



**Министерство науки и высшего образования Российской
Федерации Федеральное государственное бюджетное
образовательное учреждение высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)**

**Факультет «Информатика и системы управления»
Кафедра ИУ5 «Системы обработки информации и управления»**

**Рубежный контроль №1
по дисциплине «Базовые компоненты интернет-технологий»**

**Выполнила:
студентка группы ИУ5-33Б
Жамнова М. С.**

**Проверил:
Гапанюк Ю.Е**

2021 г.

Оглавление

Постановка задания:	3
Текст программы:	4
Экранные формы с примерами выполнения программы:	5

Постановка задания:

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.
- 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».

Вариант В.

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

Предметная область:

4	Компьютер	Дисплейный класс
---	-----------	------------------

Текст программы:

```
# используется для сортировки
from operator import itemgetter

class Comp:
    """Компьютер"""
    def __init__(self, id, brand, price, disp_cls_id):
        self.id = id
        self.brand = brand
        self.price = price
        self.disp_cls_id = disp_cls_id

class Disp_cls:
    """Дисплейный класс"""
    def __init__(self, id, name):
        self.id = id
        self.name = name

class CompDisp_cls:
    """
    'Компьютеры дисплейного класса' для реализации
    связи многие-ко-многим
    """
    def __init__(self, disp_cls_id, comp_id):
        self.disp_cls_id = disp_cls_id
        self.comp_id = comp_id

# Дисплейные классы
disp_classes = [
    Disp_cls(1, 'А-класс'),
    Disp_cls(2, 'Б-класс'),
    Disp_cls(3, 'В-класс'),
    Disp_cls(11, 'Г-класс'),
    Disp_cls(22, 'Д-класс'),
    Disp_cls(33, 'Е-класс'),
]

# Компьютеры
comps = [
    Comp(1, 'HP', 1168390, 1),
    Comp(2, 'Asus', 55390, 1),
    Comp(3, 'ThinkPad', 78940, 3),
    Comp(4, 'Acer', 97450, 2),
    Comp(5, 'Lenovo', 86980, 3),
]

# Компьютеры и Дисплейные классы для связи многие-ко-многим
comps_disp_classss = [
    CompDisp_cls(1, 1),
    CompDisp_cls(1, 2),
    CompDisp_cls(3, 3),
    CompDisp_cls(2, 4),
    CompDisp_cls(3, 5),

    CompDisp_cls(33, 1),
    CompDisp_cls(22, 2),
    CompDisp_cls(11, 3),
]
```

```

CompDisp_cls(33, 4),
CompDisp_cls(22, 5),
]

def main():
    """Основная функция"""

    # Соединение данных один-ко-многим
    one_to_many = [(c.brand, c.price, d.name)
                    for d in disp_classes
                    for c in comps
                    if c.disp_cls_id == d.id]

    # Соединение данных многие-ко-многим
    many_to_many_temp = [(d.name, dc.disp_cls_id, dc.comp_id)
                          for d in disp_classes
                          for dc in comps_disp_classes
                          if d.id == dc.disp_cls_id]

    many_to_many = [(c.brand, c.price, disp_class_name)
                    for disp_class_name, disp_cls_id, comp_id in
many_to_many_temp
                    for c in comps if c.id == comp_id]

    print('Задание B1')
    task1 = []
    for brand, price, name in one_to_many:
        if brand[0] == "A":
            task1.append((brand, name))
    print(task1)

    print('\nЗадание B2')
    task2_uns = []
    for d in disp_classes:
        # все компьютеры дисплейного класса
        d_comps = list(filter(lambda i: i[2] == d.name, one_to_many))
        if len(d_comps) > 0:
            d_price = [price for _, price, _ in d_comps]
            d_minPrice = min(d_price)
            task2_uns.append((d.name, d_minPrice))
    task2 = sorted(task2_uns, key=itemgetter(1))
    print(task2)

    print('\nЗадание B3')
    task3_uns = []
    for brand, price, name in many_to_many:
        task3_uns.append((brand, name))

    task3 = list(sorted(task3_uns, key=itemgetter(0)))
    print(task3)

if __name__ == '__main__':
    main()

```

Экранные формы с примерами выполнения программы:

```
C:\Users\A\Desktop\LABS\BKIT_2021\venv\Scripts\python.exe C:/Users/a/Desktop/LABS/BKIT_2021/code/PK1/main.py
Задание В1
[('Asus', 'А-класс'), ('Acer', 'Б-класс')]

Задание В2
[('А-класс', 55390), ('В-класс', 78940), ('Б-класс', 97450)]

Задание В3
[('Acer', 'Б-класс'), ('Acer', 'Е-класс'), ('Asus', 'А-класс'), ('Asus', 'Д-класс'), ('HP', 'А-класс'), ('HP', 'Е-класс'), ('Lenovo', 'В-класс'), ('Lenovo', 'Д-класс'), ('ThinkPad', 'В-класс'), ('ThinkPad', 'Г-класс')]

Process finished with exit code 0
```

Задание В1

[('Asus', 'А-класс'), ('Acer', 'Б-класс')]

Задание В2

[('А-класс', 55390), ('В-класс', 78940), ('Б-класс', 97450)]

Задание В3

[('Acer', 'Б-класс'), ('Acer', 'Е-класс'), ('Asus', 'А-класс'), ('Asus', 'Д-класс'), ('HP', 'А-класс'), ('HP', 'Е-класс'), ('Lenovo', 'В-класс'), ('Lenovo', 'Д-класс'), ('ThinkPad', 'В-класс'), ('ThinkPad', 'Г-класс')]

Process finished with exit code 0