

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

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

Курс «Парадигмы и конструкции языков программирования»

Отчет по рубежному контролю №1

Вариант А 29

Выполнил:
студент группы ИУ5 - 32Б:
Васильев Н. Д.
Подпись и дата:

Проверил:
преподаватель каф. ИУ5
Гапанюк Ю. Е.
Подпись и дата:

Москва, 2025 г.

Задание:

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.

Код:

```
from operator import itemgetter

class Department:
    """Кафедра"""

    def __init__(self, id, name, faculty_id):
        self.id = id
        self.name = name
        self.faculty_id = faculty_id

class Faculty:
    """Факультет"""

    def __init__(self, id, name, budget):
        self.id = id
        self.name = name
        self.budget = budget

class FacultyDepartment:
    """Связь многие-ко-многим между факультетами и Кафедрами"""

    def __init__(self, faculty_id, department_id):
        self.faculty_id = faculty_id
        self.department_id = department_id

def main():
    # Тестовые данные - факультеты
    faculties = [
        Faculty(1, 'Факультет компьютерных наук', 5000000),
        Faculty(2, 'Факультет экономики', 4500000),
        Faculty(3, 'Факультет лингвистики', 4000000),
        Faculty(4, 'Факультет математики', 4800000),
    ]

    # Тестовые данные - Кафедры
    departments = [
        Department(1, 'Кафедра математического анализа', 4),
        Department(2, 'Кафедра вычислительной техники', 1),
        Department(3, 'Кафедра иностранных языков', 3),
        Department(4, 'Кафедра физики', 4),
        Department(5, 'Кафедра информационных систем', 1),
        Department(6, 'Кафедра программной инженерии', 1),
    ]
```

```

# Тестовые данные - связи многие-ко-многим
faculties_departments = [
    FacultyDepartment(1, 2),
    FacultyDepartment(1, 5),
    FacultyDepartment(1, 6),
    FacultyDepartment(2, 1),
    FacultyDepartment(2, 3),
    FacultyDepartment(3, 3),
    FacultyDepartment(4, 1),
    FacultyDepartment(4, 4),
    FacultyDepartment(4, 5),
]

print("=" * 60)
print("РУБЕЖНЫЙ КОНТРОЛЬ №1")
print("Вариант 29: Кафедра - Факультет")
print("=" * 60)

# Соединение данных один-ко-многим
one_to_many = [
    (d.name, d.faculty_id, f.name, f.budget)
    for f in faculties
    for d in departments
    if d.faculty_id == f.id
]

# Соединение данных многие-ко-многим
many_to_many_temp = [
    (f.name, fd.faculty_id, fd.department_id)
    for f in faculties
    for fd in faculties_departments
    if f.id == fd.faculty_id
]

many_to_many = [
    (d.name, faculty_name)
    for faculty_name, faculty_id, department_id in many_to_many_temp
    for d in departments if d.id == department_id
]

print("\nЗАДАНИЕ A1")
print("Список всех связанных кафедр и факультетов (один-ко-многим),")
print("отсортированный по факультетам:")
print("-" * 50)

res_a1 = sorted(one_to_many, key=itemgetter(2)) # Сортировка по названию
факультета
for dep_name, _, fac_name, budget in res_a1:
    print(f"{fac_name:.<30} {dep_name} (бюджет: {budget:,>10} руб.)")

print("\nЗАДАНИЕ A2")
print("Список факультетов с средним бюджетом на кафедре,")
print("отсортированный по убыванию среднего бюджета:")
print("-" * 50)

res_a2_unsorted = []
for f in faculties:
    # Находим кафедры факультета
    f_deps = list(filter(lambda i: i[2] == f.name, one_to_many))
    if f_deps:
        # Вычисляем средний бюджет на кафедре
        avg_budget = f.budget / len(f_deps)
        res_a2_unsorted.append((f.name, avg_budget, len(f_deps)))

```

```

# Сортировка по убыванию среднего бюджета
res_a2 = sorted(res_a2_unsorted, key=itemgetter(1), reverse=True)

for fac_name, avg_budget, dep_count in res_a2:
    print(f"{fac_name:.<30} {avg_budget:>12,.2f} руб. ({dep_count} кафедр)")

print("\nЗАДАНИЕ A3")
print("Список всех кафедр, у которых в названии есть 'кафедра',")
print("и факультетов, с которыми они связаны:")
print("-" * 50)

res_a3 = {}
for d in departments:
    if 'кафедра' in d.name.lower():
        # Находим факультеты для этой кафедры
        d_faculties = list(filter(lambda i: i[0] == d.name, many_to_many))
        if d_faculties:
            faculty_names = [faculty for _, faculty in d_faculties]
            res_a3[d.name] = faculty_names

for dep_name, faculty_list in res_a3.items():
    print(f"\n{dep_name}:")
    for faculty in faculty_list:
        print(f"    └─ {faculty}")

print("\n" + "=" * 60)
print("ВЫПОЛНЕНИЕ ЗАВЕРШЕНО")
print("=" * 60)

if __name__ == '__main__':
    main()

```

Скриншот работы:

```

PS C:\Users\Николай\PycharmProjects> cd RK1-3Sem
PS C:\Users\Николай\PycharmProjects\RK1-3Sem> python main.py
=====
РУБЕЖНЫЙ КОНТРОЛЬ №1
Вариант 29: Кафедра - Факультет
=====

ЗАДАНИЕ A1
Список всех связанных кафедр и факультетов (один-ко-многим),
отсортированный по факультетам:
-----
Факультет компьютерных наук... Кафедра вычислительной техники (бюджет: 5,000,000 руб.)
Факультет компьютерных наук... Кафедра информационных систем (бюджет: 5,000,000 руб.)
Факультет компьютерных наук... Кафедра программной инженерии (бюджет: 5,000,000 руб.)
Факультет лингвистики..... Кафедра иностранных языков (бюджет: 4,000,000 руб.)
Факультет математики..... Кафедра математического анализа (бюджет: 4,800,000 руб.)
Факультет математики..... Кафедра физики (бюджет: 4,800,000 руб.)

ЗАДАНИЕ A2
Список факультетов с средним бюджетом на кафедре,
отсортированный по убыванию среднего бюджета:
-----
Факультет лингвистики..... 4,000,000.00 руб. (1 кафедр)
Факультет математики..... 2,400,000.00 руб. (2 кафедр)
Факультет компьютерных наук... 1,666,666.67 руб. (3 кафедр)

```

Факультет компьютерных наук... 1,666,666.67 руб. (3 кафедр)

ЗАДАНИЕ А3

Список всех кафедр, у которых в названии есть 'кафедра',
и факультетов, с которыми они связаны:

Кафедра математического анализа:

- └─ Факультет экономики
- └─ Факультет математики

Кафедра вычислительной техники:

- └─ Факультет компьютерных наук

Кафедра иностранных языков:

- └─ Факультет экономики
- └─ Факультет лингвистики

Кафедра физики:

- └─ Факультет математики

Кафедра информационных систем:

- └─ Факультет компьютерных наук
- └─ Факультет математики

Кафедра программной инженерии:

- └─ Факультет компьютерных наук

=====

ВЫПОЛНЕНИЕ ЗАВЕРШЕНО

=====