



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

ФАКУЛЬТЕТ КАФЕДРА	«Информатики и систем управления»
	ИУ5

Дисциплина «Разработка интернет-приложений»

Отчет по рубежному контролю №1
Вариант Б-22

Студент	группы ИУ5-52Б	Савченко Григорий
Преподаватель		Гапанюк Ю.Е.

Классы для предметной области

1. Класс «Библиотека», содержащий поля:
 - ID записи о библиотеке;
 - Название библиотеки;
 - ID записи о языке программирования. (для реализации связи один-ко-многим)
2. Класс «Язык программирования», содержащий поля:
 - ID записи о языке программирования;
 - Наименование языка программирования.
3. (Для реализации связи многие-ко-многим) Класс «Библиотеки языка программирования», содержащий поля:
 - ID записи о библиотеке;
 - ID записи о языке программирования.

Задание

1. «Язык программирования» и «Библиотека» связаны соотношением один-ко-многим. Выведите список всех связанных библиотек и языков программирования, отсортированный по библиотекам, сортировка по языкам программирования произвольная.
2. «Язык программирования» и «Библиотека» связаны соотношением один-ко-многим. Выведите список языков программирования с количеством библиотек в каждом языке, отсортированный по количеству библиотек.
3. «Язык программирования» и «Библиотека» связаны соотношением многие-ко-многим. Выведите список всех библиотек, у которых название заканчивается на «у», и названия языков программирования, в которые они установлены.

Код

```
#Вариант Б 22; Савченко Григорий Александрович ИУ5-52Б
```

```
# используется для сортировки
```

```
from operator import itemgetter
```

```
class Library:
```

```
    """Библиотека"""
```

```
    def __init__(self, id, name, programming_language_id):
```

```
        self.id = id
```

```
        self.name = name
```

```
        self.programming_language_id = programming_language_id
```

```
class ProgrammingLanguage:
```

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

```

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

class LibraryProgrammingLanguage:
    """
    'Библиотеки языков программирования' для реализации
    связи многие-ко-многим
    """
    def __init__(self, programming_language_id, library_id):
        self.programming_language_id = programming_language_id
        self.library_id = library_id

# Языки программирования
programming_languages = [
    ProgrammingLanguage(1, 'Python'),
    ProgrammingLanguage(2, 'C++'),
    ProgrammingLanguage(3, 'C#'),
    ProgrammingLanguage(11, 'Java'),
    ProgrammingLanguage(22, 'JavaScript'),
    ProgrammingLanguage(33, 'PHP'),
]

# Библиотеки
libraries = [
    Library(1, 'Numpy', 1),
    Library(2, 'Vector', 1),
    Library(3, 'String', 3),
    Library(4, 'Int', 11),
    Library(5, 'Double', 11),
    Library(6, 'Twig', 33),
    Library(7, 'Queue', 11),
]

libraries_programming_languages = [
    LibraryProgrammingLanguage(1,1),
    LibraryProgrammingLanguage(2,2),
    LibraryProgrammingLanguage(3,3),
    LibraryProgrammingLanguage(22,4),
    LibraryProgrammingLanguage(3,5),

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

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

```

```

# Соединение данных один-ко-многим
one_to_many = [(l.name, p.name)
                for p in programming_languages
                for l in libraries
                if l.programming_language_id==p.id]

# Соединение данных многие-ко-многим
many_to_many_temp = [(l.name, pl.programming_language_id, pl.library_id)
                     for l in programming_languages
                     for pl in libraries_programming_languages
                     if l.id==pl.programming_language_id]

many_to_many = [(p.name, programming_language_name)
                for programming_language_name, programming_language_id, library_id in many_to_many_temp
                for p in libraries if p.id==library_id]

print('Задание Б1')
res_11 = sorted(one_to_many, key=itemgetter(0))
print(res_11)

print('\nЗадание Б2')
res_12_unsorted = []
# Перебираем все языки программирования
for p in programming_languages:
    # Список библиотек
    l_languages = list(filter(lambda i: i[1]==p.name, one_to_many))
    # Если язык программирования не пустой
    if len(l_languages) > 0:
        res_12_unsorted.append((p.name, len(l_languages)))

# Сортировка по количеству библиотек
res_12 = sorted(res_12_unsorted, key=itemgetter(1), reverse=True)
print(res_12)

print('\nЗадание Б3')
res_13 = {}
# Перебираем все библиотеки
for l in libraries:
    if l.name[-1]=="y":
        # Список библиотек компьютера
        l_libraries = list(filter(lambda i: i[0]==l.name, many_to_many))
        # Только наименования языков программирования
        d_libraries_names = [x for _,x in l_libraries]
        # Добавляем результат в словарь
        # ключ - библиотека, значение - список названий языков программирован
ия
        res_13[l.name] = d_libraries_names

print(res_13)

```

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

Результат выполнения программы:

Задание Б1

[('Double', 'Java'), ('Int', 'Java'), ('Numpy', 'Pyhton'), ('Queue', 'Java'), ('String', 'C#'), ('Twig', 'PHP'), ('Vector', 'Pyhton')]

Задание Б2

[('Java', 3), ('Pyhton', 2), ('C#', 1), ('PHP', 1)]

Задание Б3

{'Numpy': ['Pyhton', 'Java']}