



Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное
учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

Факультет «Информатика, искусственный интеллект и системы управления»
Кафедра «Системы обработки информации и управления»

Рубежный контроль 1
по дисциплине «Парадигмы и конструкции языков программирования»

Вариант: 33
Вариант запросов: Б

Выполнил:
студент ИУ5-33б Левкович Л.А.

2025 г.

Код решения на Python:

```
from dataclasses import dataclass
from typing import List, Dict, Tuple


@dataclass(frozen=True)
class DataRow:
    """Строка данных (одна запись)."""
    id: int
    text: str
    length: int
    table_id: int


@dataclass(frozen=True)
class DataTable:
    """Таблица данных (контейнер строк)."""
    id: int
    name: str


@dataclass(frozen=True)
class RowTableLink:
    """Связь многие-ко-многим: строка может входить в несколько таблиц."""
    row_id: int
    table_id: int


tables: List[DataTable] = [
    DataTable(1, "Пользователи"),
    DataTable(2, "Заказы"),
    DataTable(3, "Логи"),
]
rows: List[DataRow] = [
    DataRow(1, "Иванов", 6, 1),
    DataRow(2, "Петров", 6, 1),
    DataRow(3, "Заказ #1024", 10, 2),
    DataRow(4, "Событие: вход", 13, 3),
    DataRow(5, "Сидоров", 7, 1),
]
rows_tables: List[RowTableLink] = [
    RowTableLink(1, 1),
    RowTableLink(1, 3),
    RowTableLink(2, 1),
    RowTableLink(3, 2),
    RowTableLink(4, 3),
    RowTableLink(5, 1),
    RowTableLink(5, 2),
]

table_by_id: Dict[int, DataTable] = {t.id: t for t in tables}
row_by_id: Dict[int, DataRow] = {r.id: r for r in rows}

q1: List[Tuple[str, str]] = sorted(
    [(r.text, table_by_id[r.table_id].name) for r in rows],
    key=lambda x: x[0].lower()
)

print("ЗАПРОС 1: Строки и их таблицы (1-М), сортировка по строкам")
for text, table_name in q1:
    print(f"- {text} - {table_name}")
```

```

print()

counts: Dict[int, int] = {}
for r in rows:
    counts[r.table_id] = counts.get(r.table_id, 0) + 1

q2: List[Tuple[str, int]] = sorted(
    [(table_by_id[table_id].name, cnt) for table_id, cnt in counts.items()],
    key=lambda x: x[1]
)

print("ЗАПРОС 2: Таблицы и количество строк (1-М), сортировка по количеству")
for table_name, cnt in q2:
    print(f"- {table_name}: {cnt}")
print()

pairs_mm: List[Tuple[str, str]] = [
    (row_by_id[link.row_id].text, table_by_id[link.table_id].name)
    for link in rows_tables
    if link.row_id in row_by_id and link.table_id in table_by_id
]

q3: List[Tuple[str, str]] = sorted(
    [(text, table_name) for text, table_name in pairs_mm if
     text.endswith("ов")],
    key=lambda x: (x[0].lower(), x[1].lower())
)

print('ЗАПРОС 3: Строки, оканчивающиеся на "ов", и таблицы (М-М)')
for text, table_name in q3:
    print(f"- {text} - {table_name}")

```

Результаты выполнения:

ЗАПРОС 1: Строки и их таблицы (1-М), сортировка по строкам

- Заказ #1024 – Заказы
- Иванов – Пользователи
- Петров – Пользователи
- Сидоров – Пользователи
- Событие: вход – Логи

ЗАПРОС 2: Таблицы и количество строк (1-М), сортировка по количеству

- Заказы: 1
- Логи: 1
- Пользователи: 3

ЗАПРОС 3: Строки, оканчивающиеся на "ов", и таблицы (М-М)

- Иванов – Логи
- Иванов – Пользователи
- Петров – Пользователи
- Сидоров – Заказы
- Сидоров – Пользователи