

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

Выполнил:
Студент группы ИУ5-36Б
Гаврилик Я. С.
Преподаватель:
Гапанюк Ю. Е.

Москва 2025

Листинг кода:

```
from operator import itemgetter
```

```
class Row:
```

```
    def __init__(self, id, name, value, table_id):  
        self.id = id  
        self.name = name  
        self.value = value  
        self.table_id = table_id
```

```
class Table:
```

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

```
class RowTable:
```

```
    def __init__(self, table_id, row_id):  
        self.table_id = table_id  
        self.row_id = row_id
```

```
tables = [
```

```
    Table(1, 'Аналитические данные'),  
    Table(2, 'Сырые данные'),  
    Table(3, 'Архивные записи'),  
    Table(4, 'Административные таблицы'),  
    Table(5, 'Временные данные'),
```

```
]
```

```
rows = [
```

```
    Row(1, 'Строка1', 150, 1),  
    Row(2, 'Строка2', 200, 2),  
    Row(3, 'Строка3', 180, 1),  
    Row(4, 'Строка4', 220, 3),  
    Row(5, 'Строка5', 190, 4),  
    Row(6, 'Строка6', 210, 1),
```

```

        Row(7, 'Строка7', 170, 4),
    ]

rows_tables = [
    RowTable(1, 1),
    RowTable(1, 3),
    RowTable(1, 6),
    RowTable(2, 2),
    RowTable(3, 4),
    RowTable(4, 5),
    RowTable(4, 7),
    RowTable(5, 1),
    RowTable(5, 2),
]

```



```

def main():
    one_to_many = [(r.name, r.value, t.name)
                   for t in tables
                   for r in rows
                   if r.table_id == t.id]

    many_to_many_temp = [(t.name, rt.table_id, rt.row_id)
                          for t in tables
                          for rt in rows_tables
                          if t.id == rt.table_id]

    many_to_many = [(r.name, r.value, table_name)
                    for table_name, table_id, row_id in many_to_many_temp
                    for r in rows if r.id == row_id]

    print('Задание Г1')
    res1 = {}
    for t in tables:
        if t.name.startswith('A'):
            table_rows = [row_name for row_name, _, table_name in one_to_many
                          if table_name == t.name]
            res1[t.name] = table_rows
    print(res1)

```

```
print('\nЗадание Г2')
res2_unsorted = []
for t in tables:
    table_values = [value for _, value, table_name in one_to_many
                    if table_name == t.name]
    if table_values:
        max_value = max(table_values)
        res2_unsorted.append((t.name, max_value))

res2 = sorted(res2_unsorted, key=itemgetter(1))
print(res2)

print('\nЗадание Г3')
table_groups = {}
for row_name, _, table_name in many_to_many:
    if table_name not in table_groups:
        table_groups[table_name] = []
    if row_name not in table_groups[table_name]:
        table_groups[table_name].append(row_name)

res3 = sorted(table_groups.items(), key=itemgetter(0))
print(dict(res3))

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

Результат работы программы:

Задание Г1

```
{'Аналитические данные': ['Строка1', 'Строка3', 'Строка6'], 'Архивные записи': ['Строка4'], 'Административные таблицы': ['Строка5', 'Строка7']}
```

Задание Г2

```
[('Административные таблицы', 190), ('Сырые данные', 200), ('Аналитические данные', 210), ('Архивные записи', 220)]
```

Задание Г3

```
{'Административные таблицы': ['Строка5', 'Строка7'], 'Аналитические данные': ['Строка1', 'Строка3', 'Строка6'], 'Архивные записи': ['Строка4'], 'Временные данные': ['Строка1', 'Строка2'], 'Сырые данные': ['Строка2']}
```