Style i zasoby

Czas: 2 × 45 min | praca indywidualna | Wykład 1.28

Cel: Ujednolicisz wygląd aplikacji z zadania 26 za pomocą zasobów i stylów w App.xaml oraz dodasz przełącznik motywu jasnego i ciemnego.

Po zajęciach potrafisz:

- definiuje zasoby kolorów i rozmiarów w słownikach zasobów
- tworzy style z kluczem, style niejawne i dziedziczenie BasedOn
- łączy słowniki w App.xaml (MergedDictionaries)
- przełącza motyw w trakcie działania dzięki DynamicResource

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

Przygotowanie przed zajęciami

- Przygotuj projekt Pizzeria z zadania 26 (lub inną swoją aplikację WPF z zadań 25–28).

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 Inwentaryzacja

Znajdź w XAML powtarzające się ustawienia wyglądu.

1. Wypisz w tabeli powtarzające się wartości (kolory, Margin, FontSize, Padding) i liczbę ich wystąpień.
2. Zaplanuj nazwy zasobów: `KolorAkcentu`, `KolorTla`, `KolorTekstu`, `RozmiarTekstu`...

2 Słowniki i style

Dodaj pliki: Projekt → Dodaj → Nowy element → Słownik zasobów (WPF).

```
<Application.Resources>
    <ResourceDictionary>
        <ResourceDictionary.MergedDictionaries>
            <ResourceDictionary Source="Themes/Jasny.xaml" />
            <ResourceDictionary Source="Styles.xaml" />
        </ResourceDictionary.MergedDictionaries>
    </ResourceDictionary>
</Application.Resources>
```

1. `Themes/Jasny.xaml` i `Themes/Ciemny.xaml` — te same klucze kolorów, różne wartości.
2. `Styles.xaml` — styl okna `StylOkna`, style niejawne dla TextBox, CheckBox, RadioButton, GroupBox, Label.
3. Style przycisków: `PrzyciskGlowny` i `PrzyciskUsun` z `BasedOn`.
4. W App.xaml połącz słowniki — motyw jako **pierwszy** (szablon).
5. Kolory zmieniające się z motywem wskazuj przez `DynamicResource`.

3 Zastosowanie

Przerób okno aplikacji.

1. Dodaj do okna `Style="{StaticResource StylOkna}"` i nagłówek ze stylem `Naglowek`.
2. Usuń z kontrolek wpisane na sztywno ustawienia, które przejęły style.
3. Dodaj przycisk Usuń zamówienie ze stylem `PrzyciskUsun`.
4. Policz, ile atrybutów wyglądu ubyło z MainWindow.xaml.

4 Motyw ciemny

Przełącznik CheckBox „Tryb ciemny”.

1. W obsłudze Checked/Unchecked podmień pierwszy słownik: `slowniki[0] = new ResourceDictionary { Source = new Uri(...) }`.
2. Sprawdź, które elementy zmieniły kolor, a które nie — popraw te, które używają `StaticResource` zamiast `DynamicResource`.
3. Zrób zrzuty w obu motywach. Zatwierdź pracę w Git.

W sprawozdaniu umieść

- ☐ tabela powtarzających się ustawień przed zmianą
- ☐ kod słowników i App.xaml
- ☐ liczba usuniętych atrybutów
- ☐ zrzuty w motywie jasnym i ciemnym

Pytania kontrolne

1. Czym różni się StaticResource od DynamicResource?
2. Kiedy styl bez klucza dotyczy kontrolki, a kiedy nie?
3. Dlaczego niejawny styl TextBlock w App.xaml może zmienić tekst w przyciskach?
4. Która wartość wygrywa: ustawiona w stylu czy bezpośrednio na kontrolce?

Kryteria oceny

Kryterium	Punkty
Inwentaryzacja	2
Słowniki motywów i Styles.xaml	6
Zastosowanie stylów w oknie	4
Przełącznik motywu	5
Pytania	3
Razem	**20**

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