C# Programozási Nyelv

# C# Alapok

**A nyelv története:** A C# programozási nyelv a Microsoft új fejlesztési környezetével, a 2002-ben megjelent Visual Studio.NET programcsomaggal, annak részeként jelent meg.

A nyelv jellemzői

A C# az új .NET keretrendszer bázisnyelve. Tipikusan ehhez a keretrendszerhez tervezték, nem véletlen, hogy a szabványosítási azonosítójuk is csak egy számmal tér el egymástól. A nyelv teljesen komponens orientált. A fejlesztők számára a C++ hatékonyságát, és a Visual Basic fejlesztés gyorsaságát, egyszerűségét ötvözték ebben az eszközben.

A C# nyelv legfontosabb elemei a következők:

• Professzionális, Neumann-elvű. Nagy programok, akár rendszerprogramok írására alkalmas.

• A program több fordítási egységből – modulból – vagy fájlból áll. Minden egyes modulnak vagy fájlnak azonos a szerkezete.

• Egy sorba több utasítás is írható. Az utasítások lezáró jele a pontosvessző (;). Minden változót deklarálni kell. Változók, függvények elnevezésében az ékezetes karakterek használhatóak, a kis- és nagybetűk különbözőek.

I. Alapismeretek

• Minden utasítás helyére összetett utasítás (blokk) írható. Az összetett utasítást a kapcsos zárójelek közé { } írt utasítások definiálják.

• Nincs mutatóhasználat; biztonságos a vektorhasználat.

• Boxing, unboxing. Minden típus őse az object, így például egy egész típust (int) csomagolhatunk objektumba (boxing) illetve vissza (unboxing).

• Függvények definíciói nem ágyazhatók egymásba, önmagát meghívhatja

(rekurzió). Tehát függvénydefiníció esetén nincs blokkstruktúra. Blokkon belül statikus vagy dinamikus élettartamú változók deklarálhatók.

Függvénypolimorfizmus megengedett.

• Kezdő paraméter-értékadás, változó paraméterszámú függvény deklarálása.

• Hierarchikus névterekben használt osztályok. Mivel minden osztály, ezért a „program”, a Main függvény public static hozzáférésű. Több osztály is tartalmazhat Main függvényt, de ilyen esetben a fordításkor meg kell mondani, hogy melyik osztálybeli Main függvény legyen az aktuális induló (/main:osztálynév).

• Privát konstruktor (nem akarunk egy példányt se), statikus konstruktor

(statikus mezők inicializálása, mindig példány konstruktor előtt hívódik

meg, futási időben az osztálybetöltő hívja meg) használata.

## A program szerkezete

Egy C# program tetszőleges számú fordítási egységből (modulból) állhat.

Szokás ezen modulok vagy fájlok összességét projektnek nevezni.

A program ezen modulokban definiált osztályok összessége.

Egy fájl tartalmazza az egyes modulok közti hivatkozásokat (using), osztálytípus-, változó- és függvénydeklarációkat.

Egy modul egy tetszőleges névtér része lehet. A névtér (namespace) az a

logikai egység, amiben az azonosítónknak egyedinek kell lennie. Nem kötelező

a névtér definiálása, ebben az esetben az ún. név nélküli névtér része lesz az

adott kód.

Névtér szerkezete:

namespace új\_névtérnév

{

class új\_osztálynév

{

Típusdefiníció;

Függvénydefiníció;

}

…

}

I. Alapismeretek

Függvények definíciójának formája:

visszaadott\_típus név(argumentumlista, ha van)

{

változó definíciók,

deklarációk

és utasítások

}

főprogram.

A programban használt neveknek betűvel vagy \_ jellel kell kezdődniük,

majd a második karaktertől tetszőleges betű és szám kombinációja állhat. A

nyelvben a kis- és nagybetűk különbözőek, így például a következő két név sem azonos:

Az azonosítónevek hossza általában 32 karakter lehet, de sok rendszerben az

igényeknek megfelelően lehet ezen változtatni.

Egy forrásállomány szerkesztése során magyarázatokat is elhelyezhetünk a

/\* és \*/ jelek között. Az ilyen megjegyzés több soron át is tarthat. Egy soron

belül a // jel után írhatunk magyarázatot.

A mai fejlesztőkörnyezetekben egyáltalán nem ritka, hogy valamilyen

speciális megjegyzést arra használjanak fel, hogy ennek a programszövegből

történő kigyűjtésével a forrásállománynak vagy magának a programnak egy

dokumentációját kapják. Ezt a kigyűjtést a fordító végzi el.

Ha a keretrendszerben egy változó vagy függvény definíciója elé beszúrjuk a

A C stílusú nyelvek hagyományaként a C# nyelvnek sem részei a beolvasó és

kiíró utasítások, így könyvtári szolgáltatásokra kell hagyatkoznunk, ami a

System névtér része, és ennek a névtérnek a Console osztálya biztosítja a klasszikus képernyőre írás, Console.WriteLine() és billentyűzetolvasás, Console.ReadLine() feladatát.

Változók definiálása

A nyelvben a változók definiálásának alakja:

típus változónév ;

Azonos típusú változókat egymástól vesszővel elválasztva definiálhatunk. A

típusok ismerete nélkül nézzünk néhány példát változók definiálására.

Példa:

char ch; // ch változó karakter típusú

int egy, tizenegy; // az egy és tizenegy egész típusú

Változók lehetnek külsők, azaz függvényen kívüliek, vagy belsők, azaz

függvényen belüliek. A függvényen kívüli változók is természetesen csak osztályon belüliek lehetnek. Gyakran hívják ezeket a változókat adatmezőknek is.