

**Московский государственный технический
университет им. Н.Э. Баумана**

Факультет «Информатика и системы управления»
Кафедра ИУ5 «Системы обработки информации и управления»

Курс «ПиКЯП»

Отчет по рк2

Выполнил:

студент группы ИУ5-23Б

Илюхин И. Д.

Подпись и дата: 16.12.2024

Проверил:

преподаватель каф. ИУ5

Гапанюк Ю. Е

Подпись и дата:

Москва, 2024 г.

Условия рубежного контроля №2 по курсу Пик ЯП

Рубежный контроль представляет собой разработку тестов на языке Python.

- 1) Проведите рефакторинг текста программы рубежного контроля №1 таким образом, чтобы он был пригоден для модульного тестирования.
- 2) Для текста программы рубежного контроля №1 создайте модульные тесты с применением TDD - фреймворка (3 теста).

Файл "refactored_rk1.py"

```
from operator import itemgetter

class Book:
    def __init__(self, id, name, creator, date):
        self.id = id
        self.name = name
        self.creator = creator
        self.date = date

class Library:
    def __init__(self, id, name):
        self.id = id
        self.name = name

class BookOfLibrary: # библиотека-книга
    def __init__(self, library_id, book_id):
        self.library_id = library_id
        self.book_id = book_id

# Данные
libraries = [
    Library(1, 'Центральная библиотека'),
    Library(2, 'Библиотека им. Пушкина'),
    Library(3, 'Детская библиотека'),
]

books = [
    Book(1, 'Война и мир', 'Л.Н. Толстой', 1869),
    Book(2, 'Преступление и наказание', 'Ф.М. Достоевский', 1866),
    Book(3, 'Мастер и Маргарита', 'М.А. Булгаков', 1967),
    Book(4, 'Анна Каренина', 'Л.Н. Толстой', 1877),
    Book(5, 'Братья Карамазовы', 'Ф.М. Достоевский', 1880),
]

books_of_library = [
    BookOfLibrary(1, 1),
    BookOfLibrary(1, 2),
    BookOfLibrary(2, 3),
    BookOfLibrary(2, 4),
```

```

        BookOfLibrary(3, 5),
        BookOfLibrary(1, 3),
        BookOfLibrary(2, 1),
        BookOfLibrary(3, 2),
        BookOfLibrary(3, 4),
        BookOfLibrary(2, 5),
    ]

# Функции

def get_many_to_many(libraries, books, books_of_library):
    """Создает список связей многие-ко-многим"""
    many_to_many_temp = [
        (library.name, rel.library_id, rel.book_id)
        for library in libraries for rel in books_of_library if library.id ==
rel.library_id
    ]

    many_to_many = [
        (book.name, book.creator, library_name)
        for library_name, _, book_id in many_to_many_temp
        for book in books if book.id == book_id
    ]
    return many_to_many

def task_a1(many_to_many):
    """Задание A1: Сортировка книг по названию библиотеки"""
    return sorted(many_to_many, key=itemgetter(2))

def task_a2(libraries, many_to_many):
    """Задание A2: Подсчет уникальных авторов в каждой библиотеке"""
    result = []
    for library in libraries:
        books_in_library = list(filter(lambda x: x[2] == library.name, many_to_many))
        if books_in_library:
            unique_authors = list(set(book[1] for book in books_in_library))
            result.append((library.name, len(unique_authors)))
    return sorted(result, key=itemgetter(1), reverse=True)

def task_a3(libraries, many_to_many):
    """Задание A3: Список книг по каждой библиотеке"""
    result = {}
    for library in libraries:
        books_in_library = list(filter(lambda x: x[2] == library.name, many_to_many))
        book_names = [book[0] for book in books_in_library]
        result[library.name] = book_names
    return result

def main():
    many_to_many = get_many_to_many(libraries, books, books_of_library)

    print('Задание A1')
    print(task_a1(many_to_many))

```

```

    print('Задание A2')
    print(task_a2(libraries, many_to_many))

    print('Задание A3')
    print(task_a3(libraries, many_to_many))

if __name__ == '__main__':
    main()

```

файл main.py

```

from operator import itemgetter

class book:
    def __init__(self, id, name, creator, date):
        self.id = id
        self.name = name
        self.creator = creator
        self.date = date

class library:
    def __init__(self, id, name):
        self.id = id
        self.name = name

class book_of_library: #библиотека-книга
    def __init__(self, library_id, book_id):
        self.library_id = library_id
        self.book_id = book_id

library = [
    library(1, 'Центральная библиотека'),
    library(2, 'Библиотека им. Пушкина'),
    library(3, 'Детская библиотека'),
]

book = [
    book(1, 'Война и мир', 'Л.Н. Толстой', 1869),
    book(2, 'Преступление и наказание', 'Ф.М. Достоевский', 1866),
    book(3, 'Мастер и Маргарита', 'М.А. Булгаков', 1967),
    book(4, 'Анна Каренина', 'Л.Н. Толстой', 1877),
    book(5, 'Братья Карамазовы', 'Ф.М. Достоевский', 1880),
]

book_of_library = [
    book_of_library(1, 1),
    book_of_library(1, 2),
    book_of_library(2, 3),

```

```

        book_of_library(2, 4),
        book_of_library(3, 5),

        book_of_library(1, 3),
        book_of_library(2, 1),
        book_of_library(3, 2),
        book_of_library(3, 4),
        book_of_library(2, 5),
    ]

#####
def main():
    many_to_many_temp = [
        (i.name, j.library_id, j.book_id)
        for i in library for j in book_of_library if i.id == j.library_id]

    many_to_many = [(k.name, k.creator, b_name)
                     for b_name, lib_id, book_id in many_to_many_temp
                     for k in book if k.id == book_id]

    print('Задание A1')
    task_1 = sorted(many_to_many, key=itemgetter(2))
    print(task_1)

    print('Задание A2')
    task_2 = []
    for b in library:
        b_books = list(filter(lambda i: i[2] == b.name, many_to_many))
        if len(b_books) > 0:
            b_authors = [a for _, a, _ in b_books]
            b_authors_unique = list(set(b_authors))
            task_2.append((b.name, len(b_authors_unique)))

    task_2_f = sorted(task_2, key=itemgetter(1), reverse=True)
    print(task_2_f)

    print('\nЗадание A3')
    task_3 = {}
    for b in library:
        b_books = list(filter(lambda i: i[2] == b.name, many_to_many))
        b_books_names = [x for x, _, _ in b_books]
        task_3[b.name] = b_books_names

    print(task_3)

if __name__ == '__main__':
    main()

```