 МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

ИМ. Н. Э. БАУМАНА

ФАКУЛЬТЕТ

***«***ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ***»***

КАФЕДРА

***«***АВТОМАТИЗИРОВАННЫЕ СИСТЕМЫ ОБРАБОТКИ ИНФОРМАЦИИ И УПРАВЛЕНИЯ***»***

***Рубежный контроль №1***

*по учебной дисциплине*

***«Разработка Интернет Приложений»***

*Вариант №Б27*

|  |  |
| --- | --- |
| ***Группа:*** | *ИУ5Ц-72Б* |
| ***Студент:*** | *Акимкин М.Г.* |
|  |  |
| ***Преподаватель:*** | *Гапанюк Ю.Е.* |
|  |  |
|  |  |

Москва, 2020

1. **Условия Задания**

**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.

**Вариант 27.**

|  |  |  |
| --- | --- | --- |
| **№ варианта** | **Класс 1** | **Класс 2** |
| 27 | Преподаватель | Учебный курс |

1. **Текст программы**

# используется для сортировки

from operator import itemgetter

class Teacher():

def \_\_init\_\_(self, id, name, lastname, midname, selary, course\_id):

self.id = id

self.name = name

self.lastname = lastname

self.midname = midname

self.selary = selary

self.course\_id = course\_id

class Course():

def \_\_init\_\_(self, id, name):

self.id = id

self.name = name

class TC():

def \_\_init\_\_(self, teach\_id, cour\_id):

self.teach\_id = teach\_id

self.cour\_id = cour\_id

courses = [

Course(1, 'РИП'),

Course(2, 'БКИТ'),

Course(3, 'Вычислительные средства АСОиУ'),

Course(4, 'Схемотехника дискретных устройств'),

]

teachers = [

Teacher(1, 'Юрий', 'Гапанюк', 'Евгеньевич', 80000, 1),

Teacher(2, 'Юрий', 'Гапанюк', 'Евгеньевич',80000, 2),

Teacher(3, 'Сергей', 'Спиридонов', 'Борисович', 100000, 3),

Teacher(4, 'Сергей', 'Спиридонов', 'Борисович', 100000, 4),

Teacher(5, 'Андрей', 'Аксенов', 'Николаевич', 75000, 3),

Teacher(6, 'Андрей', 'Аксенов', 'Николаевич', 75000, 4),

]

teach\_cour = [

TC(1,1),

TC(1,2),

TC(3,3),

TC(3,4),

]

def main():

print()

print('Акимкин М. Г., ИУ5ц-72Б, РИП, РК1')

print()

# Соединение данных один-ко-многим

one\_to\_many = [(e.name, e.lastname, d.name)

for d in courses

for e in teachers

if e.course\_id == d.id]

print()

print('Задание Б1')

print(sorted(one\_to\_many, key=itemgetter(0)))

# Соединение данных один-ко-многим

one\_to\_many\_2 = set()

for i in courses:

arr = ['', 0]

for j in teachers:

if j.course\_id == i.id:

if arr[0] == '':

arr[0] = i.name

arr[1] += 1

else:

arr[1] += 1

continue

one\_to\_many\_2.add((arr[0], arr[1]))

print()

print('Задание Б2')

print(sorted(one\_to\_many\_2, key=itemgetter(1)))

many\_to\_many = {}

for i in teach\_cour:

length = len(teachers[i.cour\_id-1].lastname)

#print(teachers[i.cour\_id-1].lastname[length-1])

if courses[i.cour\_id-1].name in many\_to\_many.keys():

if teachers[i.cour\_id-1].lastname[length-1] == 'в' and teachers[i.cour\_id-1].lastname[length-2] == 'о':

many\_to\_many[courses[i.cour\_id-1].name].add((teachers[i.cour\_id-1].name, teachers[i.cour\_id-1].lastname, teachers[i.cour\_id-1].midname))

else:

if teachers[i.cour\_id-1].lastname[length-1] == 'в' and teachers[i.cour\_id-1].lastname[length-2] == 'о':

many\_to\_many[courses[i.cour\_id-1].name] = set()

many\_to\_many[courses[i.cour\_id-1].name].add((teachers[i.cour\_id-1].name, teachers[i.cour\_id-1].lastname, teachers[i.cour\_id-1].midname))

print()

print('Задание Б3')

print(many\_to\_many)

if \_\_name\_\_ == '\_\_main\_\_':

main()

1. **Пример работы программы**

