

**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 А №11

Выполнил:

студент группы ИУ5-31

Карпова Ксения Павловна

Подпись и дата:

Проверил:

преподаватель каф. ИУ5

Гапанюк Юрий
Евгеньевич

Подпись и дата:

Москва, 2023г.

Постановка задачи

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 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.

11	Программа	Компьютер
----	-----------	-----------

Код программы

```
# Программа Компьютер

from operator import itemgetter

class Pr: # Program
    def __init__(self, id, name, comp_id, size_in_gb):
        self.id = id
        self.name = name
        self.comp_id = comp_id
        self.size_in_gb = size_in_gb

class Comp: # Computer
    def __init__(self, id, model, RAM, owner):
        self.id = id
        self.model = model
        self.RAM = RAM
        self.owner = owner

class Pr_Comp:
    def __init__(self, pr_id, comp_id):
        self.pr_id = pr_id
        self.comp_id = comp_id

programs = [
    Pr(1, 'Microsoft Word', 1, 2.0),
    Pr(2, 'Microsoft Excel', 1, 1.5),
    Pr(3, 'Google Docs', 3, 1.0),
    Pr(4, 'LibreOffice Writer', 1, 1.8),
    Pr(5, 'OpenOffice Calc', 4, 1.2)
]

# Rest of the code remains the same...
```

```
computers = [
    Comp(1, 'Model A-2000X', '8 GB DDR4', 'Иванов Иван Иванович'),
    Comp(2, 'UltraBook Pro 15S', '16 GB DDR4', 'Петрова Анна Сергеевна'),
    Comp(3, 'GamingBeast X9000', '32 GB DDR4', 'Смирнов Сергей
Владимирович'),
    Comp(4, 'OfficeMaster 500', '64 GB DDR4', 'Козлова Екатерина Павловна'),
    Comp(5, 'PerformanceElite 3000', '128 GB DDR4', 'Михайлов Алексей
Дмитриевич')
]
```

```
pr_comp = [Pr_Comp(1, 1), Pr_Comp(1, 2), Pr_Comp(1, 4),
            Pr_Comp(2, 3), Pr_Comp(2, 1),
            Pr_Comp(3, 4), Pr_Comp(4, 5),
            Pr_Comp(3, 5)]
```

```
def main():
    one_to_many = [(p.name, p.size_in_gb, c.owner, c.model)
                   for p in programs
                   for c in computers
                   if p.comp_id == c.id]
    one_to_many_sorted = sorted(one_to_many, key=itemgetter(0))

    many_to_many_temp = [(p.name, p_c.pr_id, p_c.comp_id)
                         for p_c in pr_comp
                         for p in programs
                         if p.id == p_c.pr_id]

    many_to_many = [(pr_name, c.model, c.owner)
                    for pr_name, dep_id, p_c_id in many_to_many_temp
                    for c in computers if c.id == p_c_id]
```

```
#####3
```

```
print('\033[91mЗадание A1\033[0m')
```

```
col_width = [40, 20, 40, 20]
```

```
# Выводим заголовок таблицы
```

```
print(
    f"{'Название'
Программы':<{col_width[0]}}{'Гбайты':<{col_width[1]}}{'Владелец
Компьютера':<{col_width[2]}}{'Модель Компьютера':<{col_width[3]}}")
print('-' * sum(col_width))
```

```
# Выводим данные в виде таблицы
```

```
for row in one_to_many_sorted:
    print(''.join(f"{str(item):<{width}}" for item, width in zip(row,
col_width)))
```

```
#####
```

```
print('\n\033[91mЗадание A2\033[0m')
```

```
res_12_unsorted = []
```

```
for c in computers:
    c_owners = list(filter(lambda i: i[2] == c.owner, one_to_many))
    if len(c_owners) > 0:
        c_Gb = [gb for _, gb, _, _ in c_owners]
        c_Gb_sum = sum(c_Gb)
        res_12_unsorted.append((c.owner, c.RAM, c_Gb_sum))
```

```
res_12 = sorted(res_12_unsorted, key=itemgetter(1), reverse=True)
col_width_2 = [40, 40, 20]
```

```

print(
    f"{'Владелец Компьютера':<{col_width_2[0]}}{'Оперативная  

память':<{col_width_2[0]}}{'Занятая память (в GB)':<{col_width_2[2]}}")
print('-' * sum(col_width_2))

for owner, ram, total_size in res_12:

print(f"{'owner':<{col_width_2[0]}}{'ram':<{col_width_2[1]}}{'total_size':<{col_wid  

th_2[2]}}")

#####

print('\n\033[91mЗадание A3\033[0m')

res_13 = {}

for p in programs:
    if 'Microsoft' in p.name:
        p_c = list(filter(lambda i: i[0] == p.name, many_to_many_temp))
        res_13[p.name] = [c.model for _, _, p_c_id in p_c for c in
computers if c.id == p_c_id]

print(res_13)

if __name__ == '__main__':
    main()

```

Результаты

Задание A1

Название Программы	Гбайты	Владелец Компьютера	Модель Компьютера
Google Docs	1.0	Смирнов Сергей Владимирович	GamingBeast X9000
LibreOffice Writer	1.8	Иванов Иван Иванович	Model A-2000X
Microsoft Excel	1.5	Иванов Иван Иванович	Model A-2000X
Microsoft Word	2.0	Иванов Иван Иванович	Model A-2000X
OpenOffice Calc	1.2	Козлова Екатерина Павловна	OfficeMaster 500

Задание A2

Владелец Компьютера	Оперативная память	Занятая память (в GB)
Иванов Иван Иванович	8 GB DDR4	5.3
Козлова Екатерина Павловна	64 GB DDR4	1.2
Смирнов Сергей Владимирович	32 GB DDR4	1.0

Задание A3

```
{'Microsoft Word': ['Model A-2000X', 'UltraBook Pro 15S', 'OfficeMaster 500'], 'Microsoft Excel': ['GamingBeast X9000', 'Model A-2000X']}
```