

**РК №1**

**ИУ5-55Б**

**Усынин Юрий**

Запрос Д

Предметная область 21 - Оператор и Язык программирования

Задание:

1. «Язык программирования» и «Оператор» связаны соотношением один-ко-многим. Выведите список всех операторов, у которых название заканчивается на «ия», и их языки программирования.

2. «Язык программирования» и «Оператор» связаны соотношением один-ко-многим. Выведите список языков программирования со средним значением «number» у соответствующих операторов, отсортированный по этому значению (отдельной функции вычисления среднего значения в Python нет, нужно использовать комбинацию функций вычисления суммы и количества значений).

«number» - любое произвольное число, необходимое для выполнения задания рубежного контроля.

3. «Язык программирования» и «Оператор» связаны соотношением многие-ко-многим. Выведите список всех языков программирования, у которых название начинается с буквы «Р», и список используемых в них операторов.

Листинг:

```
# Усынин Юрий ИУ5-55Б
# Запрос Д
# Предметная область 21 - Оператор и Язык программирования
from operator import itemgetter

class Operator:
    # Оператор
    def __init__(self, id, title, number, pl_id):
        self.id = id
        self.title = title
        # number - любое произвольное число, так как по заданию рк необходимо
        # любое числовое значение
        self.number = number
        self.pl_id = pl_id

class Programming_language:
```

```

# Язык программирования
def __init__(self, id, name):
    self.id = id
    self.name = name

class OperProg_lan:
    # Операторы языка программирования
    # для реализации связи многие-ко-многим
    def __init__(self, progLan_id, oper_id):
        self.progLan_id = progLan_id
        self.oper_id = oper_id

# Языки программирования
progLang = [
    Programming_language(1, "Basic"),
    Programming_language(2, "Pascal"),
    Programming_language(3, "C++"),
    Programming_language(4, "Python"),
    Programming_language(5, "Java"),
    Programming_language(6, "C#")
]

# операторы
oper = [
    Operator(1, "Not - инверсия", 17, 1),
    Operator(2, "And - конъюнкция", 50, 2),
    Operator(3, "Or - дизъюнкция", 22, 2),
    Operator(4, "==" - равно", 14, 3),
    Operator(5, "!" - инверсия", 38, 3),
    Operator(6, "&& - конъюнкция", 32, 3),
    Operator(7, "|| - дизъюнкция", 28, 3)
]

oper_progLang = [
    OperProg_lan(1, 1),
    OperProg_lan(2, 2),
    OperProg_lan(2, 3),
    OperProg_lan(3, 4),
    OperProg_lan(3, 5),
    OperProg_lan(3, 6),
    OperProg_lan(3, 7),

    OperProg_lan(4, 1),
    OperProg_lan(4, 2),
    OperProg_lan(4, 3),
    OperProg_lan(5, 4),
    OperProg_lan(6, 5),
    OperProg_lan(6, 6),
    OperProg_lan(6, 7),
]

def main():
    # соединение данных один-ко-многим
    one_to_many = [(c.title, c.number, o.name)
                    for o in progLang
                    for c in oper
                    if c.pl_id == o.id]

    # соединение данных многие-ко-многим
    many_to_many_temp = [(o.name, co.progLan_id, co.oper_id)
                           for o in progLang
                           for co in oper_progLang
                           if o.id == co.progLan_id]

```

```

many_to_many = [(c.title, c.number, pl_name)
                 for pl_name, progLang_id, oper_id in many_to_many_temp
                 for c in oper if c.id == oper_id]

print('Задание Д1')
res1 = []
for o in one_to_many:
    if o[0][-2:] == "ия":
        res1.append(o[0:3:2])
print(res1)

print('\nЗадание Д2')
res2_unsorted = []
for o in progLang:
    o_oper = list(filter(lambda i: i[2] == o.name, one_to_many))
    if len(o_oper) > 0:
        o_number = [number for _, number, _ in o_oper]
        o_number_sum = sum(o_number)
        o_number_count = len(o_number)
        o_number_average = o_number_sum / o_number_count
        res2_unsorted.append((o.name, int(o_number_average)))
res2 = sorted(res2_unsorted, key=itemgetter(1), reverse=True)
print(res2)

print('\nЗадание Д3')
res3 = {}
for o in progLang:
    if o.name[0] == "P":
        o_oper = list(filter(lambda i: i[2] == o.name, many_to_many))
        o_oper_titles = [x for x, _, _ in o_oper]
        res3[o.name] = o_oper_titles
print(res3)

if __name__ == '__main__':
    main()

```

## Результаты:

```

Run: RK1 x
C:\Users\1\AppData\Local\Programs\Python\Python37\python.exe C:/Work/GitHub/RIP/PK_1/RK1.py

Задание Д1
[('Not - инверсия', 'Basic'), ('And - конъюнкция', 'Pascal'), ('Or - дизъюнкция', 'Pascal'), ('! - инверсия', 'C++'), ('&& - конъюнкция', 'C++'), ('|| - дизъюнкция', 'C++')]

Задание Д2
[('Pascal', 36), ('C++', 28), ('Basic', 17)]

Задание Д3
{'Pascal': ['And - конъюнкция', 'Or - дизъюнкция'], 'Python': ['Not - инверсия', 'And - конъюнкция', 'Or - дизъюнкция']}

Process finished with exit code 0

```