**Name:Leon, Sorush, Wesam\_\_\_\_\_\_\_\_\_\_\_\_**

Konkretes Kompetenzraster

|  |  |  |  |
| --- | --- | --- | --- |
| **Handlungsziel** | **A 1.0 – 4.0** | **B 4.0 – 5.0** | **C 5.0 – 6.0** |
| **1 OO Design &  UML-Notation 10%** |  |  |  |
| Use-Case Diagramm erstellt  Klassen-Diagramm erstellt  Dokument-Struktur errichtet und in Readme.md verlinkt | Use-Cases in Diagramm sind kurz umschrieben.  Sequenz-Diagramm für einen Use-Case erstellt | Sinnvolle Sequenz-Diagramme für mehr als einen Use-Case  Klassen-Diagramm enthält mehr als 3 Klassen  Klassen-Diagramme enthalten verschiedene Assoziationen. |
|  |  | | |
| **2 Klassenbasiertes Design implementieren 40%** |  |  |  |
| Es werden Klassen benutzt  Es werden Methoden benutzt  Es werden Attribute definiert  Es werden Objekte von eigenen Klassen erzeugt. | Die Attribute werden nur aus Methoden der Klasse angesprochen. (Datenkapselung)  Klassen benutzen andere Klassen, um Aufgaben zu erledigen, d.h. sie delegieren Aufgaben an andere Klassen. | Implementation stimmt mit einem vorherigen UML-Design überein, d.h. z.B. Klassen implementieren Assoziationen richtig.  D.h. Kompositionen, Aggregationen und Assoziationen sind im Code erkennbar.  Methoden erledigen nur das, was sie müssen, der Rest wird delegiert. |
|  |  | |  |
| **3 Testen & Dokumentieren 30%** |  |  |  |
| Testplanung, Testfälle, Testprotokollierung, Testfazit kann an einem konkret durchgeführten Beispiel nachgewiesen werden.  Eine gelöste Aufgabenstellung kann sinnvoll und verständlich dokumentiert werden | Es wird mit Junit-Tests gearbeitet.  Es werden über 80% des Codes getestet.  Es wird mit JavaDoc dokumentiert. | Junit-Tests decken 100% des Codes ab.  Junit-Tests enthalten auch «negative» Tests  JavaDoc wird extensiv benutzt um Methoden, Attribute und Klassen zu beschreiben. |
| **4 Methoden- Sozialkompetenz & Selbstlernkompetenz 10%** |  |  |  |
| Der Lernende zeigt, dass er das Gelernte auch anderen Lernenden mitteilen kann (mit Publikation, Tipps, etc.). Er kann eigene Verantwortung für das Lernen aufbringen. Er zeigt seinen Entwicklungsprozess (wie z.B. aus gemachten Fehlern neue Einsichten entstehen). | Der eigene Lernfortschritt wird selber geplant und auch geprüft. Das wird entsprechend dokumentiert. Ergebnisse werden kritisch hinterfragt. Der Austausch im Tandem / Gruppen ist ebenfalls nachvollziehbar (mittels Journal, Gesprächen, etc.). | Im Projekt sind Vorteile (oder auch Nachteile) der Zusammenarbeit erkennbar. Ein Teamverhalten wird an konkreten Zeichen erkennbar. Abmachungen (Termine, produktiver Code, etc.) werden eingehalten. |

**Kompetenz gezeigt**

**Kompetenz teilweise gezeigt**

**Kompetenz nicht gezeigt**