

Dokumentacja aplikacji "Planowanie Zadań Użytkownika"

- Projekt jest napisany w obiektowym języku programowania **Java**;
- Do napisania projektu używałem IDE środowiska do pisanie kodu **IntelliJ IDEA**;
- projekt jest wykomentowany;

Cele Aplikacji:

Aplikacja Planowanie zadań użytkownika ma na celu umożliwienie użytkownikom organizacji ich codzienne zadania, a także monitorowanie ich realizacji.

Poniżej przedstawione są kluczowe aspekty związane z funkcjonalnością i strukturą projektu:

- Dodawanie, Edycja, Odczyt i Usuwanie użytkowników oraz zadań:
- Użytkownicy mogą być dodawani do systemu z określonym identyfikatorem użytkownika (ID) i nazwą użytkownika.
- Istnieje możliwość edycji informacji o użytkowniku, takich jak nazwa użytkownika.
- Odczyt danych o użytkowniku po ID oraz usuwanie użytkownika z systemu.
- Użytkownicy mogą dodawać zadania do swojej listy z określonym identyfikatorem zadania (ID) i opisem zadania.
- Jest możliwość oznaczania zadań jako ukończone lub nieukończone, czyli True lub False w konsoli.
- Odczyt danych o zadaniu po ID oraz usuwanie zadania z listy.
- Przypisywanie zadań do użytkowników.
- Użytkownicy mogą dodawać zadania do swojej listy i przypisywać je do konkretnego użytkownika.

Struktura Projektu:

User (Użytkownik):

- Reprezentuje użytkownika w systemie.
- Operacje CRUD: Dodawanie, Odczyt, Edycja, Usuwanie Użytkowników.
- Dane przechowywane w pamięci na obiektach.

Task (Zadanie):

- Reprezentuje zadanie w systemie.
- Operacje CRUD: Dodawanie, Odczyt, Edycja, Usuwanie Zadań.
- Dane przechowywane w pamięci na obiektach.

TaskPlanner (Planner Zadań):

- Główna klasa zarządzająca użytkownikami i zadaniami.
- Implementuje interfejs TaskPlannerService.
- Przechowuje dane o użytkownikach w pamięci przy użyciu InMemoryUserRepository.
- Przechowuje dane o zadaniach w pamięci przy użyciu InMemoryTaskRepository.

InMemoryUserRepository:

- Implementuje interfejs UserRepository dla użytkowników.
- Przechowuje dane o użytkownikach w pamięci za pomocą mapy.

InMemoryTaskRepository:

- Implementuje interfejs TaskRepository dla zadań.
- Przechowuje dane o zadaniach w pamięci za pomocą mapy.

TaskPlannerServiceImpl

- Implementuje interfejs TaskPlannerService.
- Zarządza operacjami związanymi z użytkownikami i zadaniami.
- Wykorzystuje Dependency Injection do wstrzykiwania repozytoriów.

Klasa **Main** w aplikacji pełni rolę punktu wejścia i koordynatora, gdzie są inicjalizowane obiekty innych klas, a następnie wywoływane są na nich konkretne operacje.

Wykorzystanie Interfejsów na 4.0

- Interfejsy UserRepository i TaskRepository definiują operacje CRUD dla użytkowników i zadań.
- Interfejs TaskPlannerService definiuje operacje związane z zarządzaniem użytkownikami i zadaniami.
- Klasa TaskPlannerServiceImpl implementuje TaskPlannerService i wykorzystuje wstrzykiwanie zależności do repozytoriów.

Walidacja Danych i Obsługa Wyjątków na 4.0

- Walidacja danych odbywa się poprzez metody validateUserId i validateTaskId, które sprawdzają poprawność ID użytkownika i zadania.
- Obsługa wyjątków realizowana jest za pomocą dwóch niestandardowych wyjątków: UserNotFoundException i TaskNotFoundException.
- W metodzie **main** przykładowo użyłem konstrukcji try-catch do przechwytywania tych wyjątków podczas operacji na użytkownikach i zadaniami.