

МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ
УНИВЕРСИТЕТ
им. Н.Э. Баумана

Факультет “Информатика и системы управления”
Кафедра “Системы обработки информации и
управления”



Дисциплина “Парадигмы и конструкции языков
программирования”

Отчет по рубежному контролю № 1

Вариант №3

Вариант запросов Г

Выполнил:

Студент группы ИУ5-35Б
Александров Г.С.

Преподаватель:

Гапанюк Ю. Е.

Москва 2025

Задание рубежного контроля

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.

Задание варианта

- 1) Язык программирования и Средство разработки связаны отношением один-ко-многим. Выведите список всех средств разработки, у которых название начинается с буквы «А», и названия их языков программирования.
- 2) Язык программирования и Средство разработки связаны отношением один-ко-многим. Выведите список языков с максимальной стоимостью лицензии средств разработки в каждом языке, отсортированный по максимальной стоимости.
- 3) Язык программирования и Средство разработки связаны отношением многие-ко-многим. Выведите список всех связанных средств разработки и языков программирования, отсортированный по языкам, сортировка по средствам произвольная.

Текст программы

Файл Alexandrov 35:

```
class Autopark:
    """Автопарк"""
    def __init__(self, id, name):
        self.id = id
        self.name = name

class Driver:
    """Водитель"""
    def __init__(self, id, fio, salary, park_id):
        self.id = id
        self.fio = fio
        self.salary = salary
        self.park_id = park_id

class DriverAutopark:
    """Водители автопарка (для связи многие-ко-многим)"""
    def __init__(self, park_id, driver_id):
        self.park_id = park_id
        self.driver_id = driver_id

autoparks = [
    Autopark(1, 'Автопарк Северный'),
    Autopark(2, 'Автобусный парк'),
    Autopark(3, 'Таксопарк Центральный'),
    Autopark(4, 'Аэропортный автопарк'),
    Autopark(5, 'Грузовой парк'),
]

drivers = [
    Driver(1, 'Иванов', 50000, 1),
    Driver(2, 'Петров', 45000, 2),
    Driver(3, 'Сидоров', 55000, 3),
    Driver(4, 'Алексеев', 48000, 3),
    Driver(5, 'Антонов', 52000, 4),
    Driver(6, 'Александров', 47000, 5),
]

drivers_autoparks = [
    DriverAutopark(1, 1),
    DriverAutopark(2, 2),
    DriverAutopark(3, 3),
    DriverAutopark(3, 4),
    DriverAutopark(4, 5),
    DriverAutopark(5, 6),
    DriverAutopark(1, 3),
]
```

```

        DriverAutopark(2, 4),
    ]

def main():

    one_to_many = [(d.fio, d.salary, p.name)
                    for p in autoparks
                    for d in drivers
                    if d.park_id == p.id]

    many_to_many_temp = [(p.name, da.park_id, da.driver_id)
                          for p in autoparks
                          for da in drivers_autoparks
                          if p.id == da.park_id]

    many_to_many = [(d.fio, d.salary, park_name)
                     for park_name, park_id, driver_id in many_to_many_temp
                     for d in drivers if d.id == driver_id]

    print('Задание Г1')

    res_g1 = {}
    for p in autoparks:
        if p.name.startswith('A'):
            p_drivers = list(filter(lambda i: i[2] == p.name, one_to_many))
            p_drivers_names = [x for x, _, _ in p_drivers]
            res_g1[p.name] = p_drivers_names

    print("Автопарки, начинающиеся на 'A', и их водители:")
    for park, drivers_list in res_g1.items():
        print(f"{park}: {drivers_list}")

    print("\nЗадание Г2")
    park_drivers_dict = {}
    for driver_fio, salary, park_name in one_to_many:
        if park_name not in park_drivers_dict:
            park_drivers_dict[park_name] = []
        park_drivers_dict[park_name].append(salary)

    res_g2_unsorted = []
    for park_name, salaries in park_drivers_dict.items():
        max_salary = max(salaries)
        res_g2_unsorted.append((park_name, max_salary))

    res_g2 = sorted(res_g2_unsorted, key=lambda x: x[1], reverse=True)
    print("Автопарки с максимальной зарплатой водителей:")
    for park, max_sal in res_g2:
        print(f"{park}: {max_sal} руб.")

    print("\nЗадание Г3")

```

```
res_g3 = sorted(many_to_many, key=lambda x: x[2])
print("Все связанные водители и автопарки:")
for driver, salary, park in res_g3:
    print(f"{park}: {driver} - {salary} руб.")

if __name__ == '__main__':
    main()
```

Экранные формы с примерами выполнения программы

Задание Г1

Автопарки, начинающиеся на 'А', и их водители:

Автопарк Северный: ['Иванов']

Автобусный парк: ['Петров']

Аэропортный автопарк: ['Антонов']

Задание Г2

Автопарки с максимальной зарплатой водителей:

Таксопарк Центральный: 55000 руб.

Аэропортный автопарк: 52000 руб.

Автопарк Северный: 50000 руб.

Грузовой парк: 47000 руб.

Автобусный парк: 45000 руб.

Задание Г3

Все связанные водители и автопарки:

Автобусный парк: Петров - 45000 руб.

Автобусный парк: Алексеев - 48000 руб.

Автопарк Северный: Иванов - 50000 руб.

Автопарк Северный: Сидоров - 55000 руб.

Аэропортный автопарк: Антонов - 52000 руб.

Грузовой парк: Александров - 47000 руб.

Таксопарк Центральный: Сидоров - 55000 руб.

Таксопарк Центральный: Алексеев - 48000 руб.