

## Вариант запросов Д

### Предметная область 12

1. «Язык Программирования» и «Средство разработки» связаны соотношением один-ко-многим. Выведите список всех Средств разработки, чьи названия заканчиваются на "Studio", и соответствующие им языки программирования.
2. «Язык Программирования» и «Средство разработки» связаны соотношением один-ко-многим. Рассчитайте среднюю стоимость средства разработки для каждого языка программирования и отсортируйте список по средней цене.
3. «Язык Программирования» и «Средство разработки» связаны соотношением многие-ко-многим. Выведите список всех Языков Программирования, у которых название начинается с буквы «Р», и связанные с ними средства разработки.

### Код программы

```
from operator import itemgetter

# Класс "Язык программирования"
class ProgramLang:
    def __init__(self, id, name):
        self.id = id
        self.name = name

# Класс "IDE" (среда разработки)
class IDE:
    def __init__(self, id, name, price, prog_lang_id):
        self.id = id
        self.name = name
        self.price = price
        self.prog_lang_id = prog_lang_id

# Класс для связи многие-ко-многим между ProgramLang и IDE
class ProgLangIDE:
    def __init__(self, prog_lang_id, ide_id):
        self.prog_lang_id = prog_lang_id
        self.ide_id = ide_id

# Список языков программирования
prog_langs = [
    ProgramLang(1, "C++"),
    ProgramLang(2, "Java"),
    ProgramLang(3, "Python"),
]

# Список IDE
ides = [
    IDE(1, "Visual Studio", 0, 1),
    IDE(2, "Eclipse", 0, 2),
```

```

        IDE(3, "PyCharm", 200, 3),
        IDE(4, "Android Studio", 0, 2),
        IDE(5, "CLion", 150, 1),
    ]

# Связь многие-ко-многим между языками программирования и IDE
pl_ide = [
    ProgLangIDE(1, 1),
    ProgLangIDE(2, 2),
    ProgLangIDE(3, 3),
    ProgLangIDE(2, 4),
    ProgLangIDE(1, 5),
]

# Задание 1: Вывод IDE, у которых название заканчивается на 'Studio', и
# соответствующие языки программирования
def first_task(one_to_many):
    res_1 = [(ide_name, pl_name) for ide_name, ide_price, pl_name in
one_to_many if ide_name.endswith('Studio')]
    return res_1

# Задание 2: Вывод языков программирования со средней стоимостью IDE,
# отсортированных по средней цене
def second_task(one_to_many):
    temp_dict = {}

    for ide_name, ide_price, pl_name in one_to_many:
        if pl_name in temp_dict:
            temp_dict[pl_name].append(ide_price)
        else:
            temp_dict[pl_name] = [ide_price]

    res_2 = [(pl_name, sum(prices) / len(prices)) for pl_name, prices in
temp_dict.items()]
    res_2.sort(key=itemgetter(1), reverse=True)
    return res_2

# Задание 3: Вывод языков программирования, начинающихся на 'P', и список
# связанных с ними IDE
def third_task(many_to_many, start_ch):
    res_3 = [(pl_name, ide_name) for pl_name, ide_name in many_to_many if
pl_name.startswith(start_ch)]
    return res_3

# Основная функция
def main():
    # Один-ко-многим
    one_to_many = [(ide.name, ide.price, pl.name)
                    for pl in prog_langs
                    for ide in ides
                    if ide.prog_lang_id == pl.id]

    # Многие-ко-многим
    many_to_many_temp = [(pl.name, ps.prog_lang_id, ps.ide_id)
                          for pl in prog_langs
                          for ps in pl_ide
                          if ps.prog_lang_id == pl.id]

    many_to_many = [(pl_name, ide.name)

```

```
        for pl_name, pl_id, ide_id in many_to_many_temp
        for ide in ides if ide.id == ide_id]

print('Задание 1')
print(first_task(one_to_many))

print("\nЗадание 2")
print(second_task(one_to_many))

print("\nЗадание 3")
print(third_task(many_to_many, 'P'))

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

## Результат выполнения

### Задание 1

[('Visual Studio', 'C++'), ('Android Studio', 'Java')]

### Задание 2

[('Python', 200.0), ('C++', 75.0), ('Java', 0.0)]

### Задание 3