

Кулыгин Е. И. РТ5-31Б

Вариант Е 13

Вариант предметной области: Книга – Библиотека

1. «Библиотека» и «Книга» связаны соотношением один-ко-многим. Выведите список всех библиотек, у которых в названии присутствует слово «Библиотека» (учитывая регистр), и список книг в каждой из них
2. «Библиотека» и «Книга» связаны соотношением один-ко-многим. Выведите список библиотек со средним количеством страниц в каждой библиотеке, отсортированный по среднему количеству страниц
3. «Библиотека» и «Книга» связаны соотношением многие-ко-многим. Выведите список всех книг, название которых начинается с заглавной русской буквы 'К', и названия их библиотек

Исходный текст программы:

Файл book_lib/book.py

```
class Book:

    # текущее кол-во книг
    counter: int = 0

    @staticmethod
    def getCounter() -> int:
        return Book.counter

    @staticmethod
    def addCounter() -> None:
        Book.counter+=1

    # конструктор по всем параметрам
    def __init__(self, name: str, numOfPages: str, libID: str) -> None:
        Book.addCounter()
        self._catalogNumber: int = Book.getCounter()
        self._name: str = name
        self._numOfPages: int = numOfPages
        self._libID: int = libID

    # свойство - номер в каталоге
    @property
    def catalogNumber(self) -> int: return self._catalogNumber
    @catalogNumber.setter
    def catalogNumber(self, num: int) -> None: self._catalogNumber = num
```

```

# свойство - название книги
@property
def name(self) -> str: return self._name
@name.setter
def name(self, name: str) -> None: self._name = name

# свойство - кол-во страниц
@property
def numOfPages(self) -> int: return self._numOfPages
@numOfPages.setter
def numOfPages(self, num: int) -> None: self._numOfPages = num

# свойство - id библиотеки
@property
def libID(self) -> int: return self._libID
@libID.setter
def libID(self, num: int) -> None: self._libID = num

# to string
def __repr__(self) -> str: return self.name

```

файл book_lib/library.py

```

class Library:

    # текущее кол-во библиотек
    counter: int = 0

    @staticmethod
    def getCounter() -> int:
        return Library.counter

    @staticmethod
    def addCounter() -> None:
        Library.counter+=1

    def __init__(self, name: str) -> None:
        Library.addCounter()
        self._id: int = Library.getCounter()
        self._name: str = name

    # свойство - id библиотеки
    @property
    def id(self) -> int: return self._id
    @id.setter
    def id(self, id: int) -> None: self._id = id

    # свойство - название библиотеки
    @property
    def name(self) -> str: return self._name

```

```

@name.setter
def name(self, name: str) -> None: self._name = name

# to string
def __repr__(self) -> str: return self.name

```

файл book_lib/booklib.py

```

class BookLib:
    def __init__(self, bookID: int, libID: int) -> None:
        self._bookID: int = bookID
        self._libID: int = libID

    @property
    def bookID(self) -> int: return self._bookID
    @bookID.setter
    def bookID(self, bookID: int) -> None: self._bookID = bookID

    @property
    def libID(self) -> int: return self._libID
    @libID.setter
    def libID(self, libID: int) -> None: self._libID = libID

```

файл main.py

```

from book_lib.book import Book
from book_lib.library import Library
from book_lib.booklib import BookLib
from operator import itemgetter

# массив, содержащий объекты "Библиотека"
libs = [
    Library("Библиотека имени Пушкина"), #id 1
    Library("Центральная библиотека"), #id 2
    Library("Детская библиотека"), #id 3
    Library("Библиотека имени Ленина"), #id 4
    Library("Библиотека МГТУ") #id 5
]

# массив, содержащий объекты "Книга"
books = [
    Book("Капитанская дочка", 112, 1), # id 1
    Book("Упражнения по математическому анализу", 218, 5), # id 2
    Book("Колобок", 15, 5), #id 3
    Book("Сказки для детей", 70, 3), #id 4
    Book("Computer Science", 280, 1), # id 5
    Book("Чистый код", 300, 5), # id 6
    Book("Солярис", 200, 4), # id 7
    Book("Краткий курс по выращиванию кабачков", 78, 4), # id 8

```

```

    Book("Обломов", 419, 1), # id 9
    Book("Красная шапочка", 45, 3) # id 10
]

# массив книг и библиотек для реализации связи многие ко многим
booklibs = [
    BookLib(1, 1),
    BookLib(1, 3),
    BookLib(2, 5),
    BookLib(3, 5),
    BookLib(4, 3),
    BookLib(3, 3),
    BookLib(5, 1),
    BookLib(5, 5),
    BookLib(6, 5),
    BookLib(7, 4),
    BookLib(7, 2),
    BookLib(8, 4),
    BookLib(9, 1),
    BookLib(9, 4),
    BookLib(9, 3),
    BookLib(8, 2),
    BookLib(10, 3)
]

# основная программа
def main() -> None:

    # списки данных - связь один ко многим
    one_to_many: list[tuple] = [(b.catalogNumber, b.name, l.id, l.name)
    for l in libs
    for b in books
    if (b.libID == l.id)
    ]
    print("\nМассив связей один ко многим: " + str(one_to_many))

    # списки данных - массив многих ко многим (временный)
    many_to_many_temp: list[tuple] = [(b.name, bl.bookID, bl.libID)
    for b in books
    for bl in booklibs
    if b.catalogNumber == bl.bookID
    ]

    # списки данных - массив многих ко многим (временный)
    many_to_many: list[tuple] = [(mm_bookID, mm_book_name, l.id, l.name)
    for l in libs
    for mm_book_name, mm_bookID, mm_libID in many_to_many_temp
    if l.id == mm_libID
    ]
    print("\nМассив связей многие ко многим: " + str(many_to_many))

```

```

# вывод - 1 задание
# вывести список всех библиотек, которые имеют в названии "Библиотека"
(учитывая регистр)
# для каждой библиотеки - список книг в ней
print("\n Задача E1\n")
res1 = [(l, list(filter(lambda i: i.libID == l.id, books)))
for l in libs
if ("Библиотека" in l.name)
]
print(res1)

# вывод - 2 задание
# вывести список библиотек со средним кол-вом страниц в книгах в каждой
библиотеке
# сортировать по кол-ву страниц по убыванию
print("\n Задача E2")
res2: tuple = []
for l in libs:
    currentBooks: list[Book] = list(filter(lambda i: i.libID == l.id, books))
    pagesCount: list[int] = []
    if (len(currentBooks) > 0):
        pagesCount = [pg.numOfPages for pg in currentBooks]

        averagePagesCount: float = 0
        if len(pagesCount) > 0: averagePagesCount =
round(sum(pagesCount)/len(pagesCount), 2)
        res2.append((l.name, averagePagesCount))

res2.sort(key=itemgetter(1), reverse=True)
print(res2)

# вывод - 3 задание
# вывести список книг, название которых начинается с заглавной русской буквы
'К'
# и название их библиотек
print("\n Задача E3")
res3: dict = {}
for mm_book_id, mm_book_name, mm_lib_id, mm_lib_name in many_to_many:
    if (mm_book_name[0] == 'К'):
        if (mm_book_name not in res3): res3[mm_book_name] = []
        res3[mm_book_name].append(mm_lib_name)

print(res3)

if (__name__ == "__main__"):
    main()

```

Пример работы программы:

Массив связей один ко многим: [(1, 'Капитанская дочка', 1, 'Библиотека имени Пушкина'), (5, 'Computer Science', 1, 'Библиотека имени Пушкина'), (9, 'Обломов', 1, 'Библиотека имени Пушкина'), (4, 'Сказки для детей', 3, 'Детская библиотека'), (10, 'Красная шапочка', 3, 'Детская библиотека'), (7, 'Солярис', 4, 'Библиотека имени Ленина'), (8, 'Краткий курс по выращиванию кабачков', 4, 'Библиотека имени Ленина'), (2, 'Упражнения по математическому анализу', 5, 'Библиотека МГТУ'), (3, 'Колобок', 5, 'Библиотека МГТУ'), (6, 'Чистый код', 5, 'Библиотека МГТУ')]

Массив связей многие ко многим: [(1, 'Капитанская дочка', 1, 'Библиотека имени Пушкина'), (5, 'Computer Science', 1, 'Библиотека имени Пушкина'), (9, 'Обломов', 1, 'Библиотека имени Пушкина'), (7, 'Солярис', 2, 'Центральная библиотека'), (8, 'Краткий курс по выращиванию кабачков', 2, 'Центральная библиотека'), (1, 'Капитанская дочка', 3, 'Детская библиотека'), (3, 'Колобок', 3, 'Детская библиотека'), (4, 'Сказки для детей', 3, 'Детская библиотека'), (9, 'Обломов', 3, 'Детская библиотека'), (10, 'Красная шапочка', 3, 'Детская библиотека'), (7, 'Солярис', 4, 'Библиотека имени Ленина'), (8, 'Краткий курс по выращиванию кабачков', 4, 'Библиотека имени Ленина'), (9, 'Обломов', 4, 'Библиотека имени Ленина'), (2, 'Упражнения по математическому анализу', 5, 'Библиотека МГТУ'), (3, 'Колобок', 5, 'Библиотека МГТУ'), (5, 'Computer Science', 5, 'Библиотека МГТУ'), (6, 'Чистый код', 5, 'Библиотека МГТУ')]

Задача E1

[(Библиотека имени Пушкина, [Капитанская дочка, Computer Science, Обломов]), (Библиотека имени Ленина, [Солярис, Краткий курс по выращиванию кабачков]), (Библиотека МГТУ, [Упражнения по математическому анализу, Колобок, Чистый код])]

Задача E2

[('Библиотека имени Пушкина', 270.33), ('Библиотека МГТУ', 177.67),
('Библиотека имени Ленина', 139.0), ('Детская библиотека', 57.5),
('Центральная библиотека', 0)]

Задача E3

{'Капитанская дочка': ['Библиотека имени Пушкина', 'Детская библиотека'],
'Краткий курс по выращиванию кабачков': ['Центральная библиотека',
'Библиотека имени Ленина'], 'Колобок': ['Детская библиотека', 'Библиотека
МГТУ'], 'Красная шапочка': ['Детская библиотека']}