

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

Факультет «Информатика и управление»

Кафедра «Системы обработки информации и управления»

Курс «Базовые компоненты интернет-технологий»

Отчет по рубежному контролю №1

Вариант №23 (Г)

Выполнил:

студент группы ИУ5-54

Фурасов В.Д.

Подпись и дата:

Проверил:

преподаватель каф. ИУ5

Гапанюк Ю. Е.

Подпись и дата:

Задание:

1. "Язык программирования" и "Синтаксические конструкции" связаны соотношением один-ко-многим. Выведите список всех связанных конструкций и языков программирования, отсортированный по названию языка программирования, сортировка по конструкциям произвольная.
2. "Язык программирования" и "Синтаксические конструкции" связаны соотношением один-ко-многим. Выведите список языков программирования с суммарным числом операндов в конструкциях в каждом языке программирования, отсортированный по суммарному числу операндов в порядке убывания.
3. "Язык программирования" и "Синтаксическая конструкция" связаны соотношением многие-ко-многим. Выведите список всех языков программирования

Код программы:

1. Импорт оператора для сортировки:

```
from operator import itemgetter
```

2. Класс "Язык программирования"

```
class ProgrammingLanguage:
    """ProgrammingLanguage"""

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

3. Класс "синтаксическая конструкция"

```
class Constr:
    """Constr"""

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

4. Класс “Сотрудники отдела” для реализации много ко многим

```
class ProgrammingLanguageConstr:
    def __init__(self, prog_lang_id, constr_id):
        self.prog_lang_id = prog_lang_id
        self.constr_id = constr_id
```

5. Языки программирования и конструкции:

```
# Языки программирования
programming_languages = [
    ProgrammingLanguage(1, 'C++'),
    ProgrammingLanguage(2, 'Python'),
    ProgrammingLanguage(3, 'JavaScript'),
]

# Конструкции
constrs = [
    Constr(1, 'cycles', 3, 1),
    Constr(2, 'functions', 2, 2),
    Constr(3, 'conditions', 4, 11),
    Constr(4, 'increments', 2, 3),
    Constr(5, 'decrements', 2, 22),
    Constr(6, 'outputs', 1, 1),
    Constr(7, 'classes', 1, 22),
]
```

```

prog_lang_constrs = [
    ProgrammingLanguageConstr(1, 1),
    ProgrammingLanguageConstr(1, 2),
    ProgrammingLanguageConstr(1, 4),
    ProgrammingLanguageConstr(1, 5),
    ProgrammingLanguageConstr(1, 6),

    ProgrammingLanguageConstr(2, 1),
    ProgrammingLanguageConstr(2, 3),
    ProgrammingLanguageConstr(2, 4),
    ProgrammingLanguageConstr(2, 5),

    ProgrammingLanguageConstr(3, 1),
    ProgrammingLanguageConstr(3, 6),
    ProgrammingLanguageConstr(3, 7),
]

```

6. Главная функция:

```

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

    # Соединение данных один-ко-многим
    one_to_many = [(o.name, o.num_of_constrs, l.name)
                    for l in programming_languages
                    for o in constrs
                    if l.id == o.prog_lang_id]

    # Соединение данных многие-ко-многим
    many_to_many_temp = [(l.name, lo.prog_lang_id, lo.constr_id)
                           for l in programming_languages
                           for lo in prog_lang_constrs
                           if l.id == lo.prog_lang_id]

    many_to_many = [(o.name, o.num_of_constrs, name)
                     for name, _, constr_id in many_to_many_temp
                     for o in constrs if o.id == constr_id]

```

```

print('Задание A1')
res_11 = sorted(one_to_many, key=itemgetter(2))
print(res_11)

print('\nЗадание A2')
res_12_unsorted = []

# Перебираем все компьютеры
for l in programming_languages:
    l_constrs = list(filter(lambda i: i[2] == l.name, one_to_many))

    if len(l_constrs) > 0:
        l_num_of_constrs = [num_of_constrs for _, num_of_constrs, _ in l_constrs]
        l_num_of_constrs_sum = sum(l_num_of_constrs)
        res_12_unsorted.append((l.name, l_num_of_constrs_sum))

res_12 = sorted(res_12_unsorted, key=itemgetter(1), reverse=True)
print(res_12)

print('\nЗадание A3')
res_13 = {}
# Перебираем все отделы
for l in programming_languages:
    if 'Python' in l.name:
        l_constrs = list(filter(lambda i: i[2] == l.name, many_to_many))
        l_constrs_names = [x for x, _, _ in l_constrs]
        res_13[l.name] = l_constrs_names

print(res_13)

if __name__ == '__main__':
    main()

```

7. Результат работы программы:

```

/Users/vfurasov/Documents/bmstu/54/ПИП/RK1_new/bin/python /Users/vfurasov/Documents/bms
Задание A1
[('cycles', 3, 'C++'), ('outputs', 1, 'C++'), ('increments', 2, 'JavaScript'), ('functi

Задание A2
[('C++', 4), ('Python', 2), ('JavaScript', 2)]

Задание A3
{'Python': ['cycles', 'conditions', 'increments', 'decrements']}

Process finished with exit code 0

```

