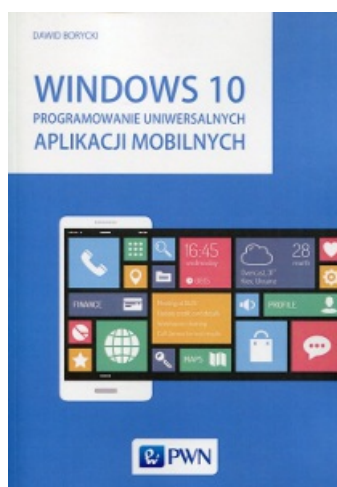


# Tytuł szkolenia: Programowanie w Windows 10 z XAML i C#

Kod szkolenia: MS-WIN10-XAML-C#-5

## Wprowadzenie



Microsoft wraz z Windows 10 wprowadził nową platformę **Universal Windows Platform**, która umożliwia tworzenie uniwersalnych aplikacji. Oznacza to, że programista pisze raz aplikację, a następnie może ją uruchomić na dowolnym urządzeniu, np. na desktopie, tablecie lub smartfonie, bez potrzeby tworzenia osobnych projektów tak jak to było wcześniej. Dzięki temu spadają koszty wytwarzania oprogramowania, a użytkownicy mogą wygodnie używać aplikacji na różnych urządzeniach.

Na szkoleniu, w ramach materiałów szkoleniowych przekazujemy uczestnikom książkę pt "**Windows 10 Programowanie uniwersalnych aplikacji mobilnych**"

Na szkoleniu dowiesz się w jaki sposób pisać uniwersalne aplikacje, które potrafią dostosować swój wygląd i zachowanie do urządzenia. Poznasz nowe kontrolki i możliwości języka XAML wprowadzone w UWP. Przejdziemy przez cały cykl tworzenia aplikacji z zachowaniem dobrych praktyk.

## Adresaci szkolenia

Grupa docelowa szkolenia to głównie osoby pełniące **kluczowe aktywności** przy **realizacji projektu** już w samym **zespole projektowym**. Adresatami szkolenia są **techniczni liderzy zespołów**, jak również **programiści**, czy też **programujący testerzy**. Znajomość podstawowych zwrotów z zakresu rozwoju oprogramowania.

Bardzo zalecane uczestnictwo w szkoleniu [\[MS-C#-5-VS\] Programowanie w C# 5.0 z Visual Studio 2013](#) (ang. Programming C# 5.0 with Visual Studio 2013).

Na szkoleniu wykorzystywane jest następujące środowisko i narzędzia:

- Windows 10
- Visual Studio 2015 Community
- Windows 10 SDK
- Microsoft Emulator for Windows 10 Mobile

## Cel szkolenia

Celem szkolenia jest nabycie umiejętności programowania **biznesowych aplikacji desktopowych w Windows 10**, korzystając z **XAML i C#**. W trakcie kursu uczestnicy zapoznają się również z zasobami **MSDN**, a także narzędziami wspierającymi rozwój oprogramowania, takimi jak **Microsoft Visual Studio**.

## Czas i forma szkolenia

- 35 godzin (5 dni x 7 godzin), w tym wykłady i warsztaty praktyczne.

## Plan szkolenia

### Wprowadzenie

- Omówienie Universal Windows Platform
- Rodziny urządzeń (Device Family)
- Porównanie UWP z Windows Mobile i Windows Phone

### Windows 10

- Urządzenia obsługiwane przez Windows 10
- Przygotowanie środowiska

### Pierwsza aplikacja

- Utworzenie pierwszej aplikacji UWP
- Struktura projektu
- Określanie obsługiwanej rodziny urządzeń (Device Family)
- Uruchamianie aplikacji na emulatorze
- Uruchamianie aplikacji na urządzeniu

### Visual Studio 2015

- Zapoznanie z narzędziem Visual Studio 2015
- Kompilacja, Debugowanie

### XAML

- Co to jest XAML
- Elementy, atrybuty, MarkupExtensions
- Konwertery

### Projektowanie interfejsu użytkownika

- Kontrolki LayoutControl (Grid, StackPanel)
- Kontrolki ContentControl (Button, TextBlock, TextBox, Image)
- Kontrolki ItemsControl (ListBox, ListView, GridView)

### Stylizacja aplikacji

- Style (Styles)
- Szablony kontrolki (Templates)
- Zasoby (Resources)
- Wyzwalacze (Triggers)

### Tworzenie uniwersalnych widoków

- VisualStateManager
- Adaptive Trigger
- Tworzenie własnego Adaptive Trigger
- RelativePanel
- Dostosowanie widoku do urządzenia (Device Specific View)
- Tworzenie aplikacji typu HamburgerMenu (SplitView)

### Tworzenie uniwersalnego kodu

- Rozszerzenia platformy (Platform Extensions)
- Sprawdzanie dostępności funkcji
- Użycie Platform Specific Analyzer

### Wiązanie danych (Binding)

- Wiązanie kontrolki z danymi

- b. Binding standardowy
- c. Binding skompilowany
- d. Bindowanie elementów
- e. Tworzenie własnego konwertera

#### **Optymalizacja XAML**

- a. Progressive Rendering
- b. Deferred Loading

#### **Nawigacja**

- a. Page
- b. NavigationService
- c. Przejścia pomiędzy widokami (Transitions)

#### **Usługi sieciowe**

- a. Dostęp do usług sieciowych REST API
- b. HttpClient

#### **Zdarzenia (Events)**

- a. Omówienie RoutedEvents
- b. Definiowanie obsługi zdarzeń
- c. Przechwytywanie zdarzeń

#### **Zastosowanie wzorca MVVM**

- a. Implementacja IPropertyChanged
- b. Implementacja ICommand

#### **Zarządzanie plikami**

- a. Tworzenie i odczytywanie plików (StorageFile)
- b. Wybór folderu

#### **Grafika i animacja**

- a. Rysowanie kształtów
- b. Używanie pędzli
- c. Animacje
- d. Transformacje
- e. Visual Layer

#### **Urządzenia, czujniki i bateria**

- a. Czujniki kierunku, orientacji, przesunięcia
- b. Obsługa Bluetooth

#### **Audio wideo**

- a. Przechwytywanie zdjęć i filmów z aparatu
- b. Wykrywanie twarzy na zdjęciach i filmach

#### **Mapy i lokalizacja**

- a. Wyświetlanie mapy
- b. Pobieranie i wyświetlanie lokalizacji użytkownika

#### **Interakcja z użytkownikiem**

- a. Klawiatura
- b. Mysz
- c. Rysik
- d. Dotyk
- e. Rozpoznawanie mowy

#### **Dynamiczne kafelki i powiadomienia**

- a. Dynamiczne kafelki
- b. Powiadomienia

#### **Lokalizacja**

- a. Tłumaczenie aplikacji z użyciem Multilingual App Toolkit

#### **Baza danych**

- a. Operacje CRUD za pomocą Entity Framework Core i SQLite

#### **Komunikacja pomiędzy aplikacjami**

- a. Współdzielenie danych takich jak tekst, linki, zdjęcia (Sharing Contract)
- b. Pobieranie współdzielonej zawartości
- c. Obsługa schowka

#### **Cykl życia aplikacji**

- a. Omówienie stanów aplikacji
- b. Przechwytywanie zdarzeń zmiany stanów aplikacji

#### **Wywoływanie aplikacji w tle**

- a. Tworzenie i rejestrowanie zadania w tle

#### **Aplikacja z usługą (App Service)**

- a. Tworzenie usługi w aplikacji
- b. Wywołanie usługi z innej aplikacji

#### **Tworzenie pakietu**

- a. Konfigurowanie pakietu
- b. Tworzenie pakietu
- c. Publikowanie pakietu poza sklepem

#### **Sklep**

- a. Monetyzacja
- b. Wyświetlanie reklam