*Федеральное государственное бюджетное образовательное учреждение* *высшего профессионального образования*

|  |  |
| --- | --- |
|  | ***«Московский государственный технический университет  имени Н.Э. Баумана»***  ***(МГТУ им. Н.Э. Баумана)*** |

ФАКУЛЬТЕТ Информатика и системы управления

КАФЕДРА Системы обработки информации и управления (ИУ5)

**Отчет**

**по рубежному контролю №1**

**Дисциплина: Разработка Интернет-Приложений**

Студент гр. ИУ5-53Б  **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** Назаров М.М.

(Подпись, дата) (И.О. Фамилия)

Преподаватель  **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** Гапанюк Ю.Е.

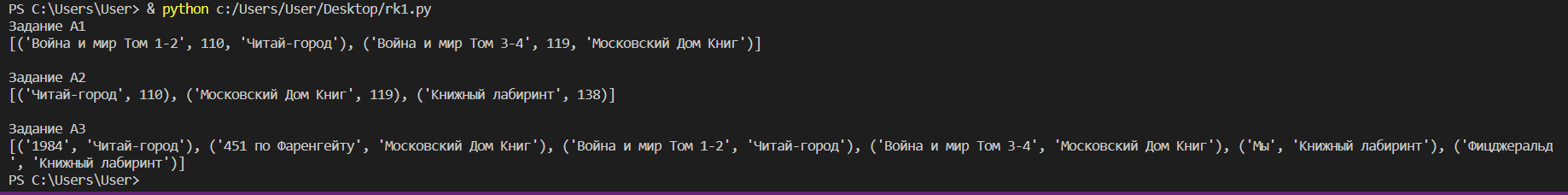
(Подпись, дата) (И.О. Фамилия)

Москва, 2020

1. **Задание**

**Вариант В.**

1. «Книжный магазин» и «Книга» связаны соотношением один-ко-многим. Выведите список всех книг, у которых название начинается с буквы «В», и названия книжных магазинов, в которых они находятся.
2. «Книжный магазин» и «Книга» связаны соотношением один-ко-многим. Выведите список книжных магазинов с минимальной стоимостью книги в каждом книжном магазине, отсортированный по минимальной цене книги.
3. «Книжный магазин» и «Книга» связаны соотношением многие-ко-многим. Выведите список всех связанных книг и книжных магазинов, отсортированный по книгам, сортировка по книжным магазинам произвольная.
4. **Текст программы**
5. from operator import itemgetter
6. class Book:
7. """Книга"""
8. def \_\_init\_\_(self, id, title, autor, price, store\_id):
9. self.id = id
10. self.title = title
11. self.autor = autor
12. self.price = price
13. self.store\_id = store\_id
14. class Store:
15. """Книжный магазин"""
16. def \_\_init\_\_(self, id, name):
17. self.id = id
18. self.name = name
19. class BookStore:
20. """'Книги книжного магазина' для реализации связи многие-ко-многим"""
21. def \_\_init\_\_(self, store\_id, book\_id):
22. self.store\_id = store\_id
23. self.book\_id = book\_id
24. # Книжные магазины
25. stores = [
26. Store(1, 'Читай-город'),
27. Store(2, 'Книжный лабиринт'),
28. Store(3, 'Московский Дом Книг'),
29. ]
30. # Книги
31. books = [
32. Book(1, 'Война и мир Том 1-2', 'Толстой', 110, 1),
33. Book(2, 'Фицджеральд', 'Скотт', 138, 2),
34. Book(3, '451 по Фаренгейту', 'Брэдбери', 217, 3),
35. Book(4, '1984', 'Оруэлл', 196, 1),
36. Book(5, 'Мы', 'Замятин', 138, 2),
37. Book(6, 'Война и мир Том 3-4', 'Толстой', 119, 3),
38. ]
39. store\_book = [
40. BookStore(1, 1),
41. BookStore(2, 2),
42. BookStore(3, 3),
43. BookStore(1, 4),
44. BookStore(2, 5),
45. BookStore(3, 6)
46. ]
47. def main():
48. """Основная функция"""
49. one\_to\_many = [(b.title, b.price, s.name)
50. for s in stores
51. for b in books
52. if b.store\_id == s.id]
53. many\_to\_many\_temp = [(s.name, bs.store\_id, bs.book\_id)
54. for s in stores
55. for bs in store\_book
56. if s.id == bs.store\_id]
57. many\_to\_many = [(b.title, store\_name)
58. for store\_name, store\_id, book\_id in many\_to\_many\_temp
59. for b in books if b.id == book\_id]
60. print('Задание А1')
61. res\_11 = list(filter(lambda x: x[0].startswith('В'), one\_to\_many))
62. print(res\_11)
63. print('\nЗадание А2')
64. res\_12\_unsorted = []
65. for s in stores:
66. s\_books = list(filter(lambda i: i[2] == s.name, one\_to\_many))
67. if len(s\_books) > 0:
68. s\_price = [price for \_, price, \_ in s\_books]
69. s\_price\_min = min(s\_price)
70. res\_12\_unsorted.append((s.name, s\_price\_min))
71. res\_12 = sorted(res\_12\_unsorted, key=itemgetter(1), reverse=False)
72. print(res\_12)
73. print('\nЗадание А3')
74. res\_13 = sorted(many\_to\_many, key=itemgetter(0))
75. print(res\_13)
76. if \_\_name\_\_ == '\_\_main\_\_':
77. main()
78. **Результаты работы программы**

****