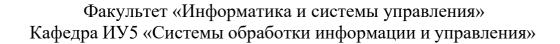
# Московский государственный технический университет им. Н.Э. Баумана



Курс «Парадигмы и конструкции языков программирования»

Отчет по рубежному контролю № 1 Вариант В-33

Выполнил: студент группы ИУ5-34Б Давшиц С. А. Проверил: преподаватель каф. ИУ5 Нардид А.Н.

## Текст программы

main.py

```
class DataRow:
    """Строка данных"""
    def init (self, id, content, table id):
        \overline{\text{self.id}} = \text{id}
        self.content = content
        self.table id = table id
class DataTable:
    """Таблица данных"""
    def init (self, id, name):
        \overline{\text{self.id}} = \text{id}
        self.name = name
class DataRowTable:
    """'Строка данных в таблице' для реализации связи многие-ко-многим"""
    def __init__(self, table_id, row_id):
        self.table_id = table_id
        self.row id = row id
# Таблицы данных
tables = [
    DataTable(1, 'Товары'),
    DataTable(2, 'Поставщики'),
    DataTable(3, 'Клиенты'),
# Строки данных
rows = [
    DataRow(1, 'Яблоки', 1),
    DataRow(2, 'Груши', 1),
    DataRow(3, '000 "Фруктовый Сад"', 2),
    DataRow(4, '000 "Свежие Поставки"', 2),
    DataRow(5, 'Иван Иванов', 3),
# Связь многие-ко-многим
rows tables = [
    DataRowTable(1, 1), # Таблица "Товары" - строка "Яблоки"
    DataRowTable(1, 2), # Таблица "Товары" - строка "Груши"
    DataRowTable(2, 3), # Таблица "Поставщики" - строка "000 'Фруктовый Сад'"
    DataRowTable(2, 4), # Таблица "Поставщики" - строка "ООО 'Свежие Поставки'"
    DataRowTable(3, 5), # Таблица "Клиенты" - строка "Иван Иванов"
1
def main():
    """Основная функция"""
    # Соединение данных один-ко-многим
    one to many = [
        (row.content, row.table id, table.name)
        for table in tables
        for row in rows
        if row.table id == table.id
    ]
```

# Соединение данных многие-ко-многим

```
many_to_many_temp = [
        (table.name, rel.table_id, rel.row_id)
        for table in tables
        for rel in rows tables
        if table.id == rel.table id
    ]
   many to many = [
        (row.content, row.table id, table name)
        for table_name, _, row_id in many_to_many temp
        for row in rows if row.id == row id
    ]
   print('Задание B1')
    # Все строки данных, которые начинаются с буквы "О"
   result_1_temp = list(filter(lambda x: x[0].startswith('O'), one_to_many))
   result_1 = [(row_content, table_name) for row_content, _, table_name in result_1_temp]
   print(result 1)
   print('\nЗадание B2')
    # Минимальный id для строк в каждой таблице
    result 2 unsorted = []
    for table in tables:
        # Все строки, принадлежащие таблице
        table rows = list(filter(lambda x: x[2] == table.name, one to many))
        if len(table rows) > 0:
            # Собираем id строк из этой таблицы
            table_row_ids = [row_id for _, row_id, _ in table_rows]
            # Находим минимальный id
            min row id = min(table row ids)
            result 2 unsorted.append((table.name, min row id))
    # Сортировка по минимальному id
    result 2 = sorted(result 2 unsorted, key=lambda x: x[1])
   print(result 2)
   print('\nЗадание ВЗ')
    # Результат многие-ко-многим, отсортированный по названию строки
   result 3 temp = sorted(many to many, key=lambda x: x[0])
   result_3 = [(row_content, table_name) for row_content, _, table_name in result_3_temp]
   print(result 3)
if name == ' main ':
   main()
```

# Результат выполнения программы

# Задание В1

[('ООО "Фруктовый Сад"', 'Поставщики'), ('ООО "Свежие Поставки"', 'Поставщики')]

#### Задание В2

[('Товары', 1), ('Поставщики', 2), ('Клиенты', 3)]

### Задание В3

[('Груши', 'Товары'), ('Иван Иванов', 'Клиенты'), ('ООО "Свежие Поставки"', 'Поставщики'), ('ООО "Фруктовый Сад"', 'Поставщики'), ('Яблоки', 'Товары')]

```
К1/RK1.py
Задание B1
[('000 "Фруктовый Сад"', 'Поставщики'), ('000 "Свежие Поставки"', 'Поставщики')]

Задание B2
[('Товары', 1), ('Поставщики', 2), ('Клиенты', 3)]

Задание B3
[('Груши', 'Товары'), ('Иван Иванов', 'Клиенты'), ('000 "Свежие Поставки"', 'Поставщики'), ('000 "Фруктовый Сад"', 'Поставщики'), ('Яблоки', 'Товары')]

PS C:\Users\sdasv\Desktop\Trash\Studies\Pickup>
```