

Software Engineering – Blatt 2

Rasmus Diederichsen Felix Breuninger
{rdiederichse, fbreunin}@uos.de

26. Oktober 2014

Aufgabe 2.1: Phasen der Software-Entwicklung (16 Punkte)

- a) Planungsphase (Projektmanagementplan, Zeitdauer, Aufwandsabschätzung)
- b) Integrationsphase (Integration der Komponenten)
- c) Spezifikationsphase (Kontrakt)
- d) Entwurfsphase (Systemarchitektur, Detailentwurf)
- e) Implementierungsphase (Modultest)
- f) Wartungsphase (Anpassung), Rückzugsphase (Datensicherung)
- g) Anforderungsphase
- h) Implementations- oder Wartungsphase

Aufgabe 2.2: Modularisierung (30 Punkte)

a) Bindung

`FieldController` Funktional da der Controller nur eine Funktion exportiert.

`GameBoard` Logisch, da zusammenhängende Funktionen angeboten werden, die aber nur teilweise auf den selben Daten operieren, sodass keine wirkliche kommunikatorische Bindung vorliegt.

`Field` Logisch, da die Operationen nicht alle auf derselben Struktur operieren; das `Field` selbst ist eher die Struktur, auf der Operiert wird und ist so nicht von anderen Typen abhängig. Die Klasse stellt somit eine Sammlung zusammengehörender Funktionen dar.

`GameView` Funktional, da es nur eine Funktion (`update()`) gibt.

`FieldButton` Wäre `refreshView()` private, so wäre das Modul funktional gebunden. So jedoch lediglich temporal oder sequentiell, da `update()` immer `refreshView()` aufruft. Da dies die einzige andere Methode ist, lässt sich dies aber schwer beurteilen.

`MineSweeper` Zufällig, da die enthaltenen Funktionen nichts miteinander zu tun haben. Betrachtet man nur die exportierte Funktionalität, kann man von funktionaler Bindung sprechen.

`GameOverListener` Funktional, da nur eine Funktion vorhanden ist.

b) Kopplung

FieldController

MouseAdapter Nicht gekoppelt, da keine Aufrufe an die Superklasse außer im Konstruktor

MineSweeper

GameView Stempelkopplung, da im Konstruktor der Listener übergeben wird, den beide kennen müssen.

GameBoard Datenkopplung (Zeile 68)

GameView

GameBoard Stempelkopplung, wegen des Konstruktoraufrufs mit einem komplexen Typ (GridLayout) sowie des Methodenaufrufs in Zeile 79.

FieldButton Ebenfalls Stempelkopplung, ebenfalls durch Methode in Zeile 79. Der Typ Field muss beiden bekannt sein.

GameOverListener Keine direkte Kopplung, da die Aufrufe keine Argumente enthalten.

FieldButton

Field Stempelkopplung aufgrund des addObserver(this)-Aufrufs

GameBoard

Field Steuerkopplung wegen des Aufrufs new Field(true), wobei der Wert eher ein Datum als einen steuernden Schalter darstellt, daher kann man auch als Datenkopplung kategorisieren.

Field

GameBoard Datenkopplung wegen des Aufrufs in Zeile 403, Stempelkopplung wegen des Aufrufs in Zeile 434.

Aufgabe 2.3: Design by Contract, C4J (34 Punkte)