

Wie entwirft man Klassen + Methoden?

Generelles Prinzip: Datenabstraktion

Schreiben + Lesen von Objektattributen sollte nicht außerhalb der eigenen Klasse erfolgen, sondern nur durch bestimmte Methoden, die in der eigenen Klasse festgelegt werden.

⇒ Selektoren (setter + getter)

Auf diese Weise kann man die genaue Implementierung "kapseln".

Zugriff von außen ist dann nur noch über bestimmte Methoden möglich.

Vorteil: Man kann die interne Implementierung einer Klasse ändern. Man muss nur dafür sorgen, dass die nach außen sichtbaren Methoden weiterhin verfügbar sind.

⇒ Modularität + Änderungsfreundlichkeit

Änderungen sind auf eine Klasse begrenzt.

### Datenabstraktion

Nach außen ist nur eine abstrahierte Sicht auf die Klasse verfügbar. Die interne Implementierung ist für die anderen Klassen verborgen.

Lässt sich mit geeigneten Zugriffsspezifikationen erzwingen.

### Geheimnisprinzip (Information Hiding)

• Client/Server - Prinzip:

## Client/Server - Prinzip:

- Anbieter publiziert Schnittstellendokumentation der öffentlich zugänglichen Komponenten  $\hat{=}$  abstrakter Datentyp
- Kunden interessiert nur die Schnittstelle, aber nicht, wie die Implementierung genau aussieht.
- Vorteile: Verständlichkeit, Änderbarkeit, Modularisierung

Java unterstützt die automatische Erstellung von Schnittstellendokumentationen:

javadoc    -d doc    oder \*.java  
                                 Redteck.java

Directory doc, in dem die entstehenden HTML-Seiten gespeichert werden

Bsp: API der Java-Bibliotheken

## Datenkapselung als Entwurfsprinzip

Entwerfer erst die öffentliche Schnittstelle und erst dann die Implementierung.