

Richtlinien zur Quellcode Dokumentierung

Klassen, Strukturen, Interfaces, Methoden und Attribute sollen mit der Xml-Dokumentierungsmethode beschrieben werden. Die Kommentare sollten dabei auf Deutsch sein.

Innerhalb von Methoden, können und sollen normale Kommentare sinnvoll verwendet werden. Code Snippets aus dem Internet sollten mit Kommentar und Quellenangabe(Link) vermerkt werden.

Verwendung der C# (Vs) Xml Dokumentation

C# bietet im Vergleich zu den normal Kommentaren (//, /*), die es auch in anderen Sprachen gibt, die Möglichkeit für eine Xml basierte Dokumentierung(///), die beim Kompiliervorgang automatisch in eine Xml-Dokumentationsdatei geschrieben wird. Desweiteren werden diese Kommentare auch von IntelliSense (Visual Studio Parser) in den Parse-Vorgang mit eingebunden und zeigt diese beim verwenden von Objekten an.

Es stehen folgende Tags zur Verfügung:

Die Wichtigsten

/// <summary></summary>

Beschreibt die allgemeine Funktion der Klasse, Methode, des Attributs usw.

/// <param name=""></param>

Nennt und Beschreibt einen Parameter einer Methode.

/// <returns></returns>

Beschreibt den Wert der von einer Methode zurückgegeben wird.

/// <exception cref=""></exception>

Die dort angegebene Exception kann von der Funktion geworfen werden.

Außerdem kann dort beschrieben werden, wann die Exception geworfen wird.

Zusätzliche Tags dessen Verwendung nicht zwingend erforderlich ist.

/// <remarks></remarks>

Kann für zusätzliche Informationen neben Summary verwendet werden.

/// <value></value>

Kann zum Beschreiben von Attributwerten verwendet werden.

/// <example></example>

Kann verwendet werden um zu zeigen, wie eine Methode aufgerufen werden soll(muss).

```
/// <code></code>
```

Kann verwendet werden um Code Beispiele zu zeigen, die über mehrere Zeilen gehen.

```
/// <seealso cref=""></seealso>
```

Kann verwendet werden, um auf andere C# Klassen zu verweisen.

Beispiele zur Verwendung.

Verwendet werden die Kommentare mit Drei / (///) vor der Deklaration von Methoden, Klassen etc. Dabei empfiehlt es sich, erst die Komplette Definition der Objekte zu definieren, da Visual Studio dann beim Einfügen von /// automatisch alle benötigten Xml Tags definiert.

```
/// <summary>
/// Lädt eine StringRessource anhand ihres Namens.
/// </summary>
/// <param name="name">Der Name der StringRessource mit Qualifier @Name.</param>
/// <returns>Die StringRessource oder <see cref="string.Empty"> wenn die Ressource nicht gefunden wurde</see></returns>
internal string LoadString( string name )
```

```
/// <summary>
/// Schreibt eine Info Nachricht in die LogDatei.
/// </summary>
/// <param name="message">Die Nachricht die geschrieben werden soll.</param>
/// <param name="className">Der Name der Klasse für die etwas dokumentiert werden soll.</param>
/// <param name="methodName">Der Name der Methode in der etwas dokumentiert werden soll.</param>
/// <exception cref="IOException">Wird geworfen wenn die Datei nicht geöffnet und beschrieben werden kann.</exception>
/// <exception cref="Exception"> Wird bei unvorhergesehenen Ereignissen geworfen.</exception>
public static void WriteInfo( string message, string className, string methodName )
```

```
/// <summary>
/// Der Pfad an dem die StringRessourcen liegen.
/// </summary>
private string Path { get; set; }
```

