

# Лабораторная работа №4

Тема: Библиотека с пользовательскими коллекциями и симуляцией

Выполнил: Попков Максим Александрович, М8О-105БВ-25

## Цель работы

- Реализовать модель библиотеки с книгами разных типов.
- Освоить создание пользовательских коллекций (справочная и словарная).
- Научиться поддерживать согласованность данных между коллекциями.
- Получить практику работы с логированием, рандомизацией и структурой модульного проекта.

## Ход выполнения

### 1. Модели книг

В модуле `models.py` реализованы три класса:

- `Book` — базовая модель с полями название, автор, год, жанр, ISBN.
- `FictionBook` — художественная книга с дополнительным возрастным ограничением.
- `ScienceBook` — научная книга с полем области науки.

Все классы проверяют корректность входных данных и переопределяют `__repr__`, чтобы объекты удобно отображались в логах.

### 2. Пользовательские коллекции

В файле `collections.py` реализованы:

`BookCollection` — справочная коллекция:

- добавление книги, удаление по ISBN, проверка наличия;
- поддержка индексации, срезов и итератора;
- автоматическая замена книги при совпадении ISBN.

`IndexDict` — словарная коллекция индексов:

- хранение книг по ключам isbn, author, year;
- быстрый поиск и синхронное удаление из всех индексов;
- доступ через `index[isbn]`, перебор элементов и проверку на вхождение.

Обе коллекции работают согласованно и обрабатывают ошибки (`ValueError`, `KeyError`).

### 3. Класс Library

В модуле `library.py` разработан интерфейс верхнего уровня:

- единая точка доступа к коллекции книг и индексам;
- добавление/удаление книг;
- поиск по ISBN, автору, году и жанру;
- удобная итерация по всем книгам.

### 4. Симуляция работы библиотеки

В модуле `simulation.py` реализовано ядро лабораторной работы — функция `run_simulation(steps, seed)`:

- создаются случайные книги (`Book`, `FictionBook`, `ScienceBook`);
- случайным образом выбираются операции ADD / REMOVE / FIND\_\*
- генерируются корректные и некорректные запросы (невалидный ISBN, несуществующий год);
- все действия записываются в лог.

Используются списки констант (названия, авторы, жанры и т.д.) из constants.py, а логирование выполняется через модуль log.py с выводом в файл shell.log.

## 5. Точка входа

В main.py организован простой интерфейс: ввод количества шагов симуляции (целое число  $\geq 10$ ) и запуск работы. При ошибке ввода программа корректно завершает работу.

## Результаты

- Реализована полная структура библиотеки с пользовательскими коллекциями.
- Все требования из задания выполнены: индексы, согласованность данных, поиск, удаление, логирование, псевдослучайная симуляция.
- Код модульный, документирован, без внешних зависимостей.
- Результат работы симуляции записывается в shell.log, что позволяет анализировать сценарии поведения программы.

## Вывод

В ходе выполнения лабораторной работы была разработана модель библиотеки с поддержкой нескольких типов книг, индексов и пользовательских коллекций. Реализованы основные операции поиска, добавления и удаления, а также полноценная симуляция работы с логированием. Все поставленные цели выполнены, структура проекта соответствует требованиям, практические навыки работы с классами, коллекциями, исключениями и логированием закреплены.