**МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ им. Н.Э.Баумана**

**Отчет по рубежному контролю №1**

**по курсу «Разработка интернет-приложений»**

**Функциональные возможности языка Python**

**Выполнил:**

Компаниец А.А.

ИУ5-51Б

**Москва, 2021**

Вариант запросов: Г





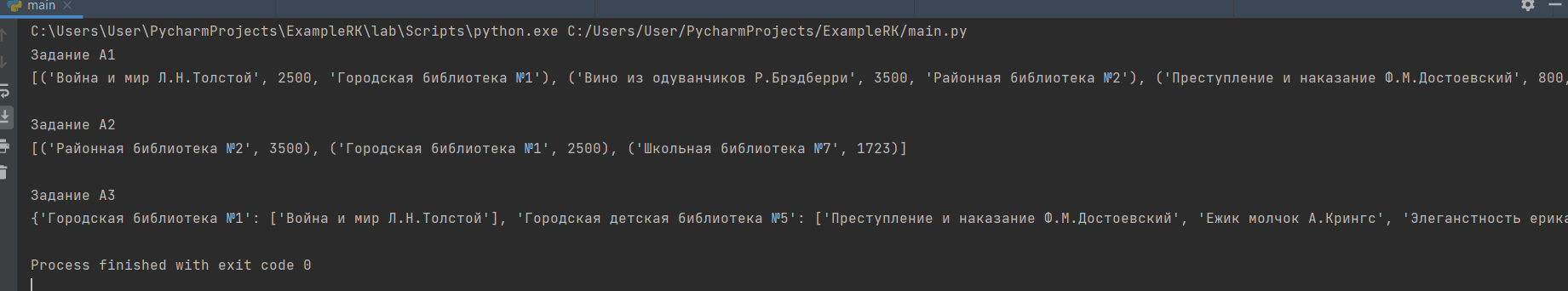
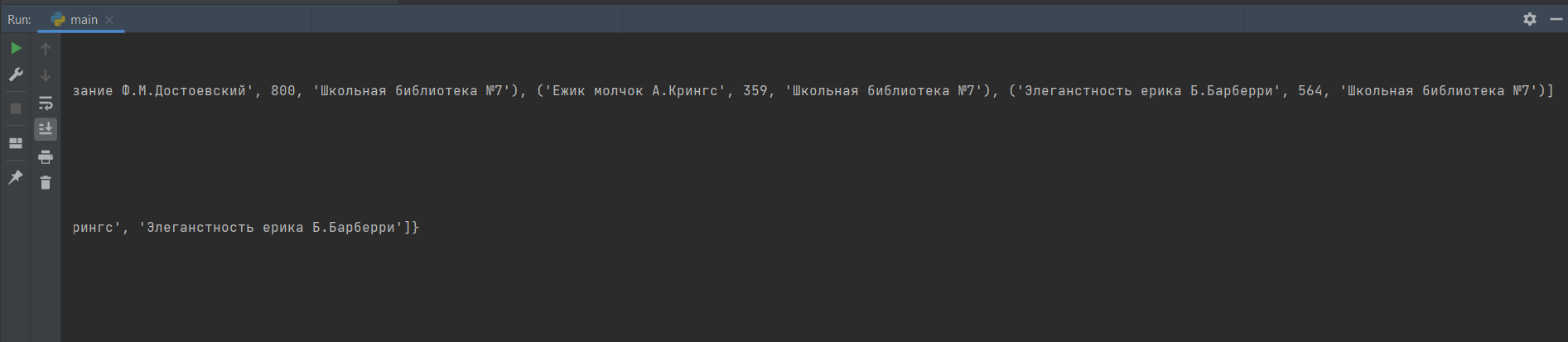
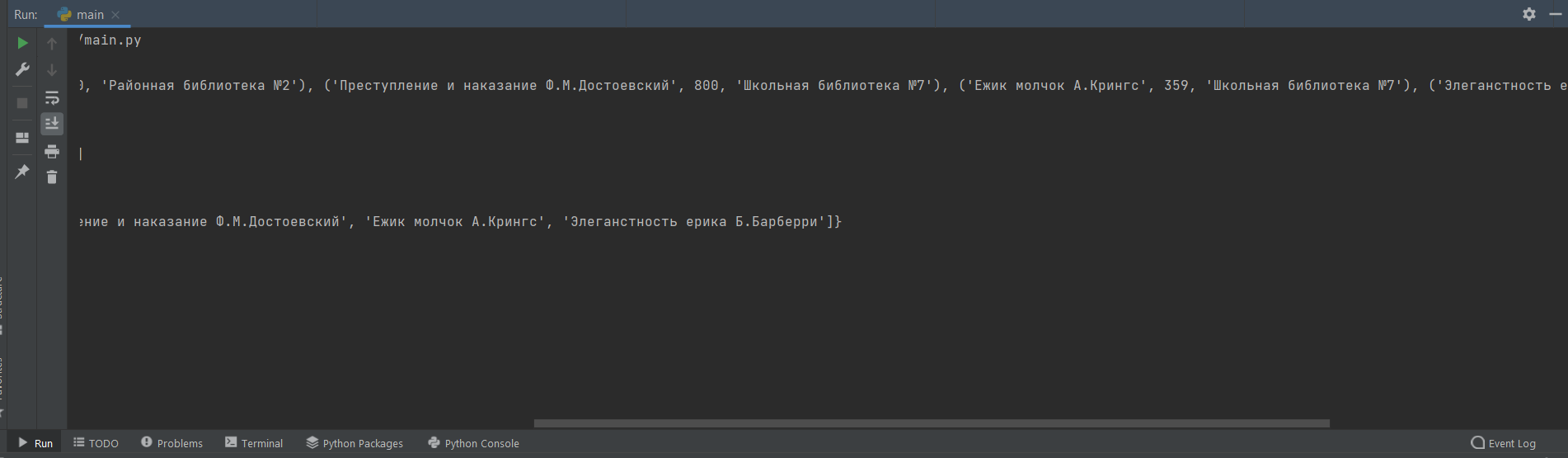
**Вариант А.**

1. «Отдел» и «Сотрудник» связаны соотношением один-ко-многим. Выведите список всех связанных сотрудников и отделов, отсортированный по отделам, сортировка по сотрудникам произвольная.
2. «Отдел» и «Сотрудник» связаны соотношением один-ко-многим. Выведите список отделов с суммарной зарплатой сотрудников в каждом отделе, отсортированный по суммарной зарплате.
3. «Отдел» и «Сотрудник» связаны соотношением многие-ко-многим. Выведите список всех отделов, у которых в названии присутствует слово «отдел», и список работающих в них сотрудников.

Код программы:

# используется для сортировки  
from operator import itemgetter  
  
  
class Book:  
 *"""кНИГА"""* def \_\_init\_\_(self, id, name\_fio, cost, lib\_id):  
 self.id = id  
 self.name\_fio = name\_fio  
 self.cost = cost  
 self.lib\_id = lib\_id  
  
  
class Lib:  
 *"""БИБЛИОТЕКА"""* def \_\_init\_\_(self, id, name):  
 self.id = id  
 self.name = name  
  
  
class BookLib:  
 *"""  
 'Книга библиотеки' для реализации  
 связи многие-ко-многим  
 """* def \_\_init\_\_(self, lib\_id, book\_id):  
 self.lib\_id = lib\_id  
 self.book\_id = book\_id  
  
  
# Библиотеки  
libs\_list = [  
 Lib(1, 'Городская библиотека №1'),  
 Lib(2, 'Районная библиотека №2'),  
 Lib(3, 'Школьная библиотека №7'),  
  
 Lib(11, 'Государственная библитека №10'),  
 Lib(22, 'Публичная библиотека № 9'),  
 Lib(33, 'Городская детская библиотека №5'),  
]  
  
# Книги  
books\_list = [  
 Book(1, 'Война и мир Л.Н.Толстой', 2500, 1),  
 Book(2, 'Вино из одуванчиков Р.Брэдберри', 3500, 2),  
 Book(3, 'Преступление и наказание Ф.М.Достоевский', 800, 3),  
 Book(4, 'Ежик молчок А.Крингс', 359, 3),  
 Book(5, 'Элеганстность ерика Б.Барберри', 564, 3),  
]  
  
bools\_libs = [  
 BookLib(1, 1),  
 BookLib(2, 2),  
 BookLib(3, 3),  
 BookLib(3, 4),  
 BookLib(3, 5),  
  
 BookLib(11, 1),  
 BookLib(22, 2),  
 BookLib(33, 3),  
 BookLib(33, 4),  
 BookLib(33, 5),  
]  
  
  
def main():  
 *"""Основная функция"""* # Соединение данных один-ко-многим  
 one\_to\_many = [(e.name\_fio, e.cost, d.name)  
 for d in libs\_list  
 for e in books\_list  
 if e.lib\_id == d.id]  
  
 # Соединение данных многие-ко-многим  
 many\_to\_many\_temp = [(d.name, ed.lib\_id, ed.book\_id)  
 for d in libs\_list  
 for ed in bools\_libs  
 if d.id == ed.lib\_id]  
  
 many\_to\_many = [(e.name\_fio, e.cost, lib\_name)  
 for lib\_name, lib\_id, book\_id in many\_to\_many\_temp  
 for e in books\_list if e.id == book\_id]  
  
 print('Задание А1')  
 res\_11 = sorted(one\_to\_many, key=itemgetter(2))  
 print(res\_11)  
  
 print('\nЗадание А2')  
 res\_12\_unsorted = []  
 # Перебираем все отделы  
 for d in libs\_list:  
 # Список сотрудников отдела  
 d\_emps = list(filter(lambda i: i[2] == d.name, one\_to\_many))  
 # Если отдел не пустой  
 if len(d\_emps) > 0:  
 # Зарплаты сотрудников отдела  
 d\_sals = [sal for \_, sal, \_ in d\_emps]  
 # Суммарная зарплата сотрудников отдела  
 d\_sals\_sum = sum(d\_sals)  
 res\_12\_unsorted.append((d.name, d\_sals\_sum))  
  
 # Сортировка по суммарной зарплате  
 res\_12 = sorted(res\_12\_unsorted, key=itemgetter(1), reverse=True)  
 print(res\_12)  
  
 print('\nЗадание А3')  
 res\_13 = {}  
 # Перебираем все отделы  
 for d in libs\_list:  
 if 'Городская' in d.name:  
 # Список сотрудников отдела  
 d\_emps = list(filter(lambda i: i[2] == d.name, many\_to\_many))  
 # Только ФИО сотрудников  
 d\_emps\_names = [x for x, \_, \_ in d\_emps]  
 # Добавляем результат в словарь  
 # ключ - отдел, значение - список фамилий  
 res\_13[d.name] = d\_emps\_names  
  
 print(res\_13)  
  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 main()

**Результат выполнения программы:**

****