**Московский государственный технический**

**университет им. Н.Э. Баумана**

Факультет «Информатика и системы управления»

Кафедра ИУ5 «Системы обработки информации и управления»

Курс «Разработка интернет приложений»

Рубежный контроль №1

Вариант №13Б

|  |  |  |
| --- | --- | --- |
| Выполнил: |  | Проверил: |
| студент группы ИУ5-52Б |  | преподаватель каф. ИУ5 |
| Корчевский Александр |  | Гапанюк Ю.Е. |
|  |  |  |

Москва, 2021 г.

**Описание задания:**

1. Необходимо создать два класса данных в соответствии с Вашим вариантом предметной области, которые связаны отношениями один-ко-многим и многие-ко-многим.
2. Необходимо создать списки объектов классов, содержащих тестовые данные (3-5 записей), таким образом, чтобы первичные и вторичные ключи соответствующих записей были связаны по идентификаторам.
3. Необходимо разработать запросы в соответствии с Вашим вариантом. Запросы сформулированы в терминах классов «Сотрудник» и «Отдел», которые используются в примере. Вам нужно перенести эти требования в Ваш вариант предметной области. При разработке запросов необходимо по возможности использовать функциональные возможности языка Python (list/dict comprehensions, функции высших порядков).

**Вариант Б.**

1. «Книга» и «Библиотека» связаны соотношением один-ко-многим. Выведите список всех связанных книг и библиотек, отсортированный по названиям книг, сортировка по библиотекам произвольная.
2. «Книга» и «Библиотека» связаны соотношением один-ко-многим. Выведите список библиотек с количеством книг в каждой библиотеке, отсортированный по количеству книг.
3. «Книга» и «Библиотека» связаны соотношением многие-ко-многим. Выведите список всех книг, фамилия автора которых заканчивается на «ов», и библиотеки, в которых они есть.

**Текст программы**

# используется для сортировки

from operator import itemgetter

# Книга и библиотека

class Book:

"""Книга"""

def \_\_init\_\_(self, id, title, author, pub\_year, lib\_id):

self.id = id

self.author = author

self.title = title

self.pub\_year = pub\_year

self.lib\_id= lib\_id

class Library:

"""Библиотека"""

def \_\_init\_\_(self, id, name, phone, email):

self.id = id

self.name = name

self.phone = phone

self.email = email

class BookLib:

"""

'Книги библиотеки' для реализации

связи многие-ко-многим

"""

def \_\_init\_\_(self, lib\_id, book\_id):

self.lib\_id = lib\_id

self.book\_id = book\_id

# Книги

books = [

Book(1, 'Преступление и наказание', 'Федор Достоевский', 1866, 1),

Book(2, 'Война и мир', 'Лев Толстой', 1865, 1),

Book(3, 'Старик и море', 'Эрнест Хемингуей', 1952, 1),

Book(4, 'Превращение', 'Франц Кафка', 1915, 1),

Book(5, 'Циники', 'Анатолий Мариенгоф', 1928, 2),

Book(7, 'Числа', 'Виктор Пелевин', 2003, 2),

Book(7, 'Хромая судьба', 'Аркадий и Борис Стругацкие', 1986, 2),

Book(8, 'Наивно супер', 'Эрланд Лу', 1996, 3),

Book(9, 'Чевенгур', 'Андрей Платонов', 1972, 3),

Book(10, 'Горе от ума', 'Александр Грибоедов', 1825, 3),

]

# Библиотеки

libs = [

Library(1, 'Библиотека Иностранной Литературы', '+7 (495) 915–36–41', 'spravka@libfl.ru'),

Library(2, 'Российская Государственная Библиотека', '+7 (800) 100-57-90', 'nbros@rsl.ru'),

Library(3, 'Библиотека им. Ф.М. Достоевского', '+7495 917 31 56', 'dostoevskylib@gmail.com'),

]

books\_libs = [

BookLib(1, 8),

BookLib(1, 3),

BookLib(1, 4),

BookLib(2, 5),

BookLib(2, 6),

BookLib(2, 7),

BookLib(3, 1),

BookLib(3, 2),

BookLib(3, 9),

BookLib(3, 10),

]

def main():

"""Основная функция"""

# Соединение данных один-ко-многим

one\_to\_many = [

(b.title, b.author, l.name)

for b in books

for l in libs

if b.lib\_id == l.id

]

# Соединение данных многие-ко-многим

many\_to\_many\_temp = [(l.name, bl.lib\_id, bl.book\_id)

for l in libs

for bl in books\_libs

if l.id == bl.lib\_id]

many\_to\_many = [(b.title, b.author, lib\_name)

for lib\_name, lib\_id, book\_id in many\_to\_many\_temp

for b in books if b.id == book\_id]

print('Задание Б1')

res1 = sorted(one\_to\_many, key=itemgetter(1)) # Сортировка по названиям

print(res1)

print('\nЗадание Б2') # Список библиотек с количеством книг в каждой библиотеке

res2 = []

# Перебираем все библиотеки

for l in libs:

# Список библиотек

l\_books = list(filter(lambda i: i[2] == l.name, one\_to\_many))

# Если библиотека не пустая

if len(l\_books) > 0:

res2.append((l.name, len(l\_books)))

res2 = sorted(res2, key=lambda item: item[1], reverse=True)

print(res2)

print('\nЗадание Б3') # Список авторов, у которых фамилия заканчивается на 'ов' и библиотеки

res3 = {}

for b in books:

if str(b.author).endswith('ов'):

l\_books = list(filter(lambda i: i[1] == b.author, many\_to\_many))

l\_books\_authors = [x for \_, \_, x in l\_books]

res3[b.author] = l\_books\_authors

print(res3)

if \_\_name\_\_ == '\_\_main\_\_':

main()

**Экранная форма с результатом выполнения программы:**

