

# Konspekt ZPOiF

## Biblioteczka domowa

Zuzanna Glinka, Joanna Kajka, Jakub Kasprzak

Grudzień 2021

### 1 Cel projektu

W ramach projektu z przedmiotu „Zaawansowane Programowanie Obiektowe i Funkcyjne” postanowiliśmy stworzyć aplikację okienkową, ułatwiającą zarządzanie biblioteczką domową. Dzięki niej z łatwością będziemy mogli zorientować się na temat m.in. ilości książek w naszym domowym zbiorze, preferowanych przez nas gatunków, czy przeczytanych przez nas książek. Naszym założeniem jest by aplikacja była jak najbardziej przyjazna dla każdego użytkownika. W związku z tym naszym głównym celem jest stworzenie jak najbardziej intuicyjnego interfejsu. Wraz z powstawaniem aplikacji, będziemy na bieżąco modyfikować wszystkie funkcjonalności, tak aby sprostać początkowym założeniom.

### 2 Charakterystyka projektu

Nasza aplikacja pozwala stworzyć jedną domową bibliotekę, zawierającą cały księgozbiór. Ów księgozbiór zawierać będzie informacje takie jak: tytuł książki, autor, rok wydania, wydawnictwo oraz liczba stron. Dodatkowo do każdej książki możemy dodać tagi ułatwiające późniejsze wyszukiwanie, informację czy książka jest obecnie czytana lub wypożyczona komuś spoza domowników. Każdy z domowników może założyć własne konto użytkownika. Dzięki temu będzie miał dostęp do listy przeczytanych książek, czytanych obecnie książek wraz ze stroną na której skończył oraz listę preferowanych tagów. Kolejnym atutem jest możliwość dodania miejsca przechowywania danej książki, np: regał w sypialni czy półka w korytarzu.

### 3 Opis biznesowy

Naszym celem jest przyciągnięcie użytkowników, którzy chcieliby lepiej zarządzać domowym księgozbiorem, np: znać listę książek danego autora czy listę książek danego gatunku, tak by w przyszłości nie zakupić ponownie takiej samej książki. Dodatkowo często zdarza się, że wraz z upływem czasu zapominamy

tytułów przeczytanych książek. Dzięki stworzonej aplikacji będziemy mogli z łatwością uzupełnić nasze braki pamięciowe. Ponadto w czasie tworzenia aplikacji będziemy na bieżąco modyfikować dane funkcjonalności, tak aby jak najbardziej pomóc użytkownikowi w lepszym zarządzaniu zbiorem książek.

## 4 Założenia

Naszym założeniem jest, aby aplikacja była dostępna na komputerze. Podczas korzystania z aplikacji nie będzie potrzeby dostępu do połączenia internetowego, ponieważ informacje z aplikacji w założeniu są dostępne w obrębie tylko jednego komputera. Dzięki temu unikniemy problemu potrzeby synchronizacji bazy danych pomiędzy wieloma urządzeniami.

## 5 Funkcje

Nasza aplikacja na ten moment będzie posiadała cztery klasy:

- Biblioteka;
- Użytkownik;
- Książka;
- Pula tagów.

### 5.1 Biblioteka

Biblioteka jest klasą, którą możemy utworzyć tylko raz, wraz z rozpoczęciem korzystania z naszej aplikacji. Posiada dwa pola:

- `listaKsiążek`, czyli listę obiektów klasy `Książka`;
- `listaUżytkowników`, czyli listę obiektów klasy `Użytkownik`.

Posiada też metody, takie jak:

- `dodajKsiążkę`, przyjmującą nowo utworzony obiekt klasy `Książka`;
- `dodajUżytkownika` przyjmującą nowo utworzony obiekt klasy `Użytkownik`;
- `wyszukajKsiążkę`, przyjmującą atrybut, po którym chcemy szukać oraz jego wartość;
- `usuńKsiążkę`, która sprawdza czy podana książka istnieje w księgozbiornie, a następnie ją usuwa;
- `sprawdźListęPożyczonychKsiążek`, która wyświetla informacje o pożyczonych książkach, oraz listę osób, które je wypożyczyły;
- `sprawdźListęCzytanychKsiążek`, która wyświetla informacje o czytanych książkach, oraz listę użytkowników, którzy je czytają.

## 5.2 Użytkownik

Użytkownik jest klasą, której obiekty możemy tworzyć wielokrotnie. Celem jest, aby każdy z domowników miał własne konto użytkownika. Klasa ta posiada pola:

- `listaPrzeczytanychKsiążek`, czyli listę obiektów klasy `Książka`;
- `czytaneKsiążki`, czyli mapę, która jako klucz przyjmuje obiekt klasy `Książka`, a jako wartość numer strony, na której użytkownik skończył czytać;
- `preferowaneTagi`, czyli mapę, która jako klucz przyjmuje tag, czyli obiekt typu `String`, a jako wartość liczbę wystąpień danego tagu w dotychczas przeczytanych książkach.

Dodatkowo obiekt klasy `Użytkownik` ma metody takie jak:

- `dodajDoListyPrzeczytanychKsiążek`, która pozwala dodać przeczytaną książkę przez użytkownika;
- `dodajDoCzytanychKsiążek`, która pozwala dodać czytaną przez użytkownika książkę.

## 5.3 Książka

`Książka` jest klasą, której obiekty możemy tworzyć wielokrotnie. Obiekty tej klasy przechowują wszystkie istotne informacje na temat danej książki, umożliwiając tym samym późniejsze wyszukiwanie książek na podstawie którejś z wartości. `Książka` automatycznie po stworzeniu będzie dodawana do listy książek. Klasa posiada pola:

- `tytuł`, czyli pole typu `String` zawierające tytuł książki;
- `autor`, czyli pole typu `String` zawierające autora książki;
- `gatunek`, czyli pole typu `String` zawierające gatunek książki;
- `rokWydania`, czyli pole typu `LocalDate`, zawierające datę wydania książki;
- `wydawnictwo`, czyli pole typu `String`, zawierające wydawnictwo książki;
- `liczbaStron`, czyli pole typu `int`, zawierające liczbę stron książki;
- `tagi`, czyli pole typu `ArrayList`, zawierające listę z tagami książki;
- `czyCzytana`, czyli pole typu `boolean` zawierające informację czy książka jest obecnie czytana;
- `czyPożyczona`, czyli pole typu `boolean`, zawierające informację czy książka jest obecnie pożyczona;

- `komuPożyczona`, czyli pole typu `String`, zawierające imię i nazwisko osoby, która pożyczyła daną książkę;
- `gdzieJest`, czyli pole typu `String`, zawierające nazwę miejsca, w którym się znajduje.

Dodatkowo klasa zawiera cztery settery:

- `ustawCzyCzytana`, dzięki której możemy ustalić wartość pola `czyCzytana`;
- `ustawCzyPożyczona`, dzięki której możemy ustalić wartość pola `czyPożyczona`;
- `ustawKomuPożyczona`, dzięki której możemy ustalić wartość pola `komuPożyczona`;
- `ustawGdzieJest`, dzięki której możemy ustalić wartość pola `gdzieJest`.

## 5.4 Pula tagów

`PulaTagów` jest klasą przechowującą listę wszystkich dostępnych tagów. Dzięki temu z łatwością będziemy mogli dodawać nieistniejące do tej pory tagi lub je modyfikować. Klasa zawiera jedno pole:

- `listaTagów`, czyli listę przechowującą wszystkie tagi.

Dodatkowo posiada metodę:

- `dodajNowyTag`, dzięki której możemy zaktualizować listę obecnych tagów.

# 6 Wymagania

## 6.1 Funkcjonalne

Do prawidłowego funkcjonowania naszej aplikacji niezbędne będzie sprawne działanie klasy `Książka` oraz `Biblioteka`. Bez tego aplikacja nie będzie miała żadnej przyszłości. Perfekcyjne działanie klas `Użytkownik` oraz `PulaTagów` nie jest koniecznością, aczkolwiek będzie sporym plusem i uzupełnieniem podstawowych funkcji aplikacji. W razie potrzeby wyboru będziemy zwracać większą uwagę na klasę `Użytkownik`, ponieważ klasę `PulaTagów` możemy w ostateczności zamienić na `Enuma`, z konsekwencją utraty dodawania nowych tagów w interfejsie użytkownika.

## 6.2 Niefunkcjonalne

Naszym celem jest, aby powstała aplikacja była jak najbardziej „przyjazna” dla użytkownika, tak aby ułatwić korzystanie z niej. W tym celu użyjemy `JavaFX`, która naszym zdaniem najlepiej zrealizuje nasze założenia. Dzięki temu, że w naszym założeniu aplikacja będzie działać na jednym komputerze nie musimy uwzględniać obciążenia serwera związanego z wykorzystaniem aplikacji przez wielu użytkowników jednocześnie.