LU16b - Prinzipien 3 (OCP, Law of Demeter)

Wir betrachten dieses Mal die Prinzipien **OCP** und das **LoD**.

Prinzipien

- *DRY (Don't Repeat Yourself)*
- *SRP (Single Responsibility Principle)*
- **OCP (Open/Closed Principle)**
- *LSP (Liskov Substitution Principle)*
- *ISP (Interface Segregation Principle)*
- *DIP (Dependency Inversion Principle)*
- *KISS (Keep It Simple, Stupid)*
- *YAGNI (You Ain't Gonna Need It)*
- *SoC (Separation of Concerns)*
- **LoD (Law of Demeter / Principle of Least Knowledge)**

P.S.: SOLID steht für SRP, OCP, LSP, ISP, DIP

OCP

Das Open/Closed Principle besagt, dass ein Programm, zwar offen sein soll für (sinnvolle) Weiterentwicklung, aber geschlossen für Änderungen (Manipulationen).

Als Faustregel kann man Klassen tendenziell schliessen, während Interfaces geöffnet werden. Der Modifier `public` sollte bei Klassen also sparsam eingesetzt werden, während `final` bei vielen Implementierungsklassen in Frage kommt.

Der Modifier `protected` bei Klassenattributen ist oft ein Indiz für eine Offenheit gegenüber Änderungen via Vererbung. Sie sollten daher immer kritisch hinterfragt werden.

LoD

Das Law of Demeter sagt aus, dass Objekte lediglich mit seinen unmittelbaren Abhängigkeiten interagieren sollte.

Ein konkretes Beispiel finden Sie auf https://luiskueng.com/5-Publish/Blog/Clean+Code/Prinzipien+OO-Design/Law+of+Demeter

Ein starkes Indiz, dass dieses Prinzip verletzt wurde sind (lange) Getter-Ketten.

```
customer.getCart().getItems().get(0).getProduct().getPrice();
```

Solche Fälle sollten in mehrere kleinere Methoden in ihren jeweiligen Entitäten aufgeteilt werden, so dass keine Klasse alle Entitäten der Kette kennen muss.

From:
https://wiki.bzz.ch/ - **BZZ - Modulwiki**

Permanent link:
https://wiki.bzz.ch/de/modul/ffit/3-jahr/java/learningunits/lu16/b?rev=1767689966

Last update: **2026/01/06 09:59**