Programmierkonzepte in der Physikalischen Chemie

24. Open-Closed-Prinzip

UNI FREIBURG

Albert-Ludwigs-Universität Freiburg

Dr. Till Biskup

Institut für Physikalische Chemie
Albert-Ludwigs-Universität Freiburg
Wintersemester 2016/17

Zentrale Aspekte

UNI FREIBURG

- Softwareeinheiten sollten offen für Erweiterungen, aber verschlossen für Veränderungen sein.
- Symptome für den Einsatz: Unflexibilität, Zerbrechlichkeit und Unbeweglichkeit
- Der Schlüssel des Prinzips ist Abstraktion in Verbindung mit Vererbung.
- Verschlossenheit gegen Veränderung ist nie vollständig.
- Kern objektorientierten Entwurfs: Führt zu Flexibilität, Wiederverwendbarkeit, Wartbarkeit.

Übersicht

UNI FREIBURG

Das Open-Closed-Prinzip

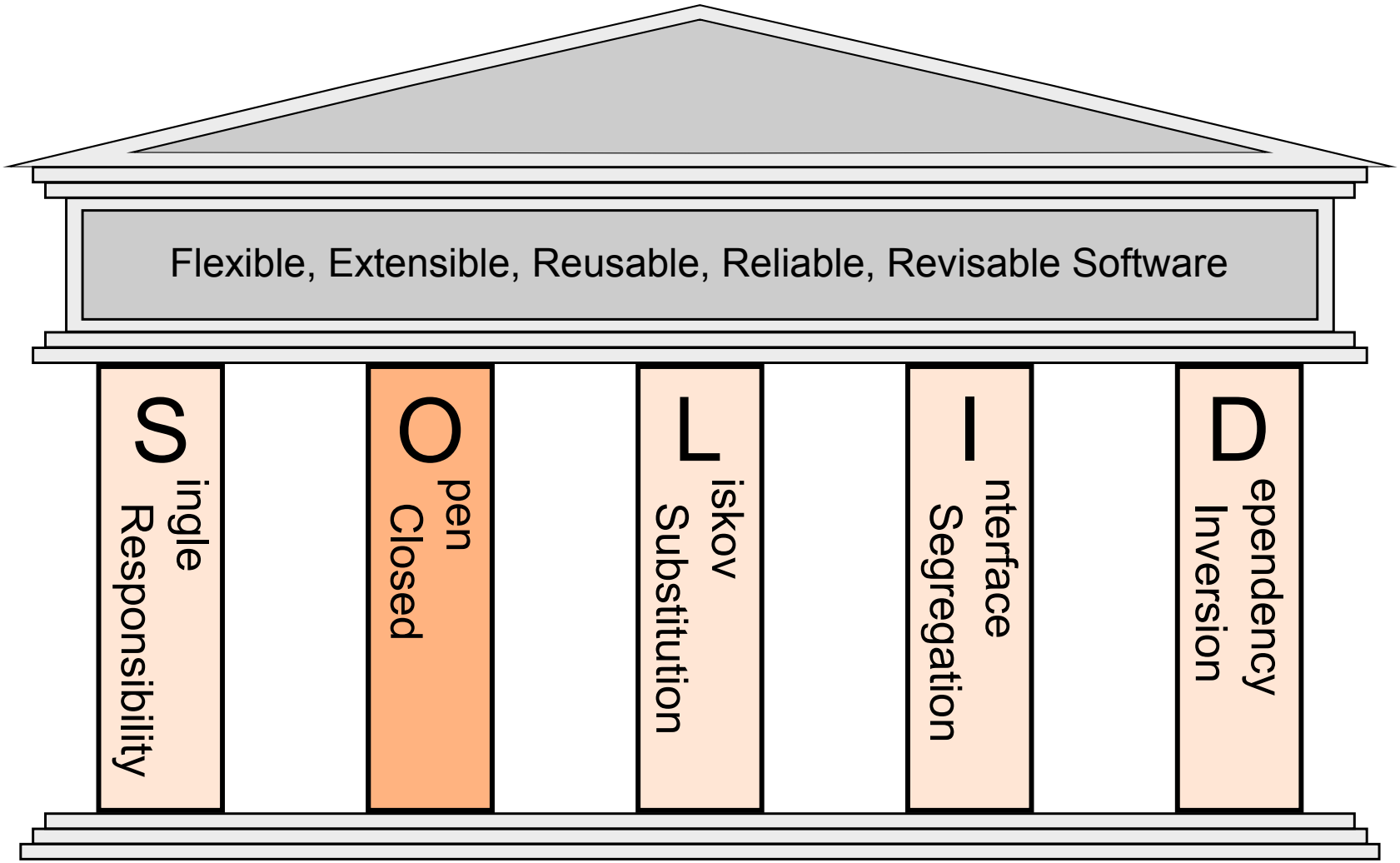
Symptome, die für seinen Einsatz sprechen

Beispiele für seinen Einsatz

Bedeutung im Gesamtkontext der Software-Architektur

Das Open-Closed-Prinzip

Übersicht über die fünf Prinzipien



Das Open-Closed-Prinzip

Offen für Erweiterungen, verschlossen gegenüber Veränderungen

UNI FREIBURG

> *Software entities (classes, modules, functions, etc.) should be open for extension, but closed for modification.*
>
> – Robert C. Martin

- Klingt zunächst wie ein Widerspruch.
  - Normalerweise bedeutet Erweiterung auch Änderung.
  - Mit nicht-objektorientierten Techniken nicht realisierbar.
- Lösung: Abstraktion und Vererbung
  - Eine Klasse hängt nur von abstrakten Klassen ab.
  - Objekte dieser Klasse nutzen Objekte konkreter Klassen, die von der abstrakten Klasse erben.

Robert C. Martin: Agile Software Development. Prentice Hall, Upper Saddle River 2003, S. 99

Der Clou: Abstraktion und Vererbung

Zentrale Konzepte objektorientierter Programmierung

UNI FREIBURG

- abstrakte Klasse
  - Klasse, die zunächst einmal nur eine Schnittstelle liefert
  - enthält (abstrakte) Methoden ohne Implementierung
  - Einsatz: konkrete Klassen erben von dieser Klasse
- Vererbung
  - abgeleitete Klasse implementiert (abstrakte) Methoden
- Polymorphismus
  - Die Zahl der abgeleiteten Klassen ist unbegrenzt.
  - ermöglicht beliebige Erweiterung der Elternklasse ohne Änderung ihres Verhaltens

☛ Dieser Einsatz von Vererbung wird von manchen Autoren als ihr einzig richtiger Einsatz bezeichnet.

Symptome, die für seinen Einsatz sprechen

Unflexibilität, Zerbrechlichkeit, Unbeweglichkeit

UNI FREIBURG

Unflexibilität (*rigidity*)

Jede Änderung zieht viele Änderungen
in anderen Teilen des Programms nach sich.

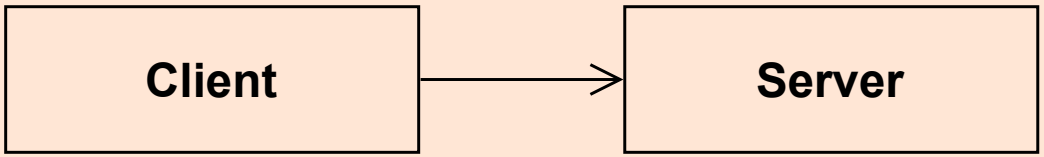
Zerbrechlichkeit (*fragility*)

Änderungen führen zu Fehlern in Bereichen,
die konzeptionell getrennt von den Änderungen sind.

Unbeweglichkeit (*immobility*)

Das System lässt sich schwer in
wiederverwendbare Komponenten aufteilen.

Verstoß gegen das Prinzip

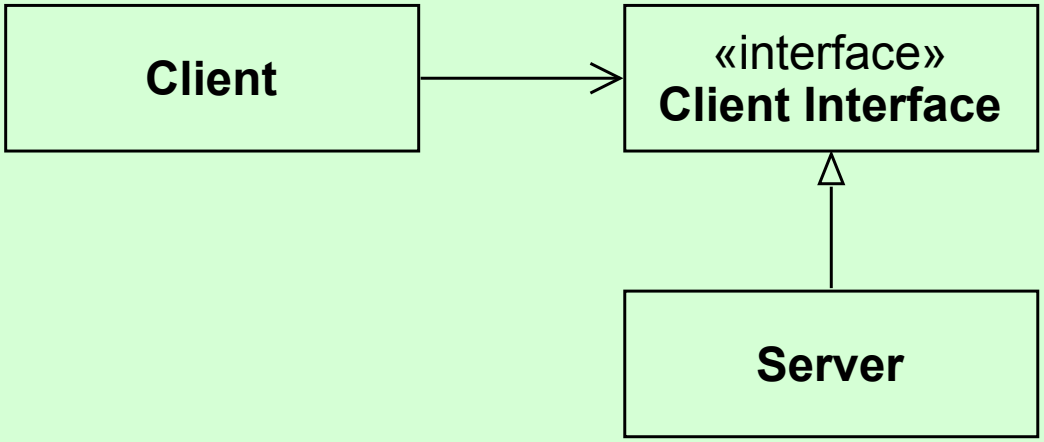


- Probleme
  - `Client` nutzt direkt `Server`.
  - Soll `Server` ausgetauscht werden, muss `Client` geändert werden.
  - Kontrollfluss erzwingt Abhängigkeit in gleicher Richtung.

- ☛ Immer gegen Schnittstellen, nicht gegen Implementierungen programmieren.

Robert C. Martin: Agile Software Development. Prentice Hall, Upper Saddle River 2003, S. 100

Lösung im Einklang mit dem Prinzip



- `Client` und `Server` sind entkoppelt.
- `Server` kann ausgetauscht werden,
  ohne dass `Client` etwas davon mitbekommt.

Robert C. Martin: Agile Software Development. Prentice Hall, Upper Saddle River 2003, S. 101

Bedeutung im Gesamtkontext

Zentral für gute objektorientierte Software-Architektur

UNI FREIBURG

- zentral für gute objektorientierte Software-Architektur
  - Schlüssel zum Erreichen der Versprechen objektorientierter Programmierung
  - führt zu Flexibilität, Wiederverwendbarkeit, Wartbarkeit
- Es reicht nicht, objektorientiert zu programmieren.
  - Der richtige und bewusste Einsatz der zur Verfügung stehenden Werkzeuge ist entscheidend.
- Übermäßiger Einsatz ist kontraproduktiv.
  - Voreiliger Abstraktion zu widerstehen ist genauso wichtig wie Abstraktion selbst.
  - Anwendung nur bei solchen Teilen des Programms, die sich häufig ändern.

Zentrale Aspekte

- Softwareeinheiten sollten offen für Erweiterungen, aber verschlossen für Veränderungen sein.
- Symptome für den Einsatz: Unflexibilität, Zerbrechlichkeit und Unbeweglichkeit
- Der Schlüssel des Prinzips ist Abstraktion in Verbindung mit Vererbung.
- Verschlossenheit gegen Veränderung ist nie vollständig.
- Kern objektorientierten Entwurfs: Führt zu Flexibilität, Wiederverwendbarkeit, Wartbarkeit.