

Текст программы:

```

# Б 12, Ионов Степан Александрович, ИУ5-52Б
# 1:М соответствует: определенное средство разработки включает много языков
программирования
from operator import itemgetter

class Lang:
    """Язык программирования"""
    def __init__(self, id, name, rating, tool_id):
        self.id = id # id языка программирования (ЯП)
        self.name = name # название ЯП
        self.rating = rating # рейтинг ЯП
        self.tool_id = tool_id # id средства разработки

class Tool:
    """Средство разработки"""
    def __init__(self, id, name):
        self.id = id # id средства разработки (СР)
        self.name = name # название СР

class LangTool:
    """Средства программирования для языков программирования'
    необходим для реализации связи многие-ко-многим"""
    def __init__(self, tool_id, lang_id):
        self.tool_id = tool_id
        self.lang_id = lang_id

# Средства разработки
tools = [
    Tool(1, 'PyCharm Community Edition 2020'),
    Tool(2, 'Embarcadero Delphi'),
    Tool(3, 'Visual Studio Code 2020'),

    Tool(11, 'PyCharm Professional'),
    Tool(22, 'Borland Delphi'),
    Tool(33, 'Visual Studio Code 2017'),
]

# Языки программирования
langs = [
    Lang(1, 'Python', 1, 1),
    Lang(2, 'Delphi', 29, 2),
    Lang(3, 'C++', 4, 3),
    Lang(4, 'C#', 23, 3),
    Lang(5, 'Java', 2, 3),
    Lang(6, 'F#', 35, 3)
]

langs_tools = [
    LangTool(1, 1),
    LangTool(2, 2),
    LangTool(3, 3),
    LangTool(3, 4),
    LangTool(3, 5),
    LangTool(3, 6),

    LangTool(11, 1),
    LangTool(22, 2),

```

```

LangTool(33, 3),
LangTool(33, 4),
LangTool(33, 5),
LangTool(33, 6)
]

def main():
    """Основная функция"""

    # Соединение данных один-ко-многим
    one_to_many = [(l.name, l.rating, t.name)
                    for t in tools
                    for l in langs
                    if l.tool_id == t.id]

    # Соединение данных многие-ко-многим
    many_to_many_temp = [(t.name, lt.tool_id, lt.lang_id)
                           for t in tools
                           for lt in langs_tools
                           if t.id == lt.tool_id]

    many_to_many = [(l.name, l.rating, tool_name)
                     for tool_name, tool_id, lang_id in many_to_many_temp
                     for l in langs if l.id == lang_id]

    print('Задание Б1')
    res_11 = sorted(one_to_many, key=itemgetter(0))
    print(res_11)  # сортировка по имени языков программирования

    print('\nЗадание Б2')
    res_12_unsorted = []
    for t in tools:
        t_langs = list(filter(lambda x: x[2] == t.name, one_to_many))
        if len(t_langs) > 0:
            l_count = len(t_langs)
            res_12_unsorted.append((t.name, l_count))
    res_12 = sorted(res_12_unsorted, key=itemgetter(1), reverse=True)
    print(res_12)  # сортировка по убыванию количества языков

    print('\nЗадание Б3')
    res_13 = {}
    for l in langs:
        if '#' == l.name[-1:]:
            t_langs = list(filter(lambda x: x[0] == l.name, many_to_many))
            t_langs_names = [x[2] for x in t_langs]
            res_13[l.name] = t_langs_names

    # Словарь всех языков программирования, у которых имя заканчивается
    print(res_13)  # на '#' и названия средств разработки для них

if __name__ == '__main__':
    main()

```

Снимок экрана с результатами выполнения программы:

```
C:\Users\User\Desktop\3 курс\РПП\Python\RIP_lab\RK_1>main.py
Задание Б1
[('C#', 23, 'Visual Studio Code 2020'), ('C++', 4, 'Visual Studio Code 2020'), ('Delphi', 29, 'Embarcadero Delphi'), ('F#', 35,
'Visual Studio Code 2020'), ('Java', 2, 'Visual Studio Code 2020'), ('Python', 1, 'PyCharm Community Edition 2020')]

Задание Б2
[('Visual Studio Code 2020', 4), ('PyCharm Community Edition 2020', 1), ('Embarcadero Delphi', 1)]

Задание Б3
{'C#': ['Visual Studio Code 2020', 'Visual Studio Code 2017'], 'F#': ['Visual Studio Code 2020', 'Visual Studio Code 2017']}
```