

**Московский государственный технический
университет им. Н.Э. Баумана**

Факультет «Информатика и системы управления»
Кафедра ИУ5 «Системы обработки информации и управления»

Курс «Базовые компоненты интернет-технологий»

Отчет по рубежному контролю №2

Выполнил:
студент группы
ИУ5И-35Б
Давиташвили Шако

Москва, 2021 г.

Описание задания:

(Вариант предметной области – 23, вариант запросов – Д)

1. «Синтаксическая конструкция» и «Язык программирования» связаны соотношением один-ко-многим. Выведите операторы, которые оканчиваются на 'оп' и сколько дней изучают язык, к которому они относятся
2. «Синтаксическая конструкция» и «Язык программирования» связаны соотношением один-ко-многим. Выведите список языков программирования со средним количеством дней уделенным для изучения каждого из них, отсортированный по среднему количеству дней.
3. «Синтаксическая конструкция» и «Язык программирования» связаны соотношением многие-ко-многим. Выведите список всех языков программирования, у которых имя начинается на «С», и названия их операторов.

Условия рубежного контроля №2 по курсу БКИТ

4. Рубежный контроль представляет собой разработку тестов на языке Python
5. 1) Проведите рефакторинг текста программы рубежного контроля №1 таким образом, чтобы он был пригоден для модульного тестирования.
6. 2) Для текста программы рубежного контроля №1 создайте модульные тесты с применением ТОО - фреймворка (3 теста).

Текст программы:

Файл **rk2.py**

```
from operator import itemgetter
```

```
class SynConstr:
```

```
    """ Синтаксическая конструкция
        операторы, которые относятся к данному языку
        количество дней в течение которого изучают этот язык """
```

```
    def __init__(self, id, oper, days, ProgLan_id):
```

```
        self.id = id
```

```
        self.oper = oper
```

```
        self.days = days
```

```
        self.ProgLan_id = ProgLan_id
```

```
class ProgLan:
```

```
    # Язык программирования
```

```

def __init__(self, id, name):
    self.id = id
    self.name = name

class SCPL:
    # синтаксическая конструкция языка программирования
    def __init__(self, ProgLan_id, SynConstr_id):
        self.ProgLan_id = ProgLan_id
        self.SynConstr_id = SynConstr_id

def test_1(SynConstrs,Langs):
    # Соединение данных один-ко-многим
    one_to_many = [(s.oper,s.days,l.name)
                    for l in Langs
                    for s in SynConstrs
                    if l.id==s.ProgLan_id]
    return list(filter(lambda x : x[0].endswith('on'), one_to_many))

def test_2(SynConstrs,Langs):
    # Соединение данных многие-ко-многим
    one_to_many = [(s.oper,s.days,l.name)
                    for l in Langs
                    for s in SynConstrs
                    if l.id==s.ProgLan_id]
    res2_unsorted = []
    for l in Langs:
        #список языковых конструкций языков программирования
        l_SynConstrs = list(filter(lambda x: x[2]==l.name, one_to_many))
        if(len(l_SynConstrs)>0):
            l_days = [days for __,days,__ in l_SynConstrs]
            l_days_all = sum(l_days)
            average_days = l_days_all/len(l_days)
            res2_unsorted.append((l.name, average_days))
    return sorted(res2_unsorted, key=itemgetter(1), reverse=True)

def test_3(SynConstrs,Langs,SCPLs):
    many_to_many = [
        (s.oper, s.days, l.name)
        for l in Langs
        for s in SynConstrs
        for sp in SCPLs
        if l.id == sp.ProgLan_id and l.id == sp.SynConstr_id
    ]
    res3={}
    for l in Langs:
        if l.name.startswith('C'):
            #список языковых конструкций языков программирования
            l_SynConstrs = list(filter(lambda x: x[2]==l.name, many_to_many))
            names = [i for i, __, _ in l_SynConstrs]
            res3[l.name] = names
    return res3

def main():

```

"""Основная функция"""

Языки программирования

```
Langs = [  
    ProgLang(1, 'Python'),  
    ProgLang(2, 'Java'),  
    ProgLang(3, 'JavaScript'),  
    ProgLang(4, 'C++'),  
    ProgLang(5, 'C#'),  
    ProgLang(6, 'R'),  
]
```

Синтаксические конструкции и количество дней, в течение которых изучают язык

```
SynConstrs = [  
    SynConstr(1, 'Condition', 273, 1),  
    SynConstr(2, 'Loop', 381, 2),  
    SynConstr(3, 'Shift', 145, 3),  
    SynConstr(4, 'Add', 11, 4),  
    SynConstr(5, 'Mul', 253, 5),  
    SynConstr(6, 'Semicolon', 3, 6),  
    SynConstr(7, 'Array', 73, 1),  
]
```

]

#Syntactic Constructions of Programming Languages/синтаксические конструкции языков программирования

```
SCPLs = [  
    SCPL(1,2),  
    SCPL(2,3),  
    SCPL(3,4),  
    SCPL(4,5),  
    SCPL(5,6),  
    SCPL(6,7),  
    SCPL(3,2),  
    SCPL(4,3),  
    SCPL(5,4),  
    SCPL(6,5),  
    SCPL(5,6),  
    SCPL(1,1),  
    SCPL(2,2),  
    SCPL(3,3),  
    SCPL(4,4),  
    SCPL(5,5),  
    SCPL(6,6),  
    SCPL(7,7),  
    SCPL(7,3),  
    SCPL(2,5),  
    SCPL(1,6),  
    SCPL(6,4),  
]
```

```
print("\nЗаданиеД1")
```

```
res1 = test_1(SynConstrs, Langs)
```

```
[print(x[0],x[1]) for x in res1]
```

```
print("\nЗаданиеД2")
```

```
res2 = test_2(SynConstrs, Langs)
print(res2)
```

```
print("\nЗаданиеД3")
res3 = test_3(SynConstrs, Langs, SCPLs)
print(res3)
```

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

Файл rk2_tests.py

```
import unittest
import rk2
```

```
class tests(unittest.TestCase):
```

```
    def test1(self):
        ans = [
            ("Condition", 273, 'Python'),
            ("Semicolon", 3, 'R')
        ]
        self.assertEqual(rk2.test_1(rk2.SynConstrs, rk2.Langs), ans)
```

```
    def test2(self):
        ans = [
            ('Java', 381.0),
            ('C#', 253.0),
            ('Python', 173.0),
            ('JavaScript', 145.0),
            ('C++', 11.0),
            ('R', 3.0)
        ]
        self.assertEqual(rk2.test_2(rk2.SynConstrs, rk2.Langs), ans)
```

```
    def test3(self):
        ans = {
            'C++': ['Condition', 'Loop', 'Shift', 'Add', 'Mul', 'Semicolon', 'Array'],
            'C#': ['Condition', 'Loop', 'Shift', 'Add', 'Mul', 'Semicolon', 'Array']
        }
        self.assertEqual(rk2.test_3(rk2.SynConstrs, rk2.Langs, rk2.SCPLs), ans)
```

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

Результат выполнения программы:

```
hakodavtahvili@MacBook-Pro-Shako RK2 % python3 rk2_tests.py
```

```
ЗаданиеД1
Condition 273
Semicolon 3
```

```
ЗаданиеД2
[('Java', 381.0), ('C#', 253.0), ('Python', 173.0), ('JavaScript', 145.0), ('C++', 11.0), ('R', 3.0)]
```

```
ЗаданиеД3
{'C++': ['Condition', 'Loop', 'Shift', 'Add', 'Mul', 'Semicolon', 'Array'], 'C#': ['Condition', 'Loop', 'Shift', 'Add', 'Mul', 'Semicolon', 'Array']}
...
```

```
-----
Ran 3 tests in 0.001s
```

```
OK
hakodavtahvili@MacBook-Pro-Shako RK2 % cd
hakodavtahvili@MacBook-Pro-Shako ~ % cd Documents/Python/RK1
hakodavtahvili@MacBook-Pro-Shako RK1 % python3 rk1.py
```