

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 № 31

Вариант запросов В

Выполнил:

Студент группы ИУ5-34Б

Солижонов У.И.

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 одинко-многим и многие-ко-многим.
- 2) 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

Классы: «Синтаксическая конструкция» и «Язык программирования»

Вариант В.

1. «Синтаксическая конструкция» и «Язык программирования» связаны соотношением один-ко-многим. Выведите список всех конструкций, у которых название начинается с буквы «А», и названия их языков программирования.
2. «Синтаксическая конструкция» и «Язык программирования» связаны соотношением один-ко-многим. Выведите список языков с минимальным временем компиляции конструкции в каждом языке программирования, отсортированный по минимальному времени компиляции.
3. «Синтаксическая конструкция» и «Язык программирования» связаны соотношением многие-ко-многим. Выведите список всех связанных конструкций и языков, отсортированный по конструкциям, сортировка по языкам программирования произвольная.

Код:

```
from operator import itemgetter
```

```
class Table:
```

```
    """Таблица данных (аналог Сотрудника)"""
```

```
    def __init__(self, id, name, row_count, db_id):
```

```
        self.id = id
```

```
        self.name = name          # аналог фамилии сотрудника
```

```
        self.row_count = row_count # аналог зарплаты
```

```
    (количественный признак)
```

```
        self.db_id = db_id       # для связи один-ко-многим
```

```

class Database:
    """База данных (аналог Отдела)"""
    def __init__(self, id, name):
        self.id = id
        self.name = name        # аналог названия отдела

class TableDatabase:
    """Связь многие-ко-многим"""
    def __init__(self, db_id, table_id):
        self.db_id = db_id
        self.table_id = table_id

# Данные для тестирования
dbs = [
    Database(1, 'AnalyticsDB'),
    Database(2, 'ArchiveDB'),
    Database(3, 'AccountingDB'),
]

tables = [
    Table(1, 'Анализ_продаж', 15000, 1),
    Table(2, 'Архив_данных', 8000, 2),
    Table(3, 'Активы', 12000, 3),
    Table(4, 'Расходы', 9500, 3),
    Table(5, 'Отчеты', 7000, 2),
]

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

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

    print("ЗАДАНИЕ В1")

```

```
print('Таблицы, начинающиеся с "А", и их БД (связь один-  
ко-многим):')
```

```
# Связь один-ко-многим для заданий В1 и В2  
one_to_many = [(t, d) for d in dbs for t in tables if t.db_id ==  
d.id]
```

```
# В1: Фильтруем по имени, начинающемуся с "А"  
result1 = [(t.name, d.name) for t, d in one_to_many if  
t.name.startswith('А')]  
result1_sorted = sorted(result1, key=itemgetter(0))
```

```
for table_name, db_name in result1_sorted:  
    print(f' Таблица: {table_name} | База данных: {db_name}')
```

```
print('\nЗАДАНИЕ В2')  
print('БД с минимальным количеством строк (связь один-ко-  
многим):')
```

```
result2 = []  
for d in dbs:  
    # Находим все таблицы этой БД (связь один-ко-многим)  
    d_tables = [t for t in tables if t.db_id == d.id]  
    if d_tables:  
        min_rows = min(t.row_count for t in d_tables)  
        result2.append((d.name, min_rows))
```

```
# Сортировка по минимальному количеству строк  
result2_sorted = sorted(result2, key=itemgetter(1))
```

```
for db_name, min_rows in result2_sorted:  
    print(f' БД: {db_name} | Мин. строк: {min_rows}')
```

```
print('\nЗАДАНИЕ В3')  
print('Все связи таблиц и БД (связь многие-ко-многим,  
сортировка по таблицам):')
```

```
# Связь многие-ко-многим для задания В3  
many_to_many = []  
for td in tables_dbs:  
    table = next(t for t in tables if t.id == td.table_id)
```

```

db = next(d for d in dbs if d.id == td.db_id)
many_to_many.append((table.name, db.name))

# Сортировка по имени таблицы (сотрудника)
result3_sorted = sorted(many_to_many, key=itemgetter(0))

for table_name, db_name in result3_sorted:
    print(f' Таблица: {table_name} | База данных: {db_name}')

if __name__ == '__main__':
    main()

```

Результат Выполнения:

ЗАДАНИЕ В1

Таблицы, начинающиеся с "А", и их БД (связь один-ко-многим):

Таблица: Активы | База данных: AccountingDB

Таблица: Анализ_продаж | База данных: AnalyticsDB

Таблица: Архив_данных | База данных: ArchiveDB

ЗАДАНИЕ В2

БД с минимальным количеством строк (связь один-ко-многим):

БД: ArchiveDB | Мин. строк: 7000

БД: AccountingDB | Мин. строк: 9500

БД: AnalyticsDB | Мин. строк: 15000

ЗАДАНИЕ В3

Все связи таблиц и БД (связь многие-ко-многим, сортировка по таблицам):

Таблица: Активы | База данных: AccountingDB

Таблица: Анализ_продаж | База данных: AnalyticsDB

Таблица: Архив_данных | База данных: ArchiveDB

Таблица: Отчеты | База данных: ArchiveDB

Таблица: Расходы | База данных: AccountingDB