

## Текст программы

Файл main.py

```
from collections import Counter
from operator import itemgetter
from typing import Dict, List

import data
from entities import CDLibrary, CD, CDLibraryCDs

def get_one_to_many_binding(libraries: List[CDLibrary], cds: List[CD]):
    return [(cd.title, cd.author, library.name)
            for library in libraries
            for cd in cds
            if cd.library_id == library.id]

def get_many_to_many_binding(libraries: List[CDLibrary], libraries_cds:
List[CDLibraryCDs], cds: List[CD]):
    many_to_many_temp = [(library.name, libraries_cd.cd_library_id,
libraries_cd.cd_id)
                        for library in libraries
                        for libraries_cd in libraries_cds
                        if library.id == libraries_cd.cd_library_id]

    return [(cd.id, cd.title, cd.author, library_name)
            for library_name, cd_library_id, cd_id in many_to_many_temp
            for cd in cds if cd.id == cd_id]

def get_quantity_of_cds_in_libraries(libraries: List[CDLibrary], cds:
List[CD]):
    libraries_counter = Counter([cd.library_id for cd in cds])
    return [(l.name, libraries_counter[l.id]) for l in libraries]

def get_library_list_for_each_cd(many_to_many, cds: List[CD]):
    result: Dict[str, List[str]] = {}
    for cd in cds:
        libraries_with_cd = [m[3] for m in filter(lambda i: i[0] == cd.id,
many_to_many)]
        result[cd.title] = libraries_with_cd

    return result

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

```

# Соединение данных один-ко-многим
one_to_many = get_one_to_many_binding(data.libraries, data.cds)

# Соединение данных многие-ко-многим
many_to_many = get_many_to_many_binding(data.libraries,
data.libraries_cds, data.cds)

# «Библиотека CD дисков» и «CD диск» связаны соотношением один-ко-
многим.
# Выведите список всех связанных CD дисков и библиотек, отсортированный
по названию диска,
# сортировка по библиотекам произвольная.
print('Задание B1')
result_a = sorted(one_to_many, key=itemgetter(0))
print(result_a)

# «Библиотека CD дисков» и «CD диск» связаны соотношением один-ко-
многим.
# Выведите список библиотек с количеством дисков в каждой библиотеке,
отсортированный по количеству дисков.
print('\nЗадание B2')
result_b_unsorted = get_quantity_of_cds_in_libraries(data.libraries,
data.cds)
print(sorted(result_b_unsorted, key=itemgetter(1), reverse=True))

# «Библиотека CD дисков» и «CD диск» связаны соотношением многие-ко-
многим.
# Выведите список всех дисков, у которых название заканчивается на «e»,
и названия их библиотек.
print('\nЗадание B3')
print(get_library_list_for_each_cd(many_to_many, list(filter(lambda x:
x.title.endswith('e'), data.cds))))

if __name__ == '__main__':
    main()

```

Файл entities.py

```

class CD:
    """CD диск"""

    def __init__(self, id_: int, title: str, author: str, library_id: int):
        self.id = id_
        self.title = title
        self.author = author

        self.library_id = library_id

class CDLibrary:
    """Библиотека CD дисков"""

```

```

def __init__(self, id_: int, name: str):
    self.id = id_
    self.name = name

class CDLibraryCDs:
    """
    'CD диски библиотеки' для реализации
    связи многие-ко-многим
    """

    def __init__(self, cd_id: int, cd_library_id: int):
        self.cd_id = cd_id
        self.cd_library_id = cd_library_id

```

Файл data.py

```

from entities import CDLibrary, CD, CDLibraryCDs

libraries = [
    CDLibrary(1, 'Tech House'),
    CDLibrary(2, 'Techno'),
    CDLibrary(3, 'Bangers'),
]

cds = [
    CD(1, 'Ferrari', 'James Hype', 1),
    CD(2, 'Dancing', 'James Hype', 1),
    CD(3, 'Music From Space', 'Horeno', 2),
    CD(4, 'The Age Of Love', 'Age Of Love', 2),
    CD(5, 'The X File', 'Matchy', 2),
]

libraries_cds = [
    CDLibraryCDs(1, 1),
    CDLibraryCDs(1, 3),

    CDLibraryCDs(2, 1),
    CDLibraryCDs(2, 3),

    CDLibraryCDs(3, 2),

    CDLibraryCDs(4, 2),
    CDLibraryCDs(4, 3),

    CDLibraryCDs(5, 2),
]

```

Файл tests.py

```

import unittest

import data
from main import get_one_to_many_binding, get_many_to_many_binding,
get_quantity_of_cds_in_libraries, \
    get_library_list_for_each_cd

class TestFunctions(unittest.TestCase):
    def setUp(self):
        self.data = data

    def test_get_one_to_many_binding(self):
        want = [('Ferrari', 'James Hype', 'Tech House'), ('Dancing', 'James
Hype', 'Tech House'),
                ('Music From Space', 'Horeno', 'Techno'), ('The Age Of
Love', 'Age Of Love', 'Techno'),
                ('The X File', 'Matchy', 'Techno')]
        actual = get_one_to_many_binding(self.data.libraries,
self.data.cds)

        self.assertEqual(want, actual)

    def test_get_many_to_many_binding(self):
        want = [(1, 'Ferrari', 'James Hype', 'Tech House'), (2, 'Dancing',
'James Hype', 'Tech House'),
                (3, 'Music From Space', 'Horeno', 'Techno'), (4, 'The Age
Of Love', 'Age Of Love', 'Techno'),
                (5, 'The X File', 'Matchy', 'Techno'), (1, 'Ferrari',
'James Hype', 'Bangers'),
                (2, 'Dancing', 'James Hype', 'Bangers'), (4, 'The Age Of
Love', 'Age Of Love', 'Bangers')]
        actual = get_many_to_many_binding(self.data.libraries,
self.data.libraries_cds, self.data.cds)

        self.assertEqual(want, actual)

    def test_get_quantity_of_cds_in_libraries(self):
        want = [('Techno', 3), ('Tech House', 2), ('Bangers', 0)]
        actual = get_quantity_of_cds_in_libraries(self.data.libraries,
self.data.cds)

        self.assertEqual(want, actual)

    def test_get_library_list_for_each_cd(self):
        many_to_many = get_many_to_many_binding(self.data.libraries,
self.data.libraries_cds, self.data.cds)
        want = {'Ferrari': ['Tech House', 'Bangers'], 'Dancing': ['Tech
House', 'Bangers'],
                'Music From Space': ['Techno'], 'The Age Of Love':
['Techno', 'Bangers'], 'The X File': ['Techno']}
        actual = get_library_list_for_each_cd(many_to_many, self.data.cds)

```

```
self.assertDictEqual(want, actual)
```

## Результаты выполнения

### Задание Б1

```
[('Dancing', 'James Hype', 'Tech House'), ('Ferrari', 'James Hype',  
'Tech House'), ('Music From Space', 'Horeno', 'Techno'), ('The Age Of  
Love', 'Age Of Love', 'Techno'), ('The X File', 'Matchy', 'Techno')]
```

### Задание Б2

```
[('Techno', 3), ('Tech House', 2), ('Bangers', 0)]
```

### Задание Б3

```
{'Music From Space': ['Techno'], 'The Age Of Love': ['Techno',  
'Bangers'], 'The X File': ['Techno']}
```

## Результаты выполнения тестов

....

-----

Ran 4 tests in 0.000s

OK