

Вариант 23 А

Синтаксическая конструкция - Язык программирования

Код программы:

```
from operator import itemgetter
```

```
class Syntax:
```

```
    """Синтаксическая конструкция"""
```

```
    def __init__(self, id, name, execution_time, lang_id):
```

```
        self.id = id
```

```
        self.name = name
```

```
        self.execution_time = execution_time
```

```
        self.lang_id = lang_id
```

```
class Language:
```

```
    """Язык программирования"""
```

```
    def __init__(self, id, name):
```

```
        self.id = id
```

```
        self.name = name
```

```
class SyntaxLang:
```

```
    """Связь синтаксических конструкций и языков
```

```
    программирования для отношения многие-ко-многим"""
```

```
    def __init__(self, lang_id, syntax_id):
```

```
        self.lang_id = lang_id
```

```
        self.syntax_id = syntax_id
```

```
# Языки программирования
```

```
langs = [
```

```
    Language(1, 'Python'),
```

```
Language(2, 'Java'),  
Language(3, 'C++'),  
Language(4, 'C')  
]
```

Синтаксические конструкции

```
syntaxes = [  
    Syntax(1, 'Функция', 0.04, 1),  
    Syntax(2, 'Цикл do while', 0.01, 3),  
    Syntax(3, 'Условный оператор if', 0.01, 3),  
    Syntax(4, 'Цикл for', 0.02, 4),  
    Syntax(5, 'Класс', 0.05, 2)  
]
```

Связи многие-ко-многим

```
syntax_langs = [  
    SyntaxLang(1, 1),  
    SyntaxLang(1, 3),  
    SyntaxLang(1, 4),  
    SyntaxLang(1, 5),  
    SyntaxLang(2, 1),  
    SyntaxLang(2, 2),  
    SyntaxLang(2, 3),  
    SyntaxLang(2, 4),  
    SyntaxLang(2, 5),  
    SyntaxLang(3, 1),  
    SyntaxLang(3, 2),  
    SyntaxLang(3, 3),  
    SyntaxLang(3, 4),  
]
```

```
SyntaxLang(3, 5),  
SyntaxLang(4, 1),  
SyntaxLang(4, 2),  
SyntaxLang(4, 3),  
SyntaxLang(4, 4)  
]
```

```
def main():
```

```
    """Основная функция"""
```

```
    # Соединение данных один-ко-многим
```

```
    one_to_many = [(s.name, s.execution_time, l.name)
```

```
                    for l in langs
```

```
                    for s in syntaxes
```

```
                    if s.lang_id == l.id]
```

```
    # Соединение данных многие-ко-многим
```

```
    many_to_many_temp = [(l.name, sl.lang_id, sl.syntax_id)
```

```
                        for l in langs
```

```
                        for sl in syntax_langs
```

```
                        if l.id == sl.lang_id]
```

```
    many_to_many = [(s.name, s.execution_time, lang_name)
```

```
                    for lang_name, lang_id, syntax_id in many_to_many_temp
```

```
                    for s in syntaxes if s.id == syntax_id]
```

```
    # Задание A1: список связанных конструкций и языков, отсортированный  
    по конструкциям
```

```
    print('Задание A1')
```

```
res_11 = sorted(one_to_many, key=itemgetter(0))  
print(res_11)
```

Задание A2: список языков с суммарным временем выполнения конструкций, отсортированный по времени

```
print('\nЗадание A2')  
res_12_unsorted = []  
  
# Перебираем все языки программирования  
for l in langs:  
    # Список синтаксических конструкций  
    l_syntaxes = list(filter(lambda i: i[2] == l.name, one_to_many))  
    # Если синтаксическая конструкция не пустая  
    if len(l_syntaxes) > 0:  
        # Время выполнения конструкций  
        l_execution_times = [time for _, time, _ in l_syntaxes]  
        # Суммарное время выполнения по всем языкам  
        l_execution_times_sum = sum(l_execution_times)  
        res_12_unsorted.append((l.name, l_execution_times_sum))
```

Сортировка по суммарному времени выполнения

```
res_12 = sorted(res_12_unsorted, key=itemgetter(1), reverse=True)  
print(res_12)
```

Задание A3: список всех языков с "C" в названии и их синтаксических конструкций

```
print('\nЗадание A3')  
res_13 = {}  
  
# Перебираем все языки программирования  
for l in langs:
```

```

if 'C' in l.name:

    # Список языков программирования
    l_syntaxes = list(filter(lambda i: i[2] == l.name, many_to_many))

    # Только названия синтаксических конструкций
    l_syntaxes_names = [x for x,_,_ in l_syntaxes]

    # Добавляем результат в словарь
    # ключ - язык, значение - список названий синтаксических конструкций
    res_13[l.name] = l_syntaxes_names

print(res_13)

if __name__ == '__main__':
    main()

```

Результаты выполнения:

Задание A1

```

[('Класс', 0.05, 'Java'), ('Условный оператор if', 0.01, 'C++'), ('Функция', 0.04, 'Python'), ('Цикл do while', 0.01, 'C++'), ('Цикл for', 0.02, 'C')]

```

Задание A2

```

[('Java', 0.05), ('Python', 0.04), ('C++', 0.02), ('C', 0.02)]

```

Задание A3

```

{'C++': ['Функция', 'Цикл do while', 'Условный оператор if', 'Цикл for', 'Класс'],
'C': ['Функция', 'Цикл do while', 'Условный оператор if', 'Цикл for']}

```