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

Выполнил:
студент группы ИУ5-36Б
Бойко Артем Андреевич
Проверил:
Нардид Анатолий Николаевич

Москва, 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;
● 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.

Мой вариант запросов - Вариант Г.

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

Мои варианты предметной области - Вариант 28 (Студенческая группа, Кафедра)

Текст программы

```
class Department:
    """Класс Кафедра"""
    def __init__(self, id, name):
        self.id = id
        self.name = name

    def __repr__(self):
        return f"Department(id={self.id}, name='{self.name}')"

class StudentGroup:
    """Класс Студенческая группа"""
    def __init__(self, id, group_name, student_count, department_id):
        self.id = id
        self.group_name = group_name
        self.student_count = student_count  # количественный признак - количество
студентов
        self.department_id = department_id

    def __repr__(self):
        return f"StudentGroup(id={self.id}, group_name='{self.group_name}',
student_count={self.student_count}, department_id={self.department_id})"

class GroupDepartment:
    """Класс для связи многие-ко-многим между Студенческой группой и Кафедрой"""
    def __init__(self, group_id, department_id):
        self.group_id = group_id
        self.department_id = department_id

    def __repr__(self):
        return f"GroupDepartment(group_id={self.group_id},
department_id={self.department_id})"

def main():
    # 2) Создание тестовых данных

    # Создаем кафедры
    departments = [
        Department(1, "Алгебры и геометрии"),
        Department(2, "Высшей математики"),
        Department(3, "Аналитической химии"),
        Department(4, "Физики"),
        Department(5, "Английского языка"),
        Department(6, "Астрономии")
    ]

    # Создаем студенческие группы (связь один-ко-многим)
    student_groups = [
```

```
            StudentGroup(1, "МАТ-101", 25, 1),
            StudentGroup(2, "МАТ-102", 30, 1),
            StudentGroup(3, "ВМ-201", 28, 2),
            StudentGroup(4, "ХИМ-301", 22, 3),
            StudentGroup(5, "АНГ-501", 35, 5),
            StudentGroup(6, "АНГ-502", 32, 5),
            StudentGroup(7, "АСТ-601", 18, 6)
        ]

        # Создаем связи многие-ко-многим
        group_departments = [
            GroupDepartment(1, 1),  # МАТ-101 - Алгебры и геометрии
            GroupDepartment(2, 1),  # МАТ-102 - Алгебры и геометрии
            GroupDepartment(3, 2),  # ВМ-201 - Высшей математики
            GroupDepartment(4, 3),  # ХИМ-301 - Аналитической химии
            GroupDepartment(5, 5),  # АНГ-501 - Английского языка
            GroupDepartment(6, 5),  # АНГ-502 - Английского языка
            GroupDepartment(7, 6),  # АСТ-601 - Астрономии
            # Дополнительные связи для демонстрации многие-ко-многим
            GroupDepartment(1, 2),  # МАТ-101 также относится к Высшей математики
            GroupDepartment(3, 1),  # ВМ-201 также относится к Алгебры и геометрии
            GroupDepartment(5, 1),  # АНГ-501 также относится к Алгебры и геометрии
        ]

        print("=" * 80)
        print("ТЕСТОВЫЕ ДАННЫЕ")
        print("=" * 80)

        print("\nКафедры:")
        for dept in departments:
            print(f"  {dept}")

        print("\nСтуденческие группы:")
        for group in student_groups:
            print(f"  {group}")

        print("\nСвязи группы-кафедры:")
        for gd in group_departments:
            print(f"  {gd}")

        # 3) Выполнение запросов

        print("\n" + "=" * 80)
        print("ЗАПРОС №1")
        print("=" * 80)
        print("Список всех кафедр, у которых название начинается с буквы «А», ")
        print("и список студенческих групп, относящихся к ним:")

        # Запрос 1: Кафедры на "А" и их студенческие группы
        departments_with_a = [dept for dept in departments if
    dept.name.startswith('А')]
```

```
    for dept in departments_with_a:
        dept_groups = [group for group in student_groups if group.department_id
== dept.id]
        print(f"\nКафедра: {dept.name}")
        if dept_groups:
            for group in dept_groups:
                print(f"  - Группа {group.group_name} ({group.student_count}
студентов)")
        else:
            print("  Нет студенческих групп")

    print("\n" + "=" * 80)
    print("ЗАПРОС №2")
    print("=" * 80)
    print("Список кафедр с максимальным количеством студентов в группах")
    print("в каждой кафедре, отсортированный по максимальному количеству
студентов:")

    # Запрос 2: Кафедры с максимальным количеством студентов, отсортированные по
количеству
    from collections import defaultdict

    # Группируем студенческие группы по кафедрам
    groups_by_department = defaultdict(list)
    for group in student_groups:
        groups_by_department[group.department_id].append(group)

    # Находим максимальное количество студентов для каждой кафедры
    department_max_students = []
    for dept_id, dept_groups in groups_by_department.items():
        if dept_groups:
            max_students = max(group.student_count for group in dept_groups)
            dept_name = next(dept.name for dept in departments if dept.id ==
dept_id)
            department_max_students.append((dept_name, max_students))

    # Сортируем по максимальному количеству студентов (по убыванию)
    department_max_students.sort(key=lambda x: x[1], reverse=True)

    for dept_name, max_students in department_max_students:
        print(f"Кафедра '{dept_name}': максимальное количество студентов =
{max_students}")

    print("\n" + "=" * 80)
    print("ЗАПРОС №3")
    print("=" * 80)
    print("Список всех связанных студенческих групп и кафедр (многие-ко-
многим),")
    print("отсортированный по кафедрам, сортировка по группам произвольная:")
```

```
    # Запрос 3: Все связи группы-кафедры, отсортированные по кафедрам
    # Создаем словари для быстрого доступа
    group_dict = {group.id: group for group in student_groups}
    department_dict = {dept.id: dept for dept in departments}

    # Группируем связи по кафедрам
    connections_by_department = defaultdict(list)
    for gd in group_departments:
        group = group_dict.get(gd.group_id)
        department = department_dict.get(gd.department_id)
        if group and department:
            connections_by_department[department].append(group)

    # Сортируем кафедры по названию
    sorted_departments = sorted(connections_by_department.keys(), key=lambda x:
x.name)

    for department in sorted_departments:
        department_groups = connections_by_department[department]
        print(f"\nКафедра: {department.name}")
        for group in department_groups:
            print(f"  - Группа {group.group_name} ({group.student_count}
студентов)")

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

Результаты выполнения программы

```
● @artem-hedgehog →/workspaces/ProgLangParadigms_BoikoA_2025 (main) $ python RK_1.py
================================================================================
ТЕСТОВЫЕ ДАННЫЕ
================================================================================

Кафедры:
  Department(id=1, name='Алгебры и геометрии')
  Department(id=2, name='Высшей математики')
  Department(id=3, name='Аналитической химии')
  Department(id=4, name='Физики')
  Department(id=5, name='Английского языка')
  Department(id=6, name='Астрономии')

Студенческие группы:
  StudentGroup(id=1, group_name='МАТ-101', student_count=25, department_id=1)
  StudentGroup(id=2, group_name='МАТ-102', student_count=30, department_id=1)
  StudentGroup(id=3, group_name='ВМ-201', student_count=28, department_id=2)
  StudentGroup(id=4, group_name='ХИМ-301', student_count=22, department_id=3)
  StudentGroup(id=5, group_name='АНГ-501', student_count=35, department_id=5)
  StudentGroup(id=6, group_name='АНГ-502', student_count=32, department_id=5)
  StudentGroup(id=7, group_name='АСТ-601', student_count=18, department_id=6)

Связи группы-кафедры:
  GroupDepartment(group_id=1, department_id=1)
  GroupDepartment(group_id=2, department_id=1)
  GroupDepartment(group_id=3, department_id=2)
  GroupDepartment(group_id=4, department_id=3)
  GroupDepartment(group_id=5, department_id=5)
  GroupDepartment(group_id=6, department_id=5)
  GroupDepartment(group_id=7, department_id=6)
  GroupDepartment(group_id=1, department_id=2)
  GroupDepartment(group_id=3, department_id=1)
  GroupDepartment(group_id=5, department_id=1)
```

```
================================================================================
ЗАПРОС №1
================================================================================
Список всех кафедр, у которых название начинается с буквы «А»,
и список студенческих групп, относящихся к ним:

Кафедра: Алгебры и геометрии
  - Группа МАТ-101 (25 студентов)
  - Группа МАТ-102 (30 студентов)

Кафедра: Аналитической химии
  - Группа ХИМ-301 (22 студентов)

Кафедра: Английского языка
  - Группа АНГ-501 (35 студентов)
  - Группа АНГ-502 (32 студентов)

Кафедра: Астрономии
  - Группа АСТ-601 (18 студентов)

================================================================================
ЗАПРОС №2
================================================================================
Список кафедр с максимальным количеством студентов в группах
в каждой кафедре, отсортированный по максимальному количеству студентов:
Кафедра 'Английского языка': максимальное количество студентов = 35
Кафедра 'Алгебры и геометрии': максимальное количество студентов = 30
Кафедра 'Высшей математики': максимальное количество студентов = 28
Кафедра 'Аналитической химии': максимальное количество студентов = 22
Кафедра 'Астрономии': максимальное количество студентов = 18

================================================================================
ЗАПРОС №3
================================================================================
Список всех связанных студенческих групп и кафедр (многие-ко-многим),
отсортированный по кафедрам, сортировка по группам произвольная:

Кафедра: Алгебры и геометрии
  - Группа МАТ-101 (25 студентов)
  - Группа МАТ-102 (30 студентов)
  - Группа ВМ-201 (28 студентов)
  - Группа АНГ-501 (35 студентов)

Кафедра: Аналитической химии
  - Группа ХИМ-301 (22 студентов)

Кафедра: Английского языка
  - Группа АНГ-501 (35 студентов)
  - Группа АНГ-502 (32 студентов)

Кафедра: Астрономии
  - Группа АСТ-601 (18 студентов)

Кафедра: Высшей математики
  - Группа ВМ-201 (28 студентов)
  - Группа МАТ-101 (25 студентов)
```