

**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»

РК №1

Вариант №19 (Г)

Выполнил:

студент группы ИУ5-34Б

Флоринский Владислав

Подпись и дата:

Проверил:

преподаватель каф. ИУ5

Гапанюк Юрий Евгеньевич

Подпись и дата:

Москва, 2023 г.

# Постановка задачи

## Условия рубежного контроля №1 по курсу ПиК ЯП

Рубежный контроль представляет собой разработку программы на языке Python, которая выполняет следующие действия:

1) Необходимо создать два класса данных в соответствии с Вашим вариантом предметной области, которые связаны отношениями один-ко-многим и многие-ко-многим.

Пример классов данных для предметной области Сотрудник-Отдел:

1. Класс «Сотрудник», содержащий поля:
  - ID записи о сотруднике;
  - Фамилия сотрудника;
  - Зарплата (количественный признак);
  - ID записи об отделе. (для реализации связи один-ко-многим)
2. Класс «Отдел», содержащий поля:
  - ID записи об отделе;
  - Наименование отдела.
3. (Для реализации связи многие-ко-многим) Класс «Сотрудники отдела», содержащий поля:
  - ID записи о сотруднике;
  - ID записи об отделе.

2) Необходимо создать списки объектов классов, содержащих тестовые данные (3-5 записей), таким образом, чтобы первичные и вторичные ключи соответствующих записей были связаны по идентификаторам.

3) Необходимо разработать запросы в соответствии с Вашим вариантом. Запросы сформулированы в терминах классов «Сотрудник» и «Отдел», которые используются в примере. Вам нужно перенести эти требования в Ваш вариант предметной области. При разработке запросов необходимо по возможности использовать функциональные возможности языка Python (list/dict comprehensions, функции высших порядков).

Для реализации запроса №2 введите в класс, находящийся на стороне связи «много», произвольный количественный признак, например, «зарплата сотрудника».

Результатом рубежного контроля является документ в формате PDF, который содержит текст программы и результаты ее выполнения.

### Вариант Г.

1. «Отдел» и «Сотрудник» связаны соотношением один-ко-многим. Выведите список всех отделов, у которых название начинается с буквы «А», и список работающих в них сотрудников.
2. «Отдел» и «Сотрудник» связаны соотношением один-ко-многим. Выведите список отделов с максимальной зарплатой сотрудников в каждом отделе, отсортированный по максимальной зарплате.
3. «Отдел» и «Сотрудник» связаны соотношением многие-ко-многим. Выведите список всех связанных сотрудников и отделов, отсортированный по отделам, сортировка по сотрудникам произвольная.

19	Деталь	Производитель
----	--------	---------------

## Код программы:

```
from prettytable import PrettyTable
from operator import itemgetter
import random

id = 1
id_area = 1

class Employees:
    #название детали - цена - производитель
    def __init__(self, sname, salary, id_area):
        global id
        self.__id = id
        id += 1
        self.__sname = sname
        self.__salary = salary
        self.__id_area = id_area
    def get_id(self): return(self.__id)
    def get_sname(self): return(self.__sname)
    def get_salary(self): return(self.__salary)
    def get_area(self): return(self.__id_area)

class Area:
    def __init__(self, name):
        global id_area
        self.__id_area = id_area
        id_area += 1
        self.__name = name
    def get_name(self): return(self.__name)
    def get_id_area(self): return(self.__id_area)

class Area_Empl:
    def __init__(self, area_id, own_id):
        self.__area_id = area_id
        self.__own_id = own_id
    def get_ownid(self): return(self.__own_id)
    def get_areaid(self): return(self.__area_id)

snames = []
anames = []
with open("snames.txt", "r") as f:
    for line in f: snames.append(line[:-1])
with open("anames.txt", "r") as f:
    for line in f: anames.append(line[:-1])
salaries = [random.randint(1000, 20000) for i in range(len(snames))]
emp_lis = [Employees(snames[i], salaries[i], id_area + (i // 5)) for i in range(len(snames))]
area_lis = [Area(anames[i]) for i in range(len(snames) // 5)]
emp_dict = { key:value for key, value in zip([i for i in range(1, id+1)], emp_lis)}
area_dict = {key:value for key,value in zip([i for i in range(1, id_area+1)], area_lis)}
table = PrettyTable()
table.field_names = ["ID детали", "Название", "Цена", "ID производителя"]
```

```

for i in emp_dict.keys():
    table.add_row([emp_dict[i].get_id(), emp_dict[i].get_sname(), emp_dict[i].get_salary(), emp_dict[i].get_area()])
print(table)
table2 = PrettyTable()
table2.field_names = ["ID детали", "Название производителя"]
for i in area_dict.keys():
    table2.add_row([area_dict[i].get_id_area(), area_dict[i].get_name()])
print(table2)
corr_id = 1
arr_corr = []
while (corr_id <= len(area_dict.keys())):
    if area_dict[corr_id].get_name()[0] == "A":
        arr_corr.append(corr_id)
        corr_id += 1
print("###Task 1###")
if len(arr_corr) == 0:
    print("Производителей, начинающихся на А нет")
else:
    print("Производители, начинающиеся на А:")
    for i in arr_corr:
        print(area_dict[i].get_name(), end=': \n')
        for g in emp_dict.keys():
            if emp_dict[g].get_area() == i:
                print(f"({emp_dict[g].get_id()}) - {emp_dict[g].get_sname()} - {emp_dict[g].get_salary()} - {emp_dict[g].get_area()}")
        print('')
print("###Task 2###")
max_sal = {}
sal_row = 0
for i in area_dict.keys():
    sal_row = 0
    for g in emp_dict.keys():
        if emp_dict[g].get_area() == i:
            if emp_dict[g].get_salary() > sal_row:
                sal_row = emp_dict[g].get_salary()
    max_sal[i] = sal_row
max_sal_sor = sorted(list(max_sal.items()), key=itemgetter(1), reverse=True)
tablesal = PrettyTable()
tablesal.field_names = ["ID производителя", "Название производителя", "Максимальная цена детали у производителя"]
for i,j in max_sal_sor:
    tablesal.add_row([i, area_dict[i].get_name(), j])
print(tablesal)
print("###Task 3###")
rarea = [random.choice(list(area_dict.keys())) for i in range(30)]
remp = [random.choice(list(emp_dict.keys())[:10]) for i in range(30)]
setarem = set([(i, j) for i, j in zip(rarea, remp)])
aremtest = [Area_Empl(i[0], i[1]) for i in setarem]
arem = sorted([(i.get_areaid(), i.get_ownid()) for i in aremtest], key = itemgetter(0))
table3 = PrettyTable()

```

```

table3.field_names = ["ID производителя", "ID детали", "Название", "Цена детали"]
for i in arem:
    table3.add_row([i[0], i[1], emp_dict[i[1]].get_sname(), emp_dict[i[1]].get_salary()])
print(table3)

```

## Результаты

Наша база данных:

ID детали	Название	Цена	ID производителя
1	Винт	5476	1
2	Гайка	5269	1
3	Шайба	9433	1
4	Втулка	2123	1
5	Пружина	1505	1
6	Шплинт	15390	2
7	Шестерня	10448	2
8	Подшипник	18297	2
9	Плита	13225	2
10	Ремень	6235	2
11	Шкив	14449	3
12	Шарнир	1700	3
13	Зубчатая рейка	2655	3
14	Вал	5633	3
15	Кронштейн	2045	3
16	Резиновая прокладка	6431	4
17	Пневмокамера	13868	4
18	Трос	17258	4
19	Регулировочная гайка	14340	4
20	Щетка	7960	4
21	Рычаг	11401	5
22	Клапан	8857	5
23	Ротор	13752	5
24	Сальник	9163	5
25	Поршень	7702	5
26	Цилиндр	16770	6
27	Линза	1168	6
28	Диск	2145	6
29	Фильтр	10518	6
30	Катушка	8065	6
31	Электрод	1406	7
32	Лента	14458	7
33	Защитный кожух	1714	7
34	Датчик	9559	7
35	Щуп	5956	7

ID детали	Название производителя
1	Звезда-Техника
2	Сибирские Детали
3	Волжский Агрегат
4	АгроПром комбинат
5	Невская Механика
6	КубаньЭлемент
7	Арбуз деталь

### Task 1

Производители, начинающиеся на А:

АгроПром комбинат : [

(16 - Резиновая прокладка - 6431 - 4)

(17 - Пневмокамера - 13868 - 4)

(18 - Трос - 17258 - 4)

(19 - Регулировочная гайка - 14340 - 4)

(20 - Щетка - 7960 - 4)

]

Арбуз деталь : [

(31 - Электрод - 1406 - 7)

(32 - Лента - 14458 - 7)

(33 - Защитный кожух - 1714 - 7)

(34 - Датчик - 9559 - 7)

(35 - Щуп - 5956 - 7)

]

### Task 2

ID производителя	Название производителя	Максимальная цена детали у производителя
2	Сибирские Детали	18297
4	АгроПром комбинат	17258
6	КубаньЭлемент	16770
7	Арбуз деталь	14458
3	Волжский Агрегат	14449
5	Невская Механика	13752
1	Звезда-Техника	9433

### Task 3

ID производителя	ID детали	Название	Цена делати
1	3	Шайба	9433
1	8	Подшипник	18297
1	7	Шестерня	10448
2	8	Подшипник	18297
2	7	Шестерня	10448
3	4	Втулка	2123
3	10	Ремень	6235
4	3	Шайба	9433
4	9	Плита	13225
4	2	Гайка	5269
4	7	Шестерня	10448
5	4	Втулка	2123
5	1	Винт	5476
5	10	Ремень	6235
5	2	Гайка	5269
5	8	Подшипник	18297
6	1	Винт	5476
6	6	Шплинт	15390
6	9	Плита	13225
7	3	Шайба	9433
7	9	Плита	13225
7	8	Подшипник	18297