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 Б1')  
 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 Б2')  
 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 Б3')  
 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.assertCountEqual(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.assertCountEqual(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.assertCountEqual(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