**Документация к приложению LibAdministration**

**LibAdministration.py**

Точка входа в программу. Присутствует метод **showMenu() -> None**, отвечающий за отображения меню, действий пользователя, вызывается после каждого обработанного запроса пользователя.

**InputManager.py**

Файл, отвечающий за обработку ввода пользователя. Метод **menuInput() -> bool** отвечает за обработку ввода для выбора функционала приложения. Возвращает True во всех случаях кроме ввода “0”, означающего завершение выполнения программы.

**addBookInput() -> None** отвечает за обработку ввода данных для новой книги.

**findBookInput() -> None** отвечает за обработку ввода данных для поиска книги.

**removeBookInput() -> None** отвечает за обработку ввода данных для удаления книги.

**changeStatusInput() -> None** отвечает за обработку ввода данных для изменения статуса существующей книги.

В случае некорректного ввода данных соответствующий метод оповещает пользователя об ошибке и переносит на экран меню. Если данные введены корректно метод передаёт их соответствующему методу из файла **LibraryController.py**

**LibraryController.py**

Файл с основными функциями программы.

**showBooksAsTable(books: List[Dict[str, Union[str, int]]]) -> None** в качестве параметра принимает лист словарей, каждый из которых описывает книгу. Отображает книги в виде таблицы.

**showAllBooks() -> None** отображает все книги, если в базе присутствует хотя бы одна, иначе оповещает об этом.

**getBooksByKey(key: str) -> None** отображает все книги, автор, название или год написания совпадает с полученным параметром или оповещает, что таковых не найдено.

**addBook(author: str, title: str, year: int) -> None** добавляет в базу книгу. Id книги генерируется автоматически, автор книги, её название и год написания передаётся в качестве параметров, статус устанавливается “in stock” (“в наличии”).

**removeBook(book\_id: int)** удаляет из базы книгу с указанным id.

**changeStatus(book\_id: int, status: str) -> None** назначает книги с указанным id, указанный статус.

**JSONManager.py**

Файл, отвечающий за работу с json файлом, хранящим данные о книгах.

**getAllBooks() -> List[Dict[str, Union[str, int]]]** получает список всех книг, описываемых словарём с ключами “id”, “author”, “title”, “year”, “status”.

**updateFile(books: Dict[str, List[Dict[str, Union[str, int]]]]) -> None** получает на вход словарь с ключом “books” и соответствующим ему значением списка словарей, описывающих книгу. Записывает эти данные в файл.

**get\_books\_quantity() -> int** получает текущее количество книг в базе библиотеки.

**checkFile() -> None** проверяет существование json файла с базой библиотеки и, в случае отсутствия, создаёт пустой файл.