Pierwsza aplikacja WPF

Czas: 2 × 45 min | praca indywidualna | Wykład 1.22–1.23

Cel: Utworzysz pierwszą aplikację okienkową WPF, poznasz pliki projektu i projektant XAML oraz obsłużysz kliknięcie przycisku.

Po zajęciach potrafisz:

- tworzy projekt WPF dla .NET 10 i uruchamia go
- rozróżnia pliki XAML i code-behind
- ustawia właściwości okna i kontrolek w XAML
- obsługuje zdarzenie Click i zmienia treść kontrolek w kodzie
- korzysta z funkcji Przeładowywanie na gorąco (Hot Reload)

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

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 Nowy projekt

Utwórz projekt z szablonu **Aplikacja WPF** (nie „WPF App (.NET Framework)”).

1. Nazwa: `Powitanie`, platforma: .NET 10.0 (LTS).
2. Obejrzyj w Eksploratorze rozwiązań pliki `App.xaml`, `MainWindow.xaml`, `MainWindow.xaml.cs` i `Powitanie.csproj`.
3. W pliku `.csproj` znajdź `UseWPF` i `TargetFramework` — zapisz ich wartości.
4. Uruchom pusty projekt (F5).

2 Okno i kontrolki

Zbuduj okno według wzoru — w XAML, nie przeciągając kontrolek.

```
<StackPanel Margin="16">
    <TextBlock Text="Jak masz na imię?" FontSize="16" />
    <TextBox x:Name="TxtImie" FontSize="16" Padding="4" />
    <Button x:Name="BtnPowitaj" Content="Powitaj" Width="120"
            HorizontalAlignment="Left" Margin="0,10,0,0"
            Click="BtnPowitaj_Click" />
    <TextBlock x:Name="TxtPowitanie" FontSize="20" FontWeight="Bold" />
</StackPanel>
```

1. Ustaw okno: `Title="Powitanie"`, wysokość 260, szerokość 420, `WindowStartupLocation="CenterScreen"`.
2. Zastąp `Grid` panelem `StackPanel` z marginesem 16.
3. Dodaj: TextBlock z pytaniem, TextBox `TxtImie`, Button `BtnPowitaj`, TextBlock `TxtPowitanie` (duża, pogrubiona czcionka w kolorze #512BD4).
4. Nadaj kontrolkom nazwy `x:Name` z przedrostkami (Txt, Btn).

3 Obsługa kliknięcia

Kod w `MainWindow.xaml.cs`.

1. W metodzie `BtnPowitaj_Click` wypisz „Witaj, [imię]!" albo „Wpisz imię.", gdy pole jest puste.
2. Dodaj pole `_kliknięcia` i wypisuj liczbę kliknięć w drugim TextBlocku.
3. Ustaw `IsDefault="True"` na przycisku i sprawdź działanie klawisza Enter.
4. Po uruchomieniu ustaw kursor w polu imienia: `TxtImie.Focus();` w konstruktorze.

4 Hot Reload

Zmiany bez ponownego uruchamiania.

1. Uruchom aplikację (F5) i nie zamykając jej, zmień w XAML kolor i rozmiar napisu. Zapisz plik i obserwuj okno.
2. Zmień tekst powitania w kodzie C# i użyj przycisku Hot Reload (ikona płomienia).
3. Zrób zrzut działającej aplikacji. Zatwierdź pracę w Git.

W sprawozdaniu umieść

- ☐ wartości `UseWPF` i `TargetFramework`
- ☐ kod XAML i code-behind
- ☐ zrzut działającej aplikacji
- ☐ opis działania Hot Reload

Pytania kontrolne

1. Do czego służy plik XAML, a do czego plik `.xaml.cs`?
2. Co oznacza `x:Name` i jak korzystamy z niego w kodzie?
3. Czym różni się TextBlock od TextBox?
4. Co robi metoda `InitializeComponent()`?

Kryteria oceny

Kryterium	Punkty
Projekt WPF .NET 10 i opis plików	3
Okno zbudowane w XAML zgodnie z opisem	6
Obsługa kliknięcia, licznik, Enter, fokus	6
Hot Reload i zrzuty	2
Pytania	3
Razem	**20**

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