

## A MESTER – online programozási feladatbank használata a Visual Basic programok teszteléséhez

Visual Studio 2015, Instant C# 18.3.21

**A Visual Basic programok C#-ra történő átalakításra, illetve az alábbi korlátozások elkerülésére a programozási versenyeken (és az érettségien) nincs szükség. Ezeken a szokásos módon megírt Visual Basic programot kell feltölteni (beadni).**

### A feladatbank webhelye

<https://mester.inf.elte.hu/> (ha nem töltődik le, akkor a https-t is írd be a URL-be!)

A bejelentkezéshez célszerű egy csak erre a célra használt *gmail* fiókot létrehozni.

A webhely tanúsítványa nem megbízható, de engedélyezheted a böngészőnek a megnyitást. ☺

Az első belépés előtt kattints a *Regisztrálok* gombra, és válaszolj a kérdésekre!

Először válassz *Témát* (azon belül *Szintet* és *Témát* ☺), majd *Feladatot*! A *Megoldom* gombra kattintva töltheted fel a megoldásodat.

### A Visual Basic programok átalakítása C#-ra

A következő webhelyről töltsd le, és telepítsd az Instant C# programot:

[http://www.tangiblesoftwaresolutions.com/Free\\_Editions.html](http://www.tangiblesoftwaresolutions.com/Free_Editions.html)

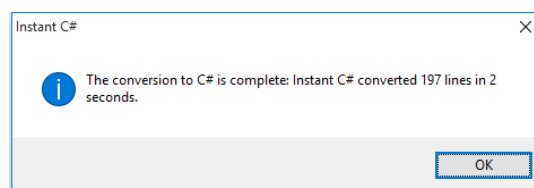
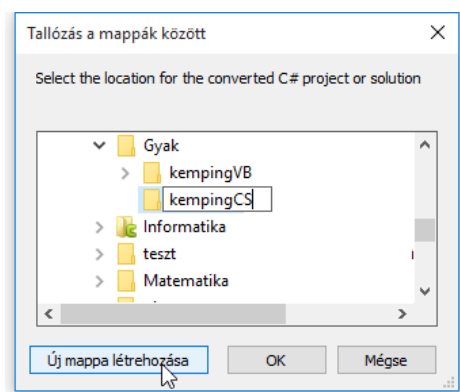
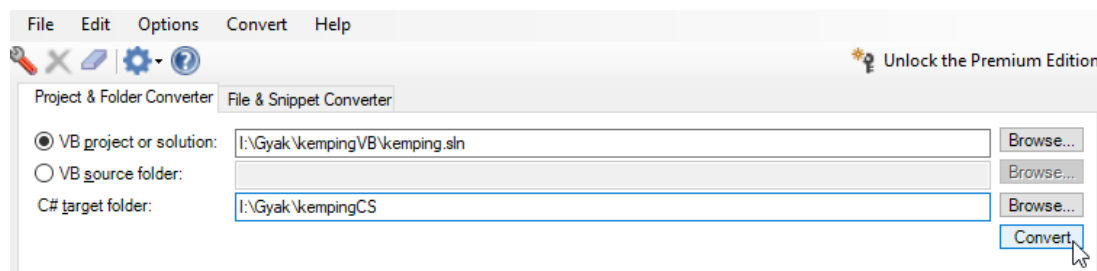
Instant C# (Convert Between VB.NET and C#)

A teljes *solution*-t át tudja alakítani VB-ről C#-ra (a példafájlok és az input/output átirányítások is átkerülnek a C# projektbe). Az ingyenes változat egyszerre legfeljebb 2000 sort konvertál, egy kódfájl mérete pedig legfeljebb 200 sor lehet. Ez bőven elegendő a versenyfeladatok megoldásának átalakításához.

A korlátozás miatt ne használj több projektet tartalmazó *solution*-t! A projekteket külön-külön is átalakíthatod (ha a *solution* esetleg mégis több projektet tartalmaz).

### Az átalakítás menete

A *Project & Folder Converter* panelen a *Browse* gombok segítségével jelöld ki a Visual Basic megoldásfájlt (*.sln* kiterjesztés), illetve a célmappát, ahová majd a C# megoldás kerül! A megoldásfájl helyett a projektmappában található *.vbproj* kiterjesztésű projektfájl is kiválasztható. Ha nem létezik megfelelő célmappa, akkor a tallózás ablakban az *Új mappa létrehozása* gombbal készítheted el.



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

## Tennivalók az átalakítás után

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

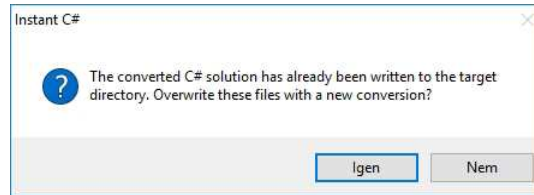
**A forráskód elejéről töröld a következő sorokat:**

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

Mentsd el, majd próbáld ki (futtasd) a C# programot! Az Instant C# a projekt beállításait is átmásolja, így megmarad a konzol átirányítása, illetve a bin/debug mappában lévő tesztfájl. Ha hiba nélkül lefut, akkor feltöltheted a megoldást (a projektmappában lévő *Module1.cs* fájlt) a MESTER webhelyére.

Célszerű a javítgatások idejére megnyitva hagyni az Instant C# programot, mert a szövegdobozok tartalmazzák a mappák elérési útját, így elegendő a *Convert* gombra kattintani az ismételt átalakításhoz.

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



Esetenként (nagyon ritkán ☺) előfordulhat, hogy hibás sort tartalmaz a C# forráskód (hullámos vonallal aláhúzva). Ekkor sajnos módosítani kell a Visual Basic program megfelelő utasítását. Az ilyen jellegű hibákkal fordulj hozzám!

## A Visual Basic programok korlátozásai

A Visual Basic programokban ügyelj a következőkre:\*

1. **A *Split* metódusnak mindig meg kell adni az elválasztókaraktert!** Például:

```
Temp = Console.ReadLine.Split(" ")
```

A C# a karakter típusú literált aposztrófjelek közé írja (' '), a sztringek viszont idézőjelek között szerepelnek.

2. Nem használható olyan metódus, amelynél a C# forráskódban megjelenik a *Microsoft.VisualBasic* vagy az *xxHelper* minősítés, például:

```
Microsoft.VisualBasic.Strings.Asc
NumericHelper.Val
ConversionHelper.StrReverse
DateHelper.Datediff stb.
```

Ez utóbbiaknál a projektfájlban találsz egy hasonló, *xxHelper.cs* kódfájlt, amire szükség van a program futtatásánál. Ezt azonban nem lehet feltölteni a MESTER-re. Ilyenkor sajnos a metódusokat mással helyettesítve, módosítanod kell a programodat.

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

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

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

*A tesztelőrendszer hibáüzenete*

*Elfelejtetted törölni a fent megadott using sorokat.*

*A tesztelőrendszer hibáüzenete*

*Számára ismeretlen azonosítókat tartalmaz a program. A VisualBasic névtérrel minősített, illetve a NumericHelper kódmodulban szereplő metódushívások helyett mást kell alkalmaznod.*

Utolsó beadás eredménye

Fordítási hiba:

```
feladat.cs(62,36): error CS0234: The type or namespace name 'Strings' does not exist in the namespace
feladat.cs(67,8): error CS0103: The name 'NumericHelper' does not exist in the current context
Compilation failed: 2 error(s), 0 warnings
```

Összpont: 0/100

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

Beadva: 2018-04-03 19:33:38.0

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

Esetenként előfordulhat, hogy az online feladatbank tesztelőprogramja vagy a teszteléshez használt adatfájlok tartalmaznak hibát. Ezek jellegzetes hibaüzenete:

#### An Error Occurred:

```
java.lang.ArrayIndexOutOfBoundsException
```

+ Stack Trace

+ Component Tree

+ Scoped Variables

Ha ilyen hibát tapasztalsz (pedig a programod biztosan jó ☺), akkor értesíts róla a feladat pontos megjelölésével (szint, téma, feladat)!

### A régebbi változatoknál előforduló korlátozások

A Visual Basic és az Instant C# (illetve a szerveren a programot futtató Mono) régebbi változatainál még a következő korlátozásokra kellett ügyelni:\*

1. A *Console* osztályt importálása helyett minden *read/write* utasításhoz kiírtuk a minősítést. Például:

```
Console.ReadLine() vagy Console.WriteLine(...)
```

2. Automatikus típuskonverzió helyett a *CInt* stb. típuskonverziós függvényeket alkalmaztuk (sztring esetén a karakter típusú kezdőértéknél a *CStr*-t). Például:

```
N = CInt(Console.ReadLine())
```

vagy:

```
Kezd = CInt(Temp(0))
```

4. Megkülönböztettük az egyetlen karakterből álló sztringet és a *Char* típust. A Visual Basic idézőjelet használ a karakterek megadására, de utána írjuk a *c* típusazonosító karaktert:

```
Dim Melyik As Char
```

```
Melyik = "A"c
```

5. A *Split* metódusnál az elválasztókarakter után is megadtuk a típusazonosító *c* karaktert! Például:

```
Temp = Console.ReadLine.Split(" "c)
```

Összeállította: Juhász Tibor

---

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