

Akka.NET

Aufgabe 6: Testen von Akteuren

TestKit: System

- TestKit:
Basisklasse
- Sys:
ActorSystem
(jedesmal neu)

```
using Akka.TestKit;
using Akka.TestKit.NUnit;
using NUnit.Framework;
using Akka.Actor;
using GuessMyNumber.actors;

namespace GuessMyNumber.Tests
{
    [TestFixture]
    public class EmptyTest : TestKit
    {
        private IActorRef chooser;

        [SetUp]
        public void Setup()
        {
            chooser = Sys.ActorOf(Props.Create<Chooser>(), "chooser");
        }

        [Test]
        public void UnitOfWork_StateUnderTest_ExpectedBehaviour()
        {
            // Arrange

            // Act

            // Assert
        }
    }
}
```

Testkit: Messages

- ❑ `ExpectMsg<>()`
`ExpectNoMsg()`
- ❑ `TestActor` ist Absender
- ❑ `BlackHoleActor` schluckt alles

```
[Test]
public void Chooser_OnStart_RepliesStarted()
{
    // Act
    chooser.Tell(new Start());

    // Assert
    ExpectMsg<Started>();
}
```


Testkit: Blick ins Innere

- ❑ TestActorRef<> – Blick „in“ den Akteur
- ❑ Zugriff auf Instanz des Akteur Objekts

```
[Setup]
public void Setup()
{
    fakeRandom = new FakeRandom { FakeNumber = 42 };
    chooser = ActorOfAsTestActorRef<Chooser>(
        Props.Create<Chooser>(fakeRandom),
        "chooser"
    );
}

[Test, TestCase(46), TestCase(97)]
public void Chooser_OnStart_ChoosesNumber(int secret)
{
    // Arrange
    fakeRandom.FakeNumber = secret;

    // Act
    chooser.Tell(new Start());

    // Assert
    Assert.AreEqual(secret, chooser.UnderlyingActor.mySecretNumber);
}
```

GuessMyNumber.Tests

- Die Tests basieren auf NUnit
- beim „ChooserTest“ fehlen Tests

```
[Test]
public void Chooser_OnTestTryTooSmall_RepliesTryTooSmall()
{
    // Arrange
    // TODO: eine Zufallszahl setzen

    // Act
    // TODO: Nachricht "TestTry" mit kleinerer Zahl versenden

    // Assert
    // TODO: gab es die passende Antwort?
}
```


Lösungsvorschlag

- zu kleine Zahl probieren
- Auf „TryTooSmall“ Antwort warten

```
[Test]
public void Chooser_OnTestTryTooSmall_RepliesTryTooSmall()
{
    // Arrange

    // Act
    chooser.Tell(new TestTry(40));

    // Assert
    ExpectMsg<TryTooSmall>();
}
```