Бакушев Илья

ИУ5-33Б

РК1 по ПиКЯП Вариант:

# Вариант Б.

1. «Отдел» и «Сотрудник» связаны соотношением один-ко-многим. Выведите список всех связанных сотрудников и отделов, отсортированный по сотрудникам, сортировка по отделам произвольная.
2. «Отдел» и «Сотрудник» связаны соотношением один-ко-многим. Выведите список отделов с количеством сотрудников в каждом отделе, отсортированный по количеству сотрудников.
3. «Отдел» и «Сотрудник» связаны соотношением многие-ко-многим. Выведите список всех сотрудников, у которых фамилия заканчивается на «ов», и названия их отделов.



# Текст программы:

from operator import itemgetter  
  
  
class Faculty:  
 *"""Факультет"""* def \_\_init\_\_(self, id, name, num\_employees):  
 self.id = id  
 self.name = name  
 self.num\_employees = num\_employees  
  
  
class University:  
 *"""Университет"""* def \_\_init\_\_(self, id, name):  
 self.id = id  
 self.name = name  
  
  
class FacultyUniversity:  
 *"""  
 'Факультеты университета' для реализации  
 связи многие ко-многим  
 """* def \_\_init\_\_(self, university\_id, faculty\_id):  
 self.university\_id = university\_id  
 self.faculty\_id = faculty\_id  
  
  
# Факультеты  
faculties = [  
 Faculty(1, 'Факультет информационных технологий', 150),  
 Faculty(2, 'Факультет экономики', 120),  
 Faculty(3, 'Факультет гуманитарных наук', 80),  
 Faculty(4, 'Факультет права', 100),  
]  
  
# Университеты  
universities = [  
 University(1, 'Московский государственный университет'),  
 University(2, 'Санкт-Петербургский государственный университет'),  
 University(3, 'Казанский федеральный университет'),  
 University(4, 'Сибирский федеральный университет'),  
 University(5, 'МГТУ им. Баумана'),  
]  
  
# Деканы  
decans = [  
 ('Иванов', 1),  
 ('Петров', 2),  
 ('Сидоров', 3),  
 ('Кузнецов', 4),  
 ('Смирнов', 1),

]  
  
faculties\_universities = [  
 FacultyUniversity(1, 1),  
 FacultyUniversity(2, 2),  
 FacultyUniversity(1, 3),  
 FacultyUniversity(2, 4),  
 FacultyUniversity(3, 1),  
 FacultyUniversity(4, 2),  
 FacultyUniversity(3, 4),  
 FacultyUniversity(5, 1),  
 FacultyUniversity(5, 2),  
]  
  
  
def main():  
 *"""Основная функция"""* # Соединение данных один-ко-многим  
 one\_to\_many = [(f.name, u.name)  
 for u in universities  
 for f in faculties  
 if f.id == u.id]  
  
 # Соединение данных многие-ко-многим  
 many\_to\_many\_temp = [(u.name, fu.university\_id, fu.faculty\_id)  
 for u in universities  
 for fu in faculties\_universities  
 if u.id == fu.university\_id]  
 many\_to\_many = [(f.name, u\_name)  
 for u\_name, university\_id, faculty\_id in many\_to\_many\_temp  
 for f in faculties if f.id == faculty\_id]  
  
 print('Задание Б1')  
 res\_11 = sorted(one\_to\_many, key=itemgetter(0))  
 print(res\_11)  
  
 print('\nЗадание Б2')  
 res\_12\_unsorted = []  
 for u in universities:  
 u\_faculties = list(filter(lambda i: i[1] == u.name, one\_to\_many))  
 if len(u\_faculties) > 0:  
 total\_employees = sum(f.num\_employees for f in faculties if f.name in [x[0] for x in u\_faculties])  
 res\_12\_unsorted.append((u.name, total\_employees))  
 res\_12 = sorted(res\_12\_unsorted, key=itemgetter(1))  
 print(res\_12)  
  
 print('\nЗадание Б3')  
 res\_13\_new = []  
 for d in decans:  
 if d[0].endswith('ов'):  
 for f\_name, u\_name, faculty\_id in many\_to\_many\_temp:  
 if d[1] == faculty\_id:  
 res\_13\_new.append((d[0], u\_name))  
 break  
 print(res\_13\_new)  
  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 main()

# Результат выполнения:

# Задание Б1

# [('Факультет гуманитарных наук', 'Казанский федеральный университет'), ('Факультет информационных технологий', 'Московский государственный университет'), ('Факультет права', 'Сибирский федеральный университет'), ('Факультет экономики', 'Санкт-Петербургский государственный университет')]

# Задание Б2

# [('Казанский федеральный университет', 80), ('Сибирский федеральный университет', 100), ('Санкт-Петербургский государственный университет', 120), ('Московский государственный университет', 150)]

# Задание Б3

# [('Иванов', 1), ('Петров', 2), ('Сидоров', 1), ('Кузнецов', 2), ('Смирнов', 1)]