# Mock Objects

# Test Doubles

- Dummy
- Fake
- Stub
- Spy
- Mock

# Mockito

# Life Cycle

- Stub   &larr;   *Arrange*
- Test   &larr;   *Act*
- Verify   &larr;   *Assert*

# Import

```
import static org.mockito.Mockito.*;
```

# Create Mock Object

```
List mockedList = mock(List.class);
```

Stub
Mock

# Setup Mock Object: Stubbing (1)

- No **static** or **final** methods!

- Methods **with** return value:

```
when(mockedList.get(0)).thenReturn("first");
when(mockedList.get(0)).thenThrow(
        new IllegalStateException());
when(mockedList.get(0)).thenAnswer(
        (invocation)->"first");
```

*Code wird ausgeführt !*

# Setup Mock Object: Stubbing (2)

- No **static** or **final** methods!
- Methods that return **void**:

```
doThrow(new RuntimeException())
      .when(mockedList)
      .clear();
```

# Verify calls on mocks

*mock object*

*methoden aufruf*

```
verify(mockedList).add("one");
verify(mockedList, times(2)).add("two");
verify(mockedList, never()).add("three");
```

# Argument Matchers

```
when(mockedList.get(eq(3)).thenReturn(…);
verify(mockedList).get(anyInt());
verify(mockedList).add(argThat(
    someString -> someString.length() > 5
));
```

# Capturing Arguments

```
ArgumentCaptor<Person> argument =
    ArgumentCaptor.forClass(Person.class);
verify(mock).doSomething(argument.capture());

assertEquals(
    "John",
    argument.getValue().getName());
```

*methode*

# …and there's more

- Automatic mocking
- Injecting mocks
- Adding user defined behaviour
- …