

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

Факультет ИУ
Кафедра ИУ5

Курс «Парадигмы и конструкции языков программирования»

Отчет по рубежному контролю №1

Вариант Б34

Выполнил:
Студент группы ИУ5-32Б
Девятов И.А.
Подпись и дата:

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

Москва, 2025

Текст программы

```
class Procedure:
    def __init__(self, id, name, size, database_id):
        self.id = id
        self.name = name
        self.size = size
        self.database_id = database_id
```

```
class Database:
    def __init__(self, id, name):
        self.id = id
        self.name = name
```

```
class ProcedureDatabase:
    def __init__(self, procedure_id, database_id):
        self.procedure_id = procedure_id
        self.database_id = database_id
```

```
# --- ТЕСТОВЫЕ ДАННЫЕ ---
```

```
databases = [
    Database(1, "HR_DB"),
    Database(2, "Finance_DB"),
    Database(3, "Analytics_DB"),
    Database(4, "Common_DB"),
]
```

```
procedures = [
    Procedure(1, "GetUsers", 120, 1),
    Procedure(2, "CalcSalary", 180, 2),
    Procedure(3, "ReportGen", 90, 3),
    Procedure(4, "LogCleanup", 60, 4),
    Procedure(5, "BackupProc", 150, 4),
    Procedure(6, "DataMov", 200, 3),
]
```

```
procedures_databases = [
    ProcedureDatabase(1, 1),
    ProcedureDatabase(2, 2),
    ProcedureDatabase(3, 3),
    ProcedureDatabase(4, 4),
    ProcedureDatabase(5, 4),
    ProcedureDatabase(6, 3),
    ProcedureDatabase(3, 4),
    ProcedureDatabase(4, 1),
]
```

```
def main():
    # --- 1) ОДИН-КО-МНОГИМ ---
```

```

one_to_many = [
    (p.name, p.size, d.name)
    for d in databases
    for p in procedures
    if p.database_id == d.id
]

print("--- Запрос Б1 ---")
print("Список связанных процедур и баз данных (1:M), отсортированный по имени процедуры:")
res1 = sorted(one_to_many, key=lambda x: x[0])
for proc, size, db in res1:
    print(f" Процедура: {proc}, Размер: {size} строк, База: {db}")

# --- 2) Количество процедур в каждой базе ---
print("\n--- Запрос Б2 ---")
print("Список баз данных с количеством процедур, отсортированный по количеству:")

res2 = []
for d in databases:
    count = len(list(filter(lambda x: x[2] == d.name, one_to_many)))
    if count > 0:
        res2.append((d.name, count))

res2.sort(key=lambda x: x[1])
for name, count in res2:
    print(f" База данных: {name}, Количество процедур: {count}")

# --- 3) Многие-ко-многим ---
print("\n--- Запрос Б3 ---")
print("Список процедур, название которых заканчивается на 'Proc', и базы данных (M:M):")

many_to_many_temp = [
    (d.name, pd.database_id, pd.procedure_id)
    for d in databases
    for pd in procedures_databases
    if d.id == pd.database_id
]

many_to_many = [
    (p.name, db_name)
    for db_name, db_id, proc_id in many_to_many_temp
    for p in procedures
    if p.id == proc_id
]

res3 = sorted(
    [(name, db) for name, db in many_to_many if name.endswith("Proc")],
    key=lambda x: x[0]
)

```

```
for proc, db in res3:
    print(f" Процедура: {proc}, База данных: {db}")

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

Скриншот работы программы

```
deviatovilya@MacBook-Air-cua-Deviatov Labs-2025-python % python -u "/Users/deviato
--- Запрос Б1 ---
Список связанных процедур и баз данных (1:M), отсортированный по имени процедуры:
Процедура: BackupProc, Размер: 150 строк, База: Common_DB
Процедура: CalcSalary, Размер: 180 строк, База: Finance_DB
Процедура: DataMov, Размер: 200 строк, База: Analytics_DB
Процедура: GetUsers, Размер: 120 строк, База: HR_DB
Процедура: LogCleanup, Размер: 60 строк, База: Common_DB
Процедура: ReportGen, Размер: 90 строк, База: Analytics_DB

--- Запрос Б2 ---
Список баз данных с количеством процедур, отсортированный по количеству:
База данных: HR_DB, Количество процедур: 1
База данных: Finance_DB, Количество процедур: 1
База данных: Analytics_DB, Количество процедур: 2
База данных: Common_DB, Количество процедур: 2

--- Запрос Б3 ---
Список процедур, название которых заканчивается на 'Proc', и базы данных (M:M):
Процедура: BackupProc, База данных: Common_DB
deviatovilya@MacBook-Air-cua-Deviatov Labs-2025-python %
```

Рис. 1 Результат работы программы