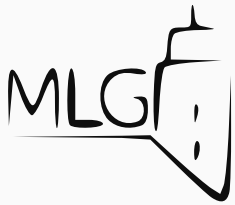


Einstieg in Testkonzepte für Java

Adrian Schrader

29. Mai 2016

Informatik 2h, Herr Endstrasser

Inhaltsübersicht

1. Unit Tests

2. Test Driven Development (TDD)

3. Beispiel: „17 und 4“

Unit Tests

Definition

Modul (Unit)

- Funktionale, isolierbare Einheit
- Meist einzelne Klassen

Modultest (Unit Test)

- sichert genau *eine* sichtbare Eigenschaft
- bezieht sich auf *eine* isolierte Komponente
- vollständig automatisierter Code (Testing Framework)

Merkmale

Unit Tests sind...

- leicht verständlich
- voneinander unabhängig
- hochwertig und gewartet
- „gemein“
- schnell

Einsatz

Vorteile

- Zuverlässiger Code
- Überprüfbare Spezifikation
- Rückversicherung des Entwicklers
- Team-Management

Nachteile

- garantiert kein Zusammenspiel von Komponenten
- garantiert keine Performance
- „doppelter“ Code

Implementierung

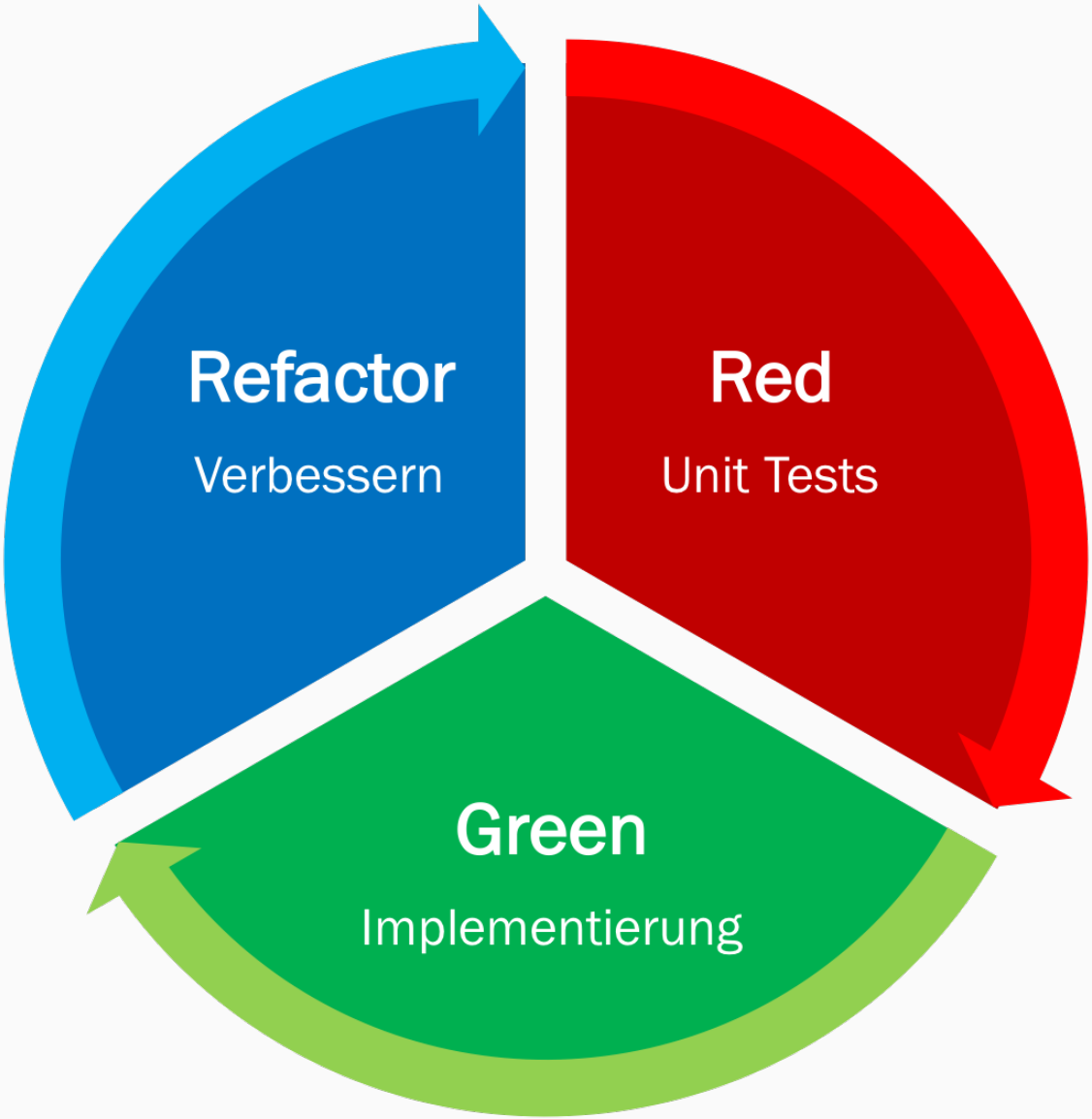
```java
import org.junit.Test;
import org.junit.Assert.*;

public class MatchBoxTest {
  public MatchBoxTest() {}

  @Test
  public void drawNotMoreThanThree() {
    MatchBox matchbox1 = new MatchBox(25);
    matchbox1.draw(4);
    assertEquals(22, matchbox1.count());
  }
}
```

Test Driven Development (TDD)

Konzept


Refactor
Verbessern
Red
Unit Tests
Green
Implementierung

Beispiel: „17 und 4“

Klassenentwurf

```
1  public class GameMaster {
2    public GameMaster(Dice dice, Class<?>[] types) { }
3    public Result playRound(int goalScore) { }
4    ...
5  }
6
7  public class GameMasterTest {
8    public GameMasterTest() { }
9    ...
10 }
```

Entwurf der Testmethode

```java
import org.junit.Test;
import org.junit.Assert.*;

public class MatchBoxTest {
  public MatchBoxTest() {}

  @Test
  public void drawNotMoreThanThree() {
    MatchBox matchbox1 = new MatchBox(25);
    matchbox1.draw(4);
    assertEquals(22, matchbox1.count());
  }
}
```

Entwurf der Testfälle

```
1  public class GameMaster {
2    public GameMaster(Dice dice, Class<?>[] types) { }
3    public Result playRound(int goalScore) { }
4    ...
5  }
6
7  public class GameMasterTest {
8    public GameMasterTest() { }
9    ...
10 }
```

```java
@Test
public void playRoundTakesTurns() throws Exception {
    GameMaster master = generateGameMaster();
    master.playRound(21);

    for (Player player : master.getPlayers()) {
        assertThat(player.getScore()).isGreaterThan(0);
    }
}
```

```java
@Test
public void playRoundSortsPlayersByScore() throws Exception {
    GameMaster master = generateGameMaster();
    PlayerScoreComparator comparator = new PlayerScoreComparator(21);
    master.playRound(21);

    assertThat(master.getPlayers())
        .isSortedAccordingTo(comparator);
}
```

```java
@Test
public void playRoundGivesResult() throws Exception {
    GameMaster master = generateGameMaster();
    Result actualResult = master.playRound(21),
          expectedResult = new Result(21, master.getPlayers());

    assertThat(actualResult).isEqualTo(expectedResult);
}
```

Implementierung der Methode

```java
/**
 * Starts one round in which every player gets one turn and
 * determines the winner.
 * @param goalScore score the players have to get close to
 * @return outcome of the round (may be a tie)
 */
public Result playRound(int goalScore) {
  // Every player gets one turn
  for (Player player : this.players) {
      player.resetScore();
      player.play(this.dice, goalScore);
  }

  // Sorting the players by their score
  this.players.sort(new PlayerScoreComparator(goalScore));

  // Determine the result and return it
  return new Result(goalScore, this.players);
}
```

Literatur


Unit-Test. URL: http://www.thur.de/philo/notizen/Kernspinresonanz.pdf (besucht am 29. 05. 2016).