C# jegyzet

# C# keletkezése, története, összehasonlítás más nyelvekkel

## Keletkezés

A C# nyelvet a Microsoft fejlesztette ki, és először 2000-ben mutatták be. A nyelvet Anders Hejlsberg vezette, aki korábban a Turbo Pascal és a Delphi nyelvek fejlesztésében is részt vett. A C# a .NET keretrendszer része, és eredetileg a Java nyelvre adott válaszként született, bár mára önálló, erősen fejlett nyelvvé vált.

## Összehasonlítás

A C# a Java-val és a C++-al hasonlítható össze leginkább jelölésrendszere alapján. A Java-hoz hasonlóan a C# is objektum-orientált, és szintén erősen típusos nyelv. A C++-hoz képest viszont a C# egyszerűbb és biztonságosabb, mivel kevésbé enged low-level műveleteket (*pl. közvetlen memóriakezelés*).

# C# és VFP

A Visual FoxPro (VFP) adatbázis-központú programozási nyelv és fejlesztői környezet, amelyet főleg üzleti alkalmazások fejlesztésére használtak. A Visual FoxPro és a C# nagyon eltérő paradigmákon alapul.

|  |  |  |
| --- | --- | --- |
| **Tulajdonság** | **C#** | **Visual FoxPro** |
| **Paradigma** | Objektumorientált (OOP) és típusbiztos nyelv | Objektumorientált, de dinamikus és laza típuskezelés |
| **Használat** | Modern alkalmazások fejlesztése (*web, mobil, asztali, játékok, cloud, AI, stb.*) | Adatbázis-központú alkalmazások fejlesztése |
| **Típusosság** | Erősen típusos (*statikus*) | Gyengén típusos (*dinamikus adattípusokkal*) |

### Objektum-orientált paradigmára való áttérés

|  |  |  |
| --- | --- | --- |
| **Tulajdonság** | **C#** | **Visual FoxPro** |
| **OOP támogatás** | Teljes OOP (osztályok, öröklődés, polimorfizmus, interfészek, absztrakt osztályok) | OOP-t támogat, de korlátozott |
| **Metódusok túlterhelése** | Igen | Nem |
| **Interfészek és absztrakt osztályok** | Igen | Nem |

### **Adatbázis-kezelés különbségei**

A C#-ban nincs beépített adatbázis-kezelés, de a .NET keretrendszer széleskörű támogatást nyújt különböző adatbázisokhoz (*pl. SQL Server, MySQL, SQLite*) az **ADO.NET**, **Entity Framework** vagy **Dapper** segítségével.

|  |  |  |
| --- | --- | --- |
| **Tulajdonság** | **C#** | **Visual FoxPro** |
| **Adatbázis-motor** | Külső adatbázisokkal (SQL Server, MySQL, PostgreSQL, SQLite, stb.) működik | Beépített adatbázis (DBF fájlok), SQL támogatás korlátozott |
| **Kapcsolat adatbázishoz** | ADO.NET, Entity Framework, Dapper | Beépített SQL Pass-Through, Cursor, DBF táblák |
| **Többfelhasználós támogatás** | SQL szerverekkel skálázható | Többfelhasználós, de fájlalapú DBF szerkezet miatt sérülékeny |

### Felhasználói felület (UI)

|  |  |  |
| --- | --- | --- |
| **Tulajdonság** | **C#** | **Visual FoxPro** |
| **UI keretrendszer** | WinForms, WPF, Blazor, MAUI, ASP.NET, Unity | Beépített UI komponensek |
| **Webes támogatás** | Igen (ASP.NET, Blazor) | Nem (csak asztali alkalmazások) |
| **Korszerű UI támogatás** | Modern UI/UX lehetőségek | Elavult felület, korlátozott lehetőségek |

1. **Miért érdemes C#-t tanulni?**: A C# széles körben használatos Windows alkalmazások fejlesztésére, játékfejlesztésre (Unity keretrendszerrel), valamint webalkalmazásokhoz (ASP.NET). Emellett a .NET keretrendszer többplatformossága miatt ma már Linuxon és macOS-en is futnak C# programok.
2. **Elemi adattípusok** – Megismerkedni a C# alapvető adattípusaival, mint az int, double, char, bool, string, stb.
3. **Összetett adatszerkezetek** – Például tömbök, listák (List<T>), szótárak (Dictionary<K,V>), halmazok (HashSet<T>), valamint saját osztályok és struktúrák.
4. **Vezérlési szerkezetek** – Feltételes elágazások (if, switch), ciklusok (for, while, foreach), kivételkezelés (try-catch).
5. **Metódusok és függvények** – Hogyan szervezd a kódot újrafelhasználható egységekké, paraméterátadás, visszatérési értékek.
6. **OOP (Objektumorientált programozás)** – Osztályok, öröklődés, interfészek, absztrakt osztályok, polimorfizmus.
7. **Fájlkezelés és adatbázis kapcsolat** – Hogyan olvashatsz és írhatsz fájlokat (StreamReader, StreamWriter), illetve használhatsz adatbázisokat (pl. SQL kapcsolat Entity Framework segítségével).
8. **Aszinkron programozás** – async és await, párhuzamos végrehajtás (Task, Thread).