self.assertEqual(self.rk.first(), self.tr1)

    def test_second(self):
        self.assertEqual(self.rk.second(), self.tr2)

    def test_third(self):

```

```
        self.assertEqual(self.rk.third(), self.tr3)

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

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

tests.py::TestCalculator::test_first PASSED [ 33%]
tests.py::TestCalculator::test_second PASSED [ 66%]
tests.py::TestCalculator::test_third PASSED [100%]

===== 3 passed in 0.02s =====

Process finished with exit code 0
```