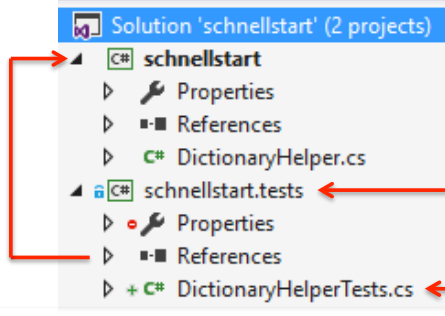


Schnellstart: Unit Tests mit MSTest

Grundlagen



The screenshot shows a solution named 'schnellstart' containing two projects: 'schnellstart' and 'schnellstart.tests'. The 'schnellstart' project contains 'DictionaryHelper.cs'. The 'schnellstart.tests' project contains 'DictionaryHelperTests.cs'. Red arrows point from text labels to these elements.

1. Projekt hinzufügen
(Unit Test Project / Komponententest)
Testprojekt pro Projekt

2. Referenz hinzufügen

3. Klasse hinzufügen
Testklasse pro Klasse

```
using Microsoft.VisualStudio.TestTools.UnitTesting;

namespace schnellstart.tests
{
    [TestClass]
    public class DictionaryHelperTests
    {
        [TestMethod]
        public void Integrationstest() {
            var sut = new DictionaryHelper();
            var result = sut.ToDictionary("a=1;b=2;c=3");
            Assert.AreEqual(3, result.Count);
        }
    }
}
```

MSTest Namespace

Entspricht dem Projektnamen

“Klasse enthält Tests”

Klassenname + “Tests”

“Dies ist eine Testmethode”

Name des Tests

“System unter Test”

1. Arrange

2. Act

3. Assert

Tests ausführen

ReSharper:

Ausführen: Im Menü unter *ReSharper - Unit Tests - Run Unit Tests*

Anzeigen: Im Menü unter *ReSharper - Windows - Unit Tests*

Visual Studio:

Ausführen: Im Menü unter *Test - Run - All Tests*

Anzeigen: Im Menü unter *Test - Windows - Test Explorer*

Assert Beispiele

Vergleiche

```
Assert.AreEqual(y, x);  
Assert.AreSame(y, x);  
Assert.AreNotEqual(y, x);  
Assert.AreNotSame(y, x);
```

Null Prüfung

```
Assert.IsNull(x);  
Assert.IsNotNull(x);
```

Strings

```
StringAssert.StartsWith(s, "abc");  
StringAssert.EndsWith(s, "abc");  
StringAssert.Contains(s, "abc");  
StringAssert.Matches(s, new Regex("[a-zA-Z]"));  
StringAssert.DoesNotMatch(s, new Regex("[a-zA-Z]"));
```

Exceptions

```
[ExpectedException(typeof(ArgumentOutOfRangeException))]  
[ExpectedException(typeof(Exception), AllowDerivedTypes = true)]
```

Aufzählungen (Array, List, IEnumerale, etc.)

```
CollectionAssert.AreEqual(y, x);  
CollectionAssert.AreNotEqual(y, x);  
CollectionAssert.Equivalent(y, x);  
CollectionAssert.AreNotEquivalent(y, x);  
CollectionAssert.AllItemsAreUnique(x);  
CollectionAssert.Contains(x, 5);
```

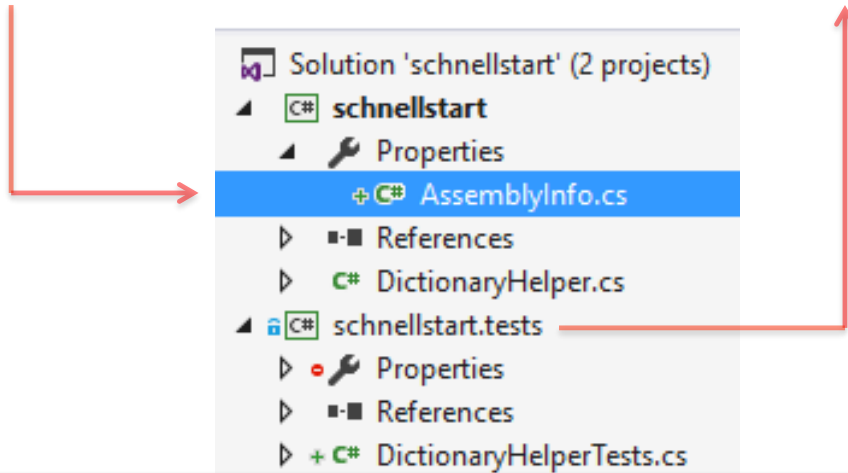
Boolesche Ausdrücke

```
Assert.IsTrue(x);  
Assert.IsFalse(x);
```

Interna testen (Whitebox)

In AssemblyInfo.cs ergänzen:

```
[assembly: InternalsVisibleTo("schnellstart.tests")]
```



```
public class DictionaryHelper
{
    ...
    internal IEnumerable<string> SplitIntoSettings(string configuration) {
        return configuration.Split(new [] {";"},
            StringSplitOptions.RemoveEmptyEntries);
    }
    ...
}
```

```
[TestClass]
public class DictionaryHelperTests
{
    [TestMethod]
    public void Empty_setting_is_ignored() {
        var sut = new DictionaryHelper();
        var result = sut.SplitIntoSettings("a;b");
        CollectionAssert.AreEqual(new[] {"a", "b"}, result);
    }
}
```

Interne Methode testen