Wstrzykiwanie zależności i repozytorium

Czas: 2 × 45 min | praca indywidualna | Wykład 2.15

Cel: Zrefaktoryzujesz dziennik z zadania 14: dostęp do danych za interfejsem repozytorium, okna dialogowe za interfejsem usługi, obiekty tworzone przez kontener DI.

Po zajęciach potrafisz:

- wydziela interfejsy usług i ich implementacje
- konfiguruje kontener Microsoft.Extensions.DependencyInjection w aplikacji WPF
- dobiera czas życia usług i używa fabryki kontekstu EF
- zastępuje implementację atrapą bez zmian w ViewModelu

Sprzęt i materiały

- komputer z Windows 11 i dostępem do internetu
- pendrive lub konto w repozytorium do zapisania projektu

Oprogramowanie

- **Visual Studio Community 2026 (obciążenie „Programowanie aplikacji klasycznych platformy .NET”)** — darmowe dla uczniów i nauczycieli; w pracowni zainstalowane visualstudio.microsoft.com/pl/downloads
- **.NET 10 SDK (LTS)** — instalowany razem z Visual Studio; sprawdzenie: `dotnet --version` dotnet.microsoft.com/pl-pl/download/dotnet/10.0
- **Pakiet NuGet CommunityToolkit.Mvvm 8.4** — instalacja: Zarządzaj pakietami NuGet nuget.org/packages/CommunityToolkit.Mvvm
- **Pakiety Microsoft.EntityFrameworkCore.Sqlite i .Tools 10.x** — wersja zgodna z .NET 10 nuget.org/packages/Microsoft.EntityFrameworkCore.Sqlite
- **Pakiet Microsoft.Extensions.DependencyInjection 10.x** nuget.org/packages/Microsoft.Extensions.DependencyInjection

Przygotowanie przed zajęciami

- Projekt dziennika z zadania 14 (nowa gałąź Git `funkcja/di`).

Bezpieczeństwo

- Zapisuj projekt w swoim folderze (np. `C:\Users\<login>\source\repos`), a na koniec zajęć skopiuj go na pendrive lub wyślij do repozytorium.
- Nie uruchamiaj programów i pakietów z nieznanych źródeł; pakiety NuGet instaluj tylko te, które wskazuje zadanie.
- Rób przerwy dla oczu (zasada 20-20-20) — praca z projektantem okien wymaga długiego patrzenia w ekran.

Przebieg zadania

1 Interfejsy

Folder `Services`.

1. `IOcenyRepo`: PobierzUczniow, PobierzPrzedmioty, DodajOcene.
2. `IDialogService`: Info, Potwierdz.
3. `EfOcenyRepo` — przyjmuje `IDbContextFactory<DziennikContext>` i tworzy kontekst na operację.
4. `DialogService` (MessageBox) i `PamiecOcenyRepo` (dane w listach).

2 ViewModel bez new

`OcenyViewModel(IOcenyRepo repo, IDialogService dialog)`.

1. Przenieś logikę z DziennikViewModel; usuń wszystkie `new DziennikContext()` i `MessageBox` z ViewModelu.
2. Przy ocenie 1 pytaj o potwierdzenie przez `dialog.Potwierdz`.
3. W kontekście dodaj konstruktor z `DbContextOptions` i warunek `if (!o.IsConfigured)` w OnConfiguring.

3 Kontener

App.xaml.cs.

```
var sc = new ServiceCollection();
sc.AddDbContextFactory<DziennikContext>(o => o.UseSqlite(baza));
sc.AddSingleton<IOcenyRepo, EfOcenyRepo>();
sc.AddSingleton<IDialogService, DialogService>();
sc.AddTransient<OcenyViewModel>();
sc.AddTransient<MainWindow>();
Uslugi = sc.BuildServiceProvider();
Uslugi.GetRequiredService<MainWindow>().Show();
```

1. Usuń `StartupUri` z App.xaml.
2. Zarejestruj: `AddDbContextFactory`, repozytorium i dialog jako Singleton, ViewModel i okno jako Transient.
3. Okno przyjmuje ViewModel w konstruktorze i ustawia DataContext.
4. Zamień jedną linią repozytorium na `PamiecOcenyRepo` i uruchom aplikację bez bazy. Wróć do EF.

4 Podsumowanie

Oceń refaktoryzację.

1. Narysuj schemat zależności (kto zna kogo) przed i po zmianach.
2. Sprawdź wyszukiwaniem (Ctrl+Shift+F), że w ViewModelach nie ma `new DziennikContext` ani `MessageBox`.
3. Pull request z gałęzi do main z opisem zmian.

W sprawozdaniu umieść

- ☐ kod interfejsów i App.xaml.cs
- ☐ schematy zależności przed i po
- ☐ zrzut aplikacji z PamiecOcenyRepo
- ☐ opis pull requesta

Pytania kontrolne

1. Co zyskujemy, gdy ViewModel zależy od interfejsu, a nie od klasy EF?
2. Dlaczego DbContext nie powinien być Singletonem?
3. Czym różni się AddDbContext od AddDbContextFactory?
4. Co to jest composition root?

Kryteria oceny

Kryterium	Punkty
Interfejsy i implementacje	5
ViewModel bez new i MessageBox	5
Konfiguracja kontenera	5
Zamiana na atrapę i schematy	3
Pytania	2
Razem	**20**

Sprawozdanie przygotuj na wzorze dołączonym do zadania (plik .docx o tym samym numerze).