



Automatisch generierte Dokumentation

Spezialkommentare (ähnlich javadoc)



Beispiel

```
/// ... comment ...
class C {
    /// ... comment ...
    public int f;

    /// ... comment ...
    public void Foo() {...}
}
```

Übersetzung `csc /doc:MyFile.xml MyFile.cs`

- *Prüft Kommentare auf Vollständigkeit und Konsistenz*
z.B. wenn Parameter dokumentiert werden, müssen alle Parameter dokumentiert werden;
Namen von Programmelementen müssen korrekt geschrieben werden.
- *Expandiert Querverweise in qualifizierte Namen*
z.B. C => T:C, f => F:C.f, Foo => M:C.Foo
- *Erzeugt XML-Datei mit kommentierten Programmelementen*
XML-Datei kann mit XSL für Browser formatiert werden

Beispiel für Quellprogramm



```
/// <summary> A counter for accumulating values and computing the mean value.</summary>
class Counter {
    /// <summary>The accumulated values</summary>
    private int value;

    /// <summary>The number of added values</summary>
    public int n;

    /// <summary>Adds a value to the counter</summary>
    /// <param name="x">The value to be added</param>
    public void Add(int x) {
        value += x; n++;
    }

    /// <summary>Returns the mean value of all accumulated values</summary>
    /// <returns>The mean value, i.e. <see cref="value"/> / <see cref="n"/></returns>
    public float Mean() {
        return (float)value / n;
    }
}
```

Daraus erzeugte XML-Datei



```
<?xml version="1.0"?>
<doc>
  <assembly>
    <name>MyFile</name>
  </assembly>
  <members>
    <member name="T:Counter">
      <summary> A counter for accumulating values and computing the mean value.</summary>
    </member>
    <member name="F:Counter.value">
      <summary>The accumulated values</summary>
    </member>
    <member name="F:Counter.n">
      <summary>The number of added values</summary>
    </member>
    <member name="M:Counter.Add(System.Int32)">
      <summary>Adds a value to the counter</summary>
      <param name="x">The value to be added</param>
    </member>
    <member name="M:Counter.Mean">
      <summary>Returns the mean value of all accumulated values</summary>
      <returns>The mean value, i.e. <see cref="F:Counter.value"/> / <see cref="F:Counter.n"/></returns>
    </member>
  </members>
</doc>
```

XML-Datei kann mit Visual Studio als HTML angezeigt werden.

- Elemente sind nicht hierarchisch geschachtelt
- Typen von Feldern und Methoden werden nicht mitgespeichert

XML-Tags



Vordefinierte Tags

Alleinstehend

- `<summary>` Kurzbeschreibung eines Programmelements `</summary>`
- `<remarks>` Ausführliche Beschreibung eines Programmelements `</remarks>`
- `<example>` Beliebiger Beispieltext (z.B. Aufrufbeispiel) `</example>`
- `<param name="ParamName">` Bedeutung des Parameters `</param>`
- `<returns>` Bedeutung des Rückgabewerts `</returns>`
- `<exception [cref="ExceptionType]>` Bei der Dokumentation einer Methode: Beschreibung der Exception `</exception>`

Teil einer anderen Beschreibung

- `<code>` Mehrzeilige Codestücke `</code>`
- `<c>` kurze Codestücke im Text `</c>`
- `<see cref="ProgramElement">` Name des Querverweises `</see>`
- `<paramref name="ParamName">` Name des Parameters `</paramref>`

Benutzerdefinierte Tags

Beliebige eigene XML-Tags, z.B. `<author>`, `<version>`, ...