Pierwsza aplikacja .NET MAUI

Czas: 2 × 45 min | praca indywidualna | Wykład 2.22

Cel: Utworzysz pierwszą aplikację .NET MAUI, uruchomisz ją na Windows i w emulatorze Androida oraz zmienisz stronę główną na licznik kliknięć.

Po zajęciach potrafisz:

- przygotowuje środowisko MAUI i emulator Androida
- rozpoznaje strukturę projektu MAUI
- buduje stronę z układem, etykietą i przyciskami
- obsługuje zdarzenie Clicked i właściwości wyglądu

Sprzęt i materiały

- komputer z Windows 11 i dostępem do internetu
- pendrive lub konto w repozytorium do zapisania projektu
- komputer z włączoną wirtualizacją sprzętową (dla emulatora) lub telefon z Androidem i kablem USB (opcjonalnie)

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

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.
- Telefon podłączaj tylko za zgodą nauczyciela; w trybie programisty nie zmieniaj innych ustawień telefonu.

Przebieg zadania

1 Środowisko

Przygotowanie (część może być już zrobiona w pracowni).

1. Instalator VS: obciążenie .NET MAUI — sprawdź, czy jest zainstalowane.
2. Narzędzia → Android → Menedżer urządzeń: utwórz emulator (np. Pixel z Androidem 15 lub 16, obraz x86_64).
3. Zapisz wersję Androida emulatora i API.

2 Projekt

Szablon **Aplikacja .NET MAUI**, nazwa `Licznik`, .NET 10.

1. Uruchom jako **Windows Machine**, potem w emulatorze — zmierz czas pierwszego i drugiego uruchomienia.
2. Opisz w sprawozdaniu foldery Platforms, Resources (AppIcon, Images, Fonts, Styles) i pliki MauiProgram, App, AppShell, MainPage.
3. Znajdź w App.xaml.cs metodę `CreateWindow` — co tworzy?

3 Licznik

Zmień MainPage.

```
private void OnKlik(object? sender, EventArgs e)
{
    _licznik++;
    LblLicznik.Text = _licznik == 1 ? "Kliknięto 1 raz" : $"Kliknięto {_licznik} razy";
    LblLicznik.TextColor = _licznik >= 10 ? Color.FromArgb("#0E7490") : Colors.Black;
}
```

1. Etykieta „Kliknięto N razy” (poprawna odmiana dla 1), przyciski Kliknij i Zeruj (czerwony).
2. Po 10 kliknięciach kolor tekstu zmienia się na #0E7490.
3. Etykieta z platformą i modelem urządzenia (`DeviceInfo.Platform`, `DeviceInfo.Model`).
4. Zmień tekst w XAML podczas działania aplikacji — sprawdź Hot Reload.

4 Ikona

Personalizacja.

1. Podmień obraz w Resources/AppIcon (SVG lub PNG) i kolor tła ikony w pliku projektu (`MauiIcon`, `Color`).
2. Sprawdź ikonę w emulatorze na liście aplikacji.
3. Zatwierdź pracę w Git.

W sprawozdaniu umieść

- ☐ wersja emulatora i czasy uruchomienia
- ☐ opis struktury projektu
- ☐ kod strony
- ☐ zrzuty: Windows i emulator, ikona aplikacji

Pytania kontrolne

1. Czym MAUI różni się od WPF?
2. Dlaczego pierwsze uruchomienie w emulatorze trwa długo?
3. Do czego służy folder Platforms?
4. Co zwraca DeviceInfo.Platform na Windows, a co w emulatorze?

Kryteria oceny

Kryterium	Punkty
Środowisko i emulator	4
Uruchomienie na dwóch platformach	4
Strona licznika	6
Ikona i opis struktury	4
Pytania	2
Razem	**20**

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