РК-1 Михалев. ИУ5Ц-72Б Вариант предметной области -- 23 — Синтаксическая конструкция / язык програмирования Вариант запросов ---------- Б

```
from operator import itemgetter


class ProgrammingLanguage:


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


class 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


class ProgrammingLanguageConstr:


    def __init__(self, prog_lang_id, constr_id):
        self.prog_lang_id = prog_lang_id
        self.constr_id = constr_id



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


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),

]


def main():



    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]


    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 Б1')
    res_11 = sorted(one_to_many, key=itemgetter(2))
    print(res_11)

    print('\nЗадание Б2')
    res_12_unsorted = []


    for l in programming_languages:
        l_count = 0
        for la in constrs:
            if la.prog_lang_id == l.id:
                l_count +=1
        res_12_unsorted.append((l.name, l_count))
```

```
    res_12 = sorted(res_12_unsorted, key=itemgetter(1), reverse=True)
    print(res_12)
    #Б3 Вывести список всех конструкций заканчивающийся на ts.
    print('\nЗадание Б3')
    sort_ProgrammingLanguage = list(filter(lambda x: x[0][-2:]== "ts", many_to_many))

    res_13 = {}
    while len(sort_ProgrammingLanguage)>0:
        name=sort_ProgrammingLanguage[0][0]
        sp=list(filter(lambda x: x[0]== name,many_to_many))
        res_13[sp[0][0]]=[x for _, _,x in sp]
        for i in sort_ProgrammingLanguage:
            if i[0]== name:
                sort_ProgrammingLanguage.remove(i)

    print(res_13)


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