

Testing und Logging

SEP Sommersemester 2019

Nicolas Brauner

14.05.2019

Wissenschaftlicher Betreuer:
Maximilian Hünemörder, Ludwig Zellner
Verantwortlicher Professor:
Prof. Dr. Peer Kröger



DBS



Übersicht

- Testing
 - Warum testen wir?
 - Arten von Tests
 - JUnit
 - TDD (Test Driven Development)
 - Live Demo
 - JUnit Test Methoden
- Logging
 - Motivation
 - Funktionsweise
 - Live Demo



Warum testen wir?



Warum testen wir?

- Zur Überprüfung, ob Testobjekt...
 - ...den Designanforderungen entspricht
 - ...korrekt auf alle (möglichen) Eingaben reagiert
 - ...einigermaßen performant ist
 - ...wie vorhergesehen eingebettet werden kann
- Hilft beim Finden von Fehlern
- Überprüfung, ob einzelne Komponenten richtig zusammenarbeiten
- Erleichtert Wartbarkeit und Erweiterbarkeit
- Kostenreduktion: Fehler im Live-Betrieb viel teurer und schwerer zu beheben als wenn zuvor getestet

Arten von Tests

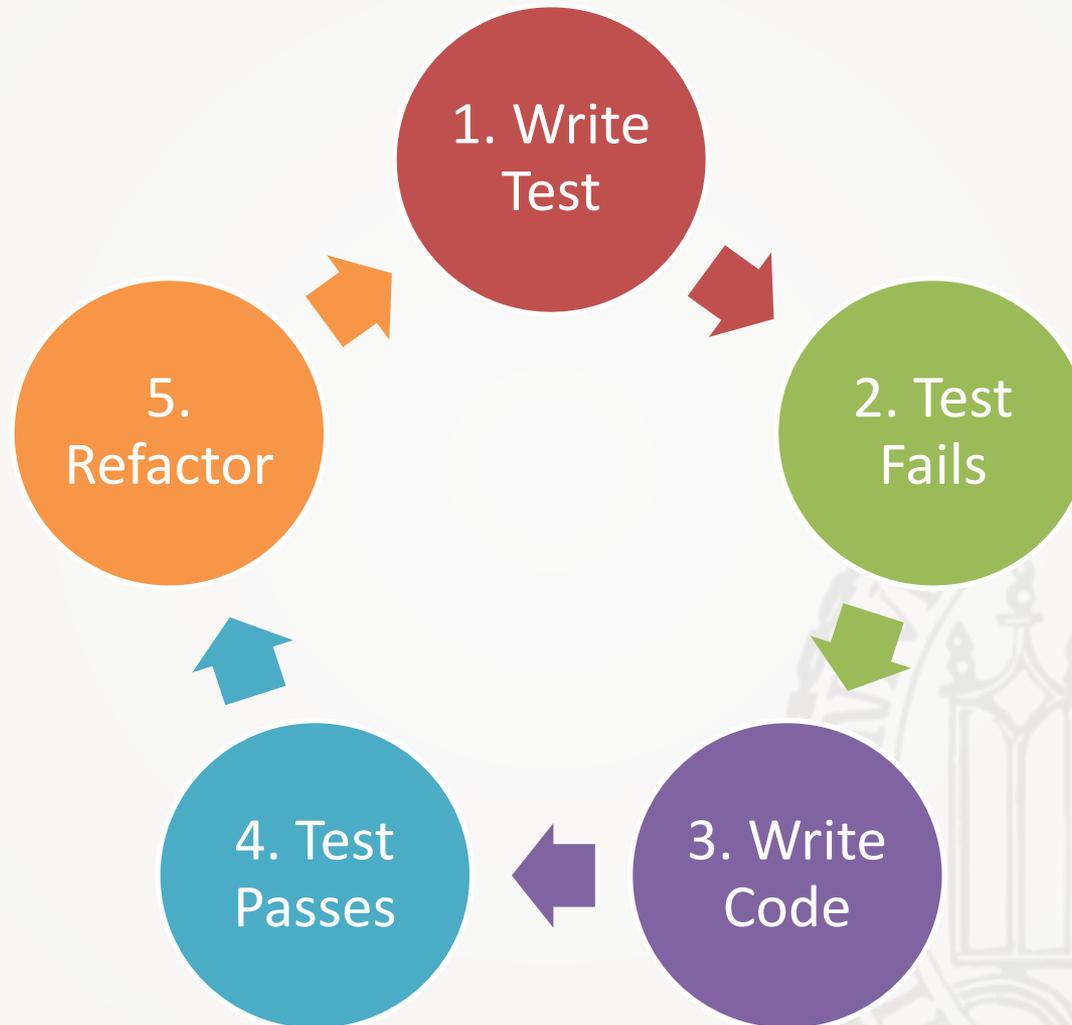
- Funktionales Testen (Basierend auf Spezifikationen, Black-Box, „Was“)
- Strukturelles Testen (Überdeckungskriterien, White-Box, „Wie“)
- Unittests
- Integrationstests
- Systemtests
- Akzeptanz/Abnahmetests

→ Softwaretechnik-Vorlesung

JUnit

- Framework zum Durchführen automatischer Unit-Tests
- Ermöglicht einfaches Testen kleiner Programmteile
- Test-Suits bestehend aus vielen kleinen Tests, möglich
- Liefert klar zu verstehendes Ergebnis **bestanden** vs. **fehlgeschlagen**
- Integriert in gängige IDEs (Eclipse, IntelliJ)

TDD (Test Driven Development)



Live Demo

A QA engineer walks into a bar. Orders a beer. Orders 0 beers. Orders 9999999999 beers. Orders a lizard. Orders -1 beers. Orders a ueicbksjdhd.

First real customer walks in and asks where the bathroom is. The bar bursts into flames, killing everyone.

JUnit Test Methoden

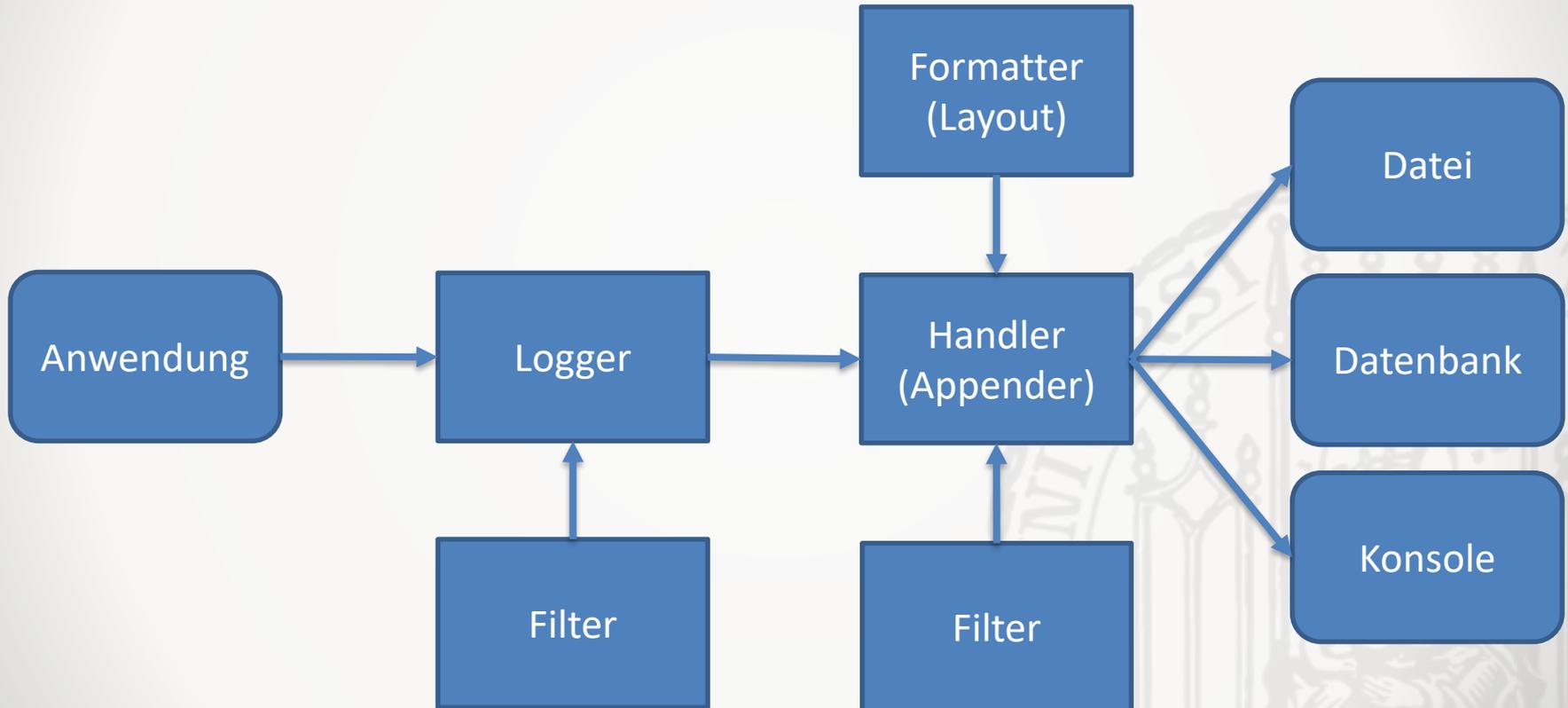
Assertion	Beschreibung
assertEquals	Testet, ob zwei Werte gleich sind
assertTrue	Testet, ob „wahr“
assertFalse	Testet, ob „falsch“
assertNotNull	Testet, ob Objekt nicht null ist
assertNull	Testet, ob Objekt null ist
assertSame	Testet, ob zwei Objekt-Referenzen gleich sind
assertNotSame	Testet, ob zwei Objekt-Referenzen nicht gleich sind
assertArrayEquals	Testet, ob zwei Arrays gleich sind
fail	Schlägt fehl

Logging

- Automatisches Erstellen eines Protokolls
- Aufzeichnung zur Nachvollziehbarkeit von (Fehler-)Zuständen
- Standardisierte Aufzeichnung ermöglicht (automatische) Auswertung
- Ausgabe auf der Konsole, in Dateien, in Datenbanken, übers Netzwerk, ...
- Formatierte Ausgaben

Funktionsweise

- Aufbau



Funktionsweise

Hierarchische „Level“:

- `java.util.logging`:
 - Severe
 - Warning
 - Info
 - Config
 - Fine
 - Finer
 - Finest
- `Log4j2`:
 - Fatal
 - Error
 - Warn
 - Info
 - Debug
 - Trace
 - All



Live Demo



Weiterführende Links

- <https://junit.org/junit5/docs/current/api/>
- <https://www.tutorialspoint.com/junit/index.htm>
- https://en.wikipedia.org/wiki/Code_coverage
- https://en.wikipedia.org/wiki/Software_testing
- <http://blog.codepipes.com/testing/software-testing-antipatterns.html>
- <https://docs.oracle.com/en/java/javase/11/docs/api/java.logging/java/util/logging/package-summary.html>
- <https://logging.apache.org/log4j/2.x/manual/index.html>