

Programowanie Aplikacji Desktopowych

środowiska programistyczne IDE (Integrated Development Environment)



zintegrowane środowisko programistyczne (posiada narzędzia potrzebne przy pisaniu programu: edytor kodu z kolorowaną składnią, kompilator, debugger, narzędzia do refaktoryzacji kodu itp.)



[Visual Studio](#) (Windows) – IDE (Integrated Development Environment), Community 2019 – free (C#, F#, Visual Basic, C++, Python, JavaScript/TypeScript)



[Visual Studio](#) dla komputerów Mac (macOS) - IDE



[Visual Studio Code](#) (Windows, Linux, macOS) – edytor open source (free) z wieloma rozszerzeniami (m.in. JavaScript, Python, Java, C/C++, C#, Powershell, HTML/CSS. PHP) //trzeba samodzielnie zainstalować kompilator i debugger



[Code::Blocks](#) - The free C/C++ and Fortran IDE

środowiska programistyczne IDE (Integrated Development Environment)



[Apache NetBeans](#) – IDE (Java, php, C++ ..., free)



[Eclipse](#) – IDE (C++, Java ..., free)



[CodeLite](#) - cross platform IDE specialized in C, C++, PHP and JavaScript (free)
//trzeba samodzielnie zainstalować kompilator i debugger



[CLion](#) - A cross-platform IDE for C and C++ (komercyjne)


[C++ compiler online](#) (to nie jest IDE, free)

środowiska programistyczne RAD (Rapid Application Development)

posiada dodatkowo narzędzia do tworzenia interfejsów graficznych



[Visual Studio](#) (Windows) – IDE (Integrated Development Environment),
Community 2019 – free (C#, F#, Visual Basic, C++, Python, JavaScript/TypeScript)

 Aplikacja klasyczna systemu Windows C++

 Aplikacja Windows Forms C#

 Aplikacja WPF C#

Instalacja Visual Studio Community

<https://visualstudio.microsoft.com/pl/>

https://visualstudio.microsoft.com/pl/

60% Szukaj



Visual Studio

Produkty

Materiały do pobrania

Kup

Pomoc techniczna

Dostęp dla subskrybentów

Bezpłatne Visual Studio

Wszystkie produkty Microsoft

Wyszukaj

Zaloguj się

To jest sposób tworzenia oprogramowania

Nad czym chcesz dzisiaj [code, build, debug, deploy, współpracować, i co przeanalizować, learn]?

Visual Studio może to zrobić.

Poznaj rodzinę programu Visual Studio



Visual Studio

Wersja 17.

Najlepsze kompleksowe środowisko IDE dla deweloperów platformy .NET i języka C++ w systemie Windows. Pełna gama narzędzi i funkcji pozwalających podnieść poziom i ulepszyć każdy etap opracowywania oprogramowania.

Pobierz Visual

Community 2022

Professional 2022



Visual Studio for Mac

Wersja 8.10

Kompleksowe środowisko IDE dla deweloperów platformy .NET, które jest natywne dla systemu macOS. Obejmuje najwyższej klasy wsparcie dla sieci, chmury i tworzenia gier oraz bardzo dobre narzędzia do tworzenia międzyplatformowych aplikacji mobilnych.

komputerów Mac



Visual Studio Code

Wersja 1.62

Autonomiczny edytor kodu źródłowego działający w systemach Windows, macOS i Linux. Najlepszy wybór dla deweloperów języka Java i internetu z mnóstwem rozszerzeń do obsługi niemal dowolnego języka programowania.

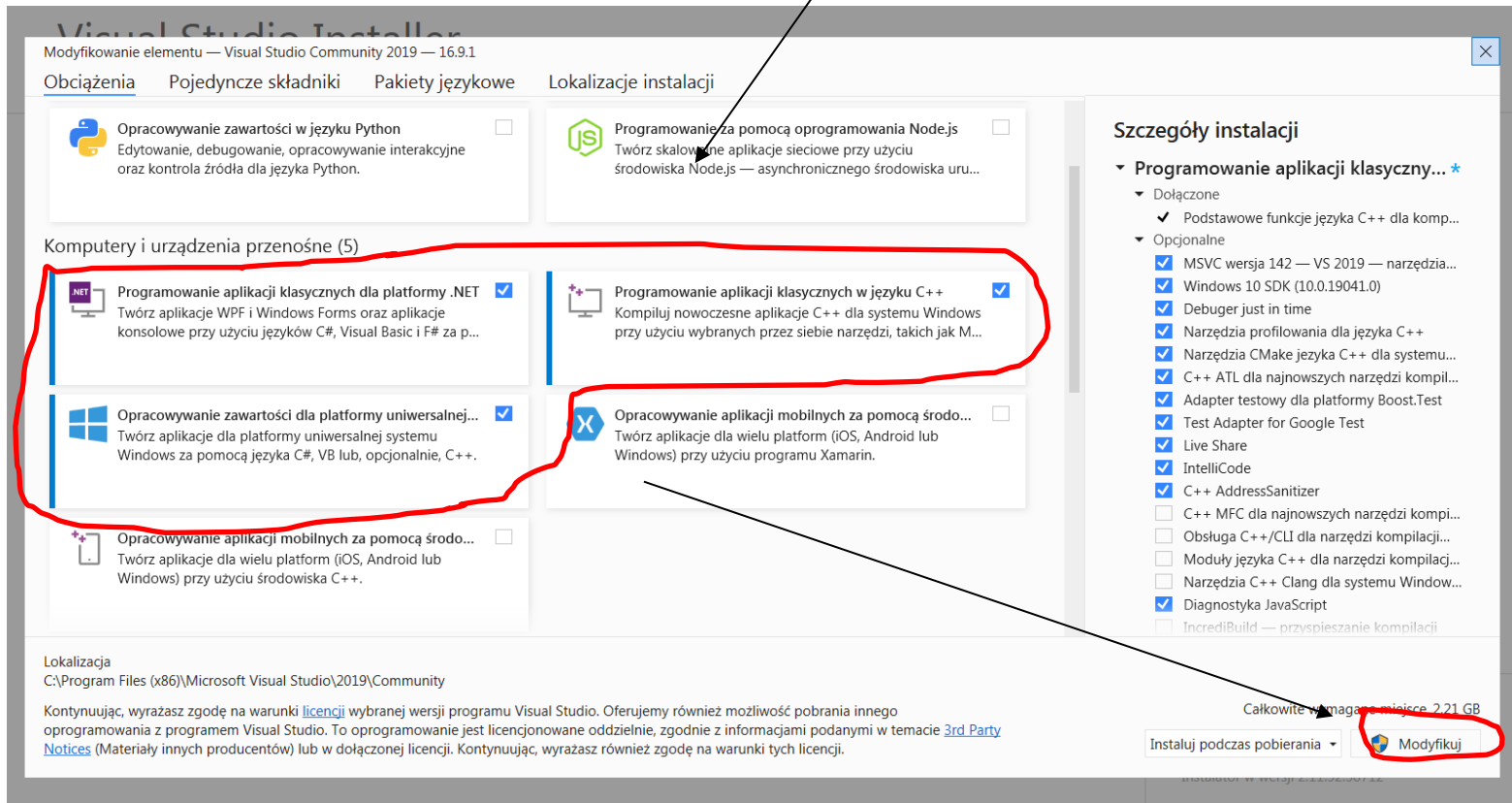
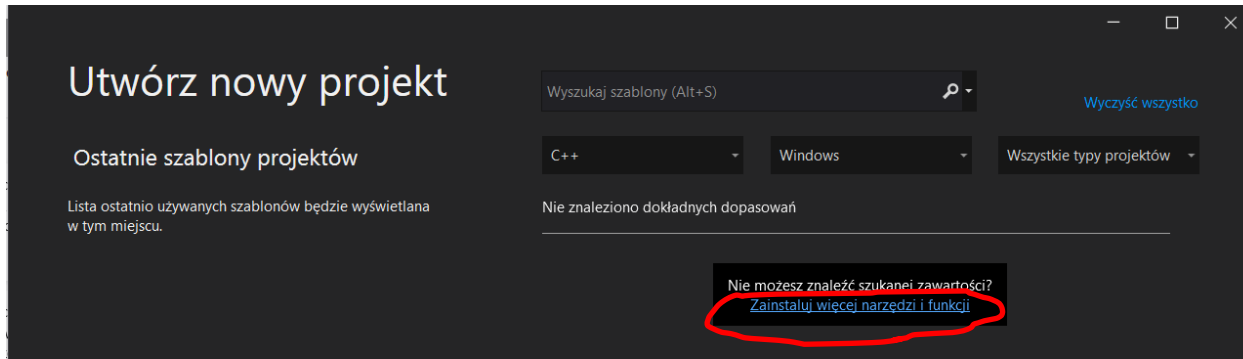
Korzystając z programu Visual Studio Code, akceptujesz jego licencję & oświadczenie o ochronie prywatności

Pobierz program Visual Studio Code

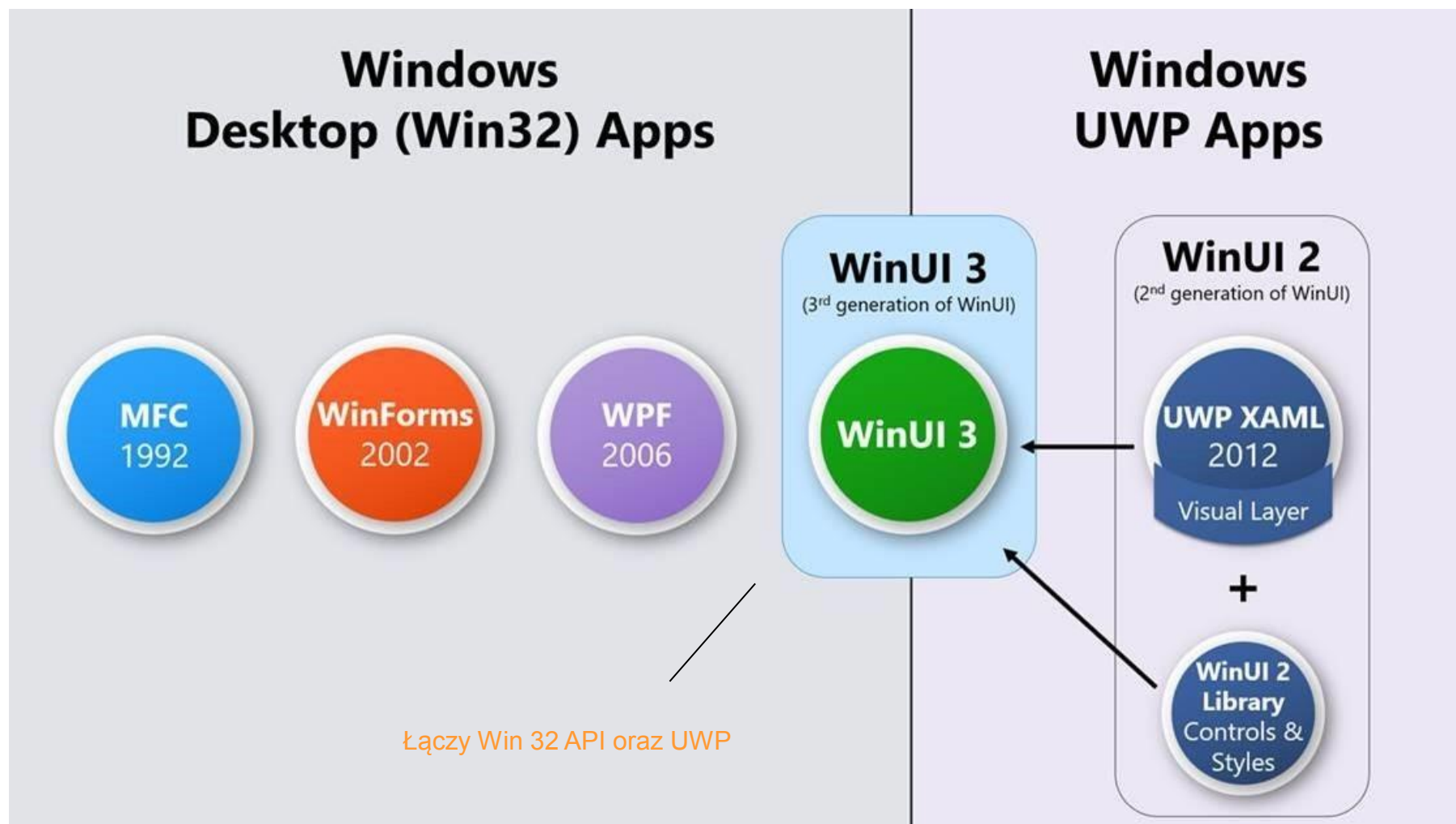
instalator

Bezpłatnie dla indywidualnych deweloperów, zastosowań akademickich i rozwiązań typu open source

Visual Studio Community 2019 instalacja środowisk do tworzenia interfejsów graficznych



Frameworki do tworzenia aplikacji okienkowej Windows



Frameworki do tworzenia aplikacji okienkowej Windows

Aplikacja klasyczna systemu Windows - Windows API, istnieje od lat 80tych.

Microsoft Foundation Class Library (MFC) – biblioteka, która opakowuje fragmenty Windows API w klasy C++, wprowadzona w 1992r.

Windows Forms (WinForms) - to bezpłatna biblioteka klas graficznych (GUI) typu open source, wprowadzona w 2002r. Uważa się ją jako następcę MFC.

Windows Presentation Foundation (WPF) - darmowy podsystem graficzny o otwartym kodzie źródłowym, został pierwotnie wydany jako część .NET Framework 3.0 w 2006 roku.

Aplikacja platformy uniwersalnej systemu Windows – ten sam kod aplikacji UWP działa na dowolnym urządzeniu z Windows 10 – telefonie, tablecie, desktopie. Wprowadzona w Win 10.

WinUI 3 - zapewnia spójne, intuicyjne i przystępne środowisko przy użyciu najnowszych wzorców interfejsu użytkownika (UI).

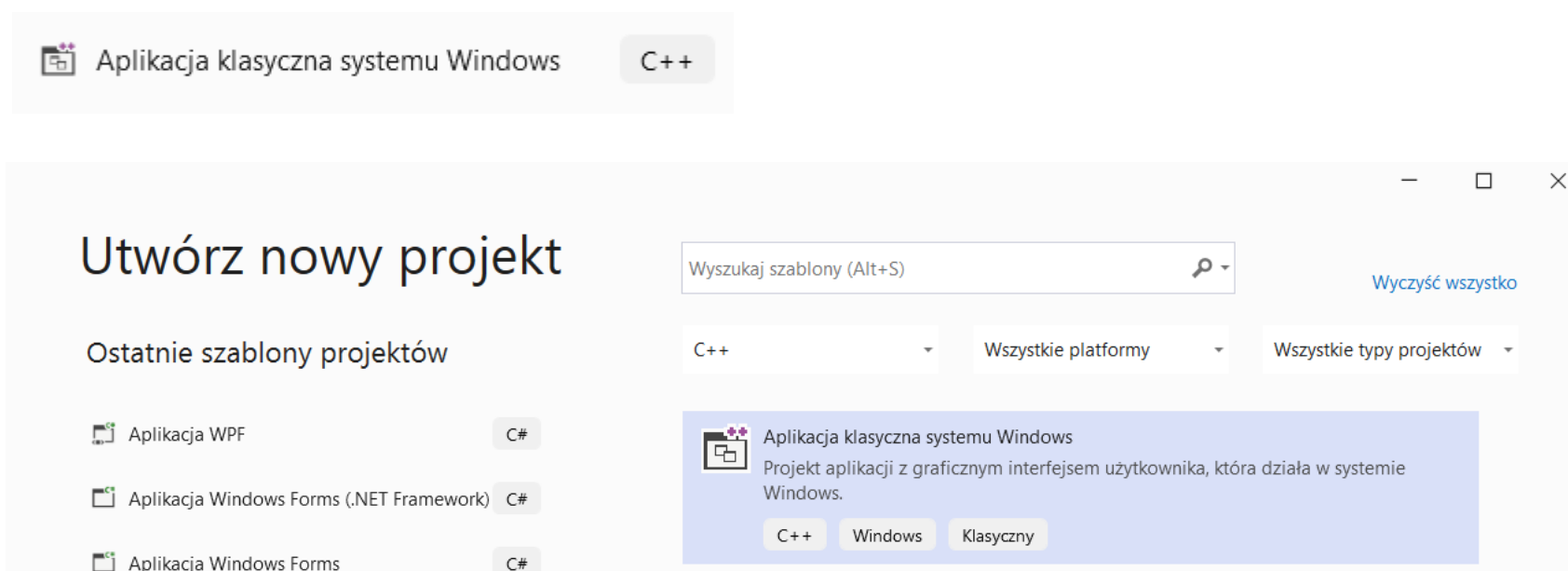
Aplikacja klasyczna systemu Windows

Application Programming Interface

Istnieje wiele sposobów tworzenia aplikacji Windows, ale ten sposób był pierwszy. Aplikacja klasyczna systemu Windows używa interfejsu API Windows (nazywanego również Win32 API, Windows Desktop API oraz Windows Classic API). Jest to oparta na języku C platforma do tworzenia aplikacji w systemie Windows. Istnieje ona od lat 80 i jest używana do tworzenia aplikacji Windows od dziesięcioleci. Na jej podstawie zostały stworzone bardziej zaawansowane i łatwiejsze w programowaniu frameworki np.: MFC, ATL, frameworki .NET. Nawet najbardziej nowoczesny kod środowiska Windows runtime dla aplikacji Uniwersalnej Platformy Windows (UWP) czy aplikacji ze sklepu napisanych w języku C++/WinRT używa interfejsu API Windows pod spodem.

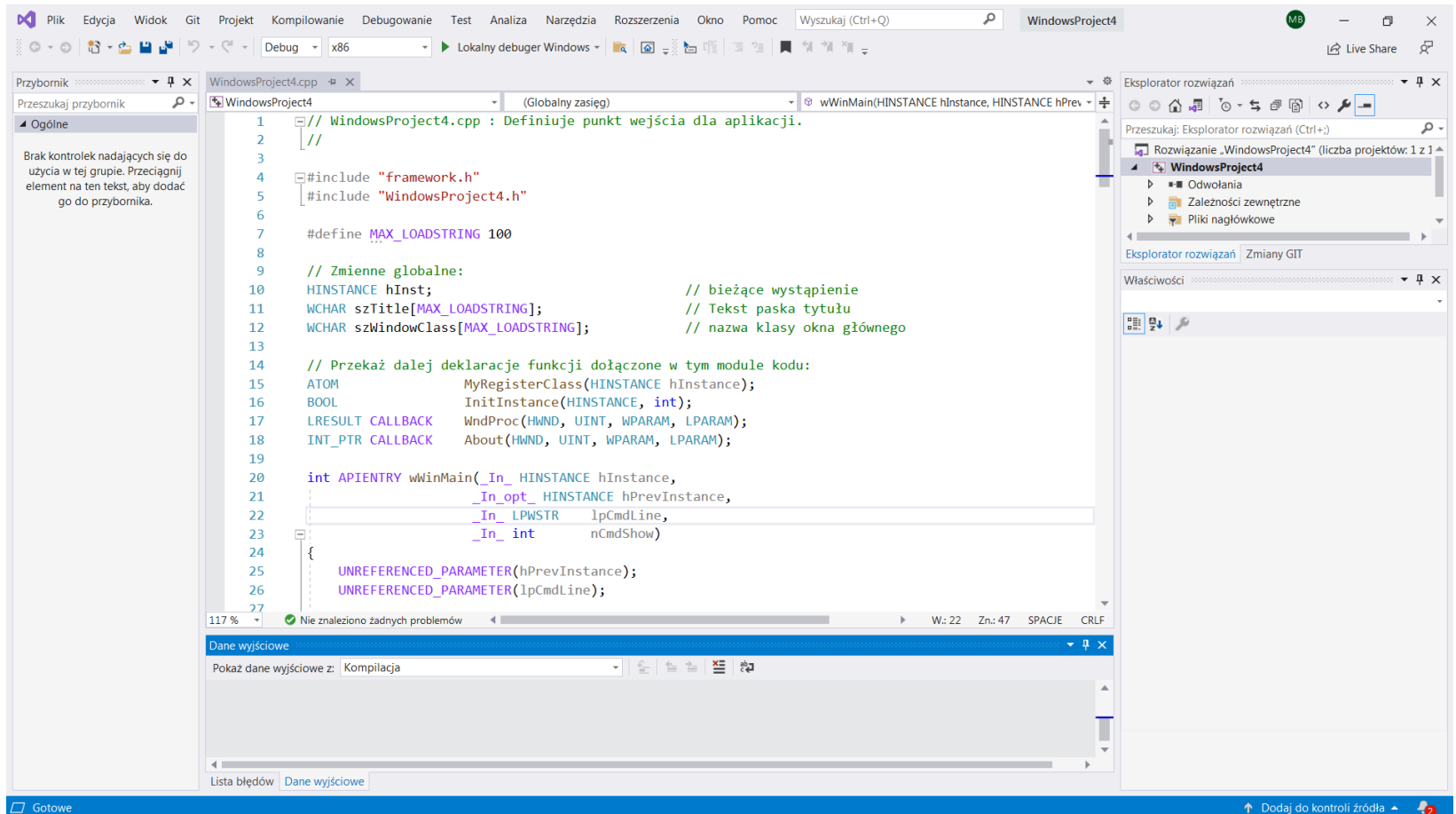
[Windows API Index](#)

Aplikacja klasyczna systemu Windows



[Przewodnik: tworzenie tradycyjnej aplikacji klasycznej Windows klasycznej \(C++\)](#)

Aplikacja klasyczna systemu Windows



Visual Studio Community 2019

programowanie bezpośrednio przy użyciu interfejsów API Win32, przy użyciu pętli komunikatów w stylu języka C, która przetwarza zdarzenia systemu operacyjnego.

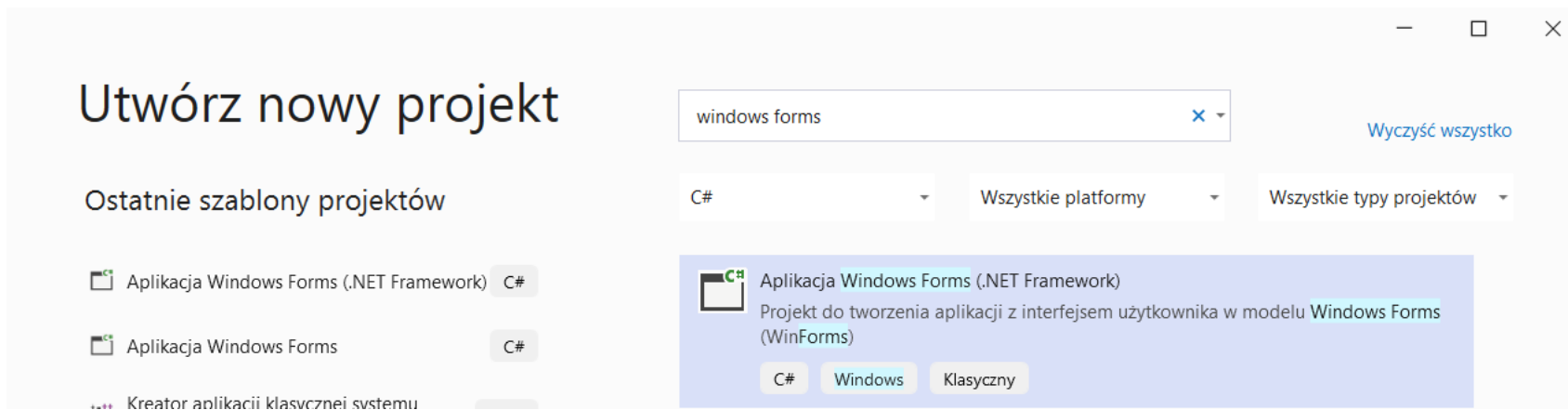
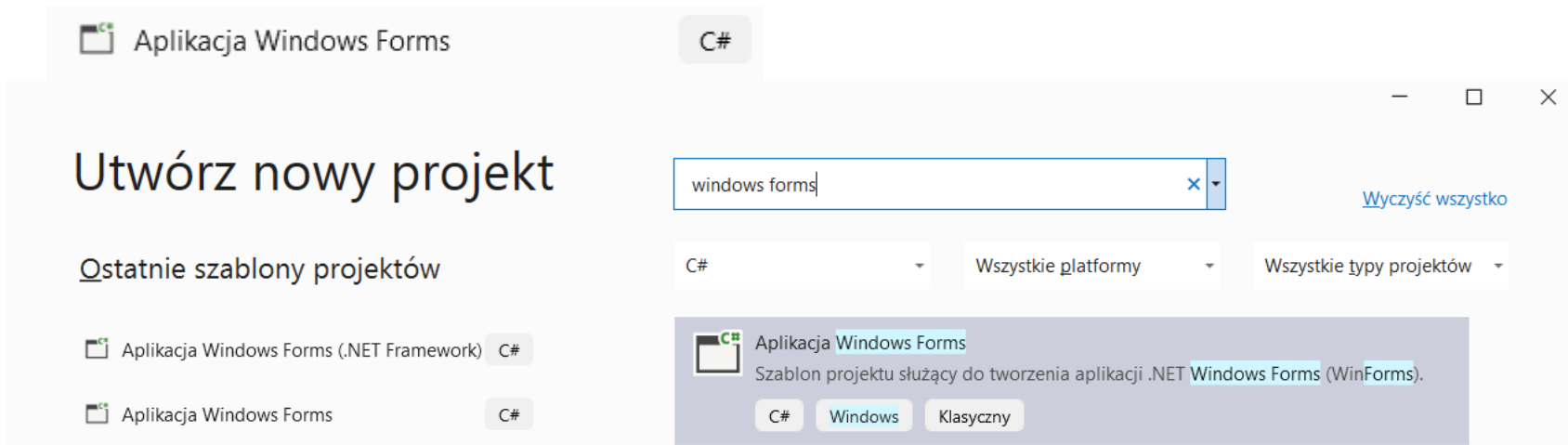
Aplikacja Windows Forms

Windows Forms (WinForms) - to bezpłatna biblioteka klas graficznych (GUI) typu open source zawarta jako część Microsoft .NET, .NET Framework lub Mono Framework[1], zapewniająca platformę do pisania aplikacji klienckich na komputery stacjonarne, laptopy i komputery typu tablet.

Chociaż jest postrzegana jako zamiennik wcześniejszej i bardziej złożonej biblioteki klas Microsoft Foundation Class Library opartej na C++, nie oferuje porównywalnego paradygmatu i działa jedynie jako platforma dla warstwy interfejsu użytkownika w rozwiązaniu wielowarstwowym.

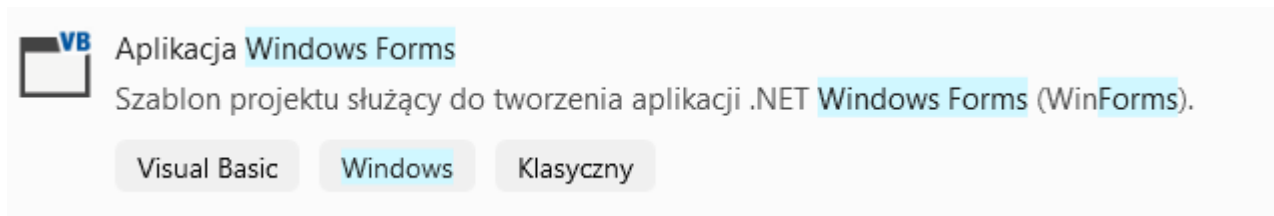
Aplikacja Windows Forms to aplikacja sterowana zdarzeniami obsługiwana przez platformę .NET Framework. W przeciwieństwie do programu konsolowego spędza większość czasu po prostu czekając, aż użytkownik coś zrobi, na przykład wypełni pole tekstowe lub kliknie przycisk. Kod aplikacji można napisać w języku programowania .NET, takim jak C# lub Visual Basic.

Aplikacja Windows Forms



[Tworzenie aplikacji Windows Forms w języku Visual Studio c#](#)

Aplikacja Windows Forms



[Tworzenie aplikacji Windows Forms w Visual Studio użyciu Visual Basic](#)

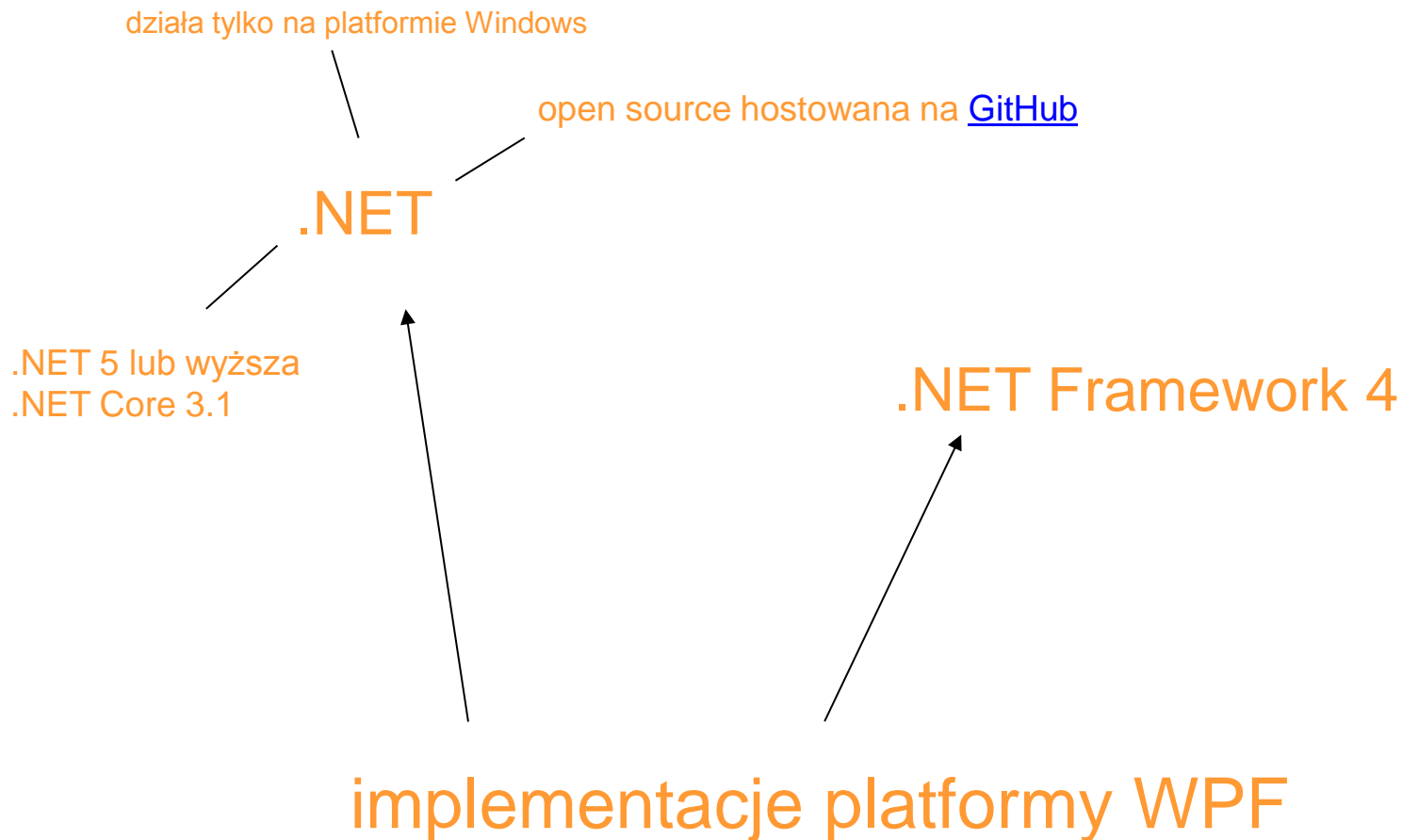
Windows Presentation Foundation (WPF)

Windows Presentation Foundation (WPF) to darmowy podsystem graficzny o otwartym kodzie źródłowym (podobny do WinForms) pierwotnie opracowany przez firmę Microsoft do renderowania interfejsów użytkownika w aplikacjach opartych na systemie Windows.

WPF, wcześniej znany jako „Avalon”, został pierwotnie wydany jako część .NET Framework 3.0 w 2006 roku.

WPF używa DirectX oraz wykorzystuje XAML, język oparty na XML, do definiowania i łączenia różnych elementów interfejsu. Aplikacje WPF można wdrażać jako samodzielne programy komputerowe lub hostować jako obiekt osadzony w witrynie internetowej. WPF ma na celu ujednoczenie wielu typowych elementów interfejsu użytkownika, takich jak renderowanie 2D/3D, dokumenty stałe i adaptacyjne, typografia, grafika wektorowa, animacja w czasie wykonywania i wstępnie renderowane multimedia. Elementy te można następnie łączyć i manipulować w oparciu o różne zdarzenia, interakcje użytkownika i powiązania danych.

Windows Presentation Foundation (WPF)

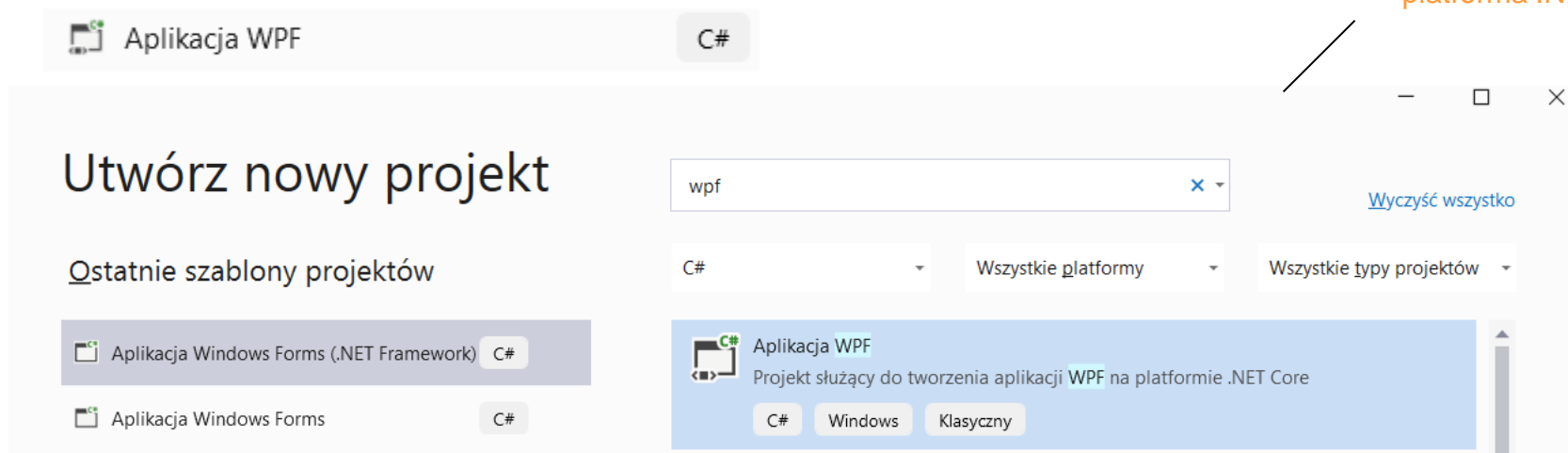


[Przewodnik klasyczny \(WPF .NET\)](#)

Windows Presentation Foundation (WPF)

Uses Extensible Application Markup Language (XAML)

platforma .NET Core



przykłady kodu WPF udostępnione przez Microsoft

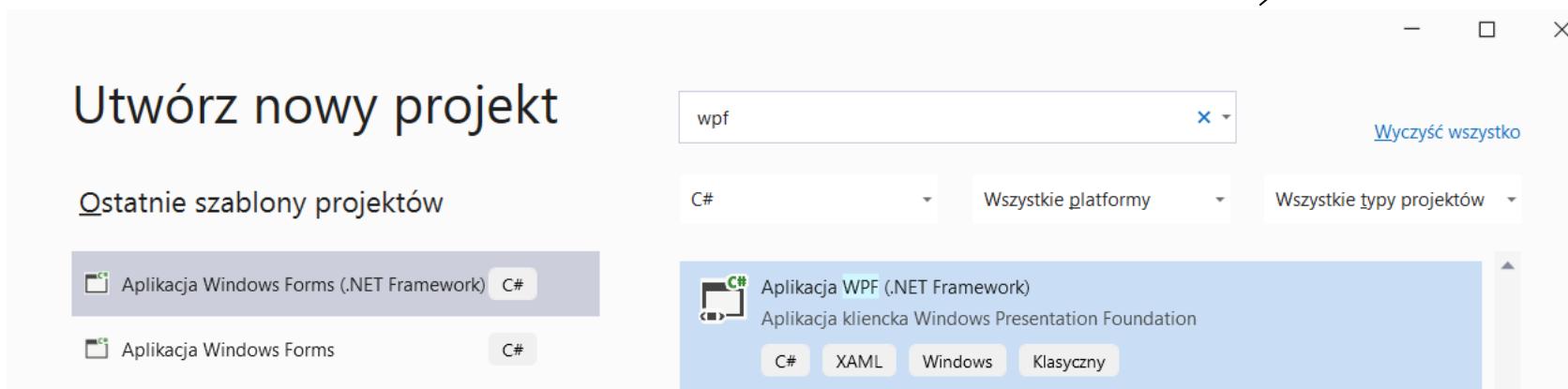
<https://github.com/microsoft/WPF-Samples>

[Samouczek: tworzenie prostej aplikacji w języku C# \(WPF\) \(Pizzeria\)](#)

[Samouczek: tworzenie nowej aplikacji WPF za pomocą platformy .NET \(Zamawiarka Prezentow\)](#)

Windows Presentation Foundation (WPF)

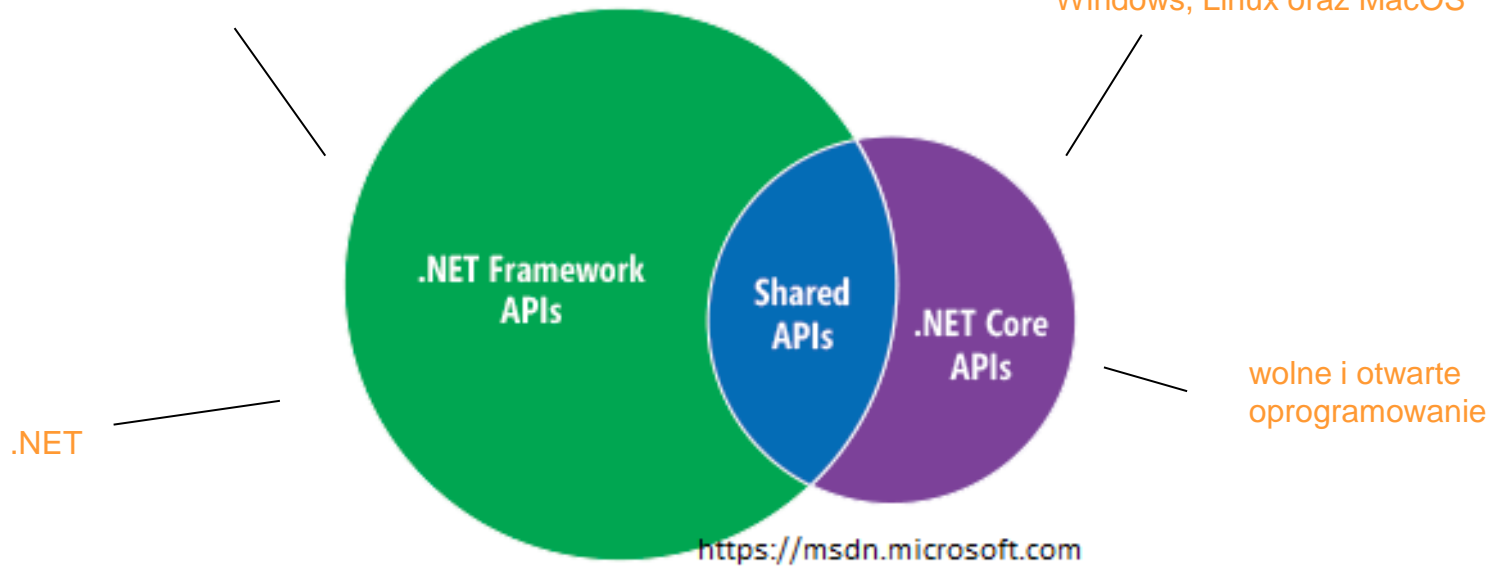
platforma .NET Framework



.NET Core vs .NET Framework

Umożliwia tworzenie aplikacji działającej tylko na platformie Windows

Umożliwia tworzenie aplikacji działających platformach: Windows, Linux oraz MacOS



[różnice między .NET Core oraz .NET Framework](#)

[.NET Core vs .NET Framework](#)

Windows Presentation Foundation (WPF)

The screenshot displays the Visual Studio 2019 interface for a WPF application named 'WpfApp2'. The main window, 'MainWindow.xaml', is shown in design view, featuring a single 'Button' control. The design view includes a visual tree on the left and a properties window on the right. The code view at the bottom shows the XAML markup for the window and button.

Visual Tree (Left):

- Wskaźnik
- Border
- Button

Properties Window (Right):

- Nazwa: <Brak nazwy>
- Typ: Button
- Rozmieść według: Kategoria
- Wygląd
- Wspólne
 - Command
 - CommandParame...
 - Content: Button
 - IsCancel
 - IsDefault
 - Cursor
 - DataContext
 - IsEnabled:

XAML Code (Code View):

```
1  xmlns:local="clr-namespace:WpfApp2"
2  mc:Ignorable="d"
3  Title="MainWindow" Height="450" Width="800"
4
5  <Grid>
6    <Button Content="Button" HorizontalAlignment="Left" Margin="56,58,0,0" VerticalAlignment="Top"
7
8
9
10
11
12  </Grid>
13 </Window>
```

Extensible Application Markup Language (XAML)

```
<Window x:Class="WpfApp.MainWindow"
        xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
        xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
        xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
        xmlns:local="clr-namespace:WpfApp"
        mc:Ignorable="d"
        Title="MainWindow" Height="450" Width="800">

    <!-- przycisk -->
    <Button Content="Button" HorizontalAlignment="Left" Margin="150,159,0,0" VerticalAlignment="Top" Height="66" Width="118"/>

</Window>
```

okno z przyciskiem

XAML to język znaczników oparty na języku XML

XAML

służy do definiowania okien, okien dialogowych, kontrolki użytkownika oraz wypełniania ich kontrolkami, kształtami i grafiką.

Aplikacja platformy uniwersalnej systemu Windows (UWP)

Uniwersalna Platforma Windows (UWP) pozwala na stworzenie uniwersalnej aplikacji, która działa na dowolnym urządzeniu z Windows 10 lub nowszym (tablecie, komputerze desktopowym, urządzeniu mobilnym (wsparcie dotyczące systemu Windows Mobile 10 zakończyło się 10 grudnia 2019 r.), holograficznym, Xbox oraz wbudowanym).

Aplikacje platformy UWP nie działają we wcześniejszych wersjach systemu Windows.

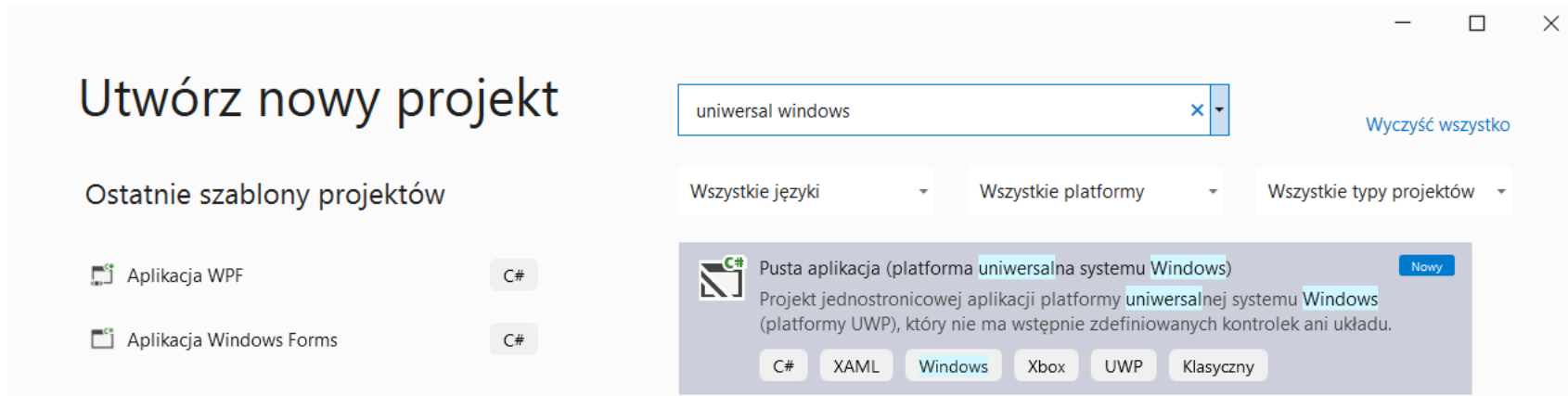
Aplikacje platformy UWP używają interfejsów API WinRT, aby zapewnić interfejs użytkownika i zaawansowane funkcje asynchroniczne, które są idealne dla urządzeń połączonych z Internetem.

Aby stworzyć program z zastosowaniem tej platformy używa się szablonu projektu platformy uniwersalnej Windows Platform (UWP), języka Extensible Application Markup Language (XAML) i języka programowania C#.

[What's a Universal Windows Platform \(UWP\) app?](#)

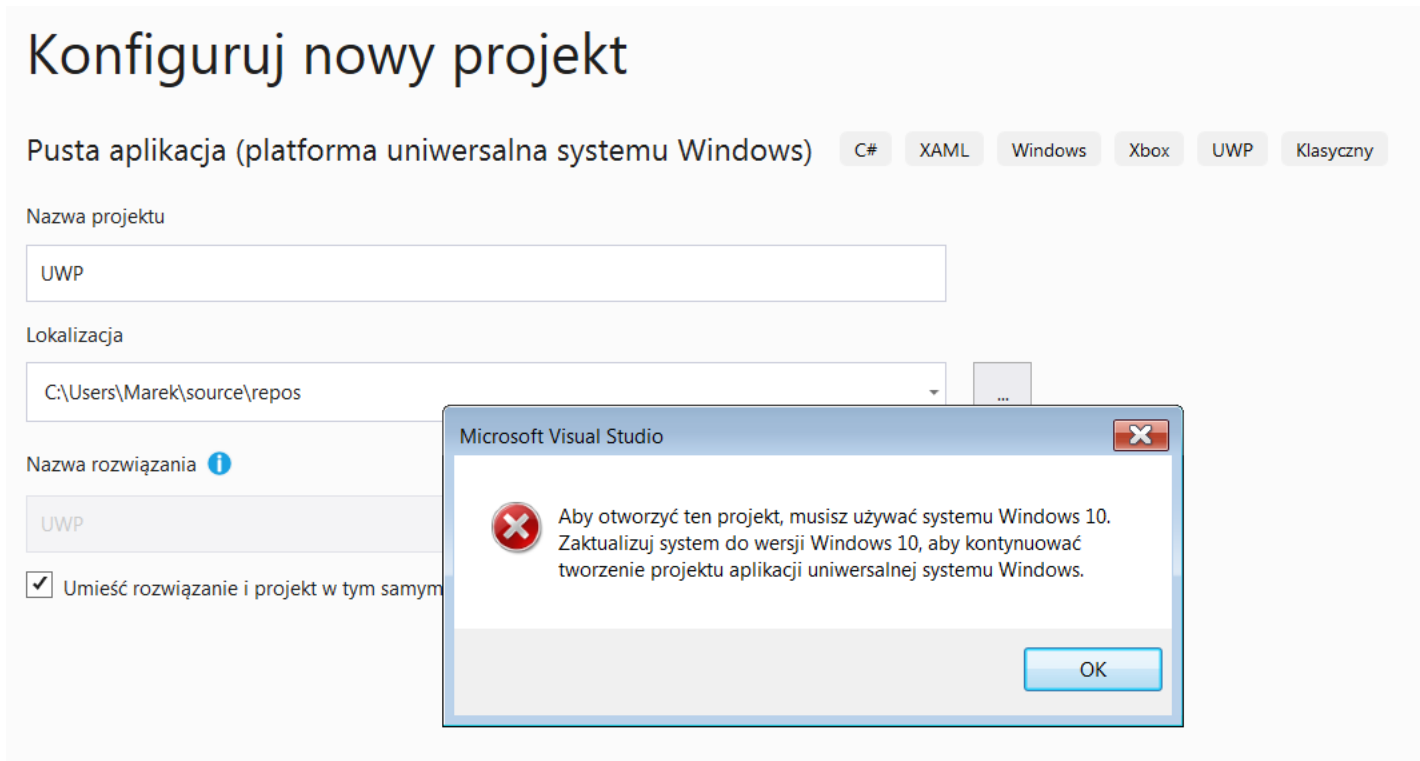
Aplikacja platformy uniwersalnej systemu Windows (UWP)

Uses Extensible Application Markup Language (XAML)



[Samouczek: tworzenie pierwszej aplikacji platformy Windows universal Visual Studio przy użyciu języka XAML i języka C#](#)

Aplikacja platformy uniwersalnej systemu Windows (UWP)



Próba uruchomienia nowego projektu w Win7

WinUI 3

Biblioteka interfejsu użytkownika systemu Windows (WinUI) 3 to najnowsza i zalecana struktura interfejsu użytkownika (UI) dla aplikacji klasycznych systemu Windows, w tym aplikacji zarządzanych, które używają C# i .NET oraz aplikacji natywnych, które używają C++ z interfejsem Win32 API. Włączając system Fluent Design do wszystkich środowisk, elementów sterujących i stylów, WinUI zapewnia spójne, intuicyjne i przystępne środowisko przy użyciu najnowszych wzorców interfejsu użytkownika. WinUI 3 jest dostępny jako część Windows App SDK. Zestaw SDK aplikacji systemu Windows zapewnia ujednolicony zestaw interfejsów API i narzędzi, których można używać w spójny sposób przez dowolną aplikację C++ Win32 lub C# .NET w szerokim zestawie docelowych wersji systemu operacyjnego Windows.

[Create a WinUI 3 app](https://docs.microsoft.com/en-us/windows/apps/get-started/?tabs=cpp-win32)

Linki

[Dokumentacja techniczna firmy Microsoft \(C#, C++...\)](#)

[Build desktop apps for Windows](#) - Ta dokumentacja zawiera najnowsze wskazówki dotyczące tworzenia aplikacji klasycznych dla systemów Windows 11 i Windows 10.