

МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
им. Н.Э. Баумана

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

Отчёт по рубежному контролю №1 по курсу
Парадигмы и конструкции языков программирования
ГУИМЦ

Исполнитель: **Колупаев Н.Н.**
студент группы
ИУ5Ц-54Б

Преподаватель: **Гапанюк Ю. Е.**

Москва, МГТУ – 2025

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;
- 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.

Варианты запросов

Мой вариант Д.

1. «Отдел» и «Сотрудник» связаны соотношением один-ко-многим. Выведите список всех сотрудников, у которых фамилия заканчивается на «ов», и названия их отделов.
2. «Отдел» и «Сотрудник» связаны соотношением один-ко-многим. Выведите список отделов со средней зарплатой сотрудников в каждом отделе, отсортированный по средней зарплате (отдельной функции вычисления среднего значения в Python нет, нужно использовать комбинацию функций вычисления суммы и количества значений).
3. «Отдел» и «Сотрудник» связаны соотношением многие-ко-многим. Выведите список всех отделов, у которых название начинается с буквы «А», и список работающих в них сотрудников.

Мой вариант предметной области:

№ варианта	Класс 1	Класс 2
30	Факультет	Университет

Текст программы:

```
from operator import itemgetter

class Факультет:
    """Факультет"""
    def __init__(self, id, название, бюджет, университет_id):
        self.id = id
        self.название = название
        self.бюджет = бюджет
        self.университет_id = университет_id

class Университет:
    """Университет"""
    def __init__(self, id, название):
        self.id = id
        self.название = название

class ФакультетУниверситет:
    """Факультеты университетов для связи многие-ко-многим"""
    def __init__(self, факультет_id, университет_id):
        self.факультет_id = факультет_id
        self.университет_id = университет_id

# Университеты
университеты = [
    Университет(1, "Академический университет"),
    Университет(2, "Московский государственный"),
    Университет(3, "Аграрный институт"),
    Университет(4, "Технический университет"),
    Университет(5, "Аналитический центр")
]

# Факультеты
факультеты = [
    Факультет(1, "Факультет информатики", 5000000, 1),
    Факультет(2, "Экономический факультет", 3000000, 1),
    Факультет(3, "Факультет Иванов", 4000000, 2),
    Факультет(4, "Юридический факультет", 2500000, 2),
    Факультет(5, "Факультет Петров", 3500000, 3),
    Факультет(6, "Аграрный факультет", 4500000, 3),
    Факультет(7, "Факультет Сидоров", 2800000, 4),
    Факультет(8, "Математический факультет", 5200000, 5)
]

# Связь многие-ко-многим
факультет_университет = [
    ФакультетУниверситет(1, 1),
    ФакультетУниверситет(2, 1),
    ФакультетУниверситет(3, 2),
    ФакультетУниверситет(4, 2),
```

```

ФакультетУниверситет(5, 3),
ФакультетУниверситет(6, 3),
ФакультетУниверситет(7, 4),
ФакультетУниверситет(8, 5),
ФакультетУниверситет(1, 3),
ФакультетУниверситет(4, 5),
ФакультетУниверситет(6, 1)
]

def main():
    # Соединение данных один-ко-многим
    один_ко_многим = [(ф.название, ф.бюджет, у.название)
                      for у in университеты
                      for ф in факультеты
                      if ф.университет_id == у.id]

    # Соединение данные многие-ко-многим
    многие_ко_многим_temp = [(у.название, fu.университет_id, fu.факультет_id)
                               for у in университеты
                               for fu in факультет_университет
                               if у.id == fu.университет_id]

    многие_ко_многим = [(ф.название, ф.бюджет, университет_название)
                         for университет_название, университет_id, факультет_id in
    многие_ко_многим_temp
                         for ф in факультеты if ф.id == факультет_id]

    print("=" * 80)
    print("РУБЕЖНЫЙ КОНТРОЛЬ №1 - ВАРИАНТ Д-30")
    print("Предметная область: Факультет и Университет")
    print("=" * 80)

    # ЗАПРОС 1: Список всех факультетов, у которых название заканчивается на
    «ов», и названия их университетов
    print("\nЗАПРОС 1: Факультеты, оканчивающиеся на 'ов', и их университеты")
    res_1 = [(факультет_название, университет_название)
              for факультет_название, _, университет_название in один_ко_многим
              if факультет_название.endswith("ов")]

    for факультет_название, университет_название in res_1:
        print(f"  Факультет: {факультет_название}, Университет:
{университет_название}")

    if not res_1:
        print("  Факультетов с названием, оканчивающимся на 'ов', не найдено")

    # ЗАПРОС 2: Список университетов со средним бюджетом факультетов в каждом
    университете, отсортированный по среднему бюджету
    print("\nЗАПРОС 2: Университеты со средним бюджетом факультетов
(отсортировано)")

```

```

университет_статистика = {}
for факультет_название, бюджет, университет_название in один_ко_многим:
    if университет_название not in университет_статистика:
        университет_статистика[университет_название] = {'сумма': 0,
'количество': 0}
        университет_статистика[университет_название]['сумма'] += бюджет
        университет_статистика[университет_название]['количество'] += 1

res_2_unsorted = []
for университет_название, stats in университет_статистика.items():
    средний_бюджет = stats['сумма'] / stats['количество']
    res_2_unsorted.append((университет_название, средний_бюджет))

res_2 = sorted(res_2_unsorted, key=itemgetter(1))

for университет_название, средний_бюджет in res_2:
    print(f" Университет: {университет_название}, Средний бюджет:
{средний_бюджет:.2f} руб.")

# ЗАПРОС 3: Список всех университетов, у которых название начинается с буквы
«А», и список работающих в них факультетов
print("\nЗАПРОС 3: Университеты, начинающиеся на 'A', и их факультеты")

a_университеты_названия = [у.название for у in университеты if
у.название.startswith("A")]

res_3 = {}
for университет_название in a_университеты_названия:
    университет_факультеты = list(filter(lambda x: x[2] ==
университет_название, многие_ко_многим))
    уникальные_факультеты = []
    просмотренные_названия = set()
    for факультет in университет_факультеты:
        факультет_название, факультет_бюджет, _ = факультет
        if факультет_название not in просмотренные_названия:
            уникальные_факультеты.append((факультет_название,
факультет_бюджет))
            просмотренные_названия.add(факультет_название)
    res_3[университет_название] = уникальные_факультеты

for университет_название, факультеты_список in res_3.items():
    print(f"\n Университет: {университет_название}")
    for факультет_название, факультет_бюджет in факультеты_список:
        print(f"      - {факультет_название}: {факультет_бюджет:,} руб.")

if __name__ == "__main__":
    main()

```

Результат выполнения программы:

РУБЕЖНЫЙ КОНТРОЛЬ №1 - ВАРИАНТ Д-30

Предметная область: Факультет и Университет

ЗАПРОС 1: Факультеты, оканчивающиеся на 'ов', и их университеты

Факультет: Факультет Иванов, Университет: Московский государственный

Факультет: Факультет Петров, Университет: Аграрный институт

Факультет: Факультет Сидоров, Университет: Технический университет

ЗАПРОС 2: Университеты со средним бюджетом факультетов (отсортировано)

Университет: Технический университет, Средний бюджет: 2,800,000.00 руб.

Университет: Московский государственный, Средний бюджет: 3,250,000.00 руб.

Университет: Академический университет, Средний бюджет: 4,000,000.00 руб.

Университет: Аграрный институт, Средний бюджет: 4,000,000.00 руб.

Университет: Аналитический центр, Средний бюджет: 5,200,000.00 руб.

ЗАПРОС 3: Университеты, начинающиеся на 'А', и их факультеты

Университет: Академический университет

- Факультет информатики: 5,000,000 руб.

- Экономический факультет: 3,000,000 руб.

- Аграрный факультет: 4,500,000 руб.

Университет: Аграрный институт

- Факультет Петров: 3,500,000 руб.

- Аграрный факультет: 4,500,000 руб.

- Факультет информатики: 5,000,000 руб.

Университет: Аналитический центр

- Математический факультет: 5,200,000 руб.

- Юридический факультет: 2,500,000 руб.