



Modul - Objektorientierte Programmierung

Bachelor Wirtschaftsinformatik

02 - Linked List

Prof. Dr. Marcel Tilly

Fakultät für Informatik, Cloud Computing



OOP in Java

Etwas zur Geschichte:

- Java ist nicht die erste objektorientierte Sprache (OO-Sprache)
- C++ war nicht die erste
- Klassischerweise gelten Smalltalk und insbesondere Simula-67 aus dem Jahr 1967 als Stammväter aller OO-Sprachen
- Die eingeführten Konzepte sind bis heute aktuell



Warum überhaupt OOP?

- Menschen nehmen die Welt in Objekten wahr
- Objektorientiertes Design mit prozeduralen Systemen ist schwierig (Programme, Unterprogramme,..)

=> Programm-Design wird durch Objekte und Klassen einfacher



OOP Prinzipien

OOP stützt sich auf die Konzepte von Objekten und Klassen (Typedefinition von Objekten).

Es gilt:

1. Alles ist ein Objekt (manchmal gibt es Ausnahmen, z.B. Basistypen)
2. Objekte kommunizieren durch das Senden und Empfangen von Nachrichten (Wie funktioniert das in Java?)
3. Jedes Objekt ist die Instanz einer Klasse.
4. Die Klasse definiert die Struktur aller ihrer Instanzen



Eigenschaften OOP

Grundsätzlich bieten Objekte die folgenden Vorteile:

- **Generalisierung** (Abstraktion): Bei der Generalisierung wird ausgenutzt, dass unterschiedliche Klassen teilweise gleiche Eigenschaften haben. Diese Eigenschaften müssen nur ein Mal beschrieben werden und können dann von allen erfindenden Klassen genutzt werden.
- **Kapselung**: Die Kapselung von Informationen wird auch als »data hiding« bezeichnet. Damit werden die Eigenschaften innerhalb einer Klasse »versteckt«, um Zugriffe von außen zu verhindern.
- **Vererbung** (Spezialisierung): Vererbung gestattet Spezialisierung auf Basis einer gemeinsamen Oberklasse. So lassen sich trotz Generalisierung spezielle Eigenschaften abbilden.
- **Polymorphie**: Das Wort Polymorphie entstammt der griechischen Sprache und bedeutet *Vielgestaltigkeit*. Die Polymorphie der objektorientierten Programmierung ist eine Eigenschaft, die in Zusammenhang mit Vererbung einhergeht. Eine Methode ist genau dann polymorph, wenn sie von verschiedenen Klassen unterschiedlich genutzt wird

OOP in Java

Das Konzept der OOP lehnt sich stark an Strukturierungs- und Klassifizierungsmethoden aus der alltäglichen (menschlichen) Betrachtungsweise unserer realen Welt an.

- OOP wird in Java mittels *Klassen* und *Objekten* realisiert.
- *Klassen* spezifizieren
 - die Struktur (*Attribute*)
 - die Hierarchie (*Vererbung*)
 - und Abhängigkeiten (*Referenzen*)

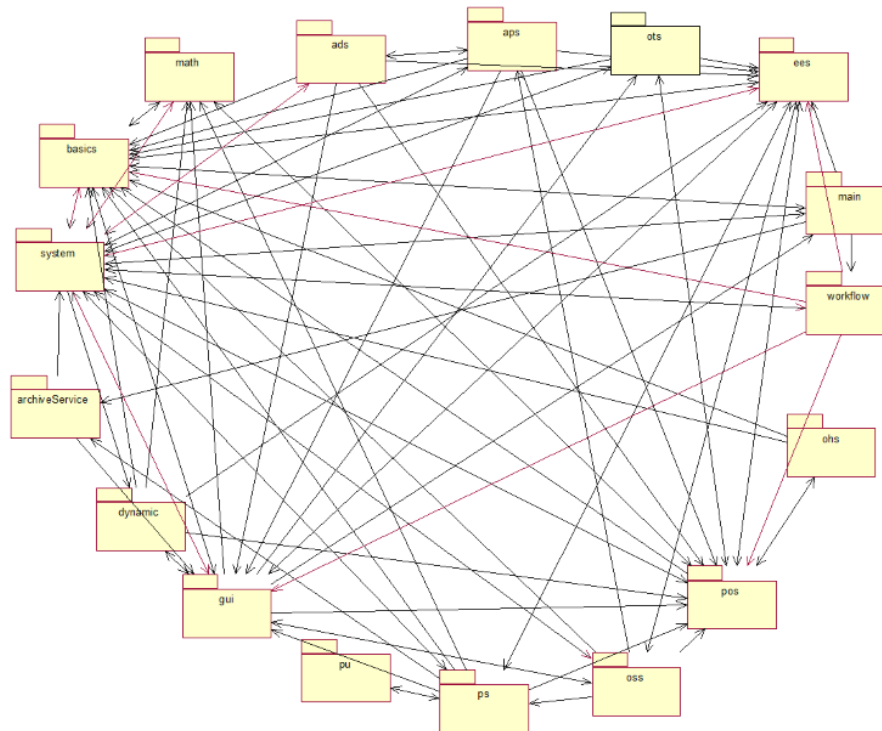


Objekte in Java

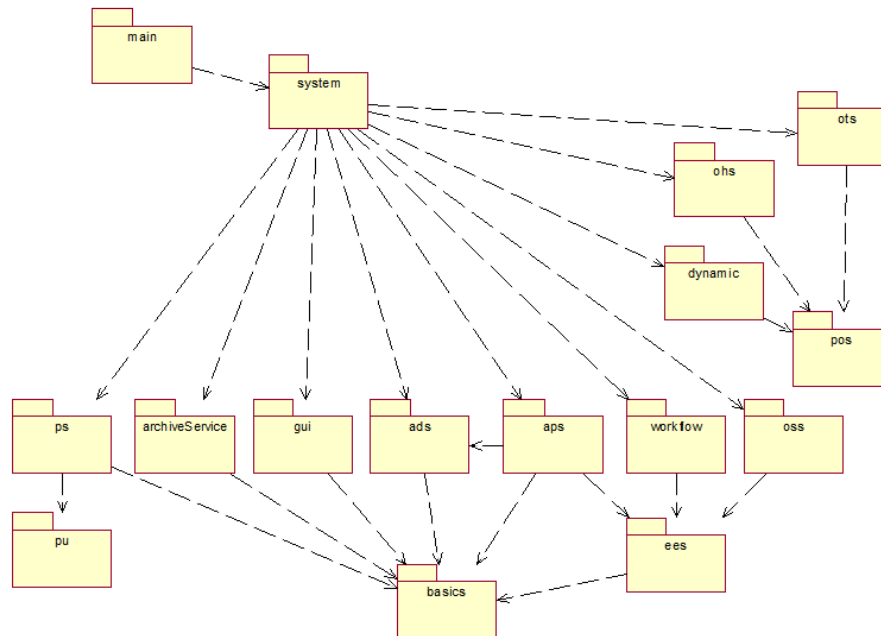
Ein *Objekt* ist die Instanz einer *Klasse* und somit die konkrete Ausprägung einer Klasse.

- Jedes Objekt hat eine **Identität** (bleibt erhalten während der Lebenszeit!)
- Jedes Objekt hat einen **Zustand** (Bildet eine Einheit von Daten und Funktionalität)
- Jedes Objekt hat ein **Verhalten**
- Jedes Objekt bietet eine Schnittstelle (Interface) zur Interaktion

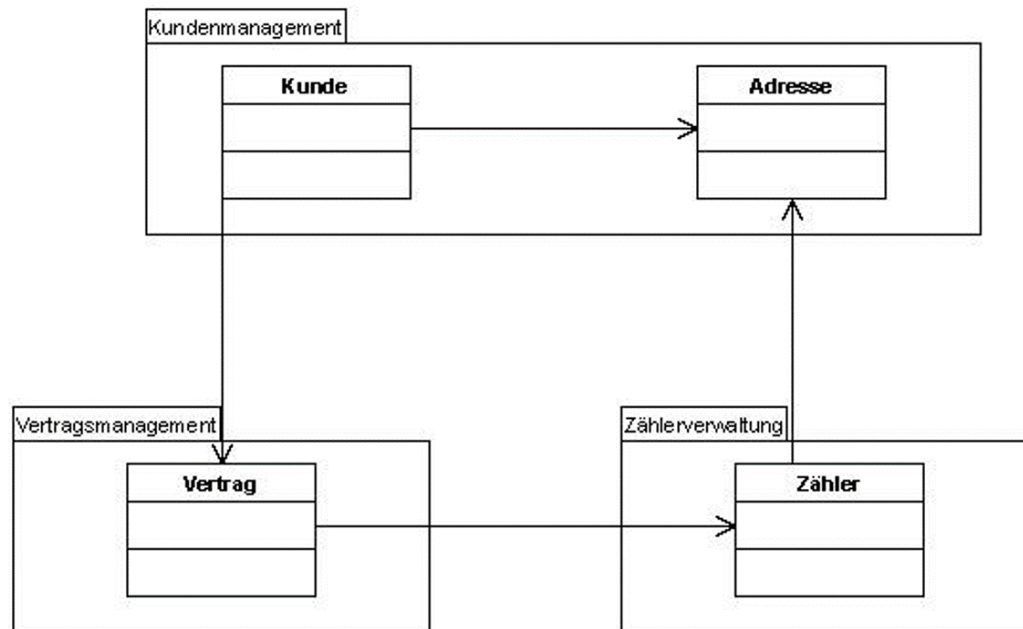
Motivation: "Bad design smells!"



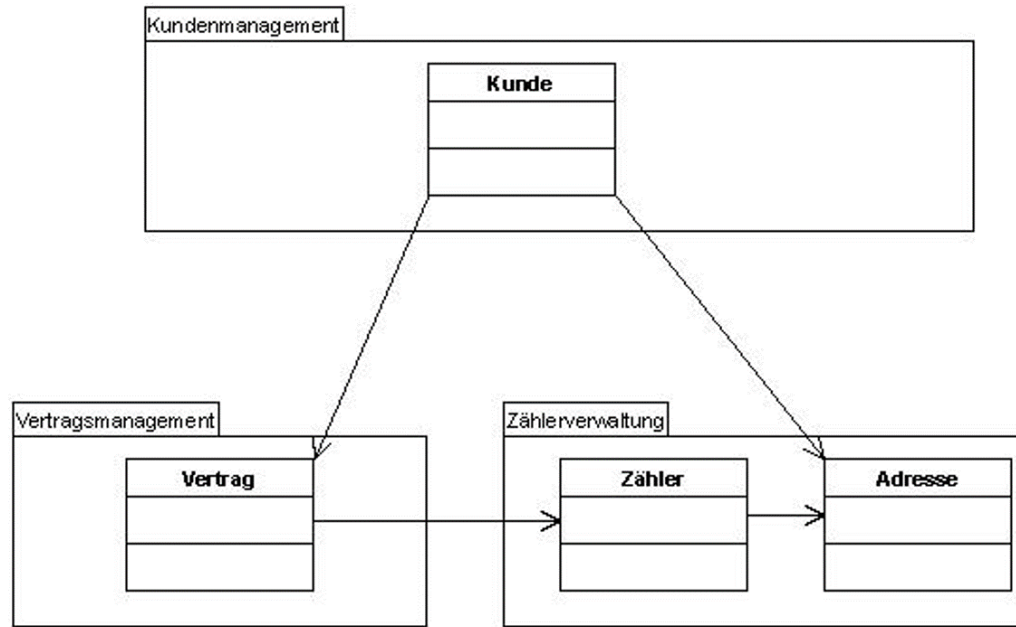
Good Design



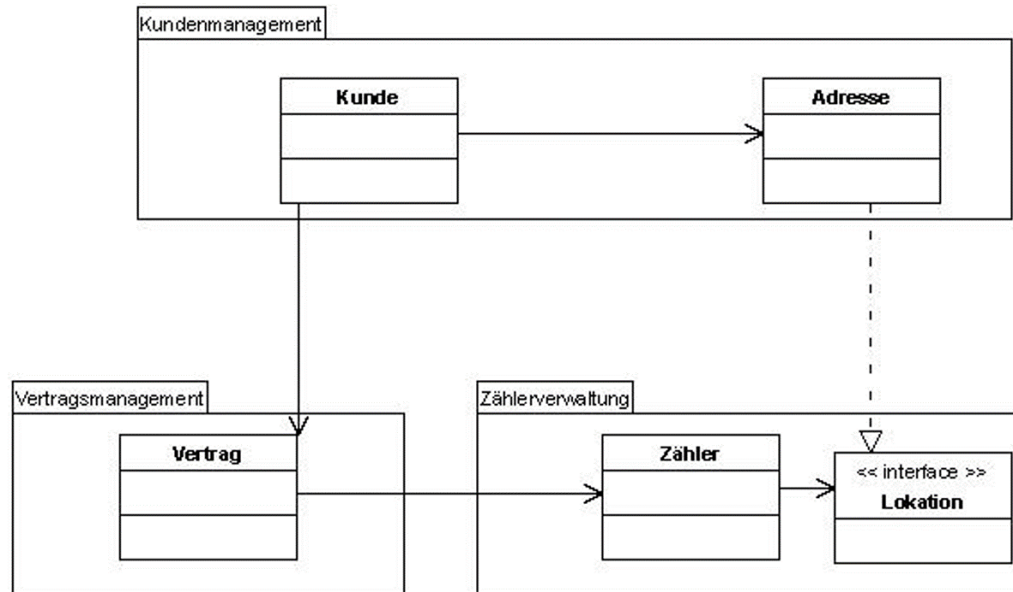
Bad Design: Wie kann das passieren?



Good Design



Better Design





Why does it matter?

Klare Struktur – klare Sprache

- Eindeutige Abhängigkeiten
- modular

Effekt

- Definierte Verantwortlichkeiten
- Einfachere Wartung
- Einfachere Änderungen
- Effizienter
- Besser zu testen



Fragen?