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: **Кузнецов Н. В.**
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
31	Таблица данных	База данных

Текст программы:

```
# используется для сортировки
from operator import itemgetter

class Table:
    """Таблица данных"""
    def __init__(self, id, name, row_count, db_id):
        self.id = id
        self.name = name
        self.row_count = row_count  # количественный признак
        self.db_id = db_id

class Database:
    """База данных"""
    def __init__(self, id, name):
        self.id = id
        self.name = name

class TableDatabase:
    """Связь многие-ко-многим между таблицами и базами данных"""
    def __init__(self, db_id, table_id):
        self.db_id = db_id
        self.table_id = table_id

# Базы данных
dbs = [
    Database(1, 'Активная база'),
    Database(2, 'Архивная база данных'),
    Database(3, 'Тестовая'),
    Database(4, 'Аналитическая база данных'),
    Database(5, 'Основная база'),
]

# Таблицы
tables = [
    Table(1, 'Пользователей', 1000, 1),
    Table(2, 'Заказов', 5000, 2),
    Table(3, 'Товаров', 800, 2),
    Table(4, 'Аналитика продаж', 300, 4),
    Table(5, 'Логов', 1200, 5),
]

# Связи многие-ко-многим
tables_dbs = [
    TableDatabase(1, 1),
    TableDatabase(2, 2),
    TableDatabase(2, 3),
    TableDatabase(4, 4),
    TableDatabase(5, 5),
    TableDatabase(1, 2),
```

```
    TableDatabase(2, 1),
]

def main():
    """Основная функция"""

    # Соединение данных один-ко-многим
    one_to_many = [(t.name, t.row_count, d.name)
                   for d in dbs
                   for t in tables
                   if t.db_id == d.id]

    # Соединение данных многие-ко-многим
    many_to_many_temp = [(d.name, td.db_id, td.table_id)
                         for d in dbs
                         for td in tables_dbs
                         if d.id == td.db_id]

    many_to_many = [(t.name, t.row_count, db_name)
                    for db_name, db_id, table_id in many_to_many_temp
                    for t in tables if t.id == table_id]

    print('Задание Д1')
    # Список всех таблиц, у которых название заканчивается на "ов", и названия их
БД
    res_1 = list(filter(lambda i: i[0].endswith('ов'), one_to_many))
    print(res_1)

    print('\nЗадание Д2')
    # Список БД со средней числом строк в таблицах в каждой БД, отсортированный
по средней
    res_2_unsorted = []
    for d in dbs:
        # Список таблиц БД
        d_tables = list(filter(lambda i: i[2] == d.name, one_to_many))
        if len(d_tables) > 0:
            # Количество строк таблиц БД
            d_rows = [row_count for _, row_count, _ in d_tables]
            # Среднее число строк (с округлением до 2 знаков)
            d_rows_avg = round(sum(d_rows) / len(d_rows), 2)
            res_2_unsorted.append((d.name, d_rows_avg))
    # Сортировка по средней числу строк
    res_2 = sorted(res_2_unsorted, key=itemgetter(1))
    print(res_2)

    print('\nЗадание Д3')
    # Список всех БД, у которых название начинается с буквы "А", и список их
таблиц
    res_3 = {}
    for d in dbs:
        if d.name.startswith('A'):
```

```
            # Список таблиц БД
            d_tables = list(filter(lambda i: i[2] == d.name, many_to_many))
            # Только названия таблиц
            d_tables_names = [name for name, _, _ in d_tables]
            res_3[d.name] = d_tables_names
    print(res_3)

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

Результат выполнения программы:

Задание Д1

[('Заказов', 5000, 'Архивная база данных'), ('Товаров', 800, 'Архивная база данных'), ('Логов', 1200, 'Основная база')]

Задание Д2

[('Аналитическая база данных', 300.0), ('Активная база', 1000.0), ('Основная база', 1200.0), ('Архивная база данных', 2900.0)]

Задание Д3

{'Активная база': ['Пользователей', 'Заказов'], 'Архивная база данных': ['Заказов', 'Товаров', 'Пользователей'], 'Аналитическая база данных': ['Аналитика продаж']}