



**Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)**

Факультет «ГУИМЦ»

Кафедра ИУ5 «Системы обработки информации и управления»

Дисциплина «Базовые компоненты ИТ»

ОТЧЕТ

Рубежный контроль №1

Студент: Бабаян А.А., группа ИУ5Ц-52Б

Преподаватель: Гапанюк Ю.Е.

2021г.

Описание задания:

Вариант Б, вариант предметной области №23.

1. «Синтаксическая конструкция» и «Язык программирования» связаны соотношением один-ко-многим. Выведите список всех связанных синтаксических конструкций и языков программирования, отсортированный по популярности ЯП, сортировка по ЯП произвольная.
2. «Синтаксическая конструкция» и «Язык программирования» связаны соотношением один-ко-многим. Выведите список синтаксической конструкции с количеством процентов популярности ЯП, отсортированный по количеству процентов популярности ЯП.
3. «Синтаксическая конструкция» и «Язык программирования» связаны соотношением многие-ко-многим. Выведите список всех ЯП, к которому относится конкретная синтаксическая конструкция, и названия их этих синтаксических конструкций.

Класс «Синтаксическая конструкция», содержащий поля:

- id конструкции (id)
- название синтаксической конструкции (name)

Класс «Язык программирования», содержащий поля:

- id ЯП (id)
- название ЯП (named)
- процент популярности ЯП (popularity)
- id раздела, для реализации связи один-ко-многим (dep_id)

Класс «Синтаксисы в ЯП» (для реализации связи один-ко-многим), содержащий поля:

- id синтаксиса (dep_id)
- id ЯП (emp_id)

Листинг программы:

```
"""РКМ1, Бабаян Артур, группа ИУ5Ц-52Б
Вариант Б, вариант предметной области 23"""

# используется для сортировки

from operator import itemgetter

class Prog:

    """Язык программирования"""

    def __init__(self, id, named, popularity, dep_id):

        self.id = id
        self.named = named
        self.popularity = popularity
        self.dep_id = dep_id

class Synt:

    """Синтаксисы"""

    def __init__(self, id, name):

        self.id = id
        self.name = name

class ProgSynt:

    """

    'Синтаксисы в ЯП' для реализации

    СВЯЗИ МНОГИЕ-КО-МНОГИМ

    """

    def __init__(self, dep_id, emp_id):

        self.dep_id = dep_id
        self.emp_id = emp_id

# Синтаксисы

deps = [

    Synt(1, 'Конструкции'),
    Synt(2, 'Функции'),
    Synt(3, 'Операции'),
    Synt(11, 'Конструкции (задания)'),
    Synt(22, 'Функции (задания)'),
    Synt(33, 'Операции (задания)'),

]

# Языки программирования

emps = [

    Prog(1, 'C++', 4.6, 1),
```

```

Prog(2, 'C#', 14.5, 2),
Prog(3, 'Python', 12.1, 3),
Prog(4, 'JavaScript', 18.1, 3),
Prog(5, 'TypeScript', 6.9, 3),

```

```

]
```

```

emps_deps = [
```

```

    ProgSynt(1, 1),
    ProgSynt(2, 2),
    ProgSynt(3, 3),
    ProgSynt(3, 4),
    ProgSynt(3, 5),
    ProgSynt(11, 1),
    ProgSynt(22, 2),
    ProgSynt(33, 3),
    ProgSynt(33, 4),

```

```

]
```

```

def main():
```

```

    """Основная функция"""
```

```

    # Соединение данных один-ко-многим
```

```

    one_to_many = [(e.named, e.popularity, d.name)
```

```

        for d in deps
        for e in emps
        if e.dep_id==d.id]
```

```

    # Соединение данных многие-ко-многим
```

```

    many_to_many_temp = [(d.name, ed.dep_id, ed.emp_id)
        for d in deps
        for ed in emps_deps
        if d.id == ed.dep_id]
```

```

    many_to_many = [(e.named, e.popularity, dep_name)
        for dep_name, dep_id, emp_id in many_to_many_temp
        for e in emps if e.id == emp_id]
```

```

    print('Задание Б1')
```

```

    res_11 = sorted(one_to_many, key=itemgetter(2))
    print(res_11)
```

```

    print('\nЗадание Б2')
```

```

    res_12_unsorted = []
```

```

    # Перебираем все синтаксисы
```

```

    for d in deps:
```

```

        # Список ЯП
```

```

        d_emps = list(filter(lambda i: i[2]==d.name, one_to_many))
```

```

        # Если отдел не пустой
```

```

        if len(d_emps) > 0:
```

```

            # Популярность каждого ЯП
```

```

            d_popularity = [popular for _,popular,_ in d_emps]
```

```

            # Суммарная популярность ЯП
```

```

d_popularity_sum = sum(d_popularity)
res_12_unsorted.append((d.name, d_popularity_sum))

# Сортировка по популярности языков программирования
res_12 = sorted(res_12_unsorted, key=itemgetter(1), reverse=True)
print(res_12)

print('\nЗадание Б3')

res_13 = {}
# Перебираем все синтаксисы

for d in deps:
    if ' ' in d.name:

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

        # Только названия ЯП
        d_emps_names = [x for x,_,_ in d_emps]
        # Добавляем результат в словарь

        # ключ - синтаксисы, значение - название ЯП
        res_13[d.name] = d_emps_names
print(res_13)

if __name__ == '__main__':
    main()

```

Результат выполнения программы:

```

C:\Program Files (x86)\Microsoft Visual Studio\Shared\Python37_64\python.exe
Задание Б1
[('C++', 4.6, 'Конструкции'), ('Python', 12.1, 'Операции'), ('JavaScript', 18.1, 'Операции'), ('TypeScript', 6.9, 'Операции'), ('C#', 14.5, 'Функции')]

Задание Б2
[('Операции', 37.1), ('Функции', 14.5), ('Конструкции', 4.6)]

Задание Б3
{'Конструкции': ['C++'], 'Функции': ['C#'], 'Операции': ['Python', 'JavaScript', 'TypeScript'], 'Конструкции (задания)': ['C++'], 'Функции (задания)': ['C#'], 'Операции (задания)': ['Python', 'JavaScript']}
Press any key to continue . . .

```