
Prof. Dr. Stephan Kleuker
Hochschule Osnabrück
Fakultät Ing-Wiss. und Informatik
- Software-Entwicklung -

Software Quality Management

Sommersemester 2026
1. Aufgabenblatt

Nutzen Sie die unter http://kleuker.iui.hs-osnabrueck.de/kleukersSEU/index.html zur Verfügung stehenden SEU, die ohne Gewähr zum Download zur Verfügung steht. Das Praktikum findet unter Windows statt, Abgaben müssen mit der gegebenen SEU lauffähig sein. Einige ergänzende Informationen für diese Veranstaltung sind in http://kleuker.iui.hs-osnabrueck.de/querschnittlich/SQMWerkzeuge.pdf enthalten.

Aufgabe 0.1
Geben Sie das Lösungswort des Quiz aus der Lernnotiz an.

Aufgabe 1
In der Veranstaltung wird davon ausgegangen, dass Sie bereits grundlegende Kenntnisse über Prozessmodellierungen, Qualität im Software-Engineering und formale Grundlagen haben. Falls dies nicht der Fall ist, zeigt dieses Aufgabenblatt Themen zum Nacharbeiten, wobei einzelne Teilbereiche auch in dieser Vorlesung kompakt zusammengefasst werden.

a) Prozessmodellierung
Beschreiben Sie den vollständigen Ablauf eines mit Scrum durchgeführten Prozesses mit mehreren Sprints mit einer visuellen Prozessmodellierungssprache, wie Aktivitätsdiagrammen oder BPMN, mit mindestens 6 Aktionen.

b) Software-Engineering
Beantworten Sie folgende Fragen jeweils mit mehreren Stichpunkten, so dass Sie die Antworten im Praktikum erläutern können.
   i) Was ist Clean-Code?
   ii) Was sind Design Pattern und welche Bedeutung haben sie im Software-Engineering?
   iii) Wofür stehen CI/CD in der Software-Entwicklung? Welche Erfahrungen haben Sie damit?
   iv) Wofür steht SOLID im Software-Engineering? Welche berechtigten Kritikpunkte kann man daran äußern?
   v) Wird in akademischen Arbeiten Software entwickelt enthält die zugehörige Dokumentation zumindest ein Kapitel Evaluation oder Validierung (oder Fragmente davon sind anderen Kapiteln zugeordnet). Welche Inhalte müssen bzw. können in einem solchen Kapitel stehen? Gehen Sie dabei auch auf die Punkte „habe ich das Richtige gemacht“ und „habe ich es richtig gemacht“ ein.

c) Testen

d) Gegeben sei folgende von der Veranstaltungsseite in einem Eclipse-Projekt sqmAufgabeTesterinnerung herunterladbare Java-Klasse.

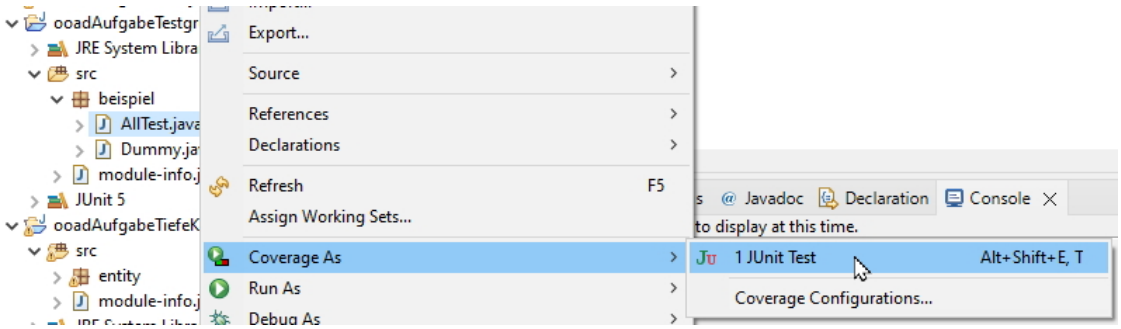
```
package control;

public class Beispiel {

  public int machen(int start, int ende) {
    int erg = 0;
    if (start > 0) {
      if (ende < 100) {
        erg = ende - start;
      } else {
        erg = -1; // Fehler: Ende ist größer als 100
      }
    } else {
```

Prof. Dr. Stephan Kleuker
Hochschule Osnabrück
Fakultät Ing-Wiss. und Informatik
- Software-Entwicklung -

Software Quality Management

Sommersemester 2026
1. Aufgabenblatt

```
            if (ende < 0) {
              erg = start - ende;
            } else {
              if (start > 10) {
                erg = -42;
              } else {
                erg = -2;
              }
            }
          }
          return erg;
        }
      }
```



Schreiben Sie JUnit-Tests, so dass die Überdeckungsmessung von Eclipse möglichst den gesamten Code grün markiert. Falls Sie dies nicht schaffen, begründen Sie dies. Nutzen Sie in Ihren JUnit-Tests die Annotation @Parameterized.

e) Formale Grundlagen
   i) Erklären Sie anschaulich die Church'sche These und gehen Sie auf ihre Bedeutung für die Praxis ein.
   ii) Erklären Sie anschaulich den Satz von Rice und gehen Sie auf seine Bedeutung für die Praxis ein.
   iii) Können folgende Programme existieren?
      α) Ein Java-Programm, dass ein beliebiges anderes Java-Programm ausführt und überprüft, ob dieses Programm nach höchsten 100 Schritten anhält.
      β) Ein Java-Programm, das ein beliebiges anderes Java-Programm ausführt und überprüft, ob dieses Programm überhaupt anhält oder nicht.
      γ) Ein Java-Programm, dass die Ausführung einer Turing-Maschine simuliert.
      δ) Eine Turing-Maschine, die die Ausführung eines Java-Programms simuliert.
      ε) Ein Java-Programm, dass überprüft, ob ein regulärer Ausdruck genau die gleiche Sprache wie ein endlicher Automat beschreibt.