

Język C# – Wprowadzenie

C# (czyt. *C Sharp*) to nowoczesny język programowania opracowany przez firmę Microsoft na początku XXI wieku. Jego głównym przeznaczeniem jest tworzenie aplikacji działających w ekosystemie .NET, ale dzięki ewolucji platformy i otwarciu jej kodu źródłowego, C# stał się językiem uniwersalnym – używanym na Windows, Linux i macOS.

C# łączy w sobie prostotę znaną z języka Java i C++, a także dodaje liczne nowoczesne mechanizmy, takie jak LINQ, async/await czy pattern matching.

Historia



Prace nad C# rozpoczęto w 1999 roku pod kierownictwem Andersa Hejlsberga (znanego również z tworzenia języka Turbo Pascal i współtworzenia Delphi).

Pierwsza wersja została wydana w 2000 roku wraz z platformą .NET Framework. Początkowo C# służył głównie do tworzenia aplikacji desktopowych w systemie Windows. Z czasem, wraz z rozwojem ASP.NET, zyskał ogromne znaczenie w budowie aplikacji webowych.

W 2016 roku Microsoft udostępnił **.NET Core** jako projekt open source, co umożliwiło uruchamianie aplikacji w C# również na Linuxie i macOS. Od 2020 roku rozwijana jest zunifikowana platforma **.NET 5+**, która zastąpiła wcześniejsze rozwiązania.

Najważniejsze właściwości

1. **Obiektość** – wszystko w C# jest klasą lub strukturą.

2. **Silne typowanie** – podobnie jak w Javie, typy są jasno określone.
 3. **Nowoczesne mechanizmy** – m.in. właściwości (properties), zdarzenia (events), delegaty, LINQ.
 4. **Asynchroniczność** – `async/await` znacznie ułatwia programowanie współbieżne.
 5. **Rozszerzalność** – możliwość pisania własnych operatorów, atrybutów i metod rozszerzających.
 6. **Integracja z .NET** – bogate API, wsparcie dla wielu platform i usług chmurowych (Azure).
-

CLR, .NET i SDK

- **CLR (Common Language Runtime)** – odpowiednik JVM w Javie. Uruchamia kod w postaci CIL (Common Intermediate Language).
 - **.NET Runtime** – środowisko uruchomieniowe zawierające CLR i biblioteki standardowe.
 - **.NET SDK** – zestaw narzędzi dla programistów, w tym kompilator `CSC`, narzędzia do budowania (`dotnet build`) i testowania (`dotnet test`).
-

C# w praktyce

C# znajduje szerokie zastosowanie:

- **Aplikacje desktopowe** – Windows Forms, WPF, MAUI.
 - **Aplikacje webowe** – ASP.NET Core, Blazor.
 - **Aplikacje mobilne** – Xamarin, .NET MAUI.
 - **Gry komputerowe** – silnik Unity w całości opiera się na C#.
 - **Chmura i mikroserwisy** – pełna integracja z Microsoft Azure i Dockerem.
-

Popularne frameworki i biblioteki

- **ASP.NET Core** – framework webowy nowej generacji.
 - **Entity Framework Core** – ORM do obsługi baz danych.
 - **Blazor** – framework do budowania aplikacji webowych z użyciem C# zamiast JavaScriptu.
 - **ML.NET** – biblioteka do uczenia maszynowego.
 - **Unity** – silnik gier, w którym cała logika pisana jest w C#.
-

Rozwój języka

C# jest aktywnie rozwijany i regularnie wzbogacany o nowoczesne funkcje:

- **C# 3.0** – wprowadzenie LINQ i wyrażeń lambda.

- **C# 5.0** – async/await.
- **C# 7.0** – krotki (tuples), pattern matching.
- **C# 9.0** – rekordy (records).
- **C# 10–12** – global using, struktury rekordowe, ulepszony pattern matching, kolejne optymalizacje.

Microsoft wprowadza nowe funkcje w zgodzie z trendami, dzięki czemu C# jest jednym z najbardziej nowoczesnych języków wysokiego poziomu.

Spółeczność i ekosystem

C# i .NET mają ogromne wsparcie wśród programistów i firm:

- **IDE** – Visual Studio, Rider, Visual Studio Code.
- **Systemy budowania** – MSBuild, dotnet CLI.
- **Repozytoria** – NuGet (odpowiednik Maven Central).
- **Chmura** – ścisła integracja z Azure, ale również wsparcie dla AWS i Google Cloud.

Zalety i wady C#

Zalety C#	Wady C#
Nowoczesne konstrukcje językowe (async, LINQ, records)	Duża zależność od ekosystemu Microsoft
Uniwersalność – desktop, web, mobile, gry	Aplikacje Windows tradycyjnie mają większe wsparcie niż Linux/macOS
Integracja z Unity i silną obecność w gamedevie	Złożoność ekosystemu .NET (różne wersje frameworków, migracje)
Silne typowanie i wysoka wydajność CLR	Nieco mniejsza popularność poza środowiskiem Microsoft niż np. Java/Python
Otwartość (open source .NET, cross-platform)	Mniejsza ilość kursów/książek niż dla Pythona czy Javy