

# Systemu Zarządzania Książkami

Jakub Opalinski

# Opis Projektu i Dokumentacja Systemu Zarządzania Książkami

## Przegląd Projektu:

System Zarządzania Książkami umożliwia użytkownikom dodawanie, usuwanie i wyszukiwanie książek na podstawie różnych atrybutów, takich jak imię autora, rok publikacji i tytuł.

## Sruktura projektu:

### 1. Klasa Book

#### Opis

Klasa **Book** służy do reprezentacji pojedynczej książki w systemie. Odzworowuje informacje o książce, takie jak id książki, tytuł, imię autora, nazwisko autora oraz rok publikacji.

### Konstruktor

```
__init__(self, book_id, title, author_first_name,  
author_last_name, publication_year)
```

- `book_id`: Unikalny identyfikator książki.
- `title`: Tytuł książki.
- `author_first_name`: Imię autora książki.
- `author_last_name`: Nazwisko autora książki.
- `publication_year`: Rok publikacji książki.

## 2. Klasa **BooksDatabase**

### Opis

Klasa **BooksDatabase** odpowiada za zarządzanie kolekcją książek. Umożliwia ładowanie danych z pliku JSON, zapisywanie zmian w bazie danych, dodawanie nowych książek, usuwanie książek oraz wyszukiwanie książek na podstawie różnych parametrów.

### Konstruktor

- ```
__init__(self, db_path):  
    self.db_path = db_path
```
- `db_path`: pole przechowujące ścieżkę do pliku bazy

### Metody

- `load_database()`: Ładuje bazę danych książek z pliku JSON.
- `save_database(books)`: Zapisuje aktualny stan bazy danych książek do pliku JSON.
- `get_next_id()`: Generuje kolejny unikalny identyfikator książki.
- `insert_book(title, author_first_name, author_last_name, publication_year)`: Dodaje nową książkę do bazy danych.
- `delete_book(book_id)`: Usuwa książkę z bazy danych na podstawie jej identyfikatora.
- `find_book_by_author_first_name(first_name)`,  
`find_book_by_author_last_name(last_name)`,  
`find_book_by_year(year)`, `find_book_by_title(title)`: Metody do wyszukiwania książek na podstawie różnych kryteriów.
- `list_books()`: Zwraca listę wszystkich książek w bazie danych.

## 3.Plik **main.py**

### Opis

Plik **main.py** odpowiada za interfejs użytkownika aplikacji do zarządzania książkami, umożliwiając wykonanie operacji.

### Metody

- `getAllRecords()`: Wyświetla wszystkie książki z bazy danych.
- `DisplayData(books)`: Formatuje i wyświetla książki w formie tabeli.
- `print_menu_table(menu_options)`: Wyświetla menu opcji dla użytkownika.
- `get_book_data_from_user()`: Pobiera dane książki od użytkownika i dodaje ją do bazy.
- `delete_book_by_id()`: Usuwa wybraną książkę z bazy danych.
- `find_book_by_name()`: Wyszukuje książki po imieniu autora.
- `find_book_by_second_name()`: Wyszukuje książki po nazwisku autora.
- `find_book_by_year()`: Wyszukuje książki po roku publikacji.
- `find_book_by_title()`: Wyszukuje książki po tytule.
- `get_number(menu_options)`: Prosi użytkownika o wybór opcji z menu.

## Interfejs Użytkownika

Interfejs użytkownika składa się z prostej aplikacji terminalowej, która pozwala na interakcję z systemem za pomocą tekstowego menu.

```
+---+-----+
|Menu|      Description      |
+---+-----+
1	Add record
2	Delete record
3	Search by name
4	Search by second name
5	Search by year
6	Search by Title
7	Show all records
8	Exit
+---+-----+
Please enter a number: 
```

## Uruchamianie Aplikacji

Aplikacje uruchamiamy poprzez uruchomienie pliku main.py (komenda: python3 main.py). Do uruchomienia potrzebujemy pythona3 oraz biblioteki json oraz.

## Interakcja z Aplikacją

Użytkownik nawiguje po aplikacji za pomocą konsoli, wybierając odpowiednie opcje z menu. Opcje menu odpowiadają za zarządzanie książkami, takie jak dodawanie nowych rekordów, usuwanie rekordów, wyszukiwanie książek lub wyświetlanie wszystkich rekordów.