CardGameApp - Testing

Gruppe International

Felix Ossmann

CardGameApp - Testing

Überblick

- Einführung
- Unit Testing
- Best Practices
- Testarten
- Test Beispiele

CardGameApp - Testing
Einführung

- *“The bitterness of poor quality remains long after the sweetness of meeting the schedule has been forgotten.”* - Karl Wiegers

- *“Never allow the same bug to bite you twice.”* - Steve Maguire

CardGameApp - Testing

Software testing

- wesentlicher Bestandteil der Softwareentwicklung
- macht Code robuster

- **Unit Test** (Modultest) -> kleine Einheiten einzeln testen
  - Ziel: Code isolieren und auf Funktion überprüfen

CardGameApp - Testing
Unit Testing

+ frühzeitiges Erkennen von Fehlern, leichteres Debugging,
- hoher Zeitaufwand, erkennt Fehler bei Integration nicht

- JUnit hat sich als Standard Framework etabliert

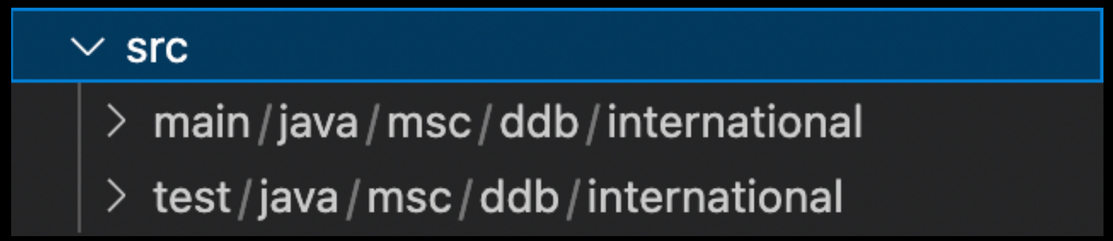
```
import static org.junit.Assert.*;

import org.junit.Test;
```

CardGameApp - Testing

Best Practices

- Source Code
  - Test Klassen vom Source Code separieren
  - Gleiche Verzeichnisstruktur anlegen



CardGameApp - Testing

Best Practices

- Kennzeichnung durch die Annotation @Test
- eindeutige und selbsterklärende Testnamen
- sollten möglichst spezifisch sein

```
13      @Test
14      public void ConstructorTest() {
15
16          assertTrue(blackjacktest instanceof BlackJack);
17      }
```

CardGameApp - Testing

Assertions

- Tests bestehen aus **Assertion**, **expected result** und **actual result**

- Die Methode *assertEquals* überprüft ob expected und actual result übereinstimmen. Trifft dies nicht zu wird der Test mit einem AssertionError abgebrochen

```
42      @Test
43      public void testGetMaximumPlayers() {
44          assertEquals(expected: 10, blackjacktest.getMaximumPlayers());
45      }
```

CardGameApp - Testing
Expected Exception

- Testet die Übergabe ungültiger Parameter

```
9   @Test(expected = MaximumPlayersBeyondAllowedMaximumException.class)
10  public void tryToSetMaximumPlayersBeyondAllowedMaximum() throws MaximumPlayersBelowAllowedMinimumException, MaximumPlayersBeyondAllowedMaximumException {
11      BlackJack blackJack;
12      try {
13          blackJack = new BlackJack();
14          blackJack.setMaximumPlayers(maximumPlayers: 18);
15      } catch (TooManyPlayersException e) {
16          e.printStackTrace();
17      }
18  }
```

CardGameApp - Testing

Test Beispiele

```
30      @Test
31      public void testGetMinimumPlayers() {
32          assertEquals(expected: 2, blackjacktest.getMinimumPlayers());
33      }
34
```

```
65      @Test
66      public void testgetValueForCard() {
67
68          assertEquals(expected: 10, blackjacktest.getValueForCard(card: "King"));
69      }
70
```

CardGameApp - Testing

Test Beispiele

```
11      @Test
12      public void testGetName() {
13          assertEquals(expected: "Dealer", dealerTest.getName());
14
15  }
```

```
17      @Test
18      public void testsetSuit() {
19          Card card = new Card(rank: "Queen", suit: "Hearts");
20          assertEquals(expected: "Hearts", card.getSuit());
21      }
22  }
23
```

Vielen Dank! :)