



**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»

ОТЧЁТ

Рубежный контроль № 1

Студент: Абрючнов Е.С. , группа ИУ5Ц-53Б

Преподаватель: Гапанюк Ю.Е.

2021 г.

Описание задания:

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

Класс «Синтаксическая конструкция» содержит поля:

- id конструкции
- название синтаксической конструкции (name)

Класс «Языки программирования» содержит поля:

- id языков программирования
- названия ЯП (name)
- процент популярности (popularity)
- id раздела, для реализации связи один-ко-многим (dep_id)

Класс «Синтаксисы и ЯП» содержит поля:

- id синтаксиса (dep_id)
- id языков программирования (emp_id)

Листинг программы:

```
"""ПК № 1, Аброчнов Егор, группа ИУ5Ц-53Б
Вариант В, № 23"""

# используется для сортировки

from operator import itemgetter

class Program:

    """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 Syntaksis:

    """Синтаксисы"""

    def __init__(self, id, name):

        self.id = id
        self.name = name

class ProgramSyntaksis:

    """
    'Синтаксисы' 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 = [
    Syntaksis(1, 'Конструкции'),
    Syntaksis(2, 'Функции'),
    Syntaksis(3, 'Операции'),
    Syntaksis(11, 'Конструкции (задания)'),
    Syntaksis(22, 'Функции (задания)'),
    Syntaksis(33, 'Операции (задания)'),
]
```

```

# Языки программирования

emps = [
    Program(1, 'Pascal', 0.3, 1),
    Program(2, 'C++', 4.5, 2),
    Program(3, 'Python', 12.1, 3),
    Program(4, 'JavaScript', 18.1, 3),
    Program(5, 'C#', 14.3, 3),
]

emps_deps = [
    ProgramSyntaksis(1,1),
    ProgramSyntaksis(2,2),
    ProgramSyntaksis(3,3),
    ProgramSyntaksis(3,4),
    ProgramSyntaksis(3,5),

    ProgramSyntaksis(11,1),
    ProgramSyntaksis(22,2),
    ProgramSyntaksis(33,3),
    ProgramSyntaksis(33,4),
    ProgramSyntaksis(33,5),
]

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 суммарной зарплате

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 ЯП

        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
[('Pascal', 0.3, 'Конструкции'), ('Python', 12.1, 'Операции'), ('JavaScript', 18.1, 'Операции'), ('C#', 14.3, 'Операции'), ('C++', 4.5, 'Функции')]
Задание Б2
[('Операции', 44.5), ('Функции', 4.5), ('Конструкции', 0.3)]
Задание Б3
{'Конструкции': ['Pascal'], 'Функции': ['C++'], 'Операции': ['Python', 'JavaScript', 'C#'], 'Конструкции (задания)': ['Pascal'], 'Функции (задания)': ['C++'], 'Операции (задания)': ['Python', 'JavaScript', 'C#']}
Press any key to continue . . .

```

