

## Prüfungssituationen für LU04

|  |   |
|--|---|
| <b>B1F:</b>  |   |
| Ich kann den Aufbau einer Software modellieren. (z.B. Klassen-, Aktivitäten- und Sequenzdiagramm)  | Eine einfache Beschreibung eines Ablaufs als Sequenzdiagramm skizzieren.                                    |
|  | Einen vorgegebenen Code analysieren und den Ablauf (oder Teile davon) mit einem Sequenzdiagramm darstellen. |
|  | ...   |
| <b>B1E:</b>  |   |
| Ich kann das Modell einer Software analysieren, kritische Punkte erkennen und Korrekturen vorschlagen. (z.B. statische und dynamische Aspekte, Vererbung, Assoziationen) | <i>Unterricht geht nicht soweit, dass eine Analyse des Codes möglich wäre.</i>                              |
| <b>C1F:</b>  |   |
| Ich kann ein- und zweiseitige Beziehungen gemäss dem statischen Entwurf implementieren.  | Bestehende Klassen um den Code für die ein- und/oder zweiseitige Beziehung ergänzen.                        |
|  | ...   |
| <b>E2G:</b>  |   |
| Ich kann erklären wozu ein Software-dokumentationswerkzeug (z.B. JavaDoc) dient und wie man es einsetzt (Anwenden von Tags, generieren der Dokumentation).               | <i>noch offen, Frage des verwendeten Tools</i>  |
|  | ...   |
| <b>E2F:</b>  |   |
| Ich kann Software mit Hilfe von einem Softwaredokumentationswerkzeug dokumentieren. (z.B. JavaDoc, anwenden von Tags, generieren der Dokumentation)                      | Klassen und Methoden gemäss der BZZ Konvention dokumentieren.   |
|  | ...   |



© René Probst

From:

<https://wiki.bzz.ch/> - **BZZ - Modulwiki**

Permanent link:

<https://wiki.bzz.ch/modul/m320/test/lu04/start>Last update: **2024/03/28 14:07**