

### Файл класса языка:

```
class Lang:
    "Язык программирования"
    def __init__(self, id, name, freq):
        self.id = id
        self.name = name
        self.freq = freq #Частота использования языка, %
```

### Файл класса оператора:

```
class Oper:
    "Оператор"
    def __init__(self, id, name, freq, id_lang):
        self.id = id
        self.name = name
        self.freq = freq #frequency of use, %
        self.id_lang = id_lang #language id
```

### Основной файл:

```
from operator import itemgetter
from operator_class import Oper
from language import Lang

# languages
langs = [
    Lang(1, 'C', 12),
    Lang(2, 'C++', 16),
    Lang(3, 'Python', 21),
    Lang(4, 'Paskal', 3),
]

# operators
opers = [
    Oper(1, 'Арифметические', 35, 1),
    Oper(2, 'Логические', 26, 1),
    Oper(3, 'Сравнения', 29, 1),
    Oper(1, 'Арифметические', 33, 2),
    Oper(2, 'Логические', 27, 2),
    Oper(3, 'Сравнения', 29, 2),
    Oper(1, 'Арифметические', 20, 3),
    Oper(2, 'Логические', 14, 3),
    Oper(3, 'Сравнения', 41, 3),
    Oper(1, 'Арифметические', 51, 4),
    Oper(2, 'Логические', 24, 4),
    Oper(3, 'Сравнения', 14, 4),
]

def main():
    # Соединение данных один-ко-многим
    one_to_many = [(o.name, o.freq, l.name, l.freq) for l in langs for o in opers if
o.id_lang==l.id]

    print('Задание 1, сортировка по частоте исп-я языка\
```

```

        и частоте исп-я оператора внутри языка')
list_1 = sorted(one_to_many, key=itemgetter(1))
list_2 = sorted(list_1, key=itemgetter(3))

#вывод через цикл
for i in range(len(list_2)):
    print('\n Название языка: {}, Частота использования языка: {}%,\n Название
оператора: {}, Частота использования оператора: {}%.'
        .format (list_2[i][2], list_2[i][3], list_2[i][0], list_2[i][1]))

#вывод через функцию высшего порядка
list(map(lambda i: print('\n Название языка: {}, Частота использования языка: {}%, \
    \n Название оператора: {}, Частота использования оператора: {}%.'
        .format (i[2], i[3], i[0], i[1])), list_2))

print('\nЗадание 2, список языков с суммарной\
    частотой использования заданных операторов')
list_2_unsorted = []
# Перебираем все языки
for l in langs:

    l_opsers = list(filter(lambda i: i[2]==l.name, one_to_many))

    l_freq = [freq for _,freq,_,_ in l_opsers]

    l_freq_sum = sum(l_freq)

    list_2_unsorted.append((l.name, l_freq_sum))

# Сортировка по суммарной частоте через sort()
list_2_unsorted.sort(key=itemgetter(1))
print(list_2_unsorted)

if __name__ == '__main__':
    main()

```

## Результаты работы программы:

### Задание 1, сортировка по частоте исп-я языка и частоте исп-я оператора внутри языка

Название языка: Paskal, Частота использования языка: 3%,

Название оператора: Сравнения, Частота использования оператора: 14%.

Название языка: Paskal, Частота использования языка: 3%,

Название оператора: Логические, Частота использования оператора: 24%.

Название языка: Paskal, Частота использования языка: 3%,

Название оператора: Арифметические, Частота использования оператора: 51%.

Название языка: C, Частота использования языка: 12%,

Название оператора: Логические, Частота использования оператора: 26%.

Название языка: C, Частота использования языка: 12%,

Название оператора: Сравнения, Частота использования оператора: 29%.

Название языка: C, Частота использования языка: 12%,

Название оператора: Арифметические, Частота использования оператора: 35%.

Название языка: C++, Частота использования языка: 16%,

Название оператора: Логические, Частота использования оператора: 27%.

Название языка: C++, Частота использования языка: 16%,

Название оператора: Сравнения, Частота использования оператора: 29%.

Название языка: C++, Частота использования языка: 16%,

Название оператора: Арифметические, Частота использования оператора: 33%.

Название языка: Python, Частота использования языка: 21%,

Название оператора: Логические, Частота использования оператора: 14%.

Название языка: Python, Частота использования языка: 21%,

Название оператора: Арифметические, Частота использования оператора: 20%.

Название языка: Python, Частота использования языка: 21%,

Название оператора: Сравнения, Частота использования оператора: 41%.

**Задание 2, список языков с суммарной частотой использования заданных операторов**

[('Python', 75), ('C++', 89), ('Paskal', 89), ('C', 90)]