



Информатика и системы управления

Системы обработки информации и управления

Курс «Парадигмы и конструкции языков программирования»

Отчет по РК №1
Вариант запросов: Г
Вариант предметной области: 23

20.10.2023

(подпись, дата)

(подпись, дата)

МГТУ им. Н.Э. Баумана

Москва, 2023 г.

Вариант Г.

Предметная область: «Синтаксическая конструкция - Язык программирования»

1. «Язык программирования» и «Синтаксическая конструкция» связаны соотношением один-ко-многим. Выведите список всех языков программирования, у которых название начинается с буквы «J», и список их синтаксических конструкций.
2. «Язык программирования» и «Синтаксическая конструкция» связаны соотношением один-ко-многим. Выведите список языков программирования с максимальным количеством синтаксических конструкций в каждом языке программирования, отсортированный по максимальному количеству синтаксических конструкций.
3. «Язык программирования» и «Синтаксическая конструкция» связаны соотношением многие-ко-многим. Выведите список всех связанных синтаксических конструкций и языков программирования, отсортированный по языкам программирования, сортировка по синтаксическим конструкциям произвольная.

Листинг программы

#Вариант 23

используется для сортировки

```
from operator import itemgetter
```

```
class syntactic:
```

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

```
    def __init__(self, id, s_name, num, lang_id):
```

```
        self.id = id
```

```
        self.s_name = s_name
```

```
        self.num = num
```

```
        self.lang_id = lang_id
```

```
class language:
```

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

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

```
        self.id = id
```

```
        self.l_name = l_name
```

```
class SynLang:
```

```
    """
```

```
    'Синтаксические конструкции языка программирования' для реализации  
    связи многие-ко-многим
```

```
    """
```

```

def __init__(self, lang_id, syn_id):
    self.lang_id = lang_id
    self.syn_id = syn_id

# Языки программирования
languages = [
    language(1, 'Python'),
    language(2, 'C#'),
    language(3, 'Java_Script'),
]

# Синтаксические конструкции
syntactics = [
    syntactic(1, 'Условные операторы', 9, 1),
    syntactic(2, 'Цикл', 3, 2),
    syntactic(3, 'Функции', 5, 3),
    syntactic(4, 'Структура данных', 1, 3),
    syntactic(5, 'Список', 1, 3),
]

#связь M:M
syn_lang = [
    SynLang(1,1),
    SynLang(2,2),
    SynLang(3,3),
    SynLang(3,4),
    SynLang(3,5),
]

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

    # Соединение данных один-ко-многим
    one_to_many = [(s.s_name, l.l_name)
                    for l in languages
                    for s in syntactics
                    if s.lang_id==l.id]

    # Соединение данных многие-ко-многим
    many_to_many_temp = [(l.l_name, s.s_name)
                          for sl in syn_lang
                          for l in languages
                          for s in syntactics

```

```

        if sl.lang_id==l.id and sl.syn_id==s.id ]

many_to_many = sorted(many_to_many_temp, key=lambda x: x[0])

#Задание 1
print('Задание Г1')
# Перебираем все языки программирования
for n in languages:
    if n.l_name.startswith('J'):
        res = [l for l, s in one_to_many if s == n.l_name]
        print(f'{n.l_name}: {res}')

#Задание 2
print('\nЗадание Г2')
max_num = [(l.l_name, max(s.num for s in syntactics if s.lang_id == l.id))
            for l in languages]
#Сортировка
res2 = sorted(max_num, key=lambda x: x[1], reverse=True)
print(f'{res2}')

#Задание 3
print('\nЗадание Г3')
for res3 in many_to_many:
    print(res3)

if __name__ == '__main__':
    main()

```

Результат

Задание Г1

Java_Script: ['Функции', 'Структура данных', 'Список']

Задание Г2

[('Python', 9), ('Java_Script', 5), ('C#', 3)]

Задание Г3

('C#', 'Цикл')

('Java_Script', 'Функции')

('Java_Script', 'Структура данных')

('Java_Script', 'Список')

('Python', 'Условные операторы')