

Task 3 – programowanie zaawansowane

Imię i nazwisko: Filip Kocima

Email: filipkocima@gmail.com

Nr albumu: 120840

Data: 28.09.2023

Link: <https://github.com/f384/advanced-programming>

Aplikacja, jest przykładem programu do zarządzania listą zadań (To-Do List) w technologii WPF (Windows Presentation Foundation). Aplikacja ta korzysta z bazy danych w pamięci oraz różnych serwisów i widoków modelu widoku (ViewModels). Poniżej omówię kluczowe elementy tego kodu oraz sposób, w jaki wpływają one na działanie aplikacji.

1. Konfiguracja i Start Aplikacji:

W momencie uruchomienia aplikacji, metoda OnStartup klasy App jest wywoływana. W tej metodzie:

Tworzona jest kolekcja usług ServiceCollection.

Wywoływana jest metoda ConfigureServices, która dodaje do kolekcji usług różne elementy, takie jak kontekst bazy danych, serwisy i modele widoku.

Kolekcja usług jest budowana do dostawcy usług (ServiceProvider), który będzie dostarczał obiekty zarejestrowanych usług do różnych części aplikacji.

Tworzone jest główne okno MainWindow, które jest modelem widoku opartym na klasie MainWindowViewModel.

Wyświetlane jest główne okno aplikacji.

2. Konfiguracja Usług:

W metodzie ConfigureServices dodawane są różne usługi do kolekcji:

TodoContext - Dodawany jest kontekst bazy danych przy użyciu Entity Framework Core. Ustawienia takie jak korzystanie z bazy danych w pamięci (UseInMemoryDatabase) oraz leniwe ładowanie proxy (UseLazyLoadingProxies) są konfigurowane.

IDialogService - Dodawany jest serwis do obsługi dialogów (okien dialogowych), który będzie dostępny w różnych częściach aplikacji.

MainWindowViewModel - Dodawany jest model widoku dla głównego okna, który będzie odpowiedzialny za zarządzanie logiką i danymi związanymi z głównym oknem aplikacji.

MainWindow - Dodawane jest główne okno jako usługa, aby można było je później pobrać z dostawcy usług i wyświetlić.

3. Obsługa Wyjątków:

Metoda `ApplicationDispatcherUnhandledException` jest wywoływana w przypadku, gdy w trakcie działania aplikacji wystąpi nieobsłużony wyjątek. W tym przypadku, obsługiwany jest wyjątek związany z interfejsem użytkownika poprzez wyświetlenie okna dialogowego z informacją o błędzie.

4. Zapis Danych Przy Zamykaniu:

W momencie zamykania aplikacji (`OnExit`), dane z kontekstu bazy danych (`TodoContext`) są zapisywane do pliku. W tym przykładzie używana jest metoda `SaveDataToFile`, która nie jest dostępna w przedstawionym kodzie, ale zakładamy, że jest to odpowiednia metoda do zapisywania danych z bazy do pliku.

Podsumowanie:

Ten fragment kodu przedstawia strukturę aplikacji WPF opartej na przykładzie To-Do List. Poprzez użycie `Dependency Injection` i różnych serwisów, kod ten tworzy aplikację zgodną z zasadami SOLID i DRY, co ułatwia zarządzanie i rozwijanie aplikacji. Aplikacja ta wykorzystuje bazę danych w pamięci do przechowywania zadań, dostarcza funkcje dialogowe oraz zarządza logiką interfejsu użytkownika poprzez modele widoku. Obsługa wyjątków i zapis danych przy zamykaniu zapewniają odpowiednią jakość użytkowania i zachowanie aplikacji.