Чугунов Александр ИУ5-32Б

```
from operator import itemgetter

class ProgrammingLanguage:
    """Язык программирования"""
    def __init__(self, id, name):
        self.id = id
        self.name = name


class SyntaticConstruction:
    """Синтаксическая конструкция"""
    def __init__(self, id, name, usage_frequency, pl_id):
        self.id = id
        self.name = name
        self.usage_frequency = usage_frequency
        self.pl_id = pl_id


class PlSc:
    """Синтаксические конструкции языков программирования"""
    def __init__(self, pl_id, sc_id):
        self.pl_id = pl_id
        self.sc_id = sc_id




languages = [
    ProgrammingLanguage(1, 'Java'),
    ProgrammingLanguage(2, 'C++'),
    ProgrammingLanguage(3, 'C#'),
]

constructions = [
    SyntaticConstruction(1, 'Идентификатор', 100, 1),
    SyntaticConstruction(2, 'Константа', 50, 1),
    SyntaticConstruction(3, 'Переменная', 100, 2),
    SyntaticConstruction(4, 'Тип', 80, 3),
    SyntaticConstruction(5, 'Метка', 20, 3),
]


languages_constructions = [
    PlSc(1, 1),
    PlSc(1, 2),
    PlSc(2, 3),
    PlSc(3, 4),
    PlSc(3, 5),
]


def main():
    """Основная функция"""
    one_to_many = [(c.name, c.usage_frequency, l.name)
    for c in constructions
    for l in languages
    if c.pl_id==l.id]

    many_to_many_temp = [(l.name, lc.pl_id, lc.sc_id)
    for l in languages
    for lc in languages_constructions
    if l.id==lc.pl_id]

    many_to_many = [(c.name, c.usage_frequency, language_name)
    for language_name, language_id, construction_id in many_to_many_temp
    for c in constructions
    if c.id==construction_id]

    print('Задание Б1')
    res_11 = sorted(one_to_many, key=itemgetter(0))
    print(res_11)

    print('\nЗадание Б2')
    language_construction_count = {}

    for l in languages:
        language_name = l.name
        constructions_count = sum(1 for c in one_to_many if c[2] == language_name)
        language_construction_count[language_name] = constructions_count
```

```
    sorted_language_construction_count = sorted(language_construction_count.items(), key=lambda
x: x[1], reverse=True)

    for language, count in sorted_language_construction_count:
        print(f'{language}: {count} конструкций')

    print('\nЗадание Б3')
    filtered_many_to_many = [(c_name, language_name) for c_name, _, language_name in many_to_many
if c_name.endswith('a')]

    for syntax, language in filtered_many_to_many:
        print(f'{syntax} ({language})')

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