Programmazione a oggetti

Alessandro Andreose'
alessandro.andreose@gmail.com

Documentazione (I)

- Online

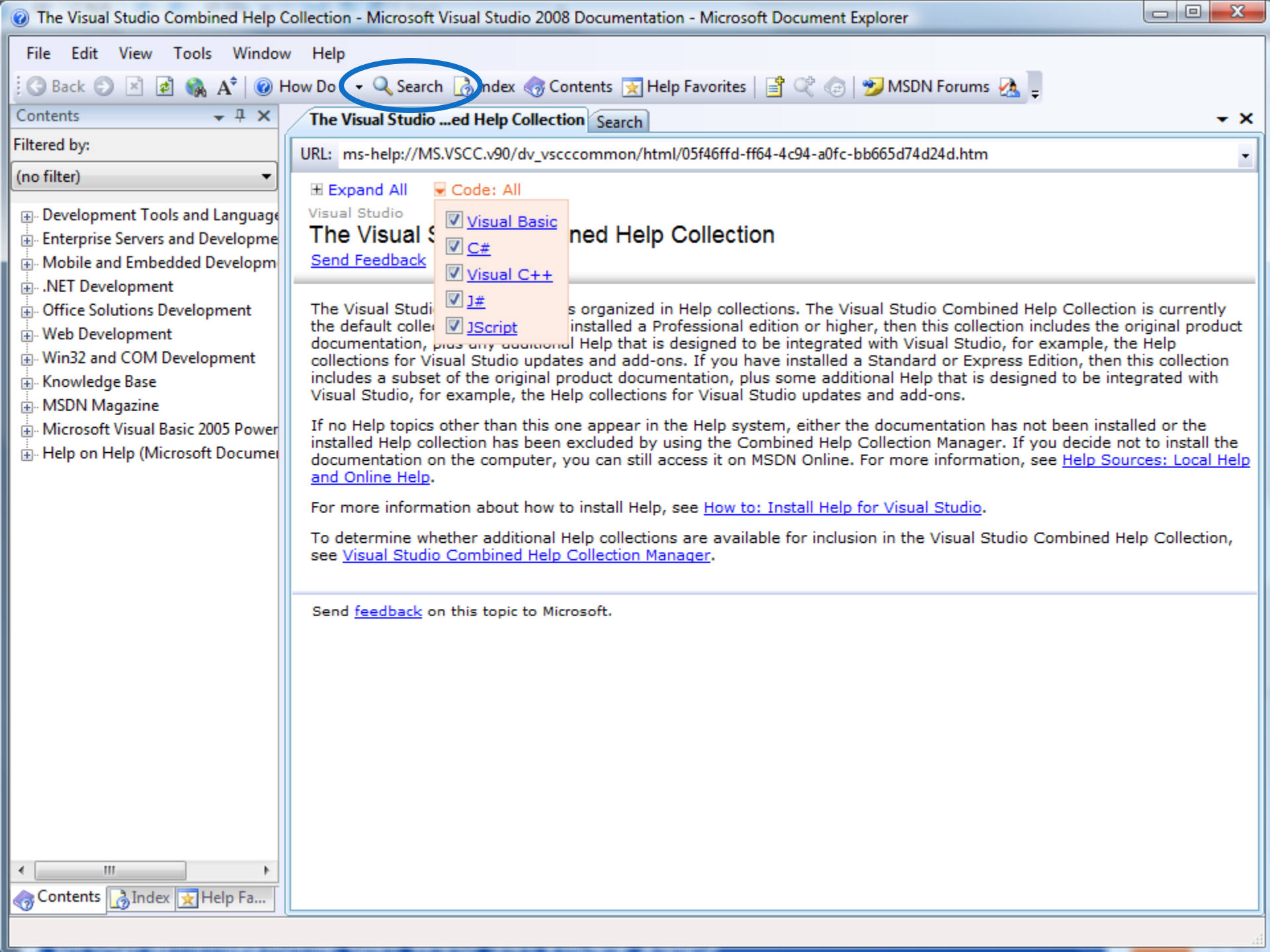
  http://msdn2.microsoft.com/en-us/default.aspx

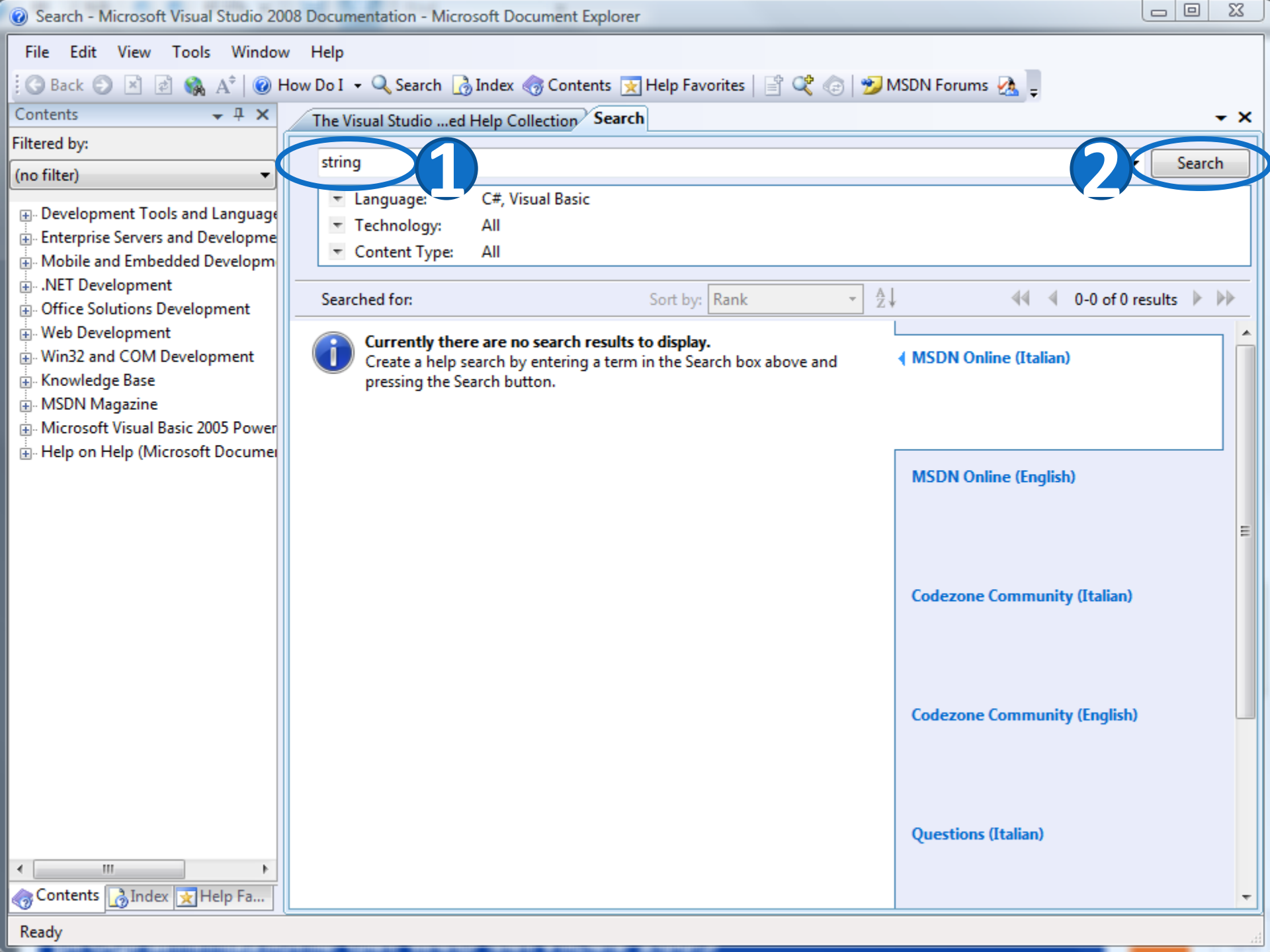
- Offline
  - Download disponibile da questo indirizzo

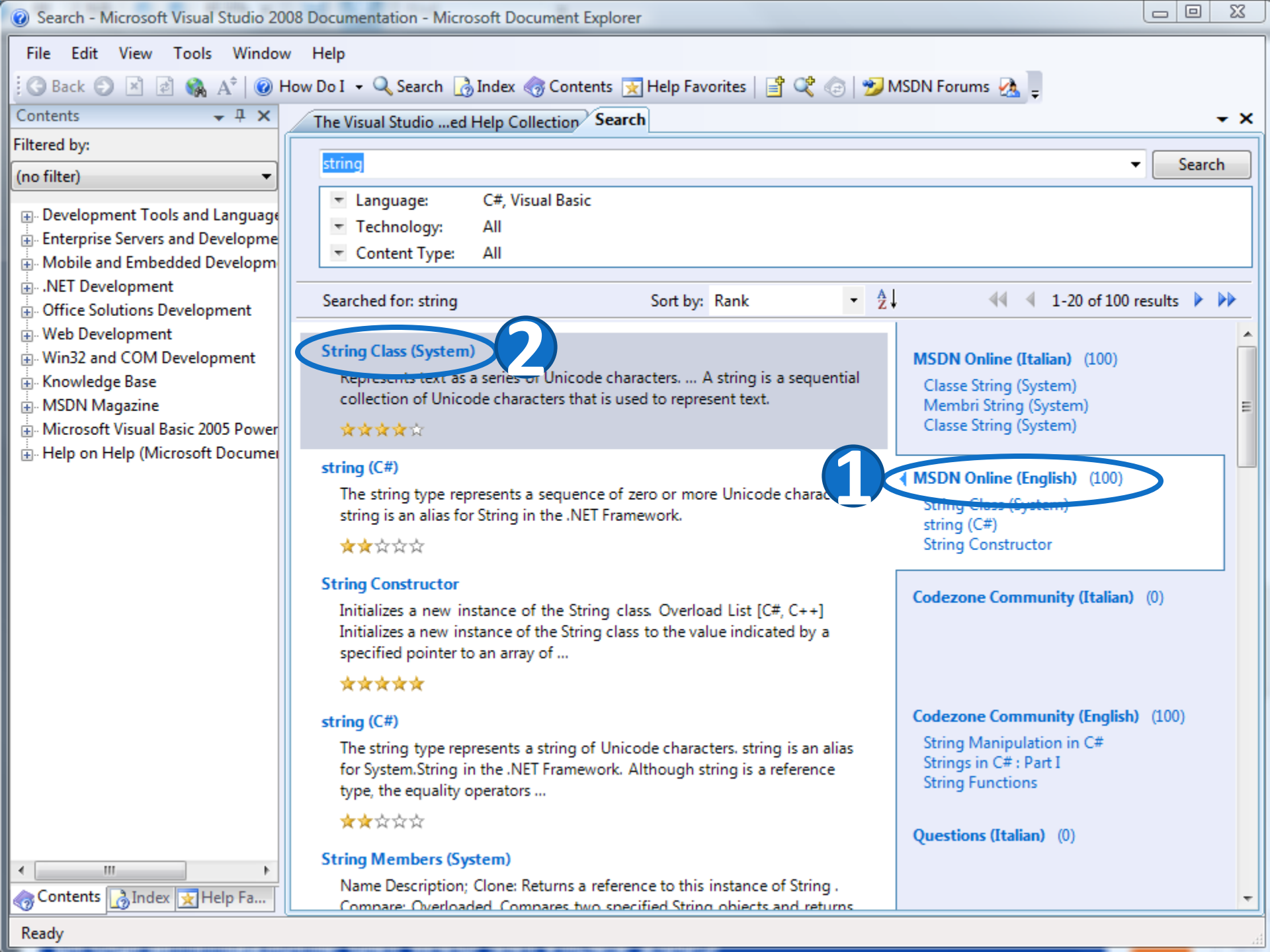
    http://www.microsoft.com/downloads/details.aspx?familyid=6FF3BC60-32C8-4C22-8591-A20BF8DFF1A2&displaylang=en

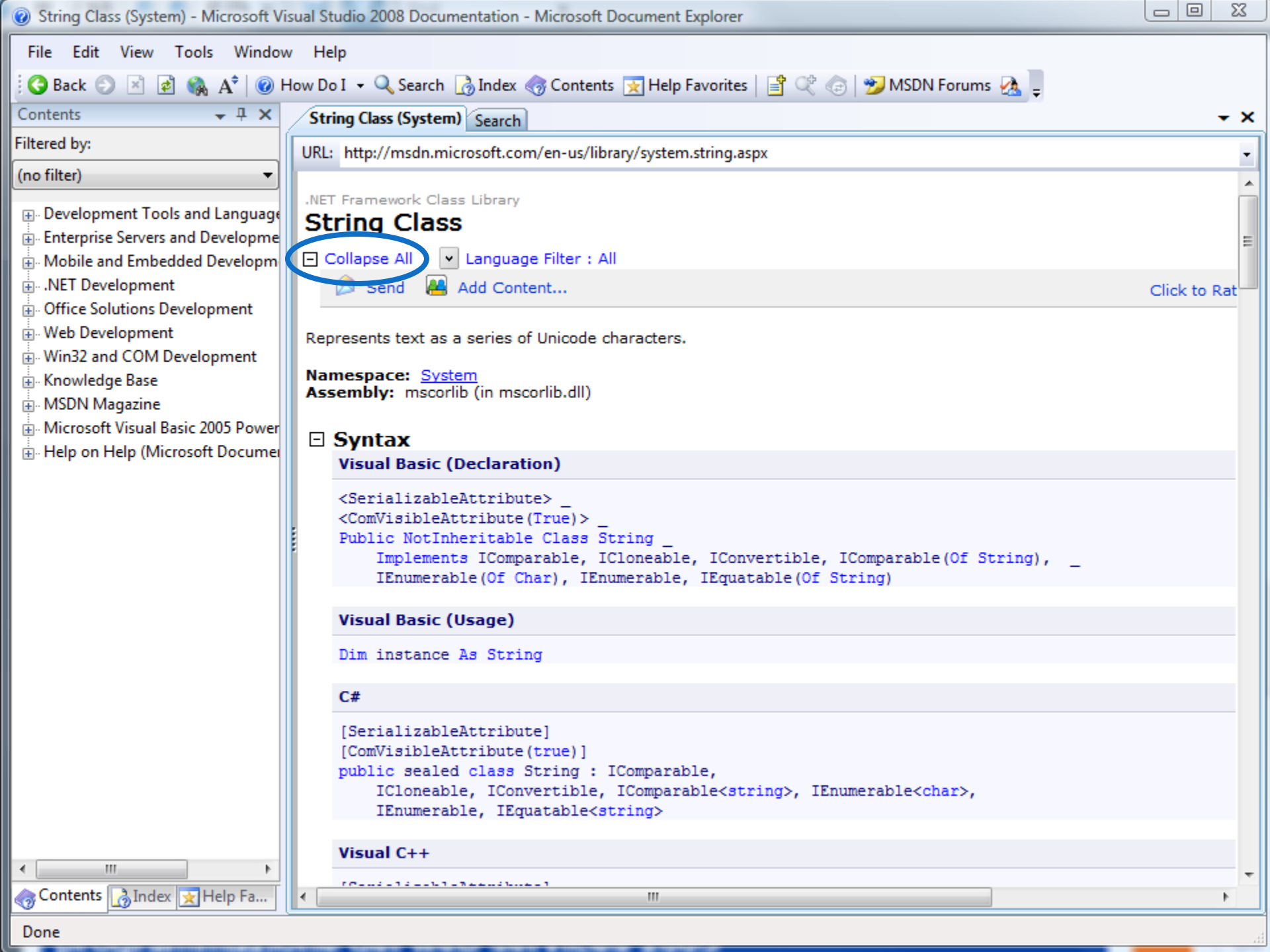
Documentazione (II)

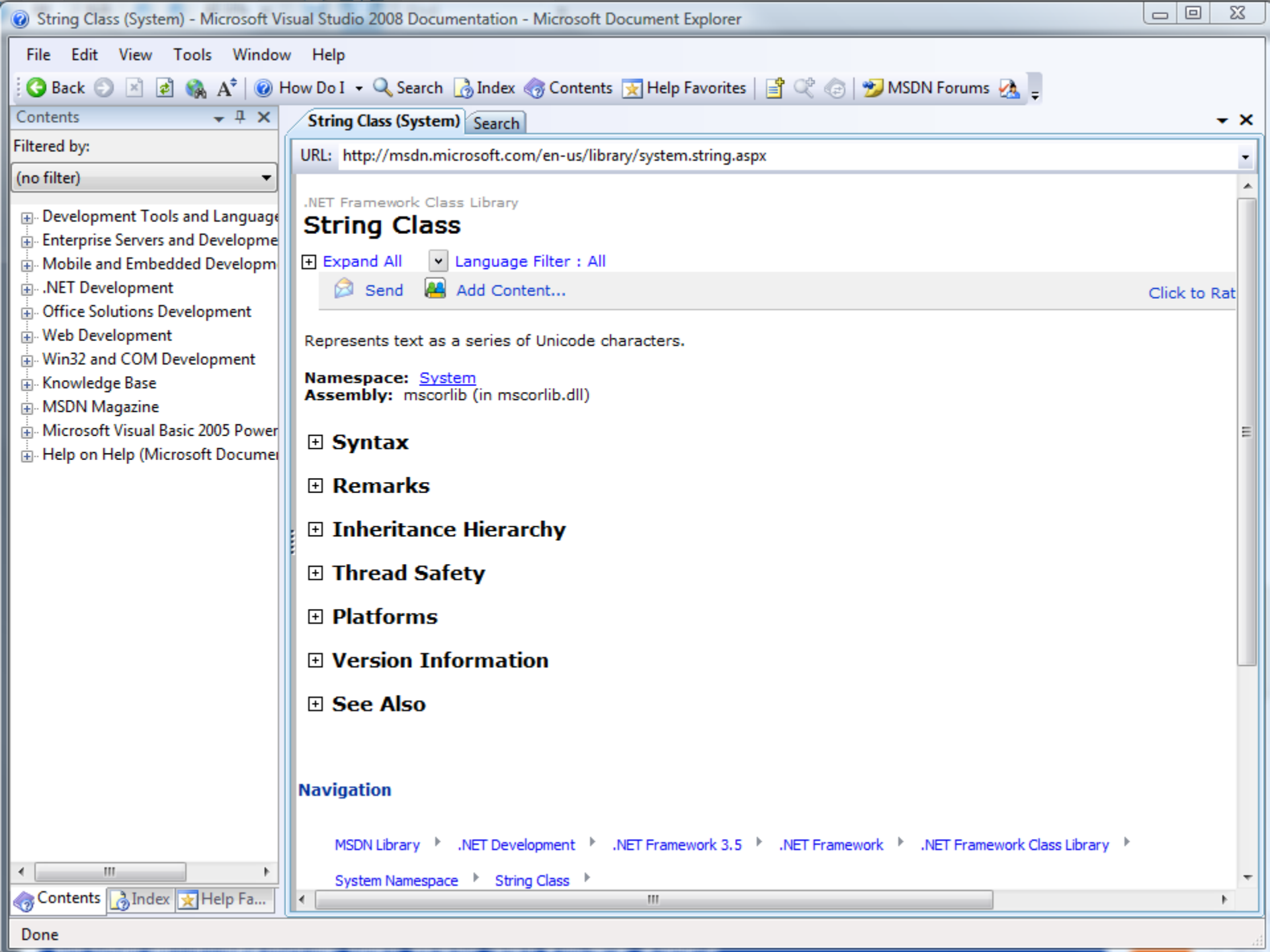
- È indispensabile
- Contiene
  - La descrizione di tutte le classi, di tutti i metodi, …
  - Molti esempi di codice
  - Descrizione del linguaggio, delle keyword, …
- Microsoft Document Explorer raccoglie informazioni
  - MSDN locale
  - MSDN online (inglese e italiano)
  - Community online (inglese e italiano)
  - Domande online (inglese e italiano)


The Visual Studio Combined Help Collection - Microsoft Visual Studio 2008 Documentation - Microsoft Document Explorer
File Edit View Tools Window Help
Back How Do Search Index Contents Help Favorites MSDN Forums
Contents
Filtered by:
(no filter)
Development Tools and Language
Enterprise Servers and Developme
Mobile and Embedded Developm
.NET Development
Office Solutions Development
Web Development
Win32 and COM Development
Knowledge Base
MSDN Magazine
Microsoft Visual Basic 2005 Power
Help on Help (Microsoft Docume
Contents Index Help Fa...
The Visual Studio ...ed Help Collection
Search
URL: ms-help://MS.VSCC.v90/dv_vscccommon/html/05f46ffd-ff64-4c94-a0fc-bb665d74d24d.htm
Expand All
Code: All
Visual Basic
C#
Visual C++
J#
JScript
Visual Studio
The Visual S ned Help Collection
Send Feedback
The Visual Studi s organized in Help collections. The Visual Studio Combined Help Collection is currently the default colle installed a Professional edition or higher, then this collection includes the original product documentation, plus any additional Help that is designed to be integrated with Visual Studio, for example, the Help collections for Visual Studio updates and add-ons. If you have installed a Standard or Express Edition, then this collection includes a subset of the original product documentation, plus some additional Help that is designed to be integrated with Visual Studio, for example, the Help collections for Visual Studio updates and add-ons.
If no Help topics other than this one appear in the Help system, either the documentation has not been installed or the installed Help collection has been excluded by using the Combined Help Collection Manager. If you decide not to install the documentation on the computer, you can still access it on MSDN Online. For more information, see Help Sources: Local Help and Online Help.
For more information about how to install Help, see How to: Install Help for Visual Studio.
To determine whether additional Help collections are available for inclusion in the Visual Studio Combined Help Collection, see Visual Studio Combined Help Collection Manager.
Send feedback on this topic to Microsoft.


Search - Microsoft Visual Studio 2008 Documentation - Microsoft Document Explorer
File Edit View Tools Window Help
Back How Do I Search Index Contents Help Favorites MSDN Forums
Contents
Filtered by:
(no filter)
Development Tools and Language
Enterprise Servers and Developme
Mobile and Embedded Developm
.NET Development
Office Solutions Development
Web Development
Win32 and COM Development
Knowledge Base
MSDN Magazine
Microsoft Visual Basic 2005 Power
Help on Help (Microsoft Docume
Contents Index Help Fa...
The Visual Studio ...ed Help Collection
Search
string
1
2
Search
Language: C#, Visual Basic
Technology: All
Content Type: All
Searched for:
Sort by: Rank
0-0 of 0 results
Currently there are no search results to display.
Create a help search by entering a term in the Search box above and pressing the Search button.
MSDN Online (Italian)
MSDN Online (English)
Codezone Community (Italian)
Codezone Community (English)
Questions (Italian)
Ready


Search - Microsoft Visual Studio 2008 Documentation - Microsoft Document Explorer
File Edit View Tools Window Help
Back How Do I Search Index Contents Help Favorites MSDN Forums
Contents
Filtered by:
(no filter)
Development Tools and Language
Enterprise Servers and Developme
Mobile and Embedded Developm
.NET Development
Office Solutions Development
Web Development
Win32 and COM Development
Knowledge Base
MSDN Magazine
Microsoft Visual Basic 2005 Power
Help on Help (Microsoft Docume
Contents Index Help Fa...
The Visual Studio ...ed Help Collection
Search
string
Search
Language: C#, Visual Basic
Technology: All
Content Type: All
Searched for: string
Sort by: Rank
1-20 of 100 results
String Class (System)
Represents text as a series of Unicode characters. ... A string is a sequential collection of Unicode characters that is used to represent text.
string (C#)
The string type represents a sequence of zero or more Unicode charac
string is an alias for String in the .NET Framework.
String Constructor
Initializes a new instance of the String class. Overload List [C#, C++] Initializes a new instance of the String class to the value indicated by a specified pointer to an array of ...
string (C#)
The string type represents a string of Unicode characters. string is an alias for System.String in the .NET Framework. Although string is a reference type, the equality operators ...
String Members (System)
Name Description; Clone: Returns a reference to this instance of String . Compare: Overloaded. Compares two specified String objects and returns
MSDN Online (Italian) (100)
Classe String (System)
Membri String (System)
Classe String (System)
MSDN Online (English) (100)
String Class (System)
string (C#)
String Constructor
Codezone Community (Italian) (0)
Codezone Community (English) (100)
String Manipulation in C#
Strings in C# : Part I
String Functions
Questions (Italian) (0)
1
2
Ready


String Class (System) - Microsoft Visual Studio 2008 Documentation - Microsoft Document Explorer
File Edit View Tools Window Help
Back How Do I Search Index Contents Help Favorites MSDN Forums
Contents
Filtered by:
(no filter)
Development Tools and Language
Enterprise Servers and Developme
Mobile and Embedded Developm
.NET Development
Office Solutions Development
Web Development
Win32 and COM Development
Knowledge Base
MSDN Magazine
Microsoft Visual Basic 2005 Power
Help on Help (Microsoft Documer
Contents Index Help Fa...
String Class (System) Search
URL: http://msdn.microsoft.com/en-us/library/system.string.aspx
.NET Framework Class Library
String Class
Collapse All Language Filter : All
Send Add Content... Click to Rat
Represents text as a series of Unicode characters.
Namespace: System
Assembly: mscorlib (in mscorlib.dll)
Syntax
Visual Basic (Declaration)
<SerializableAttribute> _
<ComVisibleAttribute(True)> _
Public NotInheritable Class String _
Implements IComparable, ICloneable, IConvertible, IComparable(Of String), _
IEnumerable(Of Char), IEnumerable, IEquatable(Of String)
Visual Basic (Usage)
Dim instance As String
C#
[SerializableAttribute]
[ComVisibleAttribute(true)]
public sealed class String : IComparable,
ICloneable, IConvertible, IComparable<string>, IEnumerable<char>,
IEnumerable, IEquatable<string>
Visual C++
Done


String Class (System) - Microsoft Visual Studio 2008 Documentation - Microsoft Document Explorer
File Edit View Tools Window Help
Back How Do I Search Index Contents Help Favorites MSDN Forums
Contents
Filtered by:
(no filter)
Development Tools and Language
Enterprise Servers and Developme
Mobile and Embedded Developm
.NET Development
Office Solutions Development
Web Development
Win32 and COM Development
Knowledge Base
MSDN Magazine
Microsoft Visual Basic 2005 Power
Help on Help (Microsoft Docume
Contents Index Help Fa...
String Class (System) Search
URL: http://msdn.microsoft.com/en-us/library/system.string.aspx
.NET Framework Class Library
String Class
Expand All Language Filter : All
Send Add Content... Click to Rat
Represents text as a series of Unicode characters.
Namespace: System
Assembly: mscorlib (in mscorlib.dll)
Syntax
Remarks
Inheritance Hierarchy
Thread Safety
Platforms
Version Information
See Also
Navigation
MSDN Library .NET Development .NET Framework 3.5 .NET Framework .NET Framework Class Library
System Namespace String Class
Done

String Class (System) - Microsoft Visual Studio 2008 Documentation - Microsoft Document Explorer

File Edit View Tools Window Help

Back How Do I Search Index Contents Help Favorites MSDN Forums

Contents

Filtered by:

(no filter)

- Development Tools and Language
- Enterprise Servers and Developme
- Mobile and Embedded Developm
- .NET Development
- Office Solutions Development
- Web Development
- Win32 and COM Development
- Knowledge Base
- MSDN Magazine
- Microsoft Visual Basic 2005 Power
- Help on Help (Microsoft Docume

Contents Index Help Fa...

String Class (System) Search

URL: http://msdn.microsoft.com/en-us/library/system.string.aspx

CE, Windows Mobile for Smartphone, Windows Mobile for Pocket PC, Xbox 360

The .NET Framework and .NET Compact Framework do not support all versions of every platform. For a list of the supported versions, see .NET Framework System Requirements.

Version Information

.NET Framework

Supported in: 3.5, 3.0 SP1, 3.0, 2.0 SP1, 2.0, 1.1, 1.0

.NET Compact Framework

Supported in: 3.5, 2.0, 1.0

XNA Framework

Supported in: 1.0

See Also

Concepts

Formatting Overview

Reference

String Members

System Namespace

IComparable

ICloneable

IConvertible

IEnumerable

System.Text.StringBuilder

CultureInfo

Done

String Members (System) - Microsoft Visual Studio 2008 Documentation - Microsoft Document Explorer

File Edit View Tools Window Help

Back How Do I Search Index Contents Help Favorites MSDN Forums

Contents

Filtered by:

(no filter)

- Development Tools and Language
- Enterprise Servers and Developme
- Mobile and Embedded Developm
- .NET Development
- Office Solutions Development
- Web Development
- Win32 and COM Development
- Knowledge Base
- MSDN Magazine
- Microsoft Visual Basic 2005 Power
- Help on Help (Microsoft Documer

Contents Index Help Fa...

String Members (System) Search

URL: http://msdn.microsoft.com/en-us/library/system.string_members.aspx

.NET Framework Class Library

String Members

Collapse All Language Filter : All

Send Add Content... Click to Rat

Represents text as a series of Unicode characters.

The String type exposes the following members.

Constructors

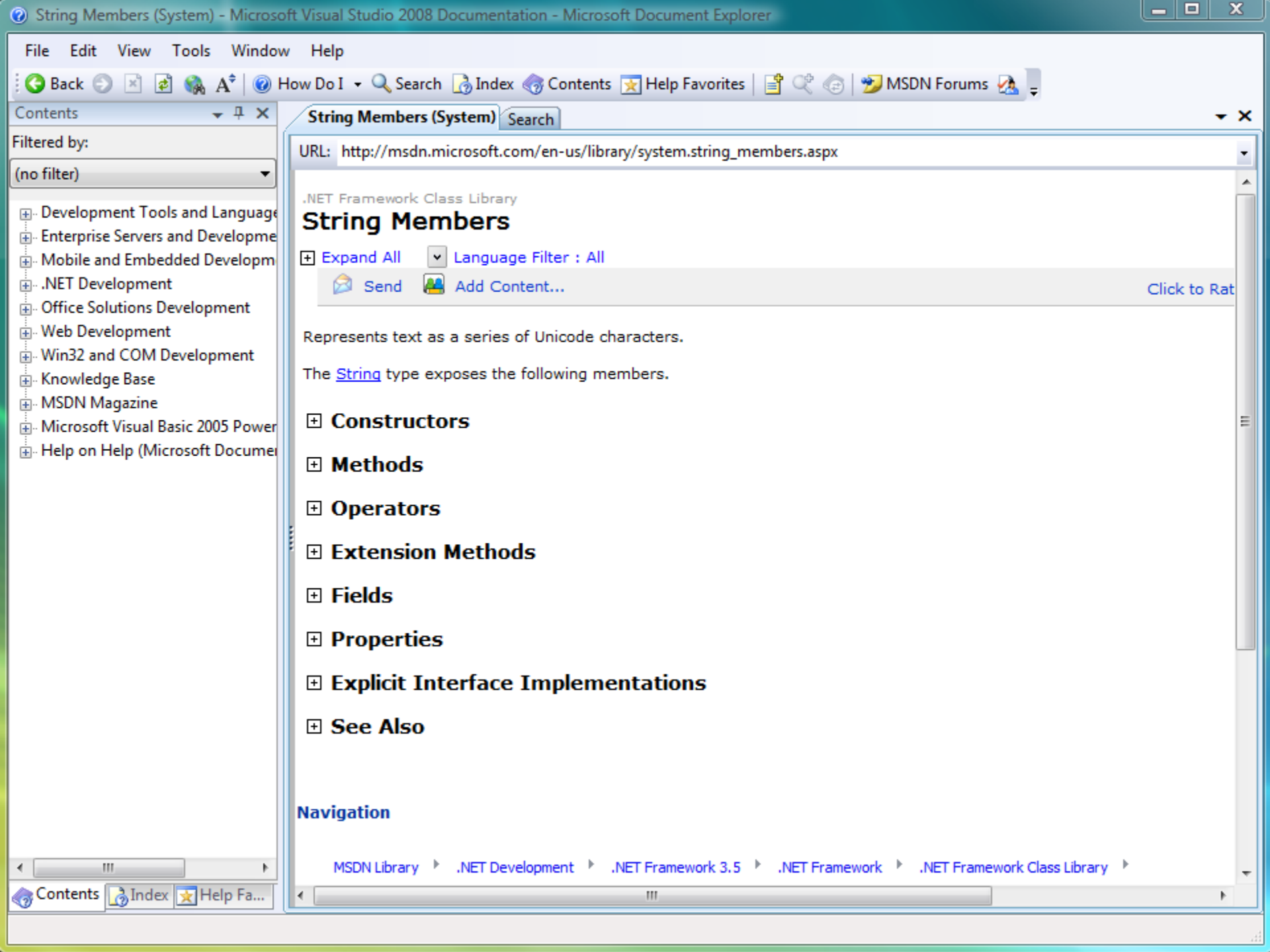
	Name	Description
	String	Overloaded. Initializes a new instance of the String class.

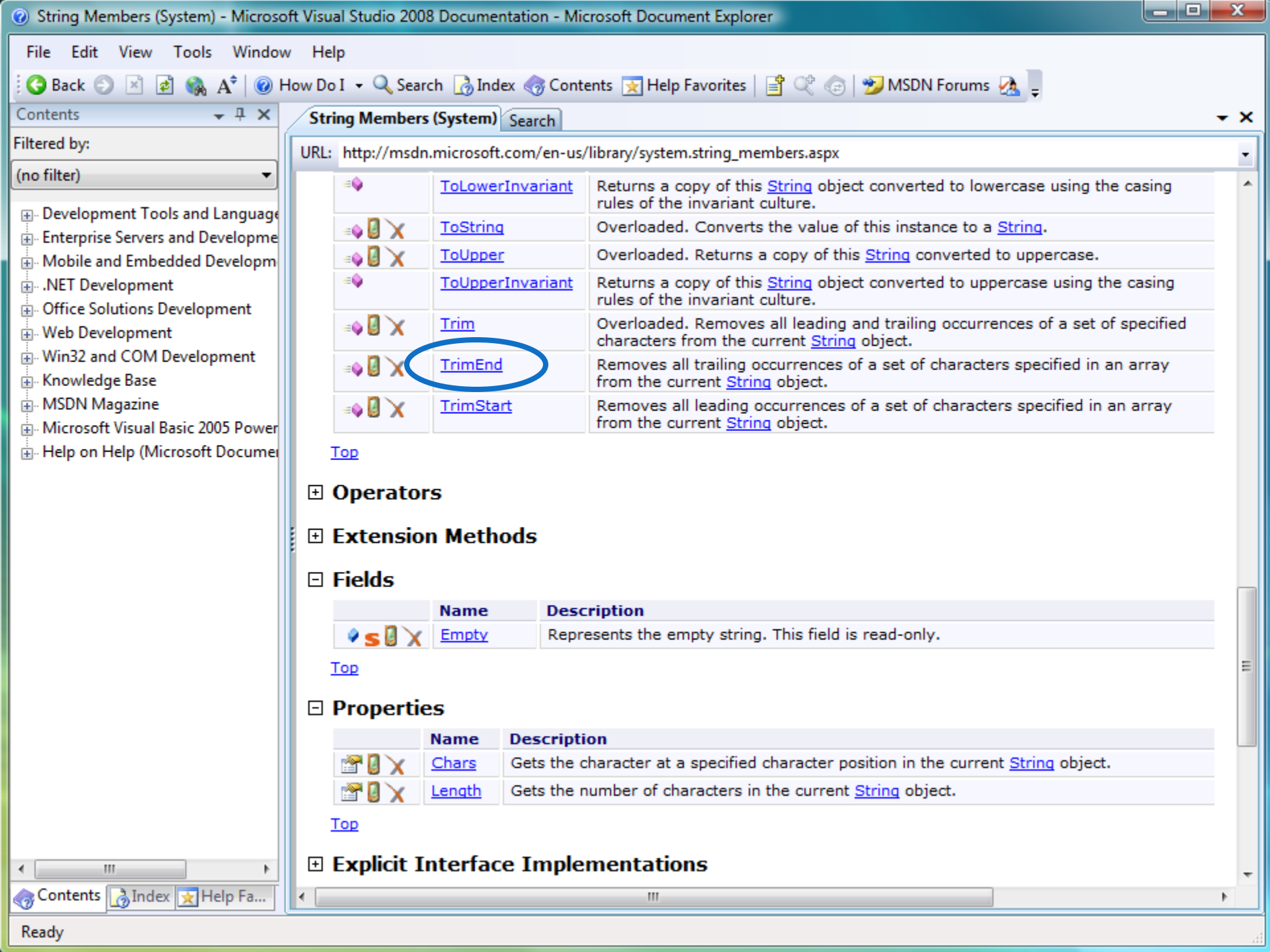
Top

Methods

	Name	Description
	Clone	Returns a reference to this instance of String.
s	Compare	Overloaded. Compares two specified String objects and returns an integer that indicates their relationship to one another in the sort order.
s	CompareOrdinal	Overloaded. Compares two String objects by evaluating the numeric values of the corresponding Char objects in each string.
	CompareTo	Overloaded. Compares this instance with a specified object or String and returns an integer that indicates whether this instance precedes, follows, or appears in the same position in the sort order as the specified object or String.
s	Concat	Overloaded. Concatenates one or more instances of String, or the String representations of the values of one or more instances of Object.
	Contains	Returns a value indicating whether the specified String object occurs within this string.
s	Copy	Creates a new instance of String with the same value as a specified String.
	CopyTo	Copies a specified number of characters from a specified position in this

javascript:EC_Click()


String Members (System) - Microsoft Visual Studio 2008 Documentation - Microsoft Document Explorer
File Edit View Tools Window Help
Back How Do I Search Index Contents Help Favorites MSDN Forums
Contents
Filtered by:
(no filter)
Development Tools and Language
Enterprise Servers and Developme
Mobile and Embedded Developm
.NET Development
Office Solutions Development
Web Development
Win32 and COM Development
Knowledge Base
MSDN Magazine
Microsoft Visual Basic 2005 Power
Help on Help (Microsoft Docume
Contents Index Help Fa...
String Members (System) Search
URL: http://msdn.microsoft.com/en-us/library/system.string_members.aspx
.NET Framework Class Library
String Members
Expand All Language Filter : All
Send Add Content... Click to Rat
Represents text as a series of Unicode characters.
The String type exposes the following members.
Constructors
Methods
Operators
Extension Methods
Fields
Properties
Explicit Interface Implementations
See Also
Navigation
MSDN Library .NET Development .NET Framework 3.5 .NET Framework .NET Framework Class Library


String Members (System) - Microsoft Visual Studio 2008 Documentation - Microsoft Document Explorer
File Edit View Tools Window Help
Back How Do I Search Index Contents Help Favorites MSDN Forums
Contents
Filtered by:
(no filter)
Development Tools and Language
Enterprise Servers and Developme
Mobile and Embedded Developm
.NET Development
Office Solutions Development
Web Development
Win32 and COM Development
Knowledge Base
MSDN Magazine
Microsoft Visual Basic 2005 Power
Help on Help (Microsoft Docume
Contents Index Help Fa...
String Members (System) Search
URL: http://msdn.microsoft.com/en-us/library/system.string_members.aspx
ToLowerInvariant Returns a copy of this String object converted to lowercase using the casing rules of the invariant culture.
ToString Overloaded. Converts the value of this instance to a String.
ToUpper Overloaded. Returns a copy of this String converted to uppercase.
ToUpperInvariant Returns a copy of this String object converted to uppercase using the casing rules of the invariant culture.
Trim Overloaded. Removes all leading and trailing occurrences of a set of specified characters from the current String object.
TrimEnd Removes all trailing occurrences of a set of characters specified in an array from the current String object.
TrimStart Removes all leading occurrences of a set of characters specified in an array from the current String object.
Top
Operators
Extension Methods
Fields
Name Description
Empty Represents the empty string. This field is read-only.
Top
Properties
Name Description
Chars Gets the character at a specified character position in the current String object.
Length Gets the number of characters in the current String object.
Top
Explicit Interface Implementations
Ready

String.TrimEnd Method (System) - Microsoft Visual Studio 2008 Documentation - Microsoft Document Explorer

File Edit View Tools Window Help

Back How Do I Search Index Contents Help Favorites MSDN Forums

Contents

Filtered by:

(no filter)

- Development Tools and Language
- Enterprise Servers and Developme
- Mobile and Embedded Developm
- .NET Development
- Office Solutions Development
- Web Development
- Win32 and COM Development
- Knowledge Base
- MSDN Magazine
- Microsoft Visual Basic 2005 Power
- Help on Help (Microsoft Docume

Contents Index Help Fa...

String.TrimEnd Method (System) Search

URL: http://msdn.microsoft.com/en-us/library/system.string.trimend.aspx

.NET Framework Class Library

String.TrimEnd Method

Collapse All Language Filter : All

Send Add Content... Click to Rat

Removes all trailing occurrences of a set of characters specified in an array from the current String object.

Namespace: System
Assembly: mscorlib (in mscorlib.dll)

Syntax

Visual Basic (Declaration)

```
Public Function TrimEnd ( _
    ParamArray trimChars As Char() _
) As String
```

Visual Basic (Usage)

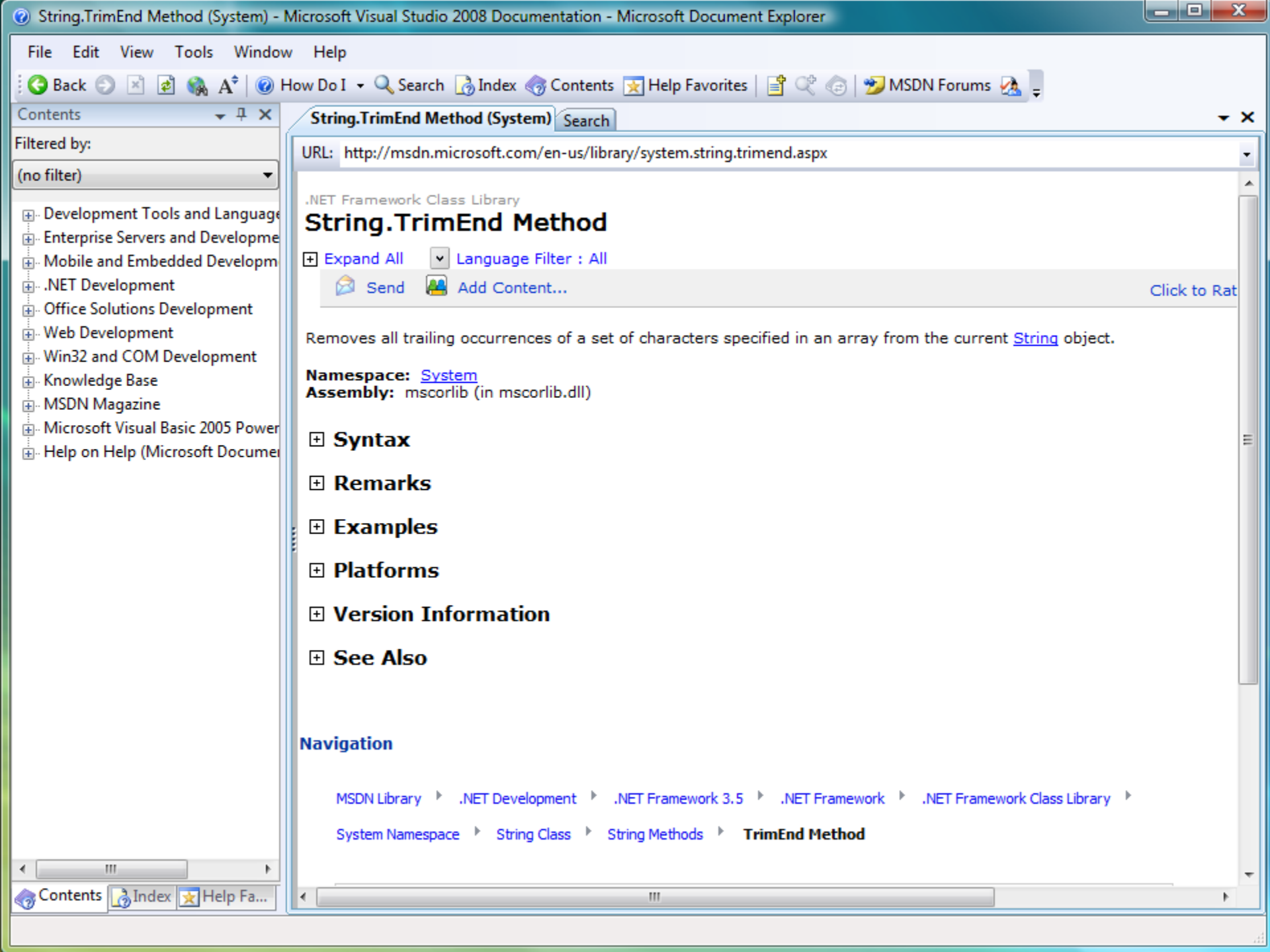
```
Dim instance As String
Dim trimChars As Char()
Dim returnValue As String

returnValue = instance.TrimEnd(trimChars)
```

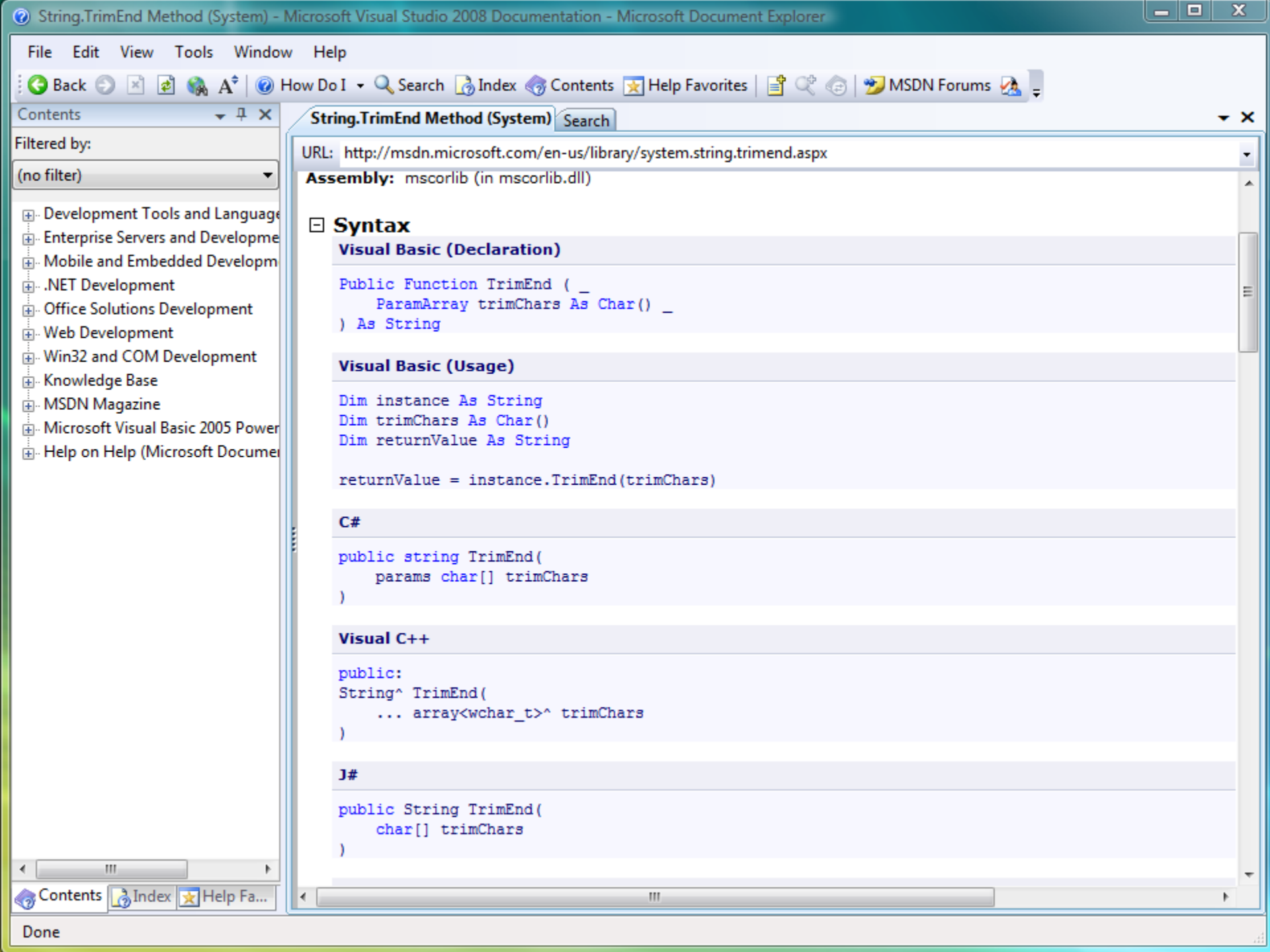
C#

```
public string TrimEnd(
    params char[] trimChars
)
```

Visual C++

Done


String.TrimEnd Method (System) - Microsoft Visual Studio 2008 Documentation - Microsoft Document Explorer
File Edit View Tools Window Help
Back How Do I Search Index Contents Help Favorites MSDN Forums
Contents
Filtered by:
(no filter)
Development Tools and Language
Enterprise Servers and Developme
Mobile and Embedded Developm
.NET Development
Office Solutions Development
Web Development
Win32 and COM Development
Knowledge Base
MSDN Magazine
Microsoft Visual Basic 2005 Power
Help on Help (Microsoft Documer
Contents Index Help Fa...
String.TrimEnd Method (System) Search
URL: http://msdn.microsoft.com/en-us/library/system.string.trimend.aspx
.NET Framework Class Library
String.TrimEnd Method
Expand All Language Filter : All
Send Add Content... Click to Rat
Removes all trailing occurrences of a set of characters specified in an array from the current String object.
Namespace: System
Assembly: mscorlib (in mscorlib.dll)
Syntax
Remarks
Examples
Platforms
Version Information
See Also
Navigation
MSDN Library .NET Development .NET Framework 3.5 .NET Framework .NET Framework Class Library
System Namespace String Class String Methods TrimEnd Method


String.TrimEnd Method (System) - Microsoft Visual Studio 2008 Documentation - Microsoft Document Explorer
File Edit View Tools Window Help
Back How Do I Search Index Contents Help Favorites MSDN Forums
Contents
Filtered by:
(no filter)
Development Tools and Language
Enterprise Servers and Developme
Mobile and Embedded Developm
.NET Development
Office Solutions Development
Web Development
Win32 and COM Development
Knowledge Base
MSDN Magazine
Microsoft Visual Basic 2005 Power
Help on Help (Microsoft Documer
Contents Index Help Fa...
String.TrimEnd Method (System) Search
URL: http://msdn.microsoft.com/en-us/library/system.string.trimend.aspx
Assembly: mscorlib (in mscorlib.dll)
Syntax
Visual Basic (Declaration)
Public Function TrimEnd (_
ParamArray trimChars As Char() _
) As String
Visual Basic (Usage)
Dim instance As String
Dim trimChars As Char()
Dim returnValue As String
returnValue = instance.TrimEnd(trimChars)
C#
public string TrimEnd(
params char[] trimChars
)
Visual C++
public:
String^ TrimEnd(
... array<wchar_t>^ trimChars
)
J#
public String TrimEnd(
char[] trimChars
)
Done

String.TrimEnd Method (System) - Microsoft Visual Studio 2008 Documentation - Microsoft Document Explorer

File Edit View Tools Window Help

Back How Do I Search Index Contents Help Favorites MSDN Forums

Contents

Filtered by:

(no filter)

- Development Tools and Language
- Enterprise Servers and Developme
- Mobile and Embedded Developm
- .NET Development
- Office Solutions Development
- Web Development
- Win32 and COM Development
- Knowledge Base
- MSDN Magazine
- Microsoft Visual Basic 2005 Power
- Help on Help (Microsoft Docume

Contents Index Help Fa...

String.TrimEnd Method (System) Search

URL: http://msdn.microsoft.com/en-us/library/system.string.trimend.aspx

Parameters

trimChars

Type: System.Char []

An array of Unicode characters to remove or a null reference (**Nothing** in Visual Basic).

Return Value

Type: System.String

The string that remains after all occurrences of the characters in the *trimChars* parameter are removed from the end of the current String object. If *trimChars* is a null reference (**Nothing** in Visual Basic), white-space characters are removed instead.

Remarks

The **TrimEnd** method removes from the current string all trailing characters that are in the *trimChars* parameter. The trim operation stops when a character that is not in *trimChars* is encountered. For example, if the current string is "123abc456xyz789" and *trimChars* contains the digits from '1' through '9', the **TrimEnd** method returns "123abc456xyz".

Note:

This method does not modify the value of the current instance. Instead, it returns a new string in which all trailing white space characters found in the current instance are removed.

For more information about which Unicode characters are categorized as white-space characters, see the Remarks section of the String.Trim() method overload.

Examples

The following code example demonstrates how you can use the **TrimEnd(Char[])** method overload to trim white space or other characters from the end of a string.

Visual Basic Copy Code

Done

String.TrimEnd Method (System) - Microsoft Visual Studio 2008 Documentation - Microsoft Document Explorer

File Edit View Tools Window Help

Back How Do I Search Index Contents Help Favorites MSDN Forums

Contents

Filtered by:

(no filter)

- Development Tools and Language
- Enterprise Servers and Developme
- Mobile and Embedded Developm
- .NET Development
- Office Solutions Development
- Web Development
- Win32 and COM Development
- Knowledge Base
- MSDN Magazine
- Microsoft Visual Basic 2005 Power
- Help on Help (Microsoft Docume

Contents Index Help Fa...

String.TrimEnd Method (System) Search

URL: http://msdn.microsoft.com/en-us/library/system.string.trimend.aspx

```
        Dim trimmedWhiteSpace As String = pathWhitespace.TrimEnd(Nothing)

        Console.WriteLine("The trimmed value is: {0}.", trimmedWhiteSpace)

    End Sub
End Module
' This code example displays the following
' to the console:
'
' The trimmed value is: c:/temp.
```

C# Copy Code

```
using System;


namespace String_Example
{
    class Application
    {
        public static void Main()
        {
            // Create a string that will be trimmed.
            string path = "c:/temp//";

            // Create an array of characters
            // that represent characters to trim.
            char[] charsToTrim = {'/'};

            // Thim the string.
            string trimmedPath = path.TrimEnd(charsToTrim);

            Console.WriteLine("The trimmed value is: {0}.", trimmedPath);

            // Create a string that will be trimmed.
```

Ready

Obiettivo:
Capire il programma di prova

```
namespace Introd {
  class Hello {
    static void Main ( string[] args )
    {
      System.Console.WriteLine ( "Hello World" );
    }
  }
}
```

Cosa fa questo programma è evidente …

… ma cosa significa **esattamente** questa roba?

Oggetti, Riferimenti, Classi, Metodi

PREMESSA

- Per il momento facciamo finta che non esistano i "tipi primitivi"
  e i "ValueType"

Type
sbyte
byte
char
short
ushort
int
uint
long
ulong
float
double

- Oggetti e riferimenti non si applica ai tipi primitivi (che sono una cosa "particolare")

Oggetti e Riferimenti (I)

- Ogni “cosa” è un **oggetto**
- Gli oggetti **non** sono associati a nomi simbolici (indicatori)

- Un programma opera sugli oggetti attraverso i **riferimenti**
- Ogni riferimento è associato ad un identificatore

IMPORTANTISSIMO:

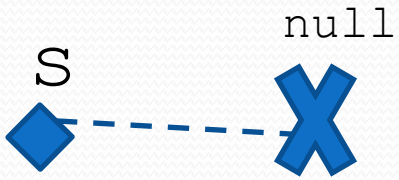
Oggetto ≠ Riferimento

Oggetti e Riferimenti (II)

```
String s;
```

Crea riferimento `s` a oggetto di tipo `String`

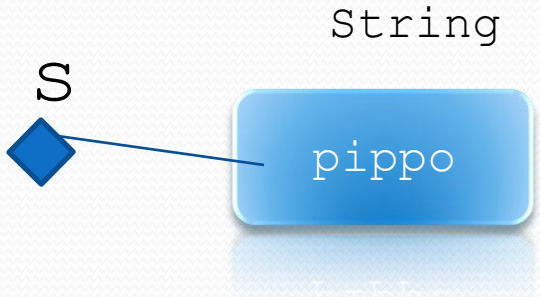
Non crea l'oggetto



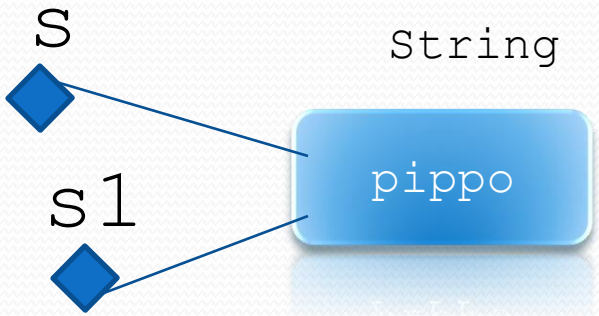

```
String s = new String("pippo");
```

Crea riferimento `s` a oggetto di tipo `String`

Crea oggetto di tipo `String` e lo **inizializza** con la sequenza di caratteri `pippo`




```
String s = new String ("pippo");
String s1 = s;
```



Oggetti e Riferimenti (III)

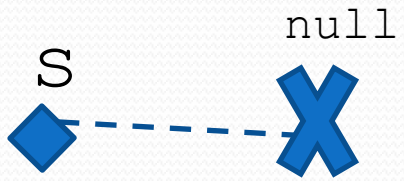
Analogia

- Oggetto ≈ Televisione
- Riferimento ≈ Telecomando

- Riferimento dice “cambia canale”
- Oggetto esegue l’operazione

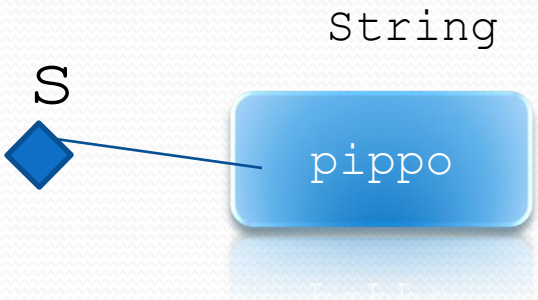
- Riferimento può esistere senza essere associato a un oggetto
- Oggetto può esistere senza essere riferito
- **Più** riferimenti possono riferire lo stesso **oggetto**

Oggetti e Riferimenti (IV)




```
string s;
```

- Manipolando il riferimento in questa situazione, si ha un errore **a tempo di esecuzione**




```
string s = new string ("pippo");
```

- In questa situazione l'oggetto può essere manipolato

Esempio (I)

```
string firstName;
firstName = new String (“Mario”);
string lastName = new String (“Rossi”);
Uri link;
string firstName2, s;
s = new String (“Luca”);
firstName2 = firstName;
firstName = s;
s = null;
lastName = new String (“Ferrari”);
link = new Uri (“http://www.google.com”);
```

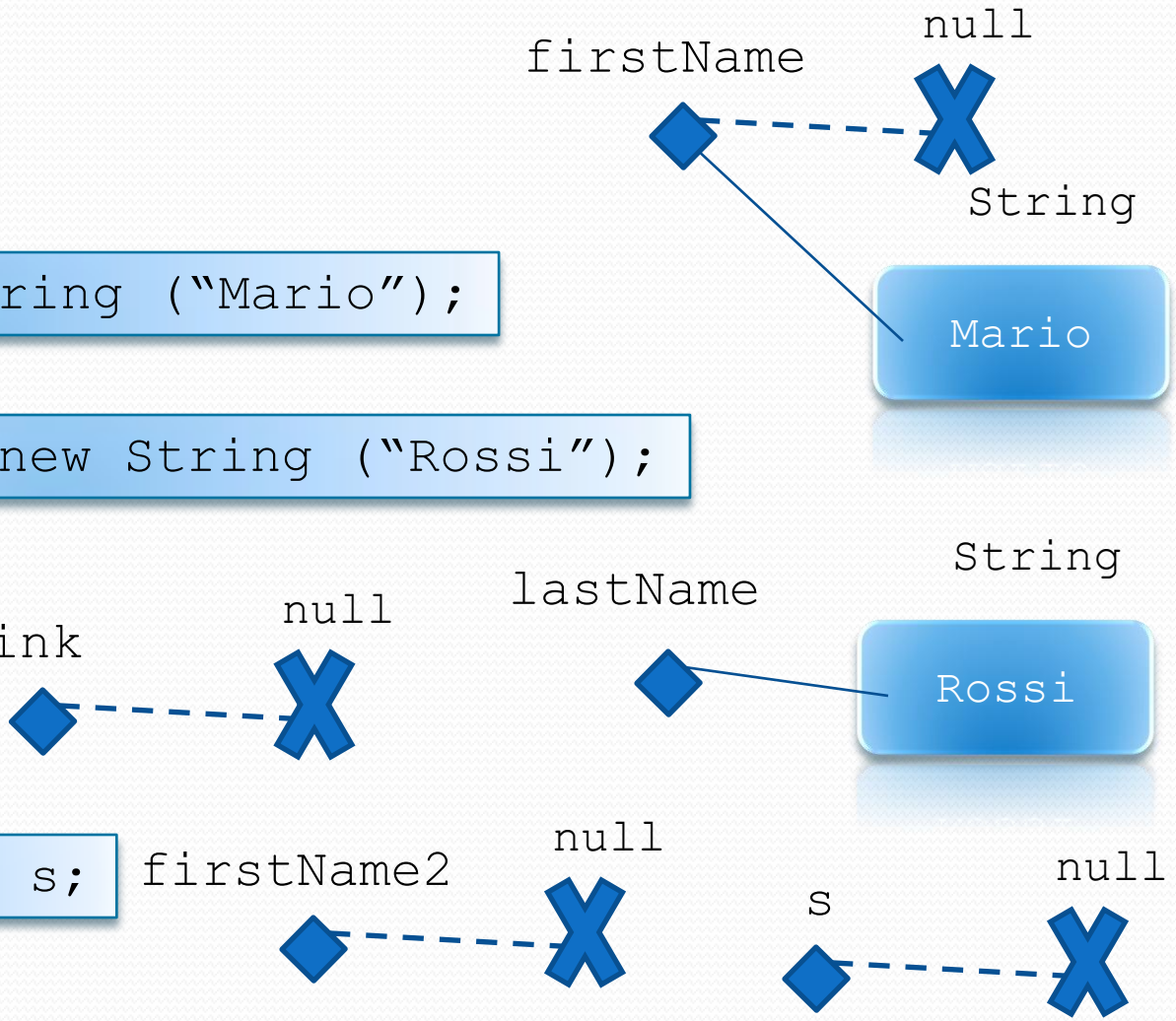
```
string firstName;
```

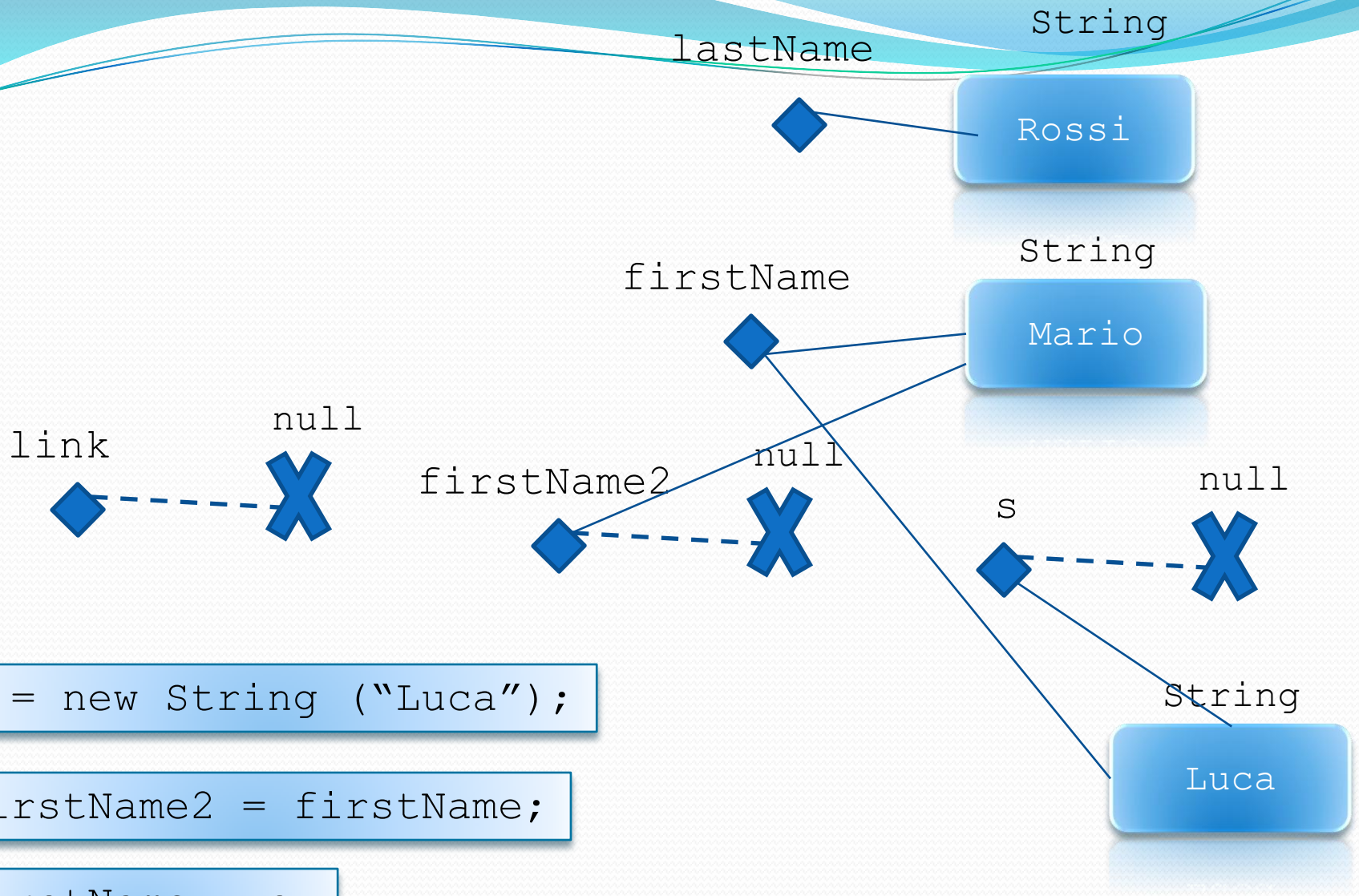
```
firstName = new String ("Mario");
```

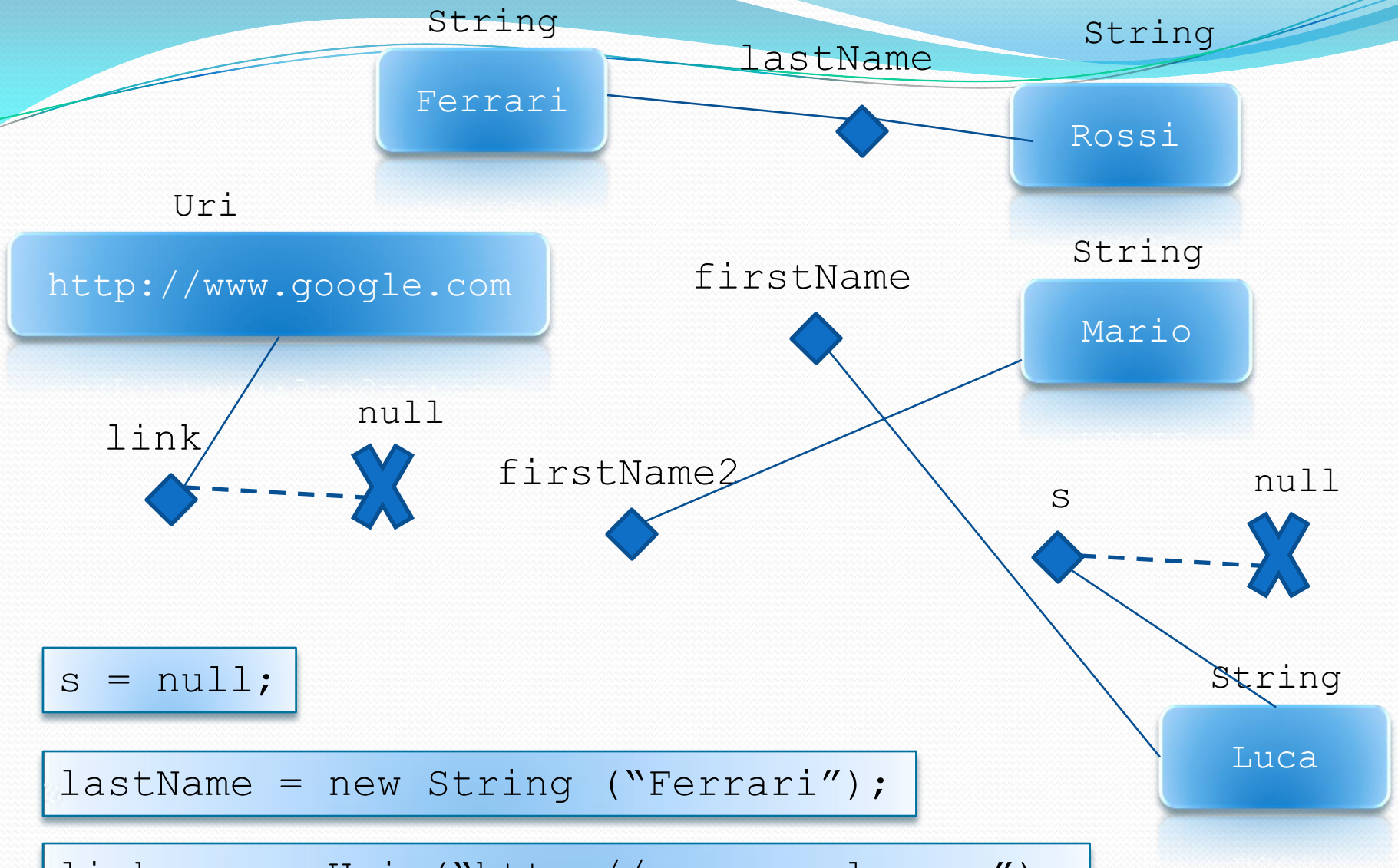
```
string lastName = new String ("Rossi");
```

```
Uri link;
```

```
string firstName2, s;
```




String
lastName
Rossi
String
firstName
Mario
link
null
firstName2
null
s
null
String
Luca
s = new String ("Luca");
firstName2 = firstName;
firstName = s;




```
s = null;
```

```
lastName = new String ("Ferrari");
```

```
link = new Uri ("http://www.google.com");
```

Esempio (II)

```
string man = "Tex Willer";
string friend;
friend = "Kit Carson";
string son = string.Empty;
string name = "Kit Willer";
son = name;
name = null;
string theForth = "Tiger Jack";
```

Classe (I)

- Classe
  - Definisce un nuovo tipo
  - Contiene dati e operazioni
  - Fondamentale nella programmazione a oggetti

```
class Stock
{
   ...
}
```

```
class CheckingAccount
{
   ...
}
```

```
class Rectangle
{
   ...
}
```

```
class Aircraft
{
   ...
}
```

```
class Rational
{
   ...
}
```

Classe (II)

- Ogni oggetto è istanza di un **tipo** (**classe**)

- **Esistono** moltissimi **tipi**
  - Predefiniti nel linguaggio
  - Definiti in librerie dell'SDK

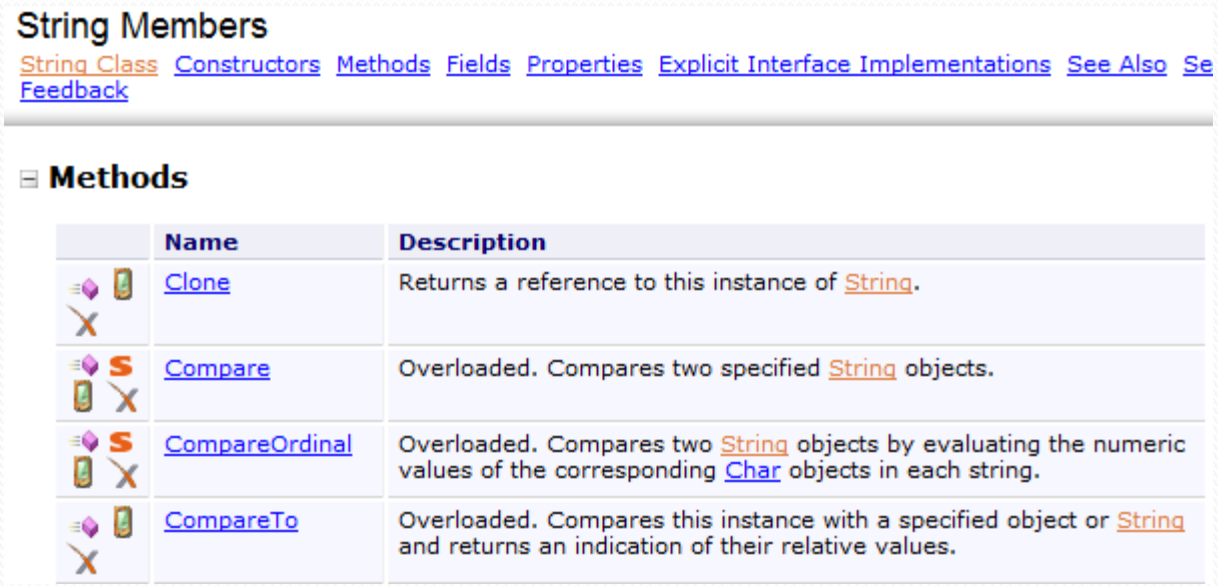
  - Se ne possono definire di nuovi
    (ogni programma ne definisce almeno uno)

  - A compile time viene creato il tipo
  - A runtime il tipo viene istanziato e "diventa" un'oggetto
  - A runtime possono esistere più oggetti dello stesso tipo

Classe (III)

- Su un oggetto posso applicare solo le operazioni permesse dal suo tipo (**metodi**)

- Nella pratica: necessario avere il manuale on-line
  - Tantissimi tipi
  - Ogni tipo esporta tanti metodi



String Members

String Class Constructors Methods Fields Properties Explicit Interface Implementations See Also Se
Feedback

Methods

	Name	Description
	Clone	Returns a reference to this instance of String.
S	Compare	Overloaded. Compares two specified String objects.
S	CompareOrdinal	Overloaded. Compares two String objects by evaluating the numeric values of the corresponding Char objects in each string.
	CompareTo	Overloaded. Compares this instance with a specified object or String and returns an indication of their relative values.

OperatingSystem
OperationCanceledException
OutOfMemoryException
OverflowException
ParamArrayAttribute
PlatformNotSupportedException
Random
RankException
ResolveEventArgs
SerializableAttribute
StackOverflowException
STAThreadAttribute
String
StringComparer
SystemException
ThreadStaticAttribute
TimeoutException
TimeZone
TimeZoneInfo
TimeZoneInfo.AdjustmentRule
TimeZoneNotFoundException
Type
TypeInitializationException

Infatti …

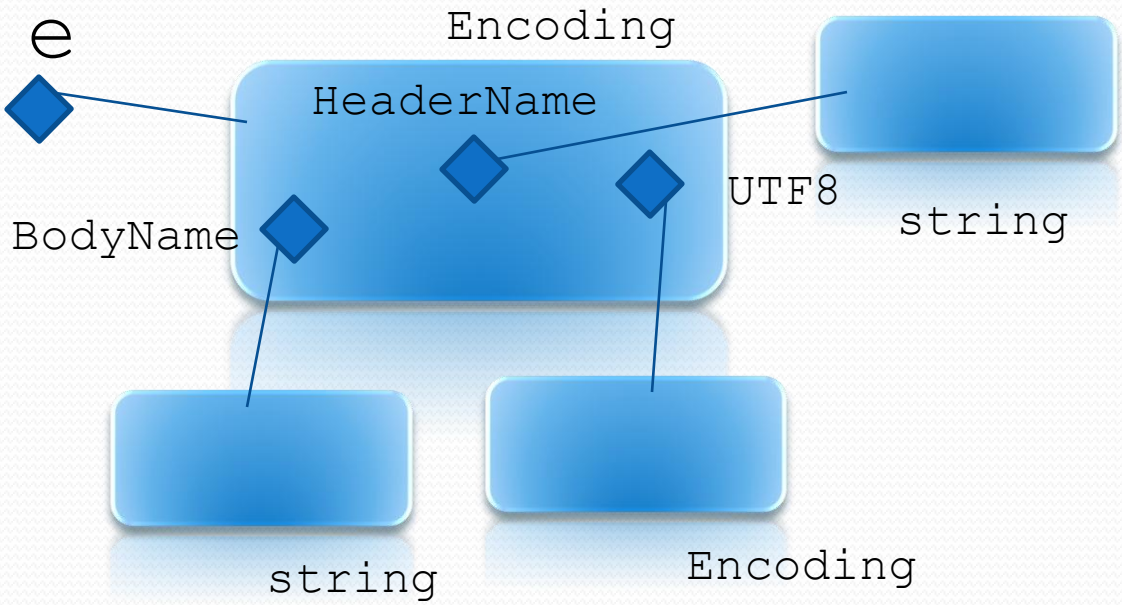
```
namespace Introd {
  class Hello {
    static void Main ( string[] args )
    {
      System.Console.WriteLine ( "Hello World" );
    }
  }
}
```

- Definisce una classe `Hello`
- Ha un solo metodo `Main()`

Classe: `Field` (I)

- Un oggetto può rendere visibili alcuni degli oggetti in esso contenuti (**field**,membri dato)
- Accade di rado (ma accade)

```
Encoding e = new Encoding (…);
```



.NET Framework Class Library

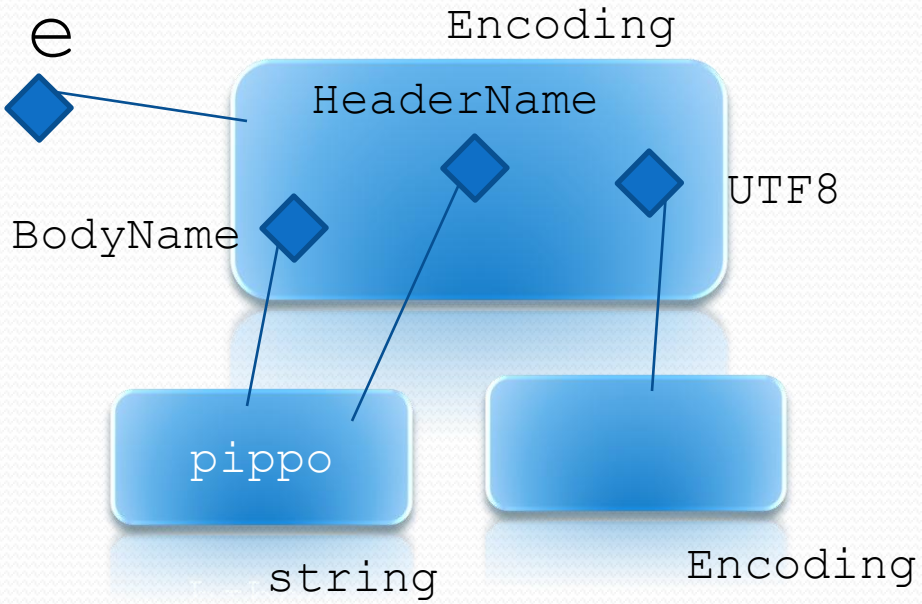
Encoding Members

```
…
public string BodyName;
…
public bool HeaderName;
…
public Encoding UTF8;
…
```

Classe: `Field` (II)

- Sintassi per l'accesso: `nomeRiferimento.NomeField`

```
e.HeaderName = new String ("pippo");
e.BodyName = e.HeaderName;
```



.NET Framework Class Library

Encoding Members

```
…
public string BodyName;
…
public bool HeaderName;
…
public Encoding UTF8;
…
```

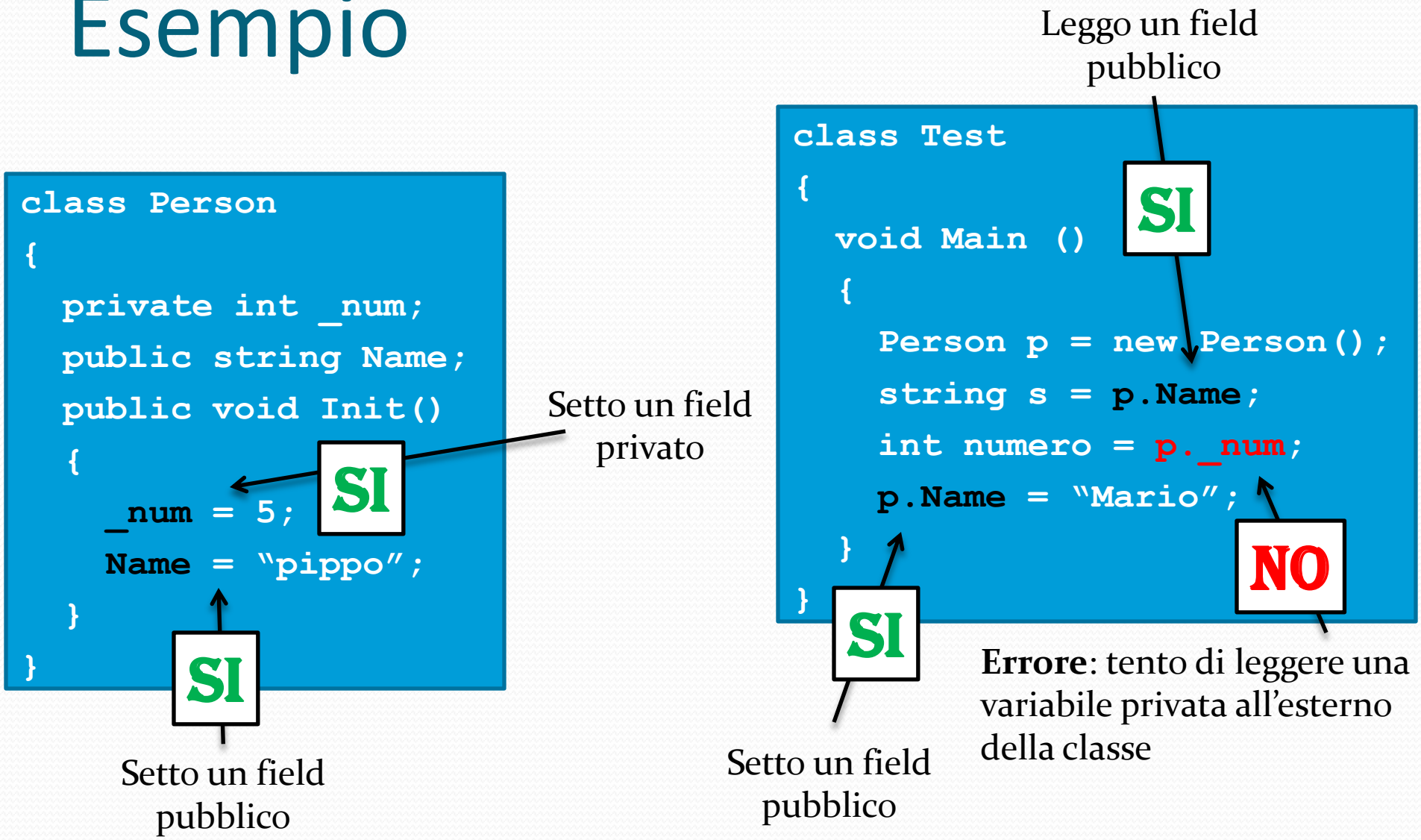
Scope

- Lo scope rappresenta l'ambito di visibilità di una variabile, field, metodo, …
- Esistono 5 diversi scope
  - `private` (default per metodi, field, property)
  - `public`
  - `internal` (default per le classi)
  - `protected`
  - `protected internal`
- Per ora vediamo solo i primi 2…
- Per ora consideriamo
  `internal` ≈≈ `public`

`public,private`

- Field, metodo, … `public`
  - Visibile ovunque
- Field, metodo, … `private`
  - Visibile solo all'interno della classe nel quale è dichiarato
  - Non può essere utilizzato all'esterno, il compilatore non lo permette

Esempio


class Person
{
private int _num;
public string Name;
public void Init()
{
_num = 5;
Name = "pippo";
}
}
SI
SI
Setto un field pubblico
Setto un field privato
class Test
{
void Main ()
{
Person p = new Person();
string s = p.Name;
int numero = p._num;
p.Name = "Mario";
}
}
SI
SI
NO
Leggo un field pubblico
Setto un field pubblico
Errore: tento di leggere una variabile privata all'esterno della classe

Field: inizializzazione

- Un field quando viene dichiarato è anche inizializzato (cosa che non accade con le variabili)
- Valori di default
  - Numeri a `0`
  - Boolean a `false`
  - Reference type a `null`

```
class Person
{
  private int _num;
  public string Name;
  private bool _isDriver;
  private int _age = 24;
}
```

Inizializzato a `0`
Equivale a
`private int _num = 0;`

Inizializzato a `null`
Equivale a
`public string Name = null;`

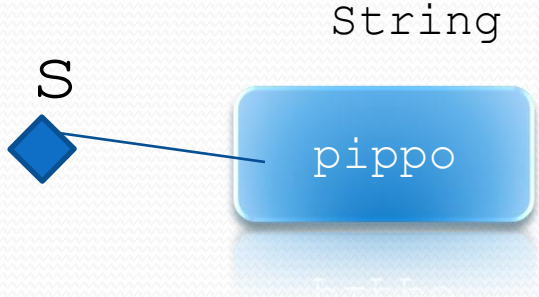
Inizializzato a `false`
Equivale a
`private bool _isDriver = false;`

Applicazione metodi: Sintassi

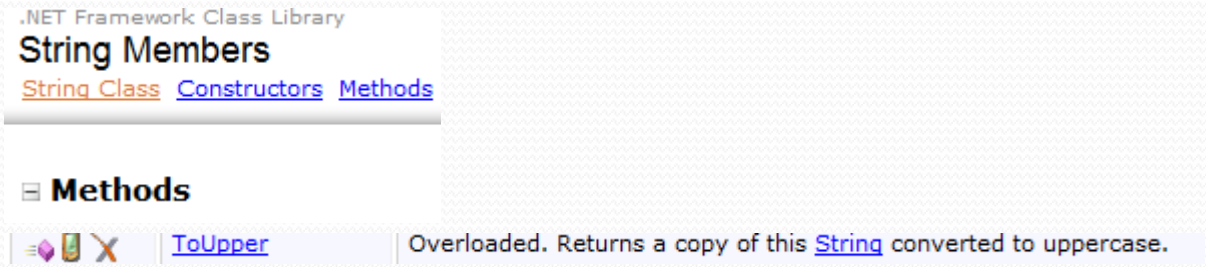
- Sintassi per **applicazione di metodi** all'oggetto:

```
riferimento.NomeMetodo(listaEventualiArgomenti);
```

```
String s = new String ("pippo");
s.ToUpper ();
```



Metodo `ToUpper ()` sull'oggetto



Metodi: Name overloading

- Un metodo di una classe è identificato dalla **coppia**:

  `MethodName` `listOfArguments`

Name Overloading

Una classe può avere più metodi:

- Con stesso `MethodName`
- Diversa `listOfArguments`

.NET Framework Class Library

StringWriter Class

Methods

Overload List

	Name	Description
	Write(Boolean)	Writes the text representation of a **Boolean** value to the text strea
	Write(Char)	Writes a character to the text stream.
	Write(Char[])	Writes a character array to the text stream.
	Write(Decimal)	Writes the text representation of a decimal value to the text stream
	Write(Double)	Writes the text representation of an 8-byte floating-point value to the text stream.
	Write(Int32)	Writes the text representation of a 4-byte signed integer to the text stream.
	Write(Int64)	Writes the text representation of an 8-byte signed integer to the text stream.
	Write(Object)	Writes the text representation of an object to the text stream by calling **ToString** on that object.
	Write(Single)	Writes the text representation of a 4-byte floating-point value to the text stream.
	Write(String)	Writes a string to the text stream
	Write(UInt32)	Writes the text representation of a 4-byte unsigned integer to the te stream.

Esempio (I)

```
StringWriter sw = new StringWriter(…);
sw.Write("This is my first C# program.");
```

Metodo `Write ()` sull'oggetto riferito a `sw`

.NET Framework Class Library

StringWriter Class

Methods

```
public void Write (bool  value);
public void Write (char  value);
public void Write (char[]  value);
…
public void Write (string value);
…
```

Esempio (II)

```
SomeType rif = new SomeType();
SomeOtherType i = rif.MethodName();
```

- Compilatore ricerca in `SomeType` un metodo con:
  - Nome `MethodName`
  - Lista di argomenti vuota
  - Valore di ritorno `SomeOtherType`
- Compilatore segnala errore se:
  - Non trova il metodo

  oppure
  - Il metodo torna un valore "incompatibile" con `SomeOtherType`

Metodi: creazione

- I metodi di una classe implementano i comportamenti della classe
  - Bisogna specificare valore di ritorno, parametri, e corpo
  - Utilizzare `void` se non ritorna nessun valore

```
class Stock
{
  void Buy ( int shares )
  {...}
  void Sell ( int shares )
  {...}
  void SetPrice ( double price )
  {...}
  double Value ()
  {...}
}
```

Metodi: implementazione

- L'implementazione di un metodo può utilizzare la keyword `this`
  - `this` rappresenta l'oggetto corrente, nel quale è definito il metodo
  - `this` è utilizzato per accedere ai field, definiti nella classe, all'interno del metodo

```
class Stock
{
   string name;
   double price;
   int shares;
   void Buy (int s)
   {
      this.shares += s;
   }
}
```

this (I)

- Si deve utilizzare `this` quando il nome di un parametro o variabile locale è in conflitto con la variabile d'istanza

SI

```
class Stock
{
   string name;
   double price;
   int shares;
   void Buy (int shares)
   {
      this.shares += shares;
   }
}
```

NO

```
class Stock
{
   string name;
   double price;
   int shares;
   void Buy (int shares)
   {
      shares += shares;
   }
}
```

Ambiguo

this (II)

- Si può omettere `this` quando non c'è ambiguità

```
class Stock
{
  string name;
  double price;
  int shares;
  void Buy (int s)
  {
    this.shares += s;
  }
}
```

SI

```
class Stock
{
  string name;
  double price;
  int shares;
  void Buy (int s)
  {
    shares += s;
  }
}
```

SI

Ordine di dichiarazione

- Metodi, field, proprietà, possono essere dichiarate in ogni ordine
  - Non è richiesto che un field sia dichiarato prima di essere utilizzato
  - Invece una variabile all'interno di un metodo deve essere dichiarata prima di essere utilizzata

```
class Stock
{
  void Buy (int s)
  {
    shares += s;
  }
  string name;
  double price;
  int shares;
}
```

SI

```
class Stock
{
  int shares;
  void Buy (int s)
  {
    shares += s;
  }
  string name;
  double price;
}
```

SI

Valore di ritorno

- Utilizzare `return` per inviare dati
  - Valore inviato indietro al chiamante
  - Tutto il codice scritto, nello stesso flusso di esecuzione, dopo `return` non verrà mai eseguito

```
class Stock
{
   double Value ()
   {
      return price * shares;
   }
   ...
}
```

```
Stock ibm = new Stock();
ibm.SetPrice(56.0);
ibm.Buy(100);

double v = ibm.Value();
```

Classe: `Property` (I)

- Viste dall'esterno sono utilizzate concettualmente come i field
- Incapsulano un field

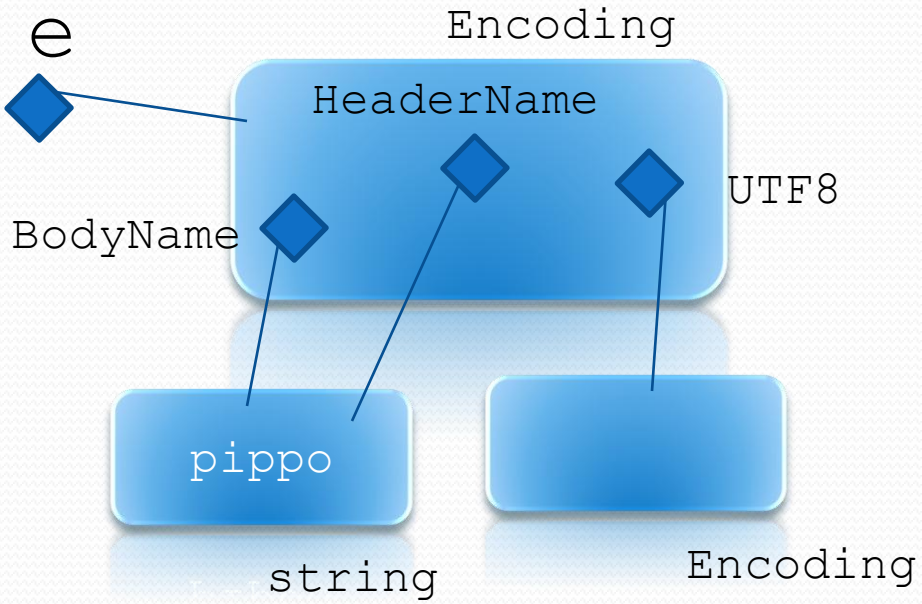
- Nella pratica:
  - Un field è messo `private`
  - Una `Property` incapsula il field ed è `public`
  - Dall'esterno si utilizzano le `Property` per accedere ai `field`

Classe: Property (II)

- Sintassi per l'accesso:

```
nomeRiferimento.NomeProperty
```

```
e.HeaderName = new String ("pippo");
e.BodyName = e.HeaderName;
```



Proprietà pubbliche

	Nome
	ASCII
	BigEndianUnicode
	BodyName
	CodePage
	DecoderFallback
	Default
	EncoderFallback
	EncodingName
	HeaderName
	IsBrowserDisplay
	IsBrowserSave
	IsMailNewsDisplay
	IsMailNewsSave

.NET Framework Class Library

Encoding Members

```
…
public string BodyName;
…
public bool HeaderName;
…
public Encoding UTF8;
…
```

Esempio (I)

```
class SomeClass
{
  private String _name;
  public String Name
  {
      get {return _name;}
      set {_name = value;}
  }
}
```

Non devono essere necessariamente simili

Ma cos'è `value`?

Chi lo ha definito?

Esempio (II)

`value` è il parametro passato al "metodo" set

Concettualmente

```
public void Set(String value)
{
  _name = value;
}
```

Domande

Perché anziché usare una Property, non espongo direttamente il Field?

Property vs Field (I)

- Le property incapsulano un field
- Nel codice delle property (**set**) è possibile validare il dato prima di essere salvato nel field
  - Maggior sicurezza
  - Controllo dell'integrità dello stato interno
- La struttura interna della classe rimane nascosta

Property vs Field (II)

```
class Person
{
  private int _age;
  public string Age
  {
    get {return _age;}
    set
    {
      if (value >= 0)
      {
        _age = value
      }
    }
}
```

Non ha senso un'età negativa
Nella proprietà si fa il test:
solo se il valore è accettabile (>=0)
viene settato il campo privato

Osservazione (I-a)

- Posso anche creare solo il metodo *get* o *set* della property
- Nella pratica non si usa creare solo il metodo *set*
- Utile quando alcune property devono essere *readonly*

SI

```
private string _name;
public string Name
{
    get {return _name;}
    set {_name = value;}
}
```

```
private string _name;
public string Name
{
    get {return _name;}
}
```

SI

```
private string _name;
public string Name
{
    set {_name = value;}
}
```

NO

Osservazione (I-b)

- get e set possono avere scope diverso
  - Se non viene indicato lo scope al get o al set si intende lo scope della proprietà

```
private string _name;
public string Name
{
  get {return _name;}
  private set {_name = value;}
}
```

SI

```
private string _name;
public string Name
{
  private get {return _name;}
  set {_name = value;}
}
```

NO

Osservazione (II)

- Viste dall'esterno sono utilizzate concettualmente come i field
- Incapsulano un field
  - NON è sempre vero
  - Possono eseguire delle operazioni è visualizzare un field fittizio

Osservazione (III)

- Con il .NET Framework 3.5 esiste una sintassi abbreviata

```
private string _name;
public string Name
{
    get {return _name;}
    set {_name = value;}
}
```

```
public string Name {get; set;}
```

```
private string _name;
public string Name
{
  get {return _name;}
  private set {_name = value;}
}
```

```
public string Name {get; private set;}
```

Esempio

```
class SomeClass
{
  private DateTime _birthdate;
  public int Age
  {
      get
      {
      return DateTime.Now.Year - _birtdate.Year;
      }
  }
}
```

Attenzione:
Non tiene conto del mese

Domande (II)

- Va bene, non devo usare i field direttamente, ma perché devo usare le property?
  Se proprio devo incapsulare un field, non posso utilizzare un metodo?
- Qualcosa del tipo

```
TypeField GetNomeField () {…}
void SetNomeField (TypeField value){…}
```

```
class Person
{
  private string _name;
  public string Name
  {
    get {return _name;}
    set {_name = value;}
  }
}
```

```
class Person
{
  private string _name;
  public string GetName()
  {
    return _name;
  }
  public void SetName (string value)
  {
    _name = value;
  }
}
```

```
Person p = new Person ();
p.Name = "pippo";
string s = p.Name;
```

```
Person p = new Person ();
p.SetName ("pippo");
string s = p.GetName()
```

Property **vs** Metodo

- **E' equivalente usare property o metodi**.
- Microsoft suggerisce di usare le property quando la logica per visualizzare il risultato è breve (**2, 3 righe** di codice al massimo)
- Altrimenti utilizzare dei metodi *getter* e *setter*
- In più se si deve creare solo il set, creare un metodo *setter* e non usare una property "*writeonly*"

Convenzioni sui nomi

Classe
Metodo
Property
public field

- Molte parole
- Tutte capitalized

```
ThisIsAClassWithLongName
ThisIsAMethod
```

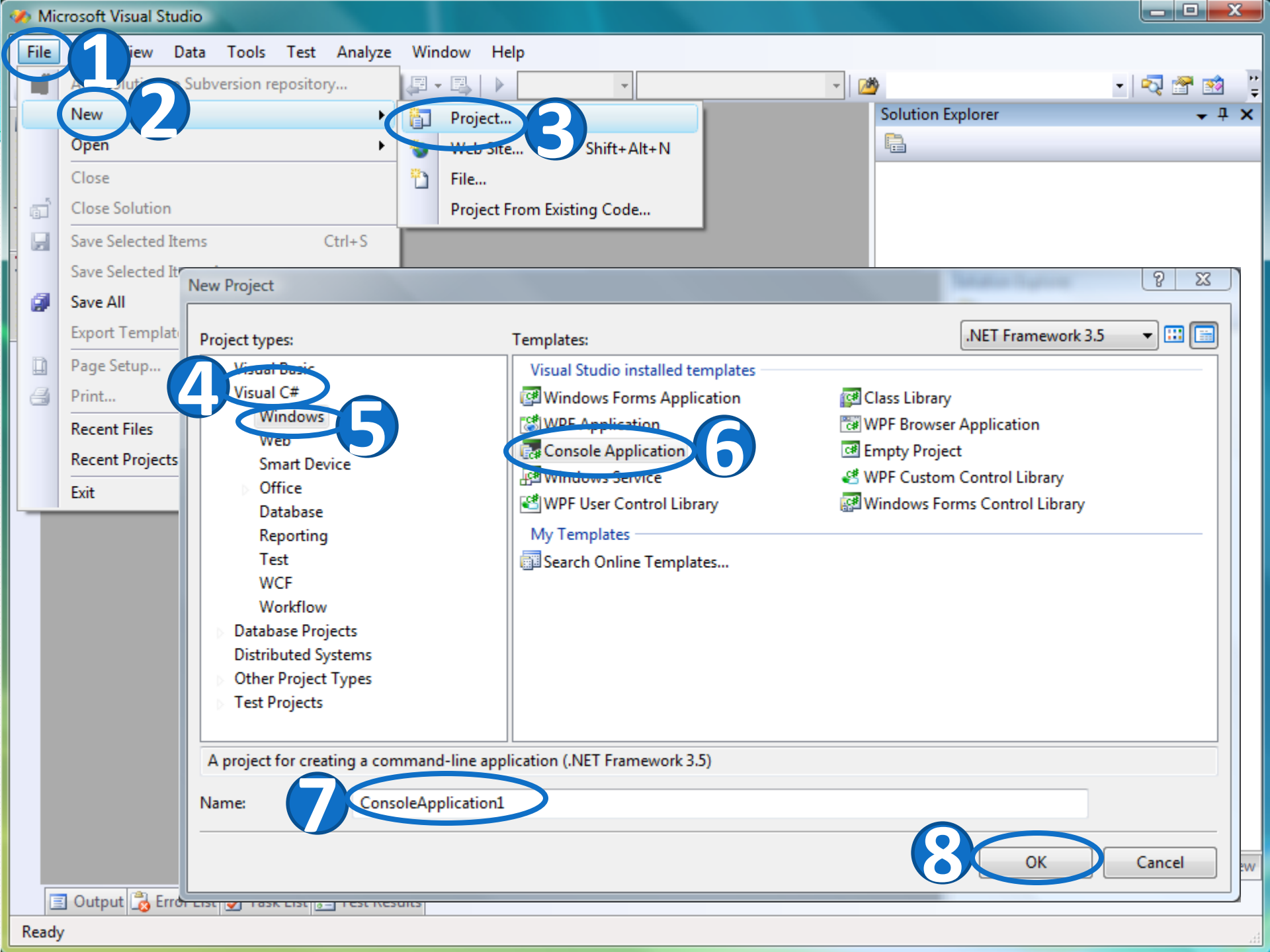
Riferimento
Field Privato
Parametro di un metodo

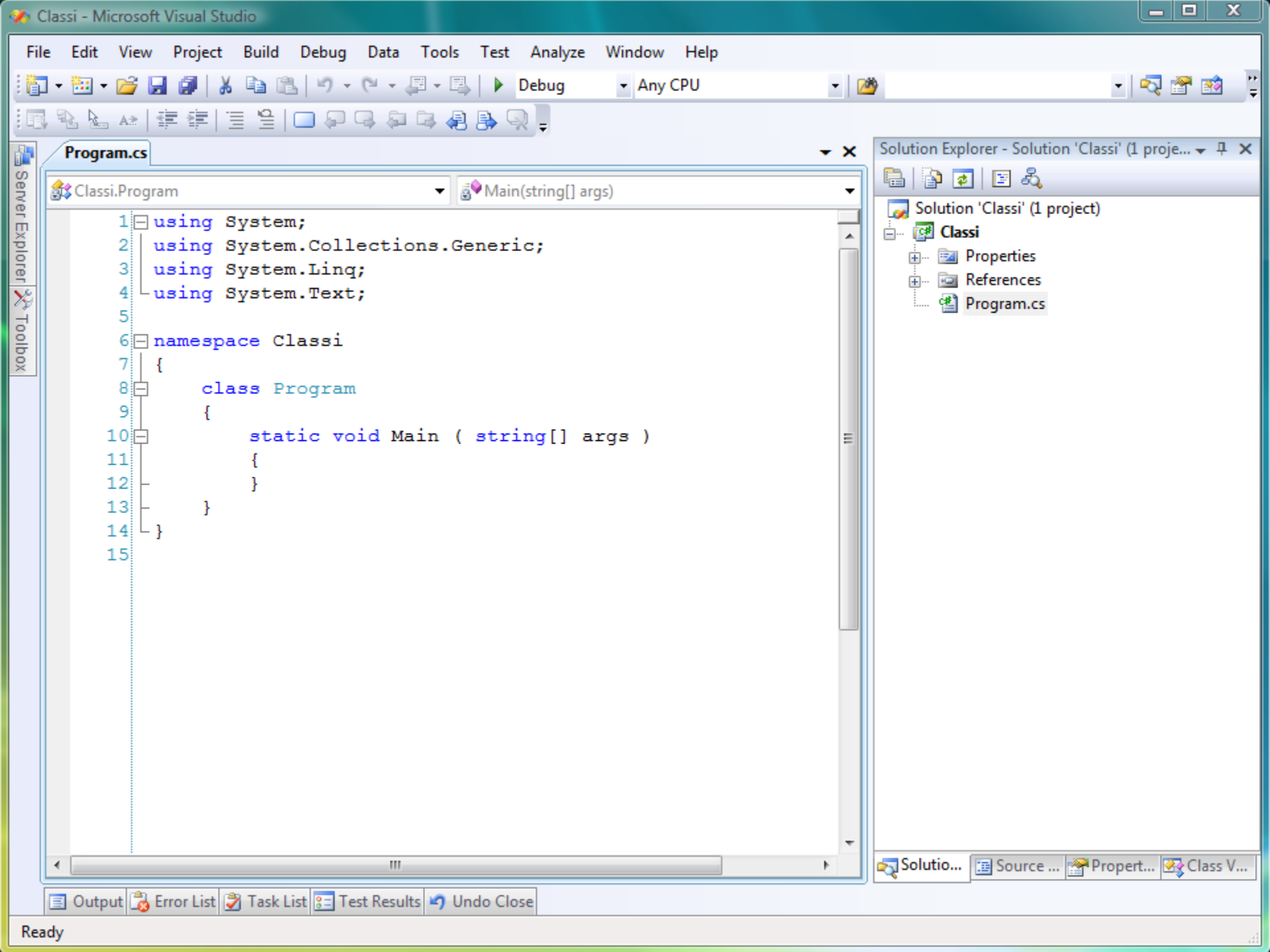
- Molte parole
- Tutte capitalized **eccetto la prima**

```
thisIsAReference
```

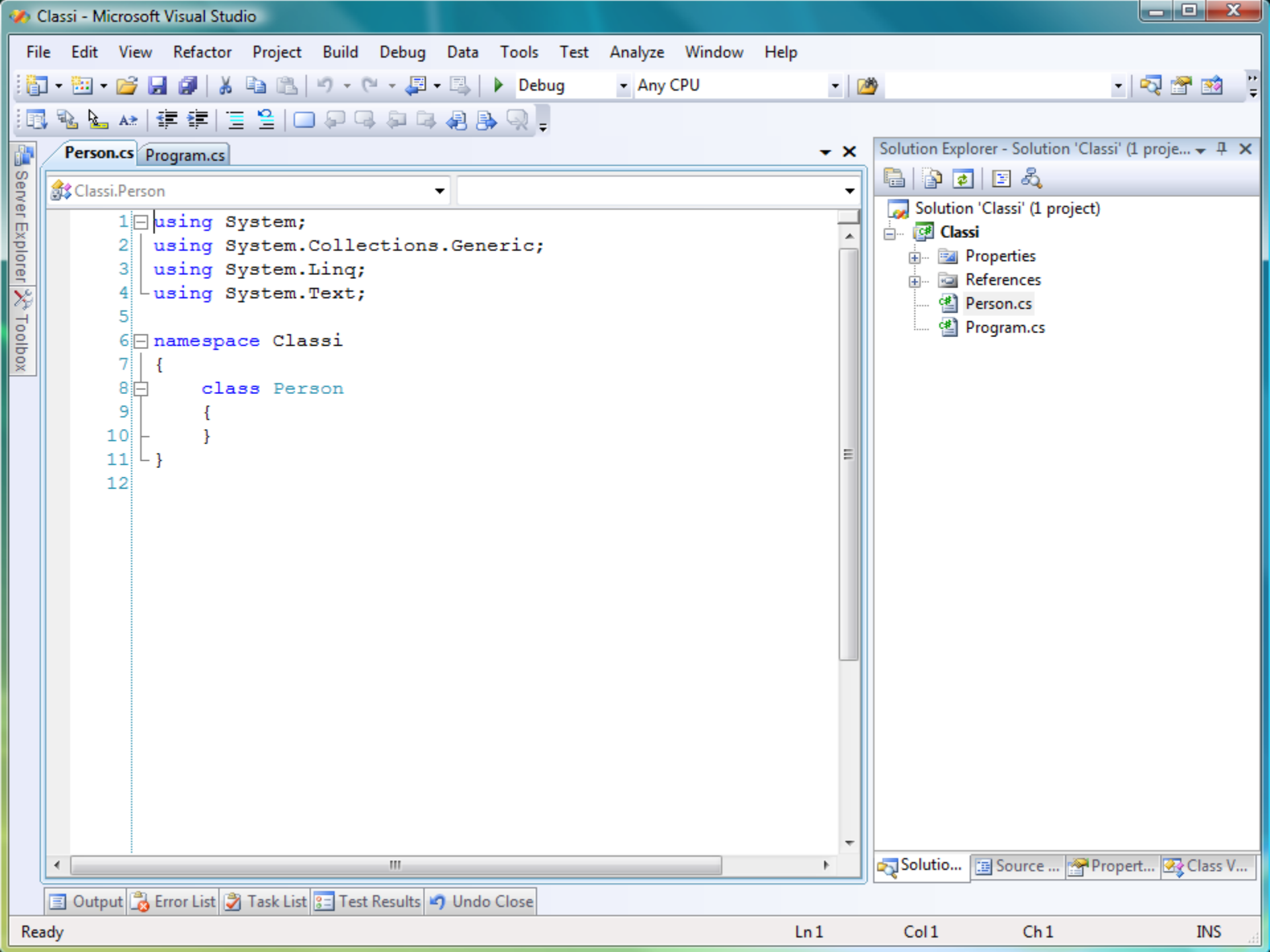
Esempio

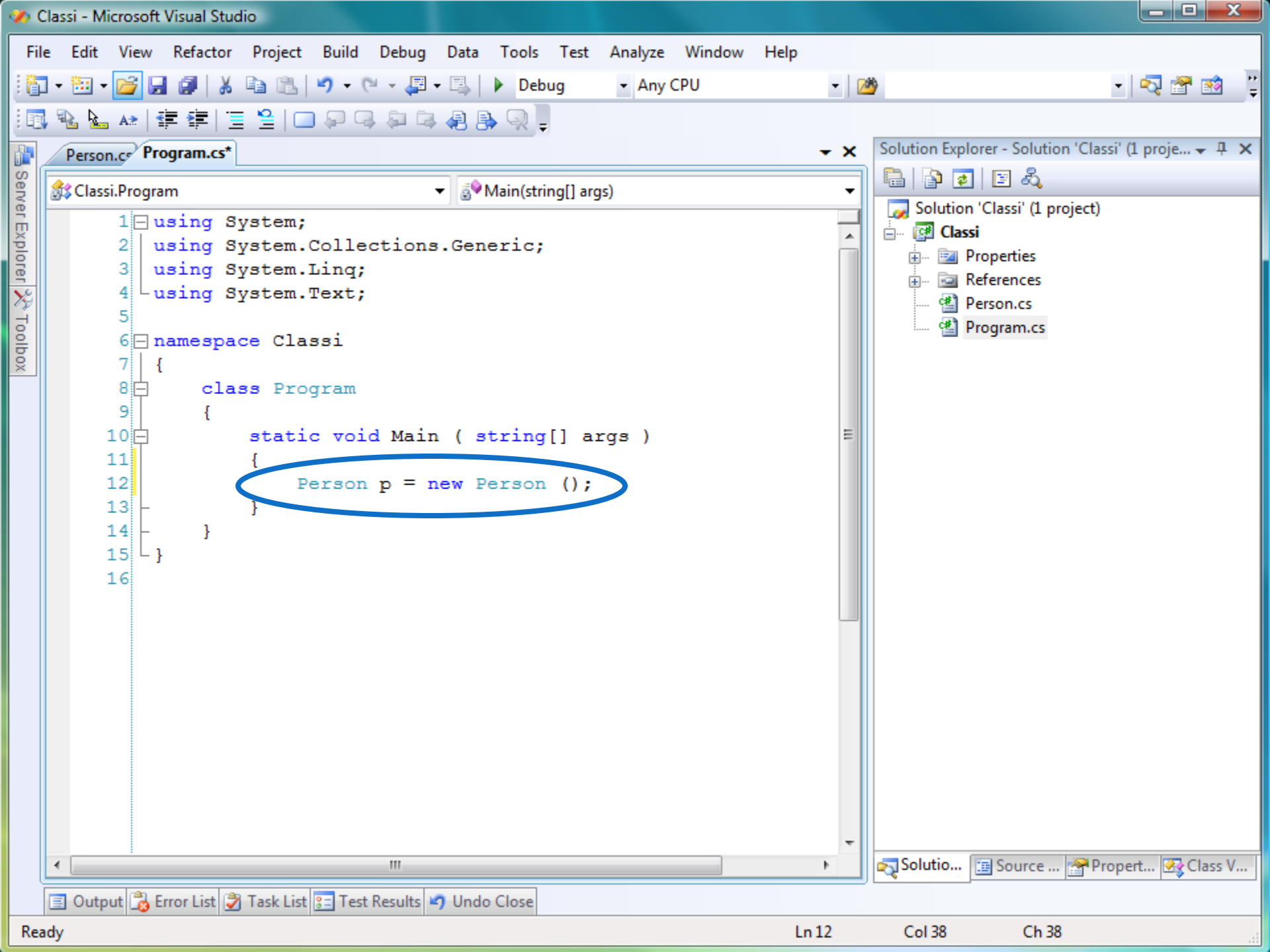
```
class SavingAccount
{
  public void Deposit (double amountOfDeposit) { ... }
  public int GetAccountNumber() { ... }

  public string Name { get; set; }

  public double Balance;
  private int _accountNumber;
}
```


Microsoft Visual Studio
File
1
Data
Tools
Test
Analyze
Window
Help
Subversion repository...
New
2
Open
Close
Close Solution
Save Selected Items
Ctrl+S
Save All
Export Templat
Page Setup...
Print...
Recent Files
Recent Projects
Exit
Project...
3
Shift+Alt+N
File...
Project From Existing Code...
Solution Explorer
New Project
Project types:
Templates:
.NET Framework 3.5
4
Visual C#
5
Windows
Web
Smart Device
Office
Database
Reporting
Test
WCF
Workflow
Database Projects
Distributed Systems
Other Project Types
Test Projects
Visual Studio installed templates
Windows Forms Application
Console Application
6
WPF User Control Library
Class Library
WPF Browser Application
Empty Project
WPF Custom Control Library
Windows Forms Control Library
My Templates
Search Online Templates...
A project for creating a command-line application (.NET Framework 3.5)
Name:
7
ConsoleApplication1
8
OK
Cancel
Output
Ready


Classi - Microsoft Visual Studio
File Edit View Project Build Debug Data Tools Test Analyze Window Help
Debug
Any CPU
Server Explorer
Toolbox
Program.cs
Classi.Program
Main(string[] args)
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5
6 namespace Classi
7 {
8 class Program
9 {
10 static void Main (string[] args)
11 {
12 }
13 }
14 }
15
Solution Explorer - Solution 'Classi' (1 proje...
Solution 'Classi' (1 project)
Classi
Properties
References
Program.cs
Solutio...
Source ...
Propert...
Class V...
Output
Error List
Task List
Test Results
Undo Close
Ready


Classi - Microsoft Visual Studio
File Edit View Project Debug Data Tools Test Analyze Window Help
1
Add Windows Form...
Add User Control...
Add Component...
Add Class...
2
Add New Item... Ctrl+Shift+A
Add Existing Item... Shift+Alt+A
New Folder
Any CPU
Program.cs
Solution Explorer - Classi
Solution 'Classi' (1 project)
Classi
Properties
References
Add New Item - Classi
Categories:
Templates:
Code
3
Class
4
Code File
Interface
General
Web
Windows Forms
WPF
Reporting
Workflow
My Templates
Search Online Templates...
An empty class definition
Name:
5
Class1.cs
6
Add
Cancel
Output
Ready


Classi - Microsoft Visual Studio
File Edit View Refactor Project Build Debug Data Tools Test Analyze Window Help
Debug
Any CPU
Person.cs
Program.cs
Classi.Person
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
namespace Classi
{
class Person
{
}
}
Server Explorer
Toolbox
Solution Explorer - Solution 'Classi' (1 proje...
Solution 'Classi' (1 project)
Classi
Properties
References
Person.cs
Program.cs
Solutio...
Source ...
Propert...
Class V...
Output
Error List
Task List
Test Results
Undo Close
Ready
Ln 1
Col 1
Ch 1
INS


Classi - Microsoft Visual Studio
File Edit View Refactor Project Build Debug Data Tools Test Analyze Window Help
Debug
Any CPU
Person.cs
Program.cs*
Classi.Program
Main(string[] args)
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
namespace Classi
{
class Program
{
static void Main (string[] args)
{
Person p = new Person ();
}
}
}
Solution Explorer - Solution 'Classi' (1 proje...
Solution 'Classi' (1 project)
Classi
Properties
References
Person.cs
Program.cs
Server Explorer
Toolbox
Output
Error List
Task List
Test Results
Undo Close
Ready
Ln 12
Col 38
Ch 38

Esercizio

- Creare una classe Calculator
  - Deve avere 4 metodi
    - Somma, Sottrai, Dividi (attenti alla divisione per zero!!), Moltiplica
    - Non accettano argomenti. I valori sono recuperati dalle proprietà X e Y
    - Restituiscono un valore double
  - Deve avere 2 property
    - X, Y
  - Deve avere un costruttore di default (public Calculator () {})
- Utilizzare questa classe nel programma dell'altra volta