

Fragen, Antworten und Kommentare zur aktuellen Vorlesung

Frage: Bei der Aufgabe 5b) geht es darum eine Datei einzulesen und die Werte daraus direkt in den Parametern zu übergeben. Der aktuell geschriebene Programm verhindert, dass Tests mit Parametern ausgeführt werden. Ist das auch für diese Aufgabe so gewollt?

Antwort: Nein, bis 5a ist es ja sinnvoll Parameter zu vermeiden, da Sie keinen Sinn haben können, ist ab 5b anders (zeigt den typischen Fall, dass bei einer Framework-Entwicklung frühe Design-Entscheidungen evtl. überdacht werden müssen). Entweder nehmen Sie die Prüfung raus oder schöner prüfen die Parameterlosigkeit nur, wenn nicht das Attribut `datei` vorhanden ist.

Frage: Um auf die Datei zuzugreifen war unser Ansatz einen `FileReader` zu implementieren. Da die Implementierung im Tester jedoch voraussetzen würde, dass wir manuell uns die Übergabeparameter anschauen und anhand dessen dann Werte übergeben, die irgendwie aus der Datei dazu passen, vermuten wir, dass der Ansatz hier nicht sinnvoll wäre. Was wäre hier der richtige Ansatz? Ist das Ziel der Aufgabe die Übergabe der Parameter in JUnit zu überführen?

Antwort: Wenn statt JUnit "ihr Tester" steht, passt das wohl. Der Positivfall sieht wie folgt aus. Ihr Tester erkennt, dass eine Methode `M` mit `datei=...` annotiert ist und `M` Parameter hat, dann liest ihr Tester aus der Datei die Werte, formatiert sie passend zu den Parametern von `M` und ruft `M` (wie eine Testmethode früher) auf. Zum Lesen einer Datei gibt es mittlerweile mehrere Versionen, `FileReader` ist ok, aber recht alt. Eine Variante ist:

```
for (String line: Files.readAllLines(path, StandardCharsets.ISO_8859_1)) {
```

Anmerkung: Das Thema Testsuiten, mit denen mehrere Testfälle oder auch Testsuiten zusammengefasst werden können, kommt in der Veranstaltung zu kurz, ist aber in der Praxis sehr wichtig. Gerade bei komplexen Systemen sollen oft aus Zeitgründen nur Tests laufen, die unmittelbar betroffene Systeme analysieren, so dass die Nutzung von TestSuiten, durchaus überschneidend für unterschiedliche Aufgaben, sinnvoll und wichtig ist. Ein einfaches Beispiel zeigt folgende Suite, alle relevanten Informationen stehen in Annotationen.

```
@Suite
@SuiteDisplayName("Tests von Blatt 6 zusammengefasst")
@SelectClasses( { PraenexnormalformTest.class
    , SkolemnormalformTest.class
    , PraedikatenlogikBasisTest.class } )
public class Blatt06Test {}
```

Neben der JUnit-Dokumentation, sind einige interessante Tipps zu JUnit auf der folgenden Seite enthalten: <https://howtodoinjava.com/junit5/junit5-test-suites-examples/>