



Информатика и системы управления

Системы обработки информации и управления

Курс «Парадигмы и конструкции языков программирования»

Отчет по РК №2

(подпись, дата)

(подпись, дата)

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.

Задание РК№2

- 1) Проведите рефакторинг текста программы рубежного контроля №1 таким образом, чтобы он был пригоден для модульного тестирования.
- 2) Для текста программы рубежного контроля №1 создайте модульные тесты с применением TDD - фреймворка (3 теста).

Листинг программы

1. Файл main.py:

#Вариант 23

```
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'
```

```
    """
```

```
    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),
```

```
]
```

```
syn_lang = [
```

```
    SynLang(1,1),
```

```
    SynLang(2,2),
```

```
    SynLang(3,3),
```

```
    SynLang(3,4),
```

```
    SynLang(3,5),
```

```
]
```

```
def get_one_to_many(languages,syntactics):
```

```
    return[(s.s_name,l.l_name)
```

```
        for l in languages
```

```
        for s in syntactics
```

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

```
def get_many_to_many(syn_lang, languages,syntactics):
```

```

        return[(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.lang_id==s.id]

def G1(languages, one_to_many):
    result = []
    for n in languages:
        if n.l_name.startswith('J'):
            s_list = [l for l, s in one_to_many if s == n.l_name]
            result.append((n.l_name, s_list))
    return result

def G2(languages, syntactics):
    lang_max_syms=[(l.l_name, max(s.num for s in syntactics if s.lang_id==l.id))
                   for l in languages]
    sorted_lang=sorted(lang_max_syms, key = lambda x: x[1], reverse=True)
    result = [(language_name, max_syms) for language_name, max_syms in
sorted_lang]
    return result

def G3(many_to_many):
    return many_to_many

def main():

    one_to_many = get_one_to_many(languages,syntactics)
    many_to_many = get_many_to_many(syn_lang, languages,syntactics)

    #Задание 1
    task_1 = G1(languages,one_to_many)
    print('Вывод языков программирования, у которых первая буква "J" и их список
синтаксических конструкций')
    for result in task_1:
        print(f'{result[0]}:{result[1]}')

    #Задание 2
    task_2 = G2(languages, syntactics)
    print('Вывод языков программирования с максимальным количеством синтаксических
конструкций в каждом языке программирования, отсортированный по максимальному
количеству синтаксических конструкций ')
    for result in task_2:
        print(f'{result[0]}:{result[1]}')

    #Задание 3
    task_3 = G3(many_to_many)
    print('Вывод списка всех связанных синтаксических конструкций и языков
программирования, отсортированный по языкам программирования')
    for result in task_3:
        print(result)

```

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

2. Файл test.py:

```
import unittest  
from main import get_one_to_many, get_many_to_many, G1, G2, G3, syntactic,  
language, SynLang  
  
class Test(unittest.TestCase):  
    def setUp(self):  
        self.languages = [  
            language(1, 'Python'),  
            language(2, 'C#'),  
            language(3, 'Java_Script')  
        ]  
  
        self.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),  
        ]  
  
        self.syn_lang = [  
            SynLang(1,1),  
            SynLang(2,2),  
            SynLang(3,3),  
            SynLang(3,4),  
            SynLang(3,5),  
        ]  
  
    def test_get_one_to_many(self):  
        one_to_many = get_one_to_many(self.languages, self.syntactics)  
        self.assertEqual(len(one_to_many),2)  
  
    def test_get_many_to_many(self):  
        many_to_many = get_many_to_many(self.syn_lang, self.languages,  
self.syntactics)  
        self.assertEqual(len(many_to_many),2)  
  
    def test_task1(self):  
        one_to_many=get_one_to_many(self.languages, self.syntactics)
```

```
task_1=G1(self.languages, one_to_many)
self.assertEqual(task_1, 'Ожидаемый результат')
```

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