

# Programozás I.

Microsoft Visual Studio 2012

Hello C# World!

Változók típusai

Gyakorló feladatok

Algoritmus alkotási feladatok

# Hallgatói Tájékoztató

**A jelen bemutatóban található adatok, tudnivalók és információk a számonkérendő anyag vázlatát képezik. Ismeretük szükséges, de nem elégséges feltétele a sikeres zárthelyinek, illetve vizsgának.**

**Sikeres zárthelyihez, illetve vizsgához a jelen bemutató tartalmán felül a kötelező irodalomként megjelölt anyag, a gyakorlatokon szóban, illetve a táblán átadott tudnivalók ismerete, valamint a gyakorlatokon megoldott példák és az otthoni feldolgozás céljából kiadott feladatok önálló megoldásának képessége is szükséges.**

# Programozás I.

Microsoft Visual Studio 2012

Hello C# World!

Változók típusai

Gyakorló feladatok

Algoritmus alkotási feladatok

# Projektek és megoldások

- **Projekt („Project”)**

A projekt egy futtatható programhoz vagy más típusú szoftvermodulhoz tartozó, együtt kezelt szoftverelemek (többségében fájlok) összessége.

- **C# forráskód („source code”) [\* .cs]**

- **Hivatkozások („references”)**

- **Beállítások („settings”) [\* .settings]**

- **Konfigurációs fájlok („configuration”) [\* .config]**

- **Egyéb erőforrások („resources”) [\* .resx, \*.rc, \*.resources]**

A projekthez tartozó elemek mappák létrehozásával hierarchikus fastruktúrába rendezhetők.

A C# projekteket a Visual Studio \*.csproj kiterjesztésű fájlokban tárolja.

# Projektek és megoldások

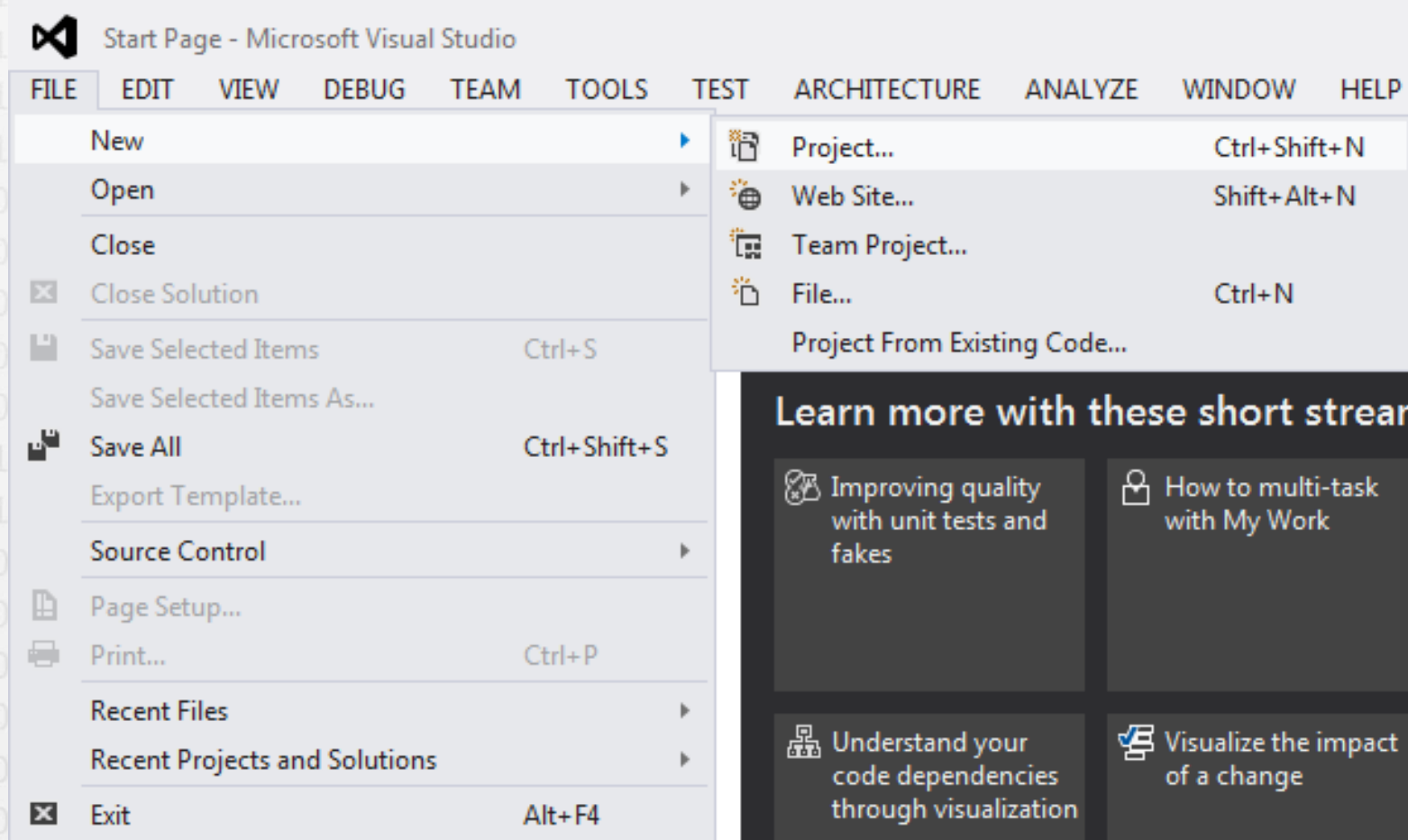
- **Megoldás („Solution”)**

**A megoldás több összefüggő projekt együttes kezelését teszi lehetővé.**

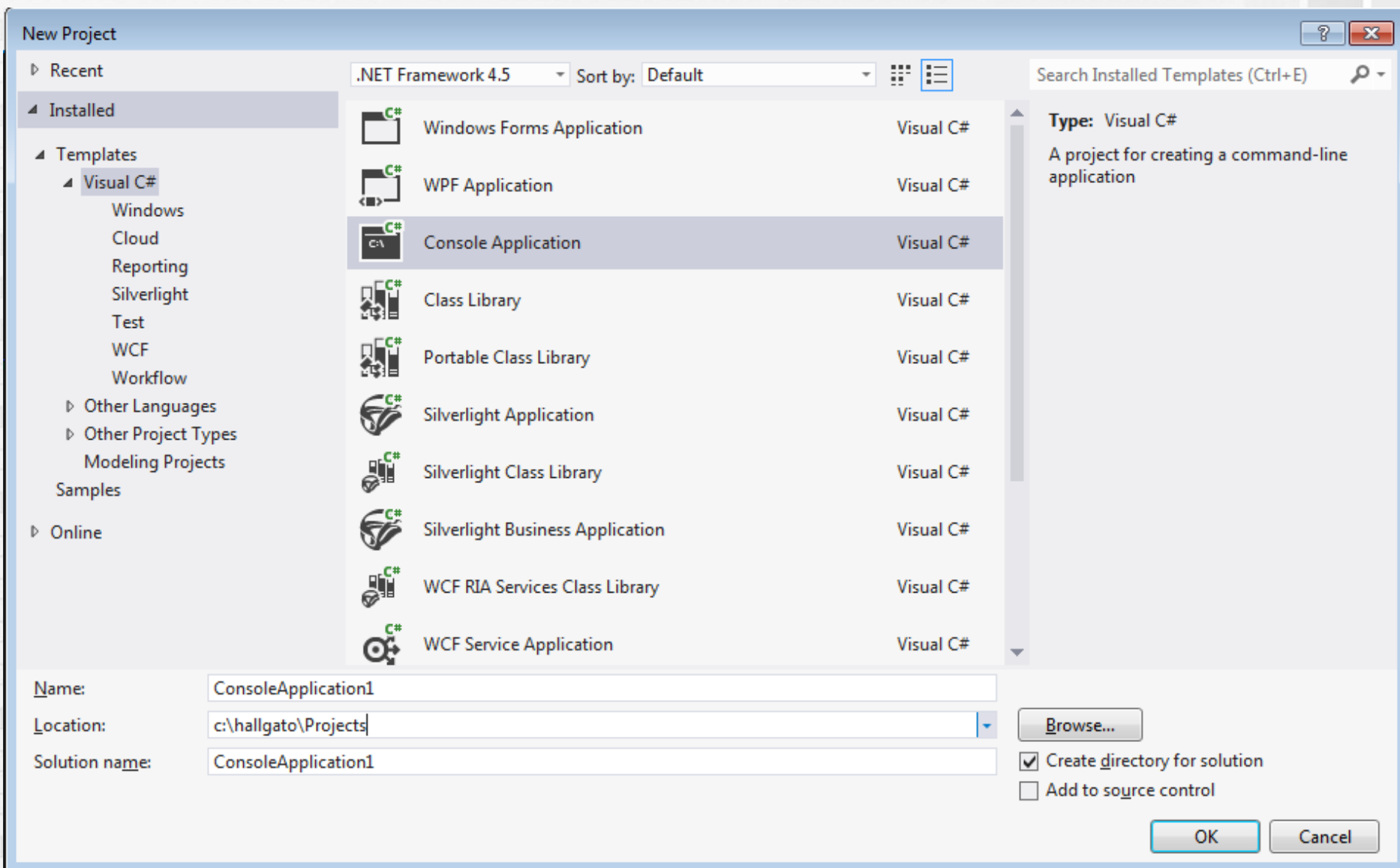
**Ezek a projektek virtuális mappák segítségével hierarchikus fastruktúrába is rendezhetők.**

**A megoldásokat a Visual Studio \*.sln kiterjesztésű fájlokban tárolja.**

# Új projekt létrehozása



# Új projekt létrehozása

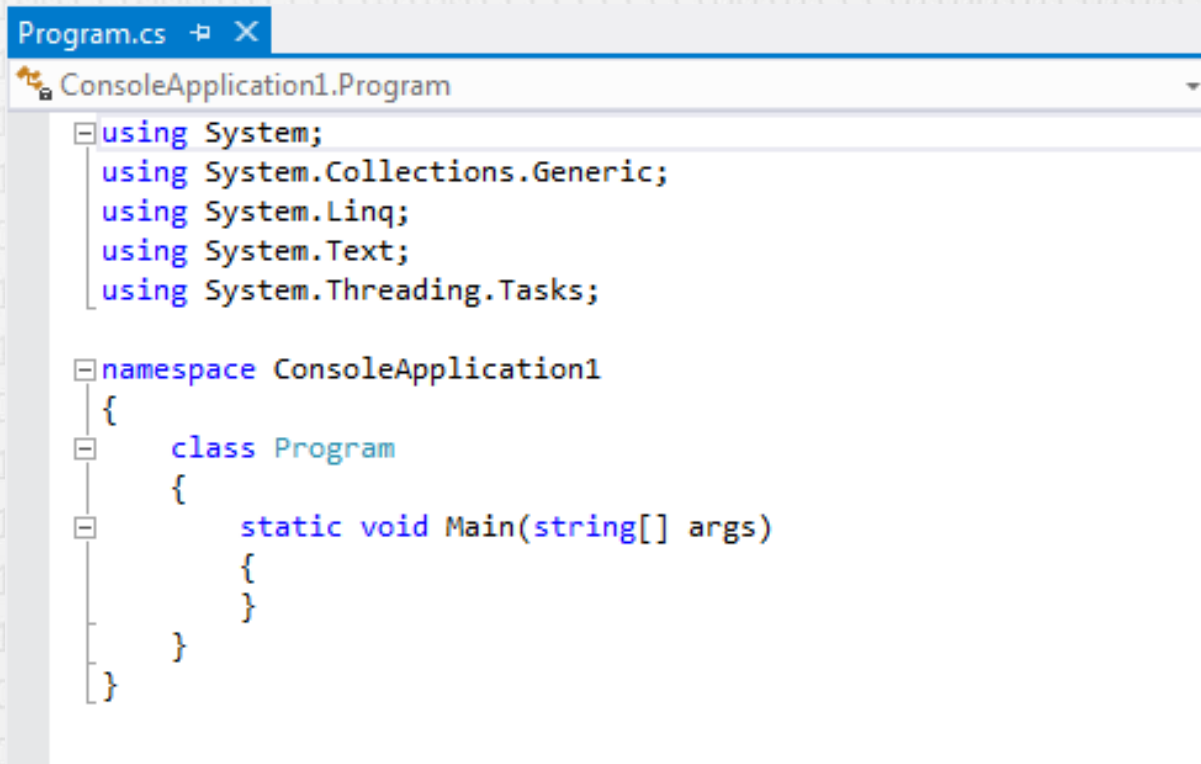


# A legfontosabb projekttypusok

- **Grafikus Windows alkalmazás („Windows Forms Application”)**  
Végeredménye egy „exe” kiterjesztésű futtatható program.
- **Parancsértelmezőben futó Windows alkalmazás („Console Application”)**  
Végeredménye egy „exe” kiterjesztésű futtatható program.
- **Osztálykönyvtár („Class Library”)**  
Végeredménye egy „dll” kiterjesztésű könyvtárfájl.
- **Üres projekt („Empty Project”)**  
Ehhez a projekttypushoz kézzel kell a megfelelő elemeket hozzáadni.



# Kód készítés



```
Program.cs  🔍  ✕  
ConsoleApplication1.Program  
  
using System;  
using System.Collections.Generic;  
using System.Linq;  
using System.Text;  
using System.Threading.Tasks;  
  
namespace ConsoleApplication1  
{  
    class Program  
    {  
        static void Main(string[] args)  
        {  
        }  
    }  
}
```

# Kód készítés

```
namespace ConsoleApplication1
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.
        }
    }
}
```

- SetWindowSize
- Title
- TreatControlCAsInput
- WindowHeight
- WindowLeft
- WindowTop
- WindowWidth
- Write
- WriteLine**

void Console.WriteLine(string format, params object[] arg) (+ 18 overload(s))  
Writes the text representation of the specified array of objects, followed by the current line terminator, to the standard output stream using the specified format information.

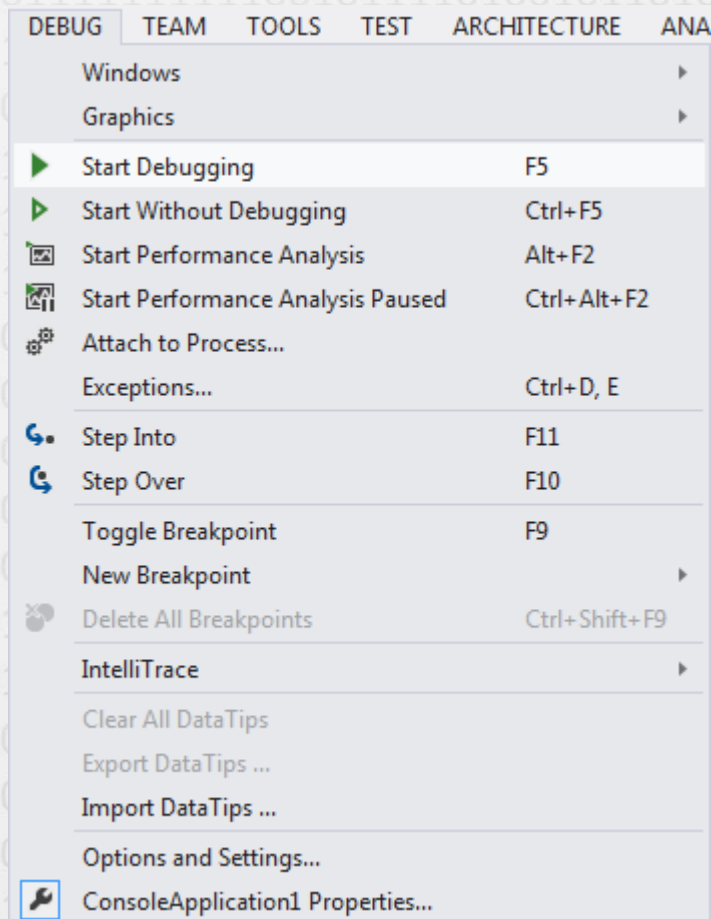
Exceptions:  
System.IO.IOException  
System.ArgumentNullException  
System.FormatException

# A program

```
namespace ConsoleApplication1
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Hello C# World!");
            Console.ReadLine();
        }
    }
}
```

# Futtatás

▶ Start ▾ Debug ▾



# Hibás program (fordítási hiba)

```
namespace ConsoleApplication1
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Hello C# World!");
            Console.ReadLine;
        }
    }
}
```



Error List				
▼ 1 Error 0 Warnings 0 Messages				
Search Error List				
	Description	File	Line	Project
✖ 1	Only assignment, call, increment, decrement, await, and new object expressions can be used as a statement	Program.cs	14	ConsoleApplication1

# Hibás program (warning)

- Nem igazi hiba, csak figyelmeztetés
- Szinte mindig érdemes hallgatni rá (a figyelmeztetés nyomán hibákra találhatunk)

```
7 namespace ConsoleApplication1
8 {
9     class Program
10    {
11        static void Main(string[] args)
12        {
13            string nev;
14            Console.WriteLine("Írd be a neved!");
15            Console.ReadLine();
16        }
17    }
18 }
```

Error List

0 Errors 1 Warning 0 Messages

Search Error List


	Description	F.	Line	Column	Project
1	The variable 'nev' is declared but never used	Prj	13	20	ConsoleApplication1

Find Symbol Results Error List

# Hibás program (futás közbeni hiba)

- Bizonyos hibákat a fordító nem tud kiszűrni a futtatás előtt

```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6
7 namespace ConsoleApplication1
8 {
9     class Program
10     {
11         static void Main(string[] args)
12         {
13             int szam1 = 10;
14             int szam2 = 0;
15             int eredmeny = szam1 / szam2;
16         }
17     }
18 }
19
```

 **DivideByZeroException was unhandled** ✕

Attempted to divide by zero.

**Troubleshooting tips:**

[Make sure the value of the denominator is not zero before performing a division operation.](#)

[Get general help for this exception.](#)

[Search for more Help Online...](#)

**Exception settings:**

☐ Break when this exception type is thrown

**Actions:**



















[View Detail...](#)

[Copy exception detail to the clipboard](#)

[Open exception settings](#)

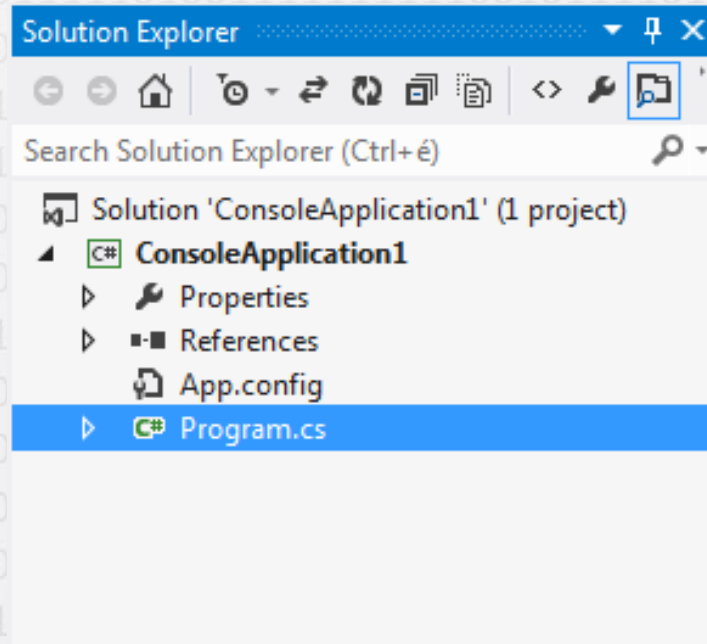


# A View menüpont

VIEW	PROJECT	BUILD	DEBUG	TEAM	TOO
<>	Code		F7		
	Solution Explorer		Ctrl+W, S		
	Team Explorer		Ctrl+Ü, Ctrl+M		
	Server Explorer		Ctrl+W, L		
	Architecture Explorer		Ctrl+Ü, Ctrl+R		
	Call Hierarchy		Ctrl+W, K		
	Class View		Ctrl+W, C		
	Code Definition Window		Ctrl+W, D		
	Object Browser		Ctrl+W, J		
	Error List		Ctrl+W, E		
	Output		Ctrl+W, O		
	Start Page				
	Task List		Ctrl+W, T		
	Toolbox		Ctrl+W, X		
	Find Results				▶
	Other Windows				▶
	Toolbars				▶
	Full Screen		Shift+Alt+Enter		
	All Windows		Shift+Alt+M		
	Navigate Backward		Ctrl+-		
	Navigate Forward		Ctrl+Shift+-		
	Next Task				
	Previous Task				
	Properties Window		Ctrl+W, P		
	Property Pages		Shift+F4		



# A Solution Explorer



# Programozás I.

Microsoft Visual Studio 2012

Hello C# World!

Változók típusai

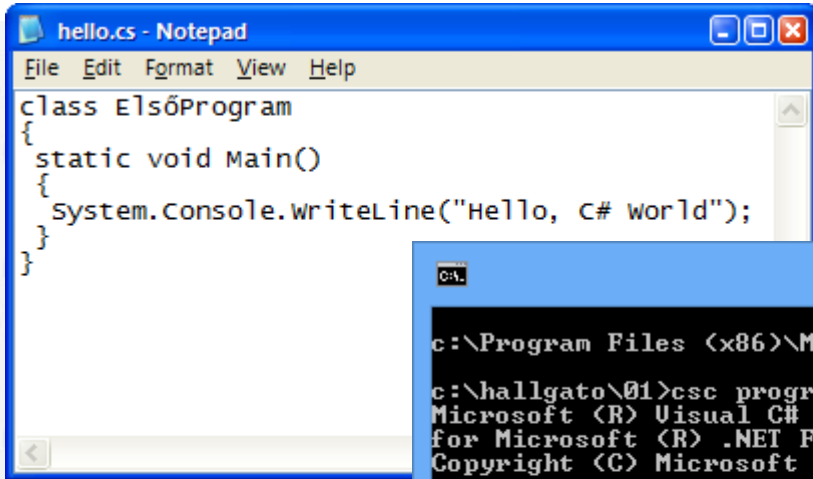
Gyakorló feladatok

Algoritmus alkotási feladatok

# Hello, C# World

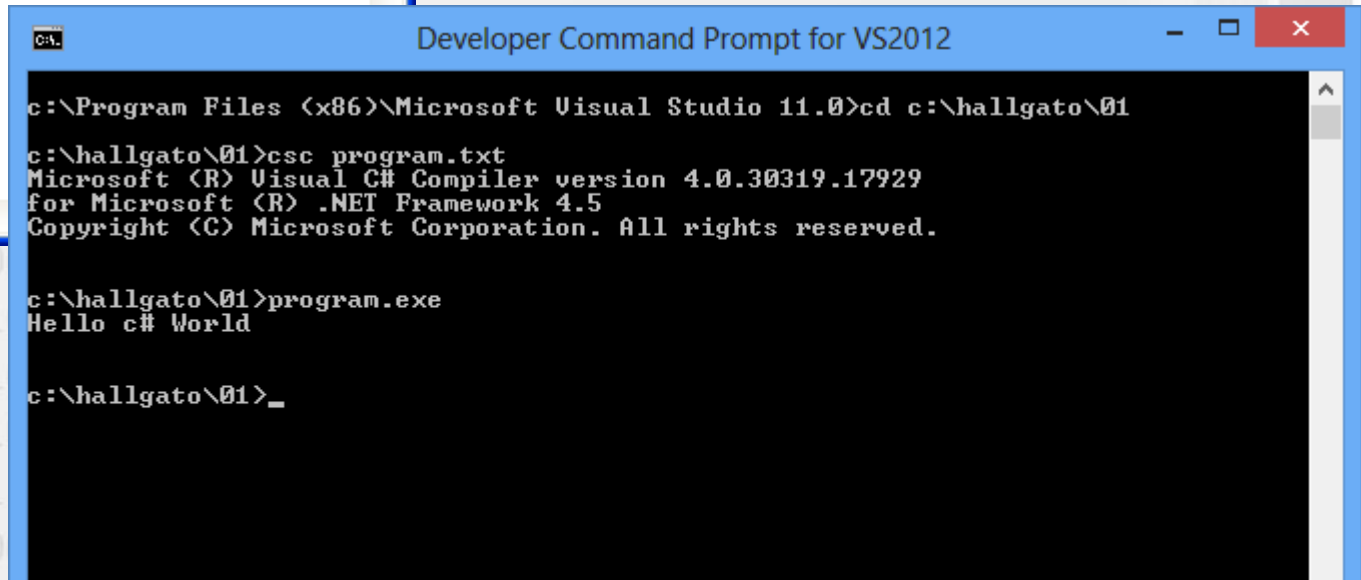
```
// Első programunk C# nyelven
class ElsőProgram
{
    static void Main()
    {
        Console.WriteLine("Hello, C# World");
    }
}
```

hello.cs



A screenshot of a Notepad window titled "hello.cs - Notepad". The window contains the following C# code:

```
class ElsőProgram
{
    static void Main()
    {
        System.Console.WriteLine("Hello, C# world");
    }
}
```



A screenshot of a Developer Command Prompt for VS2012. The prompt shows the following commands and output:

```
c:\Program Files (x86)\Microsoft Visual Studio 11.0>cd c:\hallgato\01
c:\hallgato\01>csc program.txt
Microsoft (R) Visual C# Compiler version 4.0.30319.17929
for Microsoft (R) .NET Framework 4.5
Copyright (C) Microsoft Corporation. All rights reserved.

c:\hallgato\01>program.exe
Hello c# World

c:\hallgato\01>_
```

# Hello, C# World

Készítsünk programot, amely kiírja a konzolra a „Szervusz, hallgató!” szöveget!

```
class Program
{
    static void Main()
    {
        Console.WriteLine("Szervusz, hallgató!");
        Console.ReadLine();
    }
}
```

# Programozás I.

Microsoft Visual Studio 2012

Hello C# World!

Változók típusai

Gyakorló feladatok

Algoritmus alkotási feladatok

# Egész (fixpontos) számok

Név	Leírás	Értéktartomány
<b>sbyte</b>	<b>8 bites előjeles egész</b>	<b>-128 : 127</b>
<b>byte</b>	<b>8 bites előjel nélküli egész</b>	<b>0 : 255</b>
<b>short</b>	<b>16 bites előjeles egész</b>	<b>-32 768 : 32 767</b>
<b>ushort</b>	<b>16 bites előjel nélküli egész</b>	<b>0 : 65535</b>
<b>int</b>	<b>32 bites előjeles egész</b>	<b>-2 147 483 648 : 2 147 483 647</b>
<b>uint</b>	<b>32 bites előjel nélküli egész</b>	<b>0 : 4 294 967 295</b>
<b>long</b>	<b>64 bites előjeles egész</b>	<b>-9 223 372 036 854 775 808 : 9 223 372 036 854 775 807</b>
<b>ulong</b>	<b>64 bites előjel nélküli egész</b>	<b>0 : 18 446 744 073 709 551 615</b>

# Valós (lebegőpontos) számok

Név	Leírás	Értékes jegy	Értéktartomány
<b>float</b>	<b>32 bites lebegőpontos</b>	<b>7</b>	$\pm 1,5 \cdot 10^{-45} :$ $\pm 3,4 \cdot 10^{38}$
<b>double</b>	<b>64 bites lebegőpontos</b>	<b>15</b>	$\pm 5,0 \cdot 10^{-324} :$ $\pm 1,7 \cdot 10^{308}$
<b>decimal</b>	<b>128 bites nagy pontosságú</b>	<b>28</b>	$\pm 1,0 \cdot 10^{-28} :$ $\pm 7,9 \cdot 10^{28}$

	Méret	Előjel	Kitevő	Törtrész	Eltolás
<b>Egyszeres IEEE-754 szabvány pontosság</b>	<b>32 bit</b>	<b>1 bit</b>	<b>8 bit</b>	<b>23 bit</b>	<b>127</b>
<b>Kétszeres pontosság</b>	<b>64 bit</b>	<b>1 bit</b>	<b>11 bit</b>	<b>52 bit</b>	<b>1023</b>



# Karakterek, karakterláncok

- Karakter: char (megadás: aposztróffal)
  - **char** karakter='ű';
- Karakterlánc: string (megadás: idézőjellel)
  - **string** karakterlanc="Árvíztűrő Tükörfúrógép";
- Speciális karakterek is megadhatóak (@ jellel kikapcsolható):

Jelölés	Karakter
\0	Null karakter
\a	Sípszó
\b	Visszatörlés
\f	Lapdobás
\n	Soremelés
\r	Kocsi vissza
V 1.0 \t	Vízszintes tabulátor

Jelölés	Karakter
\v	Függőleges tabulátor
\x....	Hexadecimális kód
\u....	Unicode karakter
\U....	Unicode karakter
\'	Aposztróf
\"	Idézőjel
\\	Backslash



# Logikai típus

Név	Leírás	Értéktartomány
<b>bool</b>	<b>Logikai adattípus</b>	<b>true vagy false (igaz vagy hamis)</b>

- Teljesítmény-okokból általában nem 1 biten ábrázoljuk, részletesebben lásd IEA
- Logikai műveletek:

A	B	A AND B	A OR B	A XOR B	NOT(A)
0	0	0	0	0	1
0	1	0	1	1	1
1	0	0	1	1	0
1	1	1	1	0	0

# Változók deklarálása és használata

```
int j = -10;
```

```
int x = 10, y = 20;
```

```
double pi = 3.14159;
```

```
const int száz = 100;
```

```
char d = 'x';
```

```
char UnicodePélda = '\u0170'; // "Ű" karakter
```

```
string jegy = "jeles";
```

```
string ElérésiÚt = "C:\\Program Files\\";
```

```
string ElérésiÚt2 = @"C:\Program Files\";
```

```
string vers = @"Hová merült el  
szép szemed világa";
```

```
bool igaz = true;
```

Fontos szabály: azonos  
névvel egy változót nem  
lehet kétszer deklarálni!

# Típuskonverziók

- A számtípusok közötti konverzió mikéntje attól függ, hogy történik –e értékvesztés a konverzió során
- Egyszerű értékadás használható, amennyiben biztos, hogy nincs értékvesztés:

```
byte a=5;      long c=5;      float f=3.2f;  
int b=a;      float d=c;      double g=f;
```

- Amennyiben értékvesztés történhet, akkor mindenképp jelezni kell a konverziót, ez az ún. típuskényszerítés, „kasztolás” (typecasting):

```
int a=999;      double d=3.14;      int i1=-1;  
byte b=(byte)a;  int c=(int)d;      uint i2=(uint)i1;
```

# Típuskonverziók

- A stringgé történő konverzió a C# nyelven MINDEN változónál ugyanúgy történik:

**byte** b=250;

**float** f=3.14f;

**string** s1=b.ToString();

**string** s2=f.ToString();

- Stringből számmá tudunk konvertálni:

**string** s="123";

**string** s2="123,456";

**byte** b=**byte**.Parse(s);

**float** f=**float**.Parse(s2);

- Typecasting esetén (ebben a félévben számok között):  
célváltozó = (céltípus)forrásváltozó;
- Stringgé konvertálásnál:  
célváltozó = forrásváltozó.ToString();
- Stringből konvertálásnál:  
célváltozó=céltípus.Parse(stringváltozó);

# Programozás I.

Microsoft Visual Studio 2012

Hello C# World!

Változók típusai

Gyakorló feladatok

Algoritmus alkotási feladatok

# Hello, C# World

Készítsünk programot, amely a konzolról beolvas egy nevet, majd név szerint üdvözli az illetőt!

```
class Program
{
    static void Main()
    {
        Console.Write("Hogy hívnak? ");
        string név = Console.ReadLine();
        Console.WriteLine("Szervusz, " + név + "!");
    }
}
```

# Hello, C# World

Készítsünk programot, amely a konzolról bekéri a felhasználó nevét és életkorát, majd üdvözli őt!

```
class Program
{
    static void Main()
    {
        Console.Write("Hogy hívnak? ");
        string név = Console.ReadLine();
        Console.Write("Hány éves vagy? ");
        int életkor = int.Parse(Console.ReadLine());
        Console.WriteLine("Szervusz, " + életkor +
            " éves " + név + "!");
    }
}
```



# Gyakorló feladat

**Írjon programot, amely bekéri egy téglalap két szomszédos oldalának hosszát, majd kiszámolja és kiírja a képernyőre a téglalap kerületét és területét!**



# Gyakorló feladat

**Írjon programot, amely bekér egy hőmérséklet értéket Celsius fokban, majd konvertálja azt Fahrenheit fokra.**

**A Celsius és Fahrenheit skála közötti áttéréshez használja a következő összefüggést:**

$$[^{\circ}\text{F}] = [^{\circ}\text{C}] * 9/5 + 32$$

# Gyakorló feladat

Hány utasítás van az alábbi kódban?

(A megoldáshoz ne használjon Visual Studio fejlesztői környezetet!)

```
class Program
{
    static void Main(string[] args)
    {
        ;
    }
}
```

# Gyakorló feladat

Lefordítható-e az alábbi program?

(A megoldáshoz ne használjon Visual Studio fejlesztői környezetet!)

```
using System;

namespace Feladat
{
    class Program
    {
        static void Main(string[] args)
        {
            ;;;;Console.WriteLine(); ;;;
        }
    }
}
```

# Gyakorló feladat

**Milyen típusok kerülhetnek a pontozott helyekre?**  
**(A megoldáshoz ne használjon Visual Studio fejlesztői környezetet!)**

```
using System;

namespace Feladat
{
    class Program
    {
        static void Main(string[] args)
        {
            ..... gyerekekSzama = 123; //tudjuk, hogy a gyerekek száma
10 és 200 közötti szám lehet
            ..... méretMiliméterben = 1000; // tudjuk, hogy az emberek
mérete akár 2,2 méter is lehet
            ..... folyóSzámlaEgyenleg = -144; // tudjuk, hogy a
folyószámlán lévő pénzösszeg lehet negatív is, utalva tartozásra
        }
    }
}
```

# Gyakorló feladat

Mit látunk a konzolon, ha futtatjuk az alábbi programot?  
(A megoldáshoz ne használjon Visual Studio fejlesztői környezetet!)

```
using System;

namespace Feladat
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("\\\\t\\t");
        }
    }
}
```

# Programozás I.

Microsoft Visual Studio 2012

Hello C# World!

Változók típusai

Gyakorló feladatok

Algoritmus alkotási feladatok

# Algoritmusok gyakorlása

1. Írja le a másodfokú egyenlet megoldási algoritmusát folyamatábrával, struktogrammal és pszeudokóddal!
2. Adott két síkbeli pont:  $P_1(x_1, y_1)$  és  $P_2(x_2, y_2)$ . Keressük a két adott ponton áthaladó egyenesen az  $x_0$  abszcisszájú pont  $y_0$  koordinátáját. Adjon algoritmust a feladat megoldására!
3. Készítsen algoritmust, mely eldönti, hogy egy adott év szökőév-e vagy sem!
4. Készítsen algoritmust, mely megadja, hogy egy adott év adott hónapja hány napból áll!
5. Készítsen algoritmust, mely egy pozitív egész számról eldönti, hogy prím-e vagy sem!
6. Készítsen algoritmust, mely bekéri egy tankör hallgatóinak zh eredményeit és kiszámítja azok átlagát!