Nawigacja Shell — sklep

Czas: 2 × 45 min | praca indywidualna | Wykład 2.26

Cel: Zbudujesz aplikację sklepu z TabBar (Produkty, Koszyk, O sklepie), stroną szczegółów otwieraną trasą z parametrem i koszykiem współdzielonym przez usługę.

Po zajęciach potrafisz:

- buduje strukturę aplikacji w AppShell z TabBar
- rejestruje trasy i nawiguje GoToAsync z parametrami
- odbiera parametry QueryProperty w ViewModelu
- współdzieli stan między stronami przez usługę Singleton

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
- **Obciążenie VS „Programowanie aplikacji wieloplatformowych w .NET” (.NET MAUI) z emulatorem Androida** — emulator wymaga wirtualizacji sprzętowej; awaryjnie uruchamiaj jako Windows Machine learn.microsoft.com/dotnet/maui/get-started/installation
- **Pakiet NuGet CommunityToolkit.Mvvm 8.4** — instalacja: Zarządzaj pakietami NuGet nuget.org/packages/CommunityToolkit.Mvvm

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.

Przebieg zadania

1 Usługa

Projekt MAUI `Sklep`; klasa `SklepService` bez MAUI.

1. Rekordy `Produkt(Id, Nazwa, Kategoria, Cena, Opis)` i `PozycjaKoszyka(Produkt, Ilosc)` z właściwością Wartosc.
2. Lista 5 produktów; metody Znajdz(id), Dodaj(produkt, ilosc) — scala ilości tego samego produktu, Usun(id); zdarzenie KoszykZmieniony.
3. Sprawdź: 3 × słuchawki 149 zł + 2 × pendrive 34,50 zł = 516 zł.

2 Shell

AppShell.

1. TabBar z trzema `ShellContent` (Route: produkty, koszyk, info; ContentTemplate).
2. `Routing.RegisterRoute("szczegoly", typeof(SzczegolyPage))` w konstruktorze AppShell.
3. Rejestracja usług, ViewModeli i stron w MauiProgram (usługa i koszyk — Singleton).

3 Strony

Produkty, szczegóły, koszyk.

```
[QueryProperty(nameof(Id), "id")]
public partial class SzczegolyViewModel(SklepService sklep) : ObservableObject
{
    [ObservableProperty] private int _id;
    [ObservableProperty] private Produkt? _produkt;
    partial void OnIdChanged(int value) => Produkt = sklep.Znajdz(value);
}
```

1. Produkty: CollectionView; wybór → `GoToAsync($"szczegoly?id={produkt.Id}")`; po powrocie odznacz wybór.
2. Szczegóły: `[QueryProperty(nameof(Id), "id")]`, wczytanie produktu w `OnIdChanged`, Stepper ilości, Dodaj do koszyka → komunikat i `GoToAsync("..")`.
3. Koszyk: pozycje z przyciskiem Usuń, podsumowanie, EmptyView z przyciskiem „Przejdź do produktów” (`//produkty`).

4 Sprawdzenie

Na Androidzie.

1. Systemowy przycisk Wstecz ze strony szczegółów wraca do listy.
2. Koszyk aktualizuje się po dodaniu produktu bez ręcznego odświeżania.
3. Zatwierdź pracę w Git.

W sprawozdaniu umieść

- ☐ kod SklepService i AppShell
- ☐ kod SzczegolyViewModel
- ☐ schemat nawigacji (strony i trasy)
- ☐ zrzuty trzech kart i szczegółów

Pytania kontrolne

1. Czym różni się ".." od "//produkty"?
2. Dlaczego strony szczegółów nie umieszczamy w TabBar?
3. Jak koszyk dowiaduje się o zmianach?
4. Dlaczego po powrocie trzeba odznaczyć element listy?

Kryteria oceny

Kryterium	Punkty
SklepService	4
AppShell i trasy	4
Szczegóły z parametrem	5
Koszyk z odświeżaniem	4
Pytania	3
Razem	**20**

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