

**Московский государственный технический
университет им. Н.Э. Баумана**

**Факультет ИУ
Кафедра ИУ5**

**Курс «Основы информатики»
Отчет по Рубежному контролю №1
Вариант В4**

Выполнил студент группы ИУ5-33Б:

Козлов А. А.

Подпись и дата:

Проверил преподаватель каф.:

Гапанюк Ю. Е.

Подпись и дата:

Постановка задачи

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 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.
4. **Таблица 1.** Варианты предметной области

Вариант	Класс 1	Класс 2
4	Компьютер	Дисплейный класс

Текст программы

```
from operator import itemgetter

class Computer():
    def __init__(self, id, name, cost, class_id):
        self.id = id
        self.name = name
        self.cost = cost
        self.class_id = class_id

class DisClass():
    def __init__(self, id, name):
        self.id = id
        self.name = name

class CompDC():
    def __init__(self, class_id, comp_id):
        self.class_id = class_id
        self.comp_id = comp_id

comps = [
    Computer(1, 'A123', 100, 1),
    Computer(2, 'B123', 200, 2),
    Computer(3, 'A233', 150, 3),
    Computer(4, 'D123', 200, 4),
    Computer(5, 'E123', 300, 5),]

dis_classes = [
    DisClass(1, '312л'),
    DisClass(2, '313л'),
    DisClass(3, '314л'),
    DisClass(4, '315л'),
    DisClass(5, '316л'),]

comp_dc = [
    CompDC(1, 1),
    CompDC(2, 2),
    CompDC(3, 3),
    CompDC(3, 4),
    CompDC(3, 5),
    CompDC(4, 1),
    CompDC(4, 2),
    CompDC(4, 3),
    CompDC(5, 4),
    CompDC(5, 5),]

def main():
    one_to_many = [(c.name, c.cost, dc.name)
                    for dc in dis_classes
                    for c in comps
                    if c.class_id == dc.id]
```

```

many_to_many_temp = [(dc.name, co.class_id, co.comp_id)
                      for dc in dis_classes
                      for co in comp_dc
                      if dc.id == co.class_id]

many_to_many = [(c.name, c.cost, class_name)
                 for class_name, class_id, comp_id in many_to_many_temp
                 for c in comps if c.id == comp_id]

print('Задание B1')
'''
«дисплейный класс» и «компьютер» связаны соотношением один-ко-многим. Выведите список
всех компьютеров, у которых название начинается с буквы «А», и названия их дисплейных
классов.
'''

resultB1 = sorted(x for x in one_to_many if x[0].startswith('A'))
for i in resultB1:
    print(f'Название: {i[0]}, класс: {i[2]}')

print('\nЗадание B2')
'''
«дисплейный класс» и «компьютер» связаны соотношением один-ко-многим. Выведите список
классов с минимальной ценой за компьютер в каждом дисплейном классе, отсортированный по
минимальной цене.
'''

resultB2 = {}
for class_name in set(x[2] for x in one_to_many):
    min_cost = min(x[1] for x in one_to_many if x[2] == class_name)
    resultB2[class_name] = min_cost
resultB2 = sorted(resultB2.items(), key=itemgetter(1))
for i in resultB2:
    print(f'Класс: {i[0]}, минимальная стоимость: {i[1]}')

print('\nЗадание B3')
'''
«дисплейный класс» и «компьютер» связаны соотношением многие-ко-многим. Выведите
список всех связанных компьютеров и дисплейных классов, отсортированный по компьютерам,
сортировка по дисплейным классам произвольная.
'''

resultB3 = sorted(many_to_many, key=itemgetter(0))
for i in resultB3:
    print(f'Название: {i[0]}, класс: {i[2]}')
main()

```

Анализ результатов

Задание В1

Название: A123, класс: 312л

Название: A233, класс: 314л

Задание В2

Класс: 312л, минимальная стоимость: 100

Класс: 314л, минимальная стоимость: 150

Класс: 313л, минимальная стоимость: 200

Класс: 315л, минимальная стоимость: 200

Класс: 316л, минимальная стоимость: 300

Задание В3

Название: A123, класс: 312л

Название: A123, класс: 315л

Название: A233, класс: 314л

Название: A233, класс: 315л

Название: B123, класс: 313л

Название: B123, класс: 315л

Название: D123, класс: 314л

Название: D123, класс: 316л

Название: E123, класс: 314л

Название: E123, класс: 316л

PS C:\Users\123\Documents\Work\Programs\2ndSem-RK1> █