



Jelen dokumentumra a Creative Commons Nevezd meg! – Ne add el! – Ne változtasd meg! 3.0 Unported licenc feltételei érvényesek: a művet a felhasználó másolhatja, többszörözheti, továbbadhatja, amennyiben feltünteti a szerzők nevét és a mű címét, de nem módosíthatja, és kereskedelmi forgalomba se hozhatja.

Visual Basic programok átalakítása C#-ra

Juhász Tibor–Tóth Bertalan: Programozási ismeretek kezdő versenyzőknek
(Műszaki Könyvkiadó, 2017)
Kiegészítés

A MESTER online feladatbank jelenleg* nem engedi meg a Visual Basic nyelven írt programok tesztelését. A feltöltés előtt alakítsuk át a megoldást C# nyelvre! Az átalakítás egyszerű és kiváló eszköze az *Instant C#*, amely a teljes projektet (illetve *solution*-t) tudja konvertálni, a beállításokkal (például a konzol átirányításával) együtt. Az átalakítást és tesztelést akkor is elvégezhetjük, ha nem ismerjük magát a C# programozási nyelvet.

Az *Instant C#* ingyenes változata egyszerre legfeljebb 2000 sort konvertál, egy kódfájl mérete pedig legfeljebb 200 sor lehet. Ez bőven elegendő a versenyfeladatok megoldásának átalakításához. A korlátozás miatt ne használjunk több projektet tartalmazó *solution*-t! Erre a versenyen sincs szükség. A projekteket külön-külön is átalakíthatjuk (ha a *solution* esetleg mégis több projektet tartalmaz).

Az *Instant C#* letöltése és telepítése

Az *Instant C#* programot a Tangible Software Solutions webhelyéről tölthetjük le:

http://www.tangiblesoftwaresolutions.com/Free_Editions.html

Install *Instant C#* (converts VB.NET code to C#)

A letöltés után indítsuk el a telepítőprogramot! Fogadjuk el a licencfeltételeket, majd lépkedjünk tovább! A telepítés nagyon egyszerű, semmilyen különösebb beavatkozást nem igényel.

Windows 7 (vagy korábbi Windows változatok) esetén szükség lehet a legalább 4.0 .NET Framework telepítésére. Erről külön tájékoztatást kapunk.

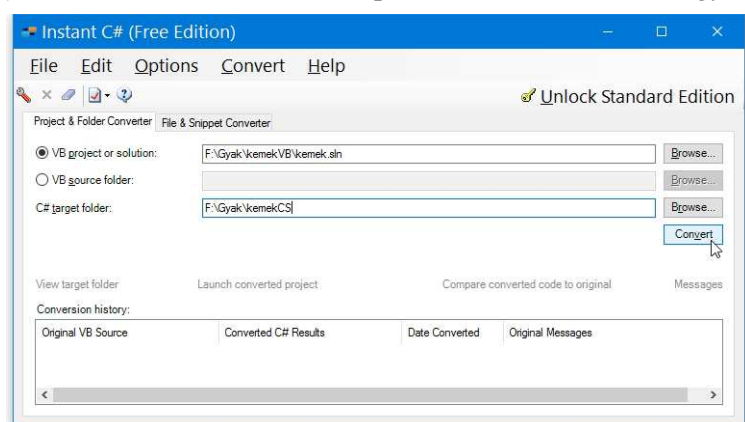
A Visual Basic programok átalakítása C#-ra

Tennivalók az átalakítás előtt

A MESTER feladatbank az online tesztelés során a C# programokat a .NET Framework *Mono* implementációjával kezeli. Visual Basic programjainkat ennek megfelelően kell elkészíteni (illetve átalakítani). Ne használjuk például a *Microsoft.VisualBasic* névtér elemeit (például a *DateAndTime* vagy a *Strings* objektumosztályt), mert ezek nem szerepelnek a *Mono*-ban! A további korlátozásokat, módosításokat ismertetőnk végén foglaljuk össze.

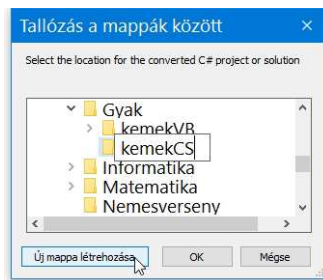
Az átalakítás menete

Indítsuk el az *Instant C#*-ot! A *Project & Folder Converter* panelen a *Browse* gombok segítségével jelöljük ki a Visual Basic megoldásfájlt (*.sln* kiterjesztés), illetve a célmappát, ahová a C# megoldás kerüljön!

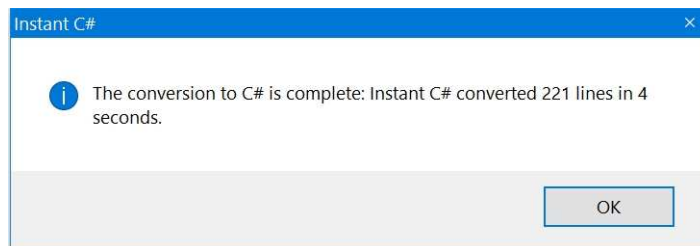


* 2016-ban

Ha nincs megfelelő mappa, akkor a tallózás ablakban az *Új mappa létrehozása* gombbal hozhatjuk létre.



Új mappa létrehozása



Az átalakítás sikeres végrehajtása

Kattintsunk a *Convert* gombra! Pár másodpercen belül megjelenik a sikeres végrehajtásról tudósító üzenet.

Tennivalók az átalakítás után

Az átalakított C# programot nyissuk meg a Visual Studióval (dupla kattintás az *.sln* fájlra)! A projekt szerkezete, használata, működése (például futtatás) teljesen megfelel a Visual Basic-nek.

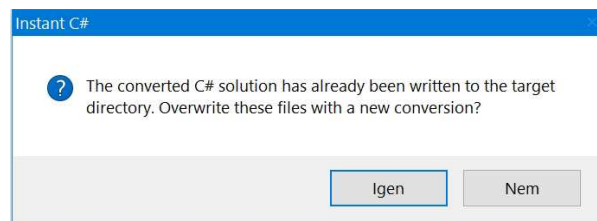
A forráskód elejéről töröljük a következő sorokat:

```
using System.Data;  
using System.Xml.Linq;  
using System.Threading.Tasks
```

Mentsük, majd futtassuk a C# programot! Ha minden hibátlan, akkor a projektmappában lévő *Module1.cs* fájlt feltölthetjük és tesztelhetjük a MESTER webhelyén.

A javítgatások idejére ne zárjuk be az Instant C# programot, mert a szövegdozokban megmarad a mappák elérési útja, így elegendő a *Convert* gombra kattintani az ismételt átalakításhoz.

Ha már létezik a C# megoldás, akkor a program rákérdez a felülírásra, amit az *Igen* gombbal fogadhatunk el.



Esetenként (nagyon ritkán ☺) előfordulhat, hogy hibás sort tartalmaz a C# forráskód (hullámos kék vonallal aláhúzva). Ekkor sajnos módosítani kell a Visual Basic program megfelelő utasítását.

A MESTER hibaüzenetei

A tesztelőrendszer *Fordítási hiba* üzenetet küld, ha olyan programelemet talált a forráskódban, amit nem ismer fel (ez természetesen nem biztos, hogy a megoldásunk hibája).

Utolsó beadás eredménye

Fordítási hiba:
feladat.cs(5,14): error CS0234: The type or namespace name 'Data' does not exist in the namespace '
feladat.cs(8,18): error CS0234: The type or namespace name 'Linq' does not exist in the namespace '
feladat.cs(9,24): error CS0234: The type or namespace name 'Tasks' does not exist in the namespace '
Compilation failed: 3 error(s), 0 warnings

Összpont: 0/100

Teszt#	Pont	...Üzenet...	Futási idő
1.0	0/0	Fordítási hiba	sec

Beadva: 2016-01-25 20:59:47.0

A tesztelőrendszer hibaiüzenete

Elfelejtettük törölni a fent említett using sorokat (a hibaiüzenet szerint az 5., 8. és 9. sort).

Utolsó beadás eredménye

Fordítási hiba:
 feladat.cs(17,6): error CS1041: Identifier expected, 'static' is a keyword
 feladat.cs(17,13): error CS1001: Unexpected symbol 'Microsoft', expecting identifier
 Compilation failed: 2 error(s), 0 warnings

Összpont: 0/100

Teszt#	Pont	...Üzenet...	Futási idő
1.0	0/0	Fordítási hiba	sec

Beadva: 2016-01-25 21:01:55.0

*A tesztelőrendszer hibaiüzenete
 Számára ismeretlen névteret tartalmaz a program.
 Ne alkalmazzunk VisualBasic névtérrel minősített metódushívásokat (lásd alább)!*

Esetenként előfordulhat, hogy az online feladatbank tesztelőprogramja vagy a teszteléshez használt adatfájlok tartalmaznak hibát. Ebben az esetben (ha a megoldásunk biztosan jó ☺) a feladat pontos megjelölésével (szint, téma, feladat) értesítjük az online feladatbank üzemeltetőit!

*Az online tesztelőrendszer üzenete
 a tesztelés közben előforduló futási hibáról*

An Error Occurred:

java.lang.ArrayIndexOutOfBoundsException

- + Stack Trace
- + Component Tree
- + Scoped Variables

Korlátozások a Visual Basic programokban

Az online feladatbank használatakor a Visual Basic programok megírásánál ügyeljünk a következőkre:*

1. Ne importáljuk a *Console* osztályt, helyette minden *read/write* utasításhoz írjuk ki a minősítést! Például: `Console.ReadLine()` vagy `Console.WriteLine(...)`
2. Automatikus típuskonverzió helyett alkalmazzuk a *CInt* stb. típuskonverziós függvényeket (ha sztringnek adunk karakter típusú kezdőértéket, akkor a *CStr*-t)! Például:
`N = CInt(Console.ReadLine())`
 vagy:
`Kezd = CInt(Temp(0))`
3. A *Char* típusú literálok után írjuk ki a *c* típusazonosító karaktert! Például:
`Dim Melyik As Char`
`Melyik = "A"c`
4. Máshol se keverjük össze az egyetlen karakterből álló sztringet a *Char* típusal! (A *C#* a karaktert aposztrófjelek közé írja, a sztringet pedig idézőjelek közé, például: `'A'`, `"A"`). Visual Basic-ben idézőjelet használunk a karakterek megadására, de írjuk utána a *c* típusazonosítót! Egy sztringhez viszont már hozzáfűzhetünk karaktert.
5. A *Split* metódusnak mindig meg kell adni az elválasztó karaktert (a típusazonosító *c*-vel együtt)! Például:
`Temp = Console.ReadLine.Split(" "c)`
6. Kollekciónkat (*List*, *SortedList*, *HashSet*, *Dictionary*, *SortedDictionary*), általában használhatunk, de **a Mono nem ismeri a rendezett halmazt (*SortedSet*)**.

Fordítási hiba:
 feladat.cs(18,18): error CS0246: The type or namespace name SortedSet could not be found.
 Compilation failed: 1 error(s), 0 warnings

* A felsorolás tapasztalati úton készült, és nem tekinthető teljesnek. A feladatbank használatáról nem áll rendelkezésre dokumentáció.

7. Nem használható olyan metódus, amelynél a C# forráskód elején megjelenik a `using static Microsoft.VisualBasic.xx` utasítás. Ilyenkor keressük meg a C# forráskódban a `Microsoft.VisualBasic.xx.xx` metódushívást (például `Microsoft.VisualBasic.DateAndTime.TimeSerial`), és ezt kell mással helyettesítenünk!
8. Nem használhatunk olyan függvényt, amelynél a C# forráskódban megjelenik előtte a `Simulate` minősítés, például `Simulate.StrReverse`. Tehát nem használható az *StrReverse*, viszont használható az *StrDup* függvény (ez utóbbi csak karaktert tud ismételni, sztringet nem).

Tapasztalataink alapján folytatni fogjuk a felsorolást. ☺ A sztringek kezelésével kapcsolatban az Olvasó figyelmébe ajánljuk még a *Programozási ismeretek kezdő versenyzőknek* feladatgyűjtemény 112. oldalát.

Felhívjuk a figyelmet arra, hogy **a programozási versenyeken (a Visual Basic programokban) ezekre a korlátozásokra általában nincs szükség**, azonban ott is kerüljük el a *VisualBasic* névtér tagjainak alkalmazását!

Összeállította: Juhász Tibor