

JUnit

Einleitung:

JUnit ist ein Framework innerhalb der Java Umgebung, welches das **automatisierte Testen** von Klassen und Applikationen erlaubt.

Es wird oft im Zusammenhang mit so genannter **Test-Driven-Development** gebraucht.

Testgetrieben heisst, dass zuerst die Tests geschrieben werden und dann dazu eine Implementation entwickelt wird, bis sie die Tests fehlerfrei besteht. (Nicht ganz einfach..)

In den Aufgaben wenden Sie Maven und **JUnit** kombiniert an.

Maven vereinfacht das automatisierte Testen, machte es von der Entwicklungsumgebung wie Eclipse oder IntelliJ unabhängig.

Sie starten mit einem ersten Warming up zum Thema.



Aufgabe: "Stöbern"

Stöbern Sie in den Dokumentationen zum *JUnit Testframework*. (ca. 10')

JUnit5:

<https://junit.org/junit5/docs/current/user-guide>

JUnit4 versus JUnit5

<https://howtodoinjava.com/junit5/junit-5-vs-junit-4/>

Aufgabe: "Cheat-Sheet"

Tauschen Sie sich mit Ihrem Lernpartner aus. (ca. 15')



Notieren Sie Ihre Antworten in einem persönlichen CheatSheet.

- Erklären Sie, für was die Annotation **@Test** benutzt wird.
- Erklären Sie, für was die **JUnit** Annotationen **@BeforeEach** und **@AfterEach** benutzt werden.
- Nennen Sie **4 Assert-Methoden**, und machen Sie je ein kurzes Beispiel dazu.
[JUnit 5 User Guide Assertions](#)
- Notieren Sie die wichtigsten Schritte für einen Test anhand der Beispiele aus:
<https://www.baeldung.com/junit-5#basic-annotations>

Automatisiert Unittest testen:

Will man Tests automatisiert ausführen, zum Beispiel auf einem Test-Server, dann muss der Build und das **Testen unabhängig von der Entwicklungsumgebung** geschehen können, also ohne Eclipse oder IntelliJ...

Hier kommt Maven ins Spiel, denn genau das kann Maven, unabhängig von einer IDE ein Projekt *builden, zusammenstellen, testen* (und noch einiges mehr 😊).