C# Bevezetés

Üdvözlök minden kedves Olvasót! Ez most egy cikksorozat első része, és a sorozat középpontjában a C# (ejtsd: C-Sharp), ill. a .Net Framework fog állni. Ajánlom mindenkinek, aki már használt objektum-orientált programozási nyelvet, de nem túl régen programozik, vagy csak ki szeretne próbálni valami újat. A célom, hogy a rendszeres olvasóval egy-egy konkrét problémával foglalkozó programot megír(at)va bevezessem őt a Framework összetevőibe, valamint a 2d/3d grafika, internetprogramozás, adatbázisok vagy a digitális képfeldolgozás alapjaiba. Mindezek közben persze egyre többet és többet tárok majd fel a .Net Framework-ről és a C#-ról. Ez az első rész egy kicsit ki fog lógni a sorból, mert a többi cikkel ellentétben sokkal általánosabb lesz: A .Net Framework-öt nem ismerőknek szól, és megpróbálja majd felkelteni az említett technológia vagy a C# nyelv iránti érdeklődésüket. A sorozat követését annak is ajánlom, aki a C#-tól eltérő, de szintén .Net nyelvvel foglalkozik!

A mi nyelvünk, a C#

A C#… van benne egy C betű. Nem meglepő, hiszen szintaktikailag sok közös vonása van a C nyelvvel. Aki mélyebbről is ismeri, az még a Java-val és a Visual Basic-kel is talál benne hasonlóságokat. A C# az említett három nyelv összeolvadásából született, és megpróbálta mindegyik előnyeit magával hozni: A C-től a tiszta, tömör, bár egy laikus számára néha átláthatatlan szintaktikát örökölte, a VB-től a programozás és a fejlesztői környezet használatának egyszerűségét, míg a Javától a futás közbeni fordítást, a cross-platform tulajdonságot, valamint az alapkönyvtár ötletét hozta magával. Ez a „cross-platform” annyit tesz, hogy több számítógép-architektúrán és több operációs rendszeren is elfutnak a C#-ban írt programok (bizonyos feltételek mellett). Már az elejétől fogva mennek ezek a programok mind PC-n, mind pedig Windows-alapú, hordozható számítógépeken is (pl. Pocket PC), és mivel a Microsoft (ez az egész az ő szüleménye) néhány dolgot közzé tett, megszületett a Project Mono, ami lehetővé teszi ezen alkalmazások Linuxon való futtatását is. Persze nem ugyanaz a bináris állomány fut el mindenütt, de a forráskódon csak minimális változtatásokat kell alkalmazni.

A .NET Framework újabb technológia az MS-tol

De mi teszi mindezt lehetővé? Ezen előnyök nagy része tulajdonképpen nem is magában a nyelvben rejlik, hanem abban a .Net Framework-nek elkeresztelt (.Net Keretrendszer) futtatási alrétegben, amit még a C#-on kívül sok-sok nyelv használ. Egy kicsit eltúlozva a Framework a Windows elavult, nehezen használható elsődleges programozási felületét, a Windows API-t hívatott leváltani, amit majd a következő windows-verzióban, a Longhornban meg is tesz. Bár egyes funkciókat megkövetel az őt használó nyelvektől, cserébe viszont annál többet kínál számukra. Ha egy program valamilyen operációs rendszertől függő eljárást szeretne elérni, azt a kérést nem a Linuxnak vagy a Windowsnak küldi el, hanem a Framework-nek, ami aztán megfelelő átalakítások után továbbküldi azt a „célszemélynek”. Így a programunk alatt csak egy a megfelelő operációs rendszerhez írt Framework-re van szükségünk (Linuxnál ez lenne a Mono), és onnantól kezdve kész van a „port”. Ennek hátránya viszont, hogy mivel a .Net Framework jelenleg egyik operációs rendszerben sincs benne gyárilag, ránk, programozókra hárul a feladat, hogy eljuttassuk a keretrendszert a felhasználóhoz.

Előnyök boncolgatása

Vegyünk sorra néhány a Keretrendszer által nyújtott dolgot:

Általános, szabványosított programozási felület. Ezt részben a minden nyelv számára elérhető, óriási BCL (base class library, alapvető osztályok könyvtára) teszi lehetővé, így tehát számítógéptől, operációs rendszertől és programozási nyelvtől függetlenül mindenki ugyanúgy hozhat létre / olvashat fájlokat, rajzolhat képeket, jeleníthet meg grafikus ablakokat, kommunikálhat az interneten stb… Persze a különböző nyelveken íródott programok/könyvtárak gond nélkül hívogathatják egymás eljárásait, mintha csak egy nyelven íródtak volna.

Futás közbeni fordítás. Ez a program futása elején eléggé lelassíthatja a dolgokat, de utána nem lesz sokkal lassabb, még mindig többször olyan gyors marad, mint egy VB-ben írt alkalmazás. Sokkal ügyesebben oldották meg ezt, mint a Java esetében, és így nem egy sima interpretert kaptunk, hanem egy másfajta JIT (Just-In-Time) fordítót, ami nagyságrendekkel hatékonyabb. Ennek az az oka, hogy egyrészt nem parancsonként, hanem egész eljárásonként fordít (tehát az eljárás meghívása elején egy picit várni kell, de utána natív kód sebességével fut a programunk), másrészt pedig ez lehetőséget ad a felhasználó hardver- és szoftverkörnyezetéhez optimalizált program létrehozására (magyarul pl. a forráskód megváltoztatása nélkül kaphatunk programot, ami szuperül megy régi gépeken is, de egy P4-en kihasználja a legújabb utasításkészleteket bár erre még éppen nem képes, de ténylegesen várható ilyen funkció a jövőben). A lefordított eljárások natív „képe” eltárolódik, így azt a következő meghíváskor már nem kell lefordítani.

Garbage Collector („szemétgyűjtő”). A C guruk tudják mennyi fejfájás származhat a memóriakezelésből. A VB-seknek és Java-soknak gőzük sincs róla, mert nekik már volt valami hasonló, ami levette a vállukról a memóriabajlódások többségét. Egyelőre csak azt kell megjegyeznünk, hogy még ha van is egy-két hátránya, képes hihetetlenül megkönnyíteni az életünket.

Persze az imént említett dolgok szorosan összefüggnek, hiszen például a „cross-platform”-ot a BCL és a JIT fordító közös erővel érik el. Eddig a .NET Framework tulajdonságairól a teljesség igénye nélkül írtam, annyit lehetne boncolgatni a témát, hogy azt egyelőre felesleges lenne az olvasóra zúdítani. Ha letöltjük az SDK-t, és elkezdjük tanulgatni a részleteket, ott úgy is mindent bőségesen kifejtenek.

Mire lesz szükségem?

Meglepő módon semmire. Vegyük elő a Jegyzettömböt és kódoljunk! Na jó, csak vicceltem. Bár tényleg megtehetnénk, hiszen a szükséges fordítók részei a Framework-nek (ami ingyenes), de azért könnyebb dolgunk lesz egy IDE-vel. Ám nem kell aggódni, most e téren is jó hírt hozok: A Visual C# 2005, Visual Basic .NET 2005, Visual Web Developer 2005 és SQL Server 2005 Express (butított, de minden meg van benne, amire csak szükségünk lehet) verzióit tanulási célokra ingyen a diákok kezébe adja a Microsoft! Így hát nem kell mást tenni, mint elbarangolni a http://lab.msdn.microsoft.com/vs2005/ címre, és egy regisztráció után már tölthetjük is a fejlesztőkörnyezetet. Minden, amit ott van az csak béta még, de 2005 eleje körül már várhatóak a végleges változatok is. Várunk mi addig? NEM!

És ami még várható…

A későbbi részekben, mint azt fentebb is említettem, konkrét feladatokat fogunk megoldani C#-ban. Ez a rész csak egy ismertető volt, de a cikksorozat későbbi részei sem fogják megtanítani, mi is a különbség egy for-ciklus és egy if-elágaztatás között. Ha viszont az olvasó érdekesnek találja a .Net-et, letöltheti az említett URL címről a szükséges programokat, és legalább minimális gyakorlatot szerez a következő számig (ha már ismer egy objektum-orientált nyelvet, ez úgysem lesz gond). A C# valóban egy olyan nyelv, aminél nem kell az általa állított akadályokra figyelni, és ténylegesen teljes erődből az aktuális problémára/programra lehet koncentrálni. Mindenkinek ajánlom, nem csak kezdőknek!

Pados Károly