

# Tools für dein SEP Testing mit JUnit 5

IEEE  
Ben Borgards

28. Juli 2022

- ▶ Unit Tests sind ...
  - ▶ die Grundlage einer gut getesteten Anwendung
  - ▶ relativ schnell in der Ausführung
  - ▶ auf kleine, isolierte Teile der Anwendung fokussiert
- ▶ Problem: Interaktion zwischen verschiedenen Klassen
- ▶ Lösung: Mocking/Stubbing/...
  - ▶ Abhängigkeiten zwischen Klassen werden durch Mock o. Stub Objekte ersetzt
  - ▶ Diese bilden simple Funktionalität der benötigten Klassen nach

# JUnitTestCaseExample

```
public class JUnitTestCaseExample {  
    private List<String> students = new ArrayList();  
  
    public void add(String name) {  
        students.add(name);  
    }  
  
    public int sizeOfStudent() {  
        return students.size();  
    }  
}
```

```
public class TestJUnitTestCaseExample {  
  
    JUnitTestCaseExample obj = new JUnitTestCaseExample();  
  
    @Test  
    public void testAdd() {  
        obj.add("Emma");  
        obj.add("Paul");  
        assertEquals(2, obj.sizeOfStudent());  
    }  
}
```

```
public class TestJUnitTestCaseExample {  
  
    JUnitTestCaseExample obj = new JUnitTestCaseExample();  
  
    @Test  
    public void testAdd() {  
        obj.add("Emma");  
        List students = obj.getStudents();  
        assertThat(list).isNotNull().isNotEmpty();  
        assertThat(userList).contains("Emma")  
    }  
}
```

```
public class TestJUnitTestCaseExample {  
  
    JUnitTestCaseExample obj = new JUnitTestCaseExample();  
  
    @Test  
    public void testAdd() {  
        obj.add("Emma");  
        List students = obj.getStudents();  
        assertThat(userList).contains("Emma")  
    }  
}
```

```
@Test
public void whenNotUseMockAnnotation_thenCorrect() {
    List mockList = Mockito.mock(ArrayList.class);

    Mockito.when(mockList.size()).thenReturn(100);
    assertEquals(100, mockList.size());
}
```

# Selenium

```
public class FirstScriptTest {
    public WebDriver driver;

    @Test
    public void eightComponents() {
        driver = new ChromeDriver();

        driver.get("https://google.com");

        String title = driver.getTitle();
        Assertions.assertEquals("Google", title);

        driver.manage().timeouts().implicitlyWait(Duration.ofMillis(500));

        WebElement searchBox = driver.findElement(By.name("q"));
        WebElement searchButton = driver.findElement(By.name("btnK"));

        searchBox.sendKeys("Selenium");
        searchButton.click();

        searchBox = driver.findElement(By.name("q"));
        String value = searchBox.getAttribute("value");
        Assertions.assertEquals("Selenium", value);

        driver.quit();
    }
}
```



# Tools...

**JUnit 5** <https://junit.org/junit5/>

**AssertJ** <https://joel-costigliola.github.io/assertj/>

**Mockito** <https://site.mockito.org/>

**PowerMock** <http://powermock.github.io/>

**Hamcrest** <http://hamcrest.org/JavaHamcrest/>

**Selenium** <https://www.seleniumhq.org/download/>