Software-Entwicklung mit C#

Komponententests (Unit-Tests)

Immo Köster

kr@gso-koeln.de

www.gso-koeln.de

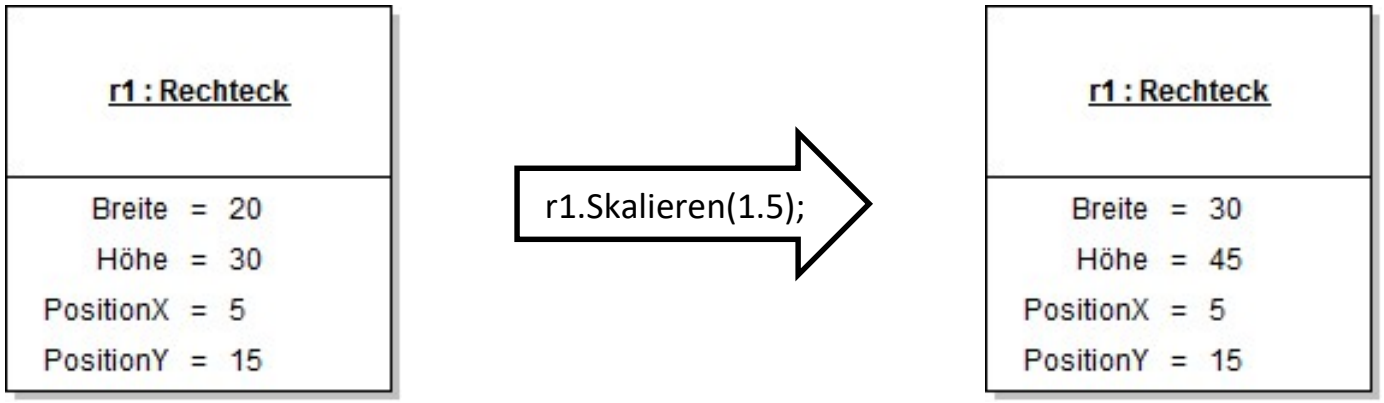
Komponententests (Unit-Tests)

- dienen zum Testen einzelner Klassen und deren Methoden
- Empfehlung: Bestandteil des Entwicklungsprozesses
  ➔ Test Driven Development (TDD)
- seit Visual Studio 2012 auch in Express-Version unterstützt
  - Microsoft Test-Explorer (vorher nur über Drittanbieter, z.B. NTest)
- Projektvorlage „Komponententestprojekt“
  - Verweis Microsoft.VisualStudio.QualityTools.UnitTestFramework
  - Namensraum Microsoft.VisualStudio.TestTools.UnitTesting
- Namenskonventionen:
  - Projekt: <zu testendes Projekt>Tests, z.B. LagerverwaltungTests
  - Klasse: <zu testende Klasse>Tests, z.B. LagerTests
  - Methode: <zu testende Methode>_Informationen_zum_Verhalten
    z.B. WarenEinkaufen_erhoeht_Bestand()

Testmethoden implementieren

- Muster „Arrange, Act, Assert“ (AAA)
  - **Arrange** (Vorbereitung)
    z.B. Objekte initialisieren, Testdaten (Parameter) erstellen
  - **Act** (Aktion)
    Aufruf der zu testenden Methode mit den Test-Parametern
  - **Assert** (Sicherstellung/Prüfung/Bestätigung)
    Verhalten/Ergebnis der Methode überprüfen
    - **Positivtest** (Standardfall)
      Ablauf der Methode im Normalfall z.B. anhand des Rückgabewerts oder der Auswirkungen auf die Daten prüfen.
    - **Negativtest** (Fehlerfall)
      Sicherstellen, dass fehlerhafte/ungültige Testdaten oder Konstellationen angemessen behandelt werden.

Beispiel Positivtest



Beispiel Positivtest

Attribut TestMethod

keine Rückgabe

keine Parameter

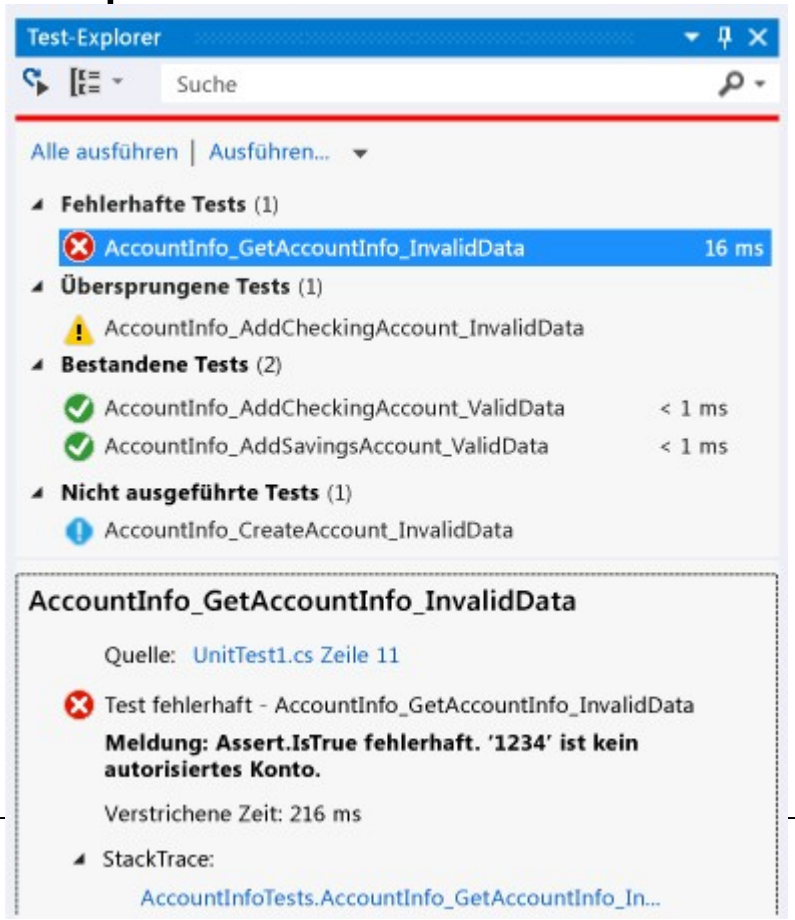
```
[TestMethod]
public void Skalieren_AendertBreiteUndHoehe()
{
    // arrange
    Rechteck r1 = new Rechteck();
    r1.Breite = 20;
    r1.Hoehe = 30;

    // act
    r1.Skalieren(1.5);

    // assert
    Assert.AreEqual(30, r1.Breite);
    Assert.AreEqual(45, r1.Hoehe);
}
```

Tests ausführen

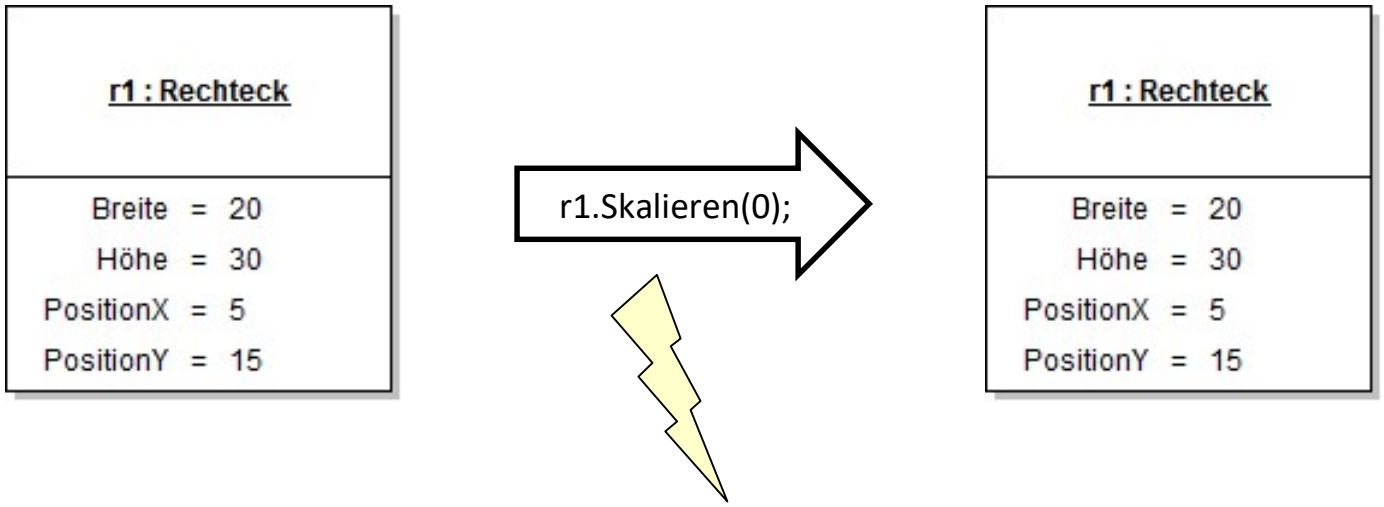
- Menü TEST → Fenster → Test-Explorer



Richtlinien für Komponententests

- jeder Test prüft nur **genau einen** Aspekt
- keine Abhängigkeiten zu anderen Tests
  ➔ Ausführung der Tests in zufälliger Reihenfolge!
- Test = Dokumentation des Codes, Verwendungsbeispiel

Beispiel Negativtest



Beispiel Negativtest

Erwartet eine Ausnahme…

…vom Typ…

…Argument außerhalb des gültigen Bereichs

```
[Test…thod]
[ExpectedException(typeof(ArgumentOutOfRangeException))]
public void Skalieren_FaktorNull_ErzeugtAusnahme()
{
    // arrange
    Rechteck r1 = new Rechteck();
    r1.Breite = 20;
    r1.Hoehe = 30;

    // act
    r1.Skalieren(0);

    // assert
    Assert.AreEqual(20, r1.Breite);
    Assert.AreEqual(30, r1.Hoehe);
}
```