Table of Contents

Klassen 1
Interfaces 3
Unit Tests 4

Alle unsere C# Klassen verwenden einen einheitlichen Style.

Klassen

Für Klassen verwenden wir folgenden Style:

ClassTemplate.cs

```
/*
 * Copyright 2018 SIB Visions GmbH
 *
 * Licensed under the Apache License, Version 2.0 (the "License"); you
may not
 * use this file except in compliance with the License. You may obtain
a copy of
 * the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT
 * WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See
the
 * License for the specific language governing permissions and
limitations under
 * the License.
 *
 *
 * History
 *
 * dd.MM.yyyy - [XX] - creation
 */

namespace com.sibvisions.foo
{
    /// <summary>This is the base bar class.</summary>
    /// <author>First Last</author>
    public class Bar
    {
        #region Fields

        /// <summary>the property name for the serializer.</summary>
        public const int TYPE_FOO = 1;

        /// <summary>the value of foo bar.</summary>
        protected Object oValue = null;

        #endregion
```

```
        #region Constructors

        /// <summary>Creates a new instance of Foo</summary>
        /// <param name="pName">the simple name.</param>
        public Foo(String pName)
        {
        }

        #endregion

        #region Properties

        /// <summary>Returns the value object.</summary>
        /// <returns>the value</returns>
        public Object Value
        {
            get
            {
                return oValue;
            }
        }

        #endregion

        #region Interface implementation

        #endregion

        #region Methods

        #endregion

    }    // Foo

    /// <summary>The FooBar handles everything.</summary>
    /// <author>First Last</author>
    sealed class FooBar
    {
        #region Fields

        #endregion

        #region Constructor

        #endregion

    }    // FooBar

}
```

Folgende Regeln werden durch diese Vorlage definiert:

- Variablendeklaration zu Beginn (zuerst Konstante, danach veränderbare Variablen)
- danach Konstrukten und Initialisierungsmethoden
- danach die Definition von Properties
- danach die Implementierungen von Interface Methoden
- danach alle überschriebenen Methoden (gekennzeichnet mit @Override)
- danach alle Methoden der Klasse
- Sub/Inner Klassen am Ende

- Jeder Parameter einer Methode wird mit dem Prefix “p” gekennzeichnet
- Für Instanz Variablen wird ebenfalls ein Prefix verwendet wie z.B.:

```
String sValue = "bar";
```

- Im Header erstellen wir bei wichtigen Änderungen einen Hinweis mit Zeitstempel und Autor
- Dokumentation für die Klassen Deklaration, ALLE Methoden, Properties und Instanz Variablen bzw. Konstante

Interfaces

Folgende Regeln werden durch diese Vorlage definiert:

InterfaceTemplate.cs

```
/*
 * Copyright 2018 SIB Visions GmbH
 *
 * Licensed under the Apache License, Version 2.0 (the "License"); you
may not
 * use this file except in compliance with the License. You may obtain
a copy of
 * the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT
 * WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See
the
 * License for the specific language governing permissions and
limitations under
 * the License.
 *
 *
 * History
 *
 * dd.MM.yyyy - [XX] - creation
 */
```

```
namespace com.sibvisions.foo
{
    /// <summary>This is the base bar interface</summary>
    /// <author>First Last</author>
    public interface IBar
    {
        #region Method definitions

        /// <summary>Gets the value.</summary>
        /// <returns>the value.</returns>
        public Object getValue();

        #endregion

    }    // IBar
}
```

The following rules are defined by this template:

- Interface Methoden werden zu Beginn definiert
- Sub/Inner Interfaces am Ende

- Jedes Interface beginnt mit “I”
- Im Header erstellen wir bei wichtigen Änderungen einen Hinweis mit Zeitstempel und Autor
- Dokumentation für die Interface Deklaration und ALLE Methoden

Unit Tests

Durch den Einsatz von Unit Tests stellen wir sicher das die Basisfunktionalitäten wie erwartet funktionieren. Ein Unit Test kann niemals die komplette Funktionalität in allen erdenkbaren Konstellationen testen, doch ohne Unit Tests können die Qualitätsanforderungen nicht erfüllt werden. Wir setzen daher ein funktionierendes Set an Unit Tests voraus.

Die Unit Tests werden getrennt vom Core Source Code gespeichert:

```
<jvxnet>/trunk/net/JVxWin/src/com/sibvisions/foo
<jvxnet>/trunk/net/JVxWin/test/com/sibvisions/foo
```

Als Testing Framework kommt NUnit zum Einsatz.

Für Unit Tests verwenden wir folgenden Style:

TestTemplate.cs

```
/*
 * Copyright 2010 SIB Visions GmbH
 *
 * Licensed under the Apache License, Version 2.0 (the "License"); you
```

```
may not
 * use this file except in compliance with the License. You may obtain
a copy of
 * the License at
 *
 * http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT
 * WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See
the
 * License for the specific language governing permissions and
limitations under
 * the License.
 *
 *
 * History
 *
 * dd.MM.yyyy - [XX] - creation
 */

namespace com.sibvisions.foo
{
    /// <summary>Tests the functionality of ...</summary>
    /// <author>First Last</author>
    [TestFixture]
    public class TestBar
    {
        #region Fields

        #endregion

        #region Initialization

        ///<summary>Initializes the unit test.</summary>
        [SetUp]
        public void SetUp()
        {
        }

        ///<summary>Sets values before each test.</summary>
        [TestFixtureSetUp]
        public virtual void FixtureSetUp()
        {
        }

        #endregion

        #region Test Methods
```

```
        /// <summary>Tests the ... method.</summary>
        [Test]
        public void testGet()
        {
        }

        #endregion

    }    // TestBar
}
```

Folgende Regeln werden durch diese Vorlage definiert:

- Variablendeklaration zu Beginn (zuerst Konstante, danach veränderbare Variablen)
- danach Methoden für die Test Initialisierung
- danach alle Test Methoden (gekennzeichnet mit [Test])

- Jede Test Klasse beginnt mit “Test”
- Jede Test Methode beginnt mit “test”
- Im Header erstellen wir bei wichtigen Änderungen einen Hinweis mit Zeitstempel und Autor
- Dokumentation für die Klassen Deklaration, ALLE Methoden, Properties und Instanz Variablen bzw. Konstante

From:
http://doc.sibvisions.com/ - **Documentation**

Permanent link:
http://doc.sibvisions.com/de/jvx/join/style_csharp

Last update: **2018/02/06 09:38**