

Praktikum Software-Engineering 1

Wintersemester 2014/2015 – Aufgabenblatt 2

Prof. Dr. Stefan Sarstedt <stefan.sarstedt@haw-hamburg.de>

Raum 1085, Tel. 040/42875-8434

Norbert Kasperczyk-Borgmann <nkb@informatik.haw-hamburg.de>

Raum 1189, Tel. 040/42875-8417

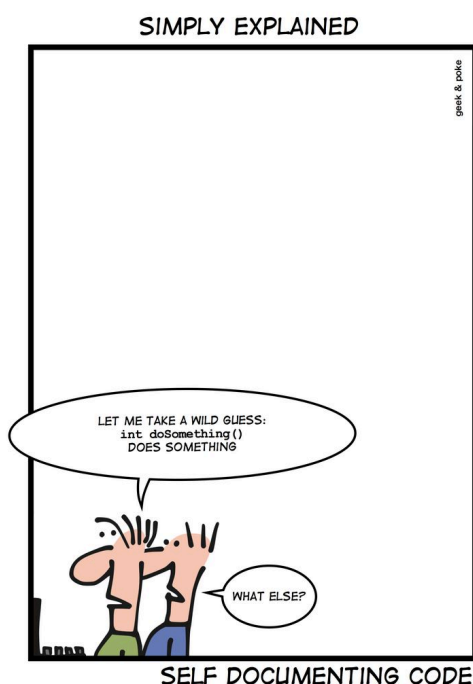
Bearbeitungshinweise

- Die Bearbeitung der Aufgaben findet in **festen Zweiergruppen** statt.
- Zu Praktikumsbeginn erfolgt eine **Befragung** der Teams und die **Abnahme** aller Aufgaben. Eine erfolgreiche Befragung und Abnahme ist Voraussetzung für die weitere Teilnahme am Praktikum.
- Die Lösungen der Aufgaben müssen zum Praktikumstermin fertiggestellt sein und **bis 12:00 Uhr des Folgetages über die e-learning-Plattform emil als eine PDF-Datei** abgegeben werden.
- Außerdem ist eine **ausgedruckte Fassung** der Lösungen in mein Postfach zu legen (Namen der Teammitglieder nicht vergessen!).

Aufgabe 2.1: Entwurf – Komponentenschnitt

Erstellen Sie einen **Komponentenschnitt** für Ihr System. Dazu:

- a) Verfeinern/ergänzen Sie zunächst Ihr Datenmodell nach neuen Erkenntnissen. Entscheiden sich für **Assoziationsrichtungen** in Ihrem Datenmodell und trage Sie diese dort ein.
- b) Entwickeln Sie einen Komponentenschnitt und begründen Sie diesen textuell! Falls es mehrere „sinnvolle“ Schnitt-Varianten gibt, listen Sie diese auf. Sie können den Schnitt in Ihrem fachlichen Datenmodell mit **Farben** markieren.
- c) Zeichnen Sie mit Microsoft Visio ein **UML Komponentendiagramm**, das Ihren Schnitt und die Abhängigkeiten zwischen Ihren Komponenten reflektiert.



Aufgabe 2.2: Implementierung – Entitäten, FDT, Komponenten, DI

In dieser Aufgabe beginnen Sie eine Basis-Implementierung.

- a) Implementieren Sie Ihre **Entitäten** und **fachlichen Datentypen in Java**. Fügen Sie dort sinnvolle **Operationen** ein.
- b) Implementieren Sie Ihre **Komponenten** aus Aufgabe 2.1 als entsprechende Klassen-Skelette in Java. Achten Sie auf passende Konstruktor-Parameter für den Aufgabenteil c).
- c) Erstellen Sie eine `main()`-Methode in Java in der Ihre Komponenten instanziiert und mittels **Dependency-Injection** verbunden werden. Benutzen Sie eine Constructor-Injection. Alle Komponenten sollen von einer Dummy-Persistenz-Komponente mit der (leeren) Schnittstelle `IPersistenceServices` abhängen.

Viel Spaß!