Кокозов С.И. ИУ5-53Б 13 Вариант РК1

Вариант 13

Вариант В.

* «Библиотеки» и «Книги» связаны соотношением один-ко-многим. Выведите список всех программ, у которых название начинается с буквы «И», и названия их компьютеров.
* «Библиотеки» и «Книги» связаны соотношением один-ко-многим. Выведите список программ с минимальным размером в каждом компьютере, отсортированный по минимальному размеру.
* «Библиотеки» и «Книги» связаны соотношением многие-ко-многим. Выведите список всех связанных программ и компьютеров, отсортированный по размеру, сортировка по компьютерам произвольная.

Исходный код:

# используется для сортировки

from operator import itemgetter

class Book:

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

self.id = id

self.name = name

class Libertycity:

def \_\_init\_\_(self, id, name,size, id\_of\_book):

self.id = id

self.name = name

self.size = size

self.bk\_id = id\_of\_book

class Libertycity\_Book:

def \_\_init\_\_(self, id\_of\_book, id\_of\_Libertycity):

self.id\_of\_book = id\_of\_book

self.id\_of\_Libertycity = id\_of\_Libertycity

Book = [

Book(1, 'Книга 1'),

Book(2, 'Книга 2'),

Book(3, 'Книга 3'),

Book(11, 'Четвертая книга'),

Book(22, 'Пятая книга'),

Book(33, 'Шестая книга')

]

Libertycity = [

Libertycity(1,'Имени Ленина', 2200, 1),

Libertycity(2,'1 библиотека', 2500, 3),

Libertycity(3,'Бауманская библиотека', 2200, 3),

Libertycity(4,'Главная библиотека всеся руси', 5000, 2),

Libertycity(5,'Бибофф', 8000, 3)

]

Libertycity\_Book = [

Libertycity\_Book(1,1),

Libertycity\_Book(2,2),

Libertycity\_Book(3,3),

Libertycity\_Book(3,4),

Libertycity\_Book(5,5),

Libertycity\_Book(11,1),

Libertycity\_Book(22,2),

Libertycity\_Book(33,3),

Libertycity\_Book(33,4),

Libertycity\_Book(33,5)

]

def main():

one\_to\_many = [( b.name, b.size, a.name)

for a in Book

for b in Libertycity

if b.bk\_id == a.id]

# Соединение данных многие-ко-многим

many\_to\_many\_temp = [(c.name, d.id\_of\_book, d.id\_of\_Libertycity)

for c in Book

for d in Libertycity\_Book

if c.id == d.id\_of\_book]

many\_to\_many = [(b.name, b.size, name\_of\_book)

for name\_of\_book, id\_of\_book, id\_of\_Libertycity in many\_to\_many\_temp

for b in Libertycity if b.id== id\_of\_Libertycity]

print('Задание В1')

first\_res = {}

for l in Libertycity:

if 'И' == l.name[0]:

libertycity = list((filter(lambda i: i[0] == l.name, one\_to\_many)))

l\_stud\_names = [f[2] for f in libertycity]

first\_res[l.name] = l\_stud\_names

print(first\_res)

print('\nЗадание B2')

second\_res\_unsort = []

for d in Book:

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

if len(libertycity) > 0:

second\_res\_unsort.append((d.name, min([a[1] for a in libertycity])))

second\_res = sorted(second\_res\_unsort, key=itemgetter(1))

print(second\_res)

print('\nЗадание B3')

third\_res = sorted(many\_to\_many, key=itemgetter(0))

print(third\_res)

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

main()

Вывод программы:

Задание В1

{'Имени Ленина': ['Книга 1']}

Задание B2

[('Книга 1', 2200), ('Книга 3', 2200), ('Книга 2', 5000)]

Задание B3

[('1 библиотека', 2500, 'Книга 2'), ('1 библиотека', 2500, 'Пятая книга'), ('Бауманская библиотека', 2200, 'Книга 3'), ('Бауманская библиотека', 2200, 'Шестая книга'), ('Бибофф', 8000, 'Шестая книга'), ('Главная библиотека всеся руси', 5000, 'Книга 3'), ('Главная библиотека всеся руси', 5000, 'Шестая книга'), ('Имени Ленина', 2200, 'Книга 1'), ('Имени Ленина', 2200, 'Четвертая книга')]