

# Program do obsługi biblioteki - dokumentacja

Jakub Parat

## Spis treści

<b>1</b>	<b>Wstęp</b>	<b>2</b>
<b>2</b>	<b>Struktura projektu</b>	<b>2</b>
<b>3</b>	<b>Jak uruchomić program?</b>	<b>2</b>
<b>4</b>	<b>Opis klas</b>	<b>3</b>
4.1	Klasa Ksiazka . . . . .	3
4.2	Klasa Czytelnik . . . . .	3
4.3	Klasa Biblioteka . . . . .	4
4.4	Klasa Administrator . . . . .	5
4.5	Pomocnicze . . . . .	5
<b>5</b>	<b>Szczegółowy opis działania programu</b>	<b>6</b>
5.1	Opcje dostępne dla użytkownika . . . . .	6
5.2	Opcje dostępne dla administratora . . . . .	7
<b>6</b>	<b>Podsumowanie</b>	<b>7</b>

# 1 Wstęp

Projekt ten to program służący do zarządzania biblioteką, który umożliwia obsługę użytkowników, zarządzanie książkami oraz wypożyczanie ich przez czytelników. Program został napisany w języku C++. Zawiera on następujące funkcjonalności:

- Dodawanie nowych użytkowników,
- Logowanie użytkowników na podstawie imienia i ID,
- Wypożyczanie i zwracanie książek,
- Wyświetlanie listy wypożyczonych książek,
- Wyszukiwanie książek (sprawdzenie ich autora, ID oraz dostępności),
- Wyświetlanie listy dostępnych książek w kolejności alfabetycznej.

## 2 Struktura projektu

Projekt składa się z następujących plików:

- `biblioteka.h` - Plik nagłówkowy zawierający klasy `Ksiazka`, `Czytelnik`, oraz `Biblioteka` wraz z ich implementacjami.
- `biblioteka.cpp` - Plik źródłowy zawierający implementację logiki programu.
- `Makefile` - Plik pozwalający na automatyczne kompilowanie projektu. Zawiera flagi `-Wall` oraz `-std=c++11`.
- `ksiazki.txt` - Plik przechowujący dane o książkach (zawiera ID, tytuł, autora, status)
- `czytelnicy.txt` - Plik przechowujący dane o czytelnikach (zawiera ID, imię, listę wypożyczonych książek)

## 3 Jak uruchomić program?

W celu uruchomienia programu należy:

- W terminalu otwartym w miejscu znajdowania się plików projektu wykonać polecenie `make` w celu automatycznego skompilowania projektu.
- Po skompilowaniu wykonać polecenie `./biblioteka.out` w celu uruchomienia programu.

Aby usunąć wszelkie pliki powstałe na skutek kompilacji, należy wykonać polecenie `make clean`.

## 4 Opis klas

### 4.1 Klasa Ksiazka

Reprezentuje książkę w bibliotece.

- **Pola prywatne:**

- `int identyfikator` - Unikalny ID książki.
- `std::string tytul` - Tytuł książki.
- `std::string autor` - Autor książki.
- `bool czyDostepna` - Informacja, czy książka jest dostępna.

- **Metody publiczne:**

- `Ksiazka(...)` - Konstruktor klasy Ksiazka.
- `int pobierzIdentyfikator()` - Zwraca ID książki.
- `std::string pobierzTytul()` - Zwraca tytuł książki.
- `std::string pobierzAutora()` - Zwraca autora książki.
- `bool sprawdzDostepnosc()` - Sprawdza dostępność książki.
- `void wypożycz()` - Ustawia status książki jako wypożyczony.
- `void zwroc()` - Ustawia status książki jako dostępny.
- `void wyswietl()` - Wyświetla szczegóły książki.
- `void ustawDostepnosc(bool)` - Ustawia dostępność książki na podaną.
- `void zapiszDoPliku(std::ofstream)` - Zapisuje dane książki do pliku.
- `static Ksiazka wczytajZPliku(std::ifstream)` - Wczytuje książkę z pliku.

### 4.2 Klasa Czytelnik

Reprezentuje użytkownika biblioteki.

- **Pola prywatne:**

- `int identyfikator` - Unikalny ID użytkownika.
- `std::string imie` - Imię użytkownika.
- `std::vector<Ksiazka>` - Lista wypożyczonych książek.

- **Metody publiczne:**

- `Czytelnik(...)` - Konstruktor klasy Czytelnik
- `int pobierzIdentyfikator()` - Zwraca ID użytkownika.
- `std::string pobierzImie()` - Zwraca imię użytkownika.

- `void wypożyczKsiążke(Książka&)` - Dodaje książkę do listy wypożyczonych.
- `void zwrocKsiążke(Książka&)` - Zwraca książkę i usuwa ją z listy.
- `void wyswietlWypożyczoneKsiążki()` - Wyświetla książki wypożyczone przez użytkownika.
- `void zapiszDoPliku(std::ofstream)` - Zapisuje dane użytkownika do pliku.
- `static Czytelnik wczytajZPliku(std::string, std::vector<Książka>)` - Wczytuje dane czytelnika z pliku.
- `std::vector<Książka> pobierzWypożyczoneKsiążki()` - Pobiera książki wypożyczone przez użytkownika.

### 4.3 Klasa Biblioteka

Zarządza użytkownikami i książkami w bibliotece.

- **Pola prywatne:**

- `std::vector<Książka>` - Lista wszystkich książek.
- `std::vector<Czytelnik>` - Lista wszystkich użytkowników.

- **Metody publiczne:**

- `void dodajKsiążke(Książka&)` - Dodaje nową książkę.
- `int sprawdzCzyWBazieImie(std::string&)` - Sprawdza, ilu użytkowników ma dane imię.
- `bool sprawdzCzyWBazieID(int)` - Sprawdza, czy użytkownik o danym ID istnieje.
- `void dodajCzytelnika(Czytelnik&)` - Dodaje nowego użytkownika do listy użytkowników biblioteki
- `void wyszukajKsiążke(std::string&)` - Wyszukuje książkę po tytule.
- `void wypożyczKsiążke(int, int)` - Wypożycza książkę dla użytkownika.
- `void zwrocKsiążke(int, int)` - Zwraca książkę przez użytkownika.
- `void zamianaKsiążek(Książka&, Książka&)` - Zamienia miejscami dwie książki w liście książek.
- `void pokazDostępneKsiążki()` - Wyświetla listę dostępnych książek.
- `void pokazWypożyczoneKsiążki(int)` - Wyświetla książki wypożyczone przez użytkownika.
- `void zapiszDane()` - Zapisuje dane książek i czytelników do plików.
- `void wczytajDane()` - Wczytuje dane książek i użytkowników z plików.
- `int ileCzytelnikow()` - Zwraca aktualną liczbę użytkowników biblioteki.

- `std::vector<Ksiazka> pobierzKsiazki` - Zwraca listę książek w bibliotece.
- `std::vector<Czytelnik> pobierzCzytelnikow()` - Zwraca listę czytelników w bibliotece.

## 4.4 Klasa Administrator

Odpowiada za zarządzanie książkami.

- **Metody publiczne:**

- `void dodajKsiazke(Biblioteka, int, std::string, std::string)` - Dodaje nową książkę do biblioteki.
- `void zmienDostepnoscKsiazki(Biblioteka, int)` - Zmienia dostępność książki na przeciwną.
- `void pokazStatusKsiazek(Biblioteka)` - Wyświetla status wszystkich książek biblioteki.
- `void sprawdzKtoWypozyczylKsiazke(Biblioteka, int)` - Sprawdza, kto wypożyczył daną książkę.

## 4.5 Pomocnicze

- **Funkcje globalne:**

- `void czyszczenieBufora()` - Czyści bufor wejściowy, aby zapobiec problemom z przechowywanymi znakami.
- `void czyszczenieEkranu()` - Czyści ekran terminalu.
- `void kontynuacja()` - Zatrzymuje działanie programu do czasu wciśnięcia klawisza przez użytkownika.
- `std::string naMaleLitery(std::string&)` - Zamienia podany ciąg znaków na ten sam ciąg, składający się tylko z małych znaków.

## 5 Szczegółowy opis działania programu

Po uruchomieniu programu użytkownik proszony jest o podanie swojego **imienia**, które będzie pomocne przy jego kolejnym logowaniu do systemu. Możliwe są tutaj cztery scenariusze:

1. **Użytkownik o podanym imieniu nie widnieje w systemie** W tym przypadku tworzony jest nowy profil użytkownika. Użytkownik otrzymuje numer identyfikacyjny (ID) przypisany do jego konta. Po zakończeniu tego procesu użytkownik jest automatycznie zalogowany do systemu.
2. **Użytkownik o podanym imieniu widnieje w systemie** W tym przypadku użytkownik proszony jest o podanie swojego numeru identyfikacyjnego (ID), aby go odróżnić od innych użytkowników o tym samym imieniu. W zależności od poprawności wpisanego ID użytkownik zostaje zalogowany do systemu lub proszony o ponowne podanie poprawnego ID.
  - Jeśli użytkownik nie posiada jeszcze swojego ID (np. loguje się po raz pierwszy), może wpisać 0, aby system utworzył dla niego nowy profil.
3. **Użytkownik rezygnuje z próby logowania** Jeśli zamiast podania imienia użytkownik wpisze "exit", program zakończy swoje działanie.
4. **Administrator loguje się do systemu** Jeśli użytkownik wpisze "TajnyAdmin" jako imię, zostaje zalogowany do trybu administratora, w którym ma dostęp do funkcji zarządzania biblioteką.

### 5.1 Opcje dostępne dla użytkownika

Po poprawnym zalogowaniu użytkownik widzi listę dostępnych operacji:

1. **Wyszukaj książkę** Użytkownik może wyszukać książkę, wpisując jej tytuł. System przeszukuje bazę książek bez względu na wielkość liter w tytule. Jeśli książka istnieje, wyświetlane są jej szczegóły (autor, ID, status dostępności). W przypadku błędnego wpisania tytułu lub braku książki w bazie użytkownik zostanie o tym poinformowany. **Uwaga:** Można wyszukać także książki aktualnie niedostępne.
2. **Wypożycz książkę** Użytkownik może wypożyczyć książkę, podając jej ID. Jeśli książka jest dostępna, zostanie ona przypisana do konta użytkownika, a jej status zmieniony na „niedostępna”. Jeśli książka jest niedostępna lub ID jest nieprawidłowe, użytkownik zostanie o tym poinformowany.
3. **Zwróć książkę** Użytkownik może zwrócić książkę, wpisując jej ID. Jeśli książka została wypożyczona przez tego użytkownika, zwrot zostanie zarejestrowany, a książka ponownie stanie się dostępna. W przeciwnym razie użytkownik zostanie poinformowany o błędnej operacji.

4. **Pokaż swoje wypożyczone książki** Wyświetla listę książek wypożyczonych przez użytkownika w kolejności ich wypożyczenia. Jeśli użytkownik nie posiada wypożyczonych książek, zostanie wyświetlony stosowny komunikat.
5. **Pokaż dostępne książki** Wyświetla listę książek dostępnych w bibliotece w kolejności alfabetycznej według tytułów. Jeśli żadna książka nie jest aktualnie dostępna, system poinformuje użytkownika o braku dostępnych książek.
6. **Wyloguj** Powoduje wylogowanie użytkownika z systemu i przenosi go z powrotem do ekranu logowania.
7. **Wyjście** Kończy działanie programu i wyświetla komunikat pożegnalny.

## 5.2 Opcje dostępne dla administratora

W trybie administratora wyświetlana jest specjalna lista operacji:

1. **Dodaj książkę** Administrator może dodać nową książkę do biblioteki, podając jej ID, tytuł oraz autora.
2. **Zmień dostępność książki** Administrator może zmienić status książki na dostępną lub niedostępną, podając jej ID. System automatycznie przełącza status książki na przeciwny.
3. **Pokaż status książek** Wyświetla listę wszystkich książek w bibliotece wraz z ich szczegółami (ID, tytuł, autor, status dostępności).
4. **Sprawdź kto wypożyczył książkę** Administrator może sprawdzić, kto wypożyczył daną książkę, podając jej ID. System wyświetla imię i ID użytkownika, który aktualnie posiada książkę.
5. **Wyloguj** Wylogowuje administratora i przenosi go z powrotem do ekranu logowania.

Przy zamykaniu programu dane są automatycznie zapisywane, a przy ponownym uruchomieniu odczytywane.

Wszystkie z powyższych operacji wyposażone są w obsługę błędów przy wprowadzaniu konkretnych danych, w celu uniknięcia występowania w programie zachowań nieoczekiwanych.

## 6 Podsumowanie

Prosty program do obsługi biblioteki wyposażony w funkcjonalności konieczne do jej sprawnego i wygodnego działania. Umożliwia rejestrację nowych użytkowników oraz logowanie istniejących na podstawie imienia i ID. Pozwala na dodawanie książek do bazy biblioteki, wyszukiwanie książek po tytule niezależnie od wielkości liter, a także ich wypożyczanie i zwracanie. Użytkownik ma możliwość wyświetlenia

listy dostępnych książek w kolejności alfabetycznej oraz przeglądania książek aktualnie przez niego wypożyczonych. Program został wyposażony w mechanizmy walidacji danych wejściowych, co zwiększa jego niezawodność. Intuicyjny interfejs terminalowy sprawia, że korzystanie z programu jest proste i wygodne.