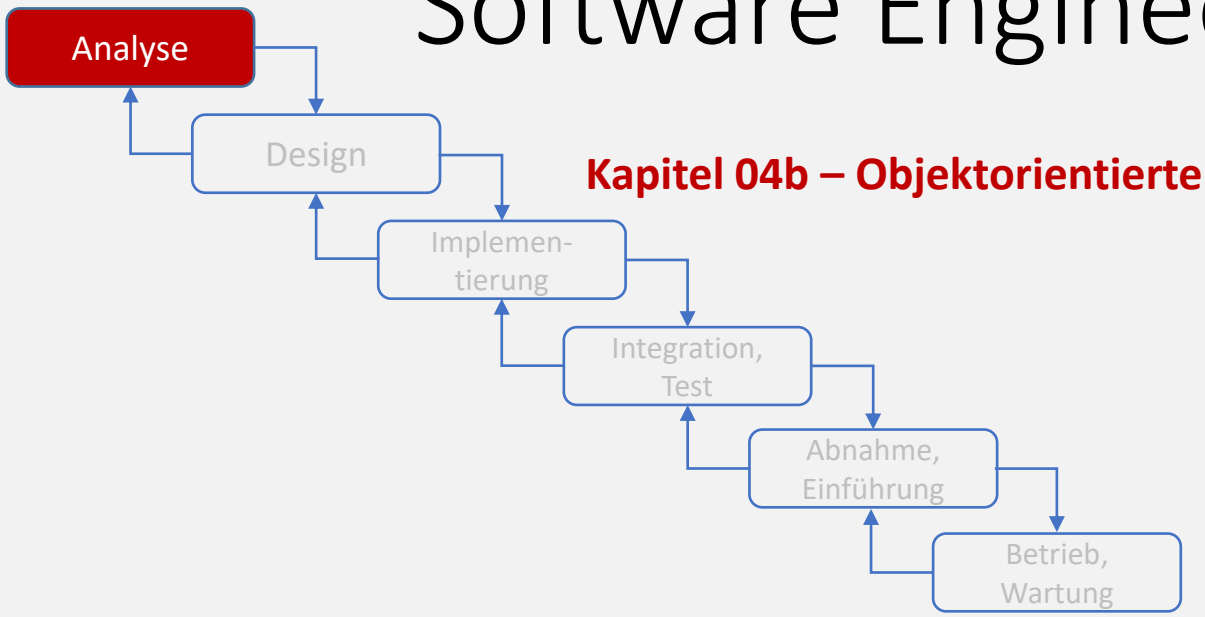
Software Engineering

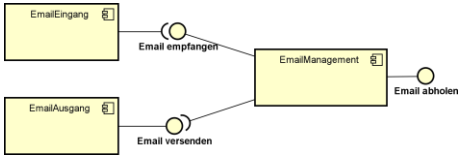


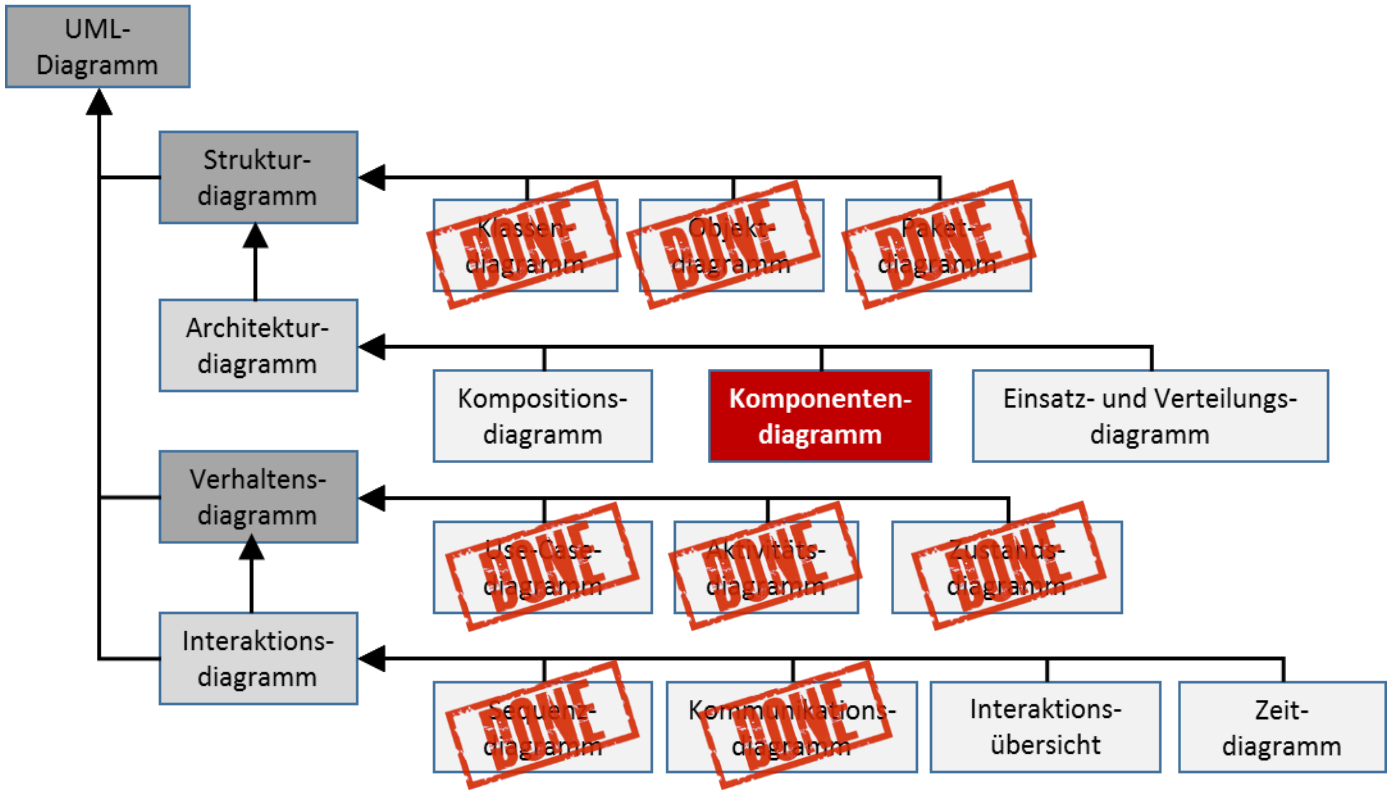
Kapitel 04b – Objektorientierte Analyse (OOA)

Komponentendiagramme

OSTBAYERISCHE TECHNISCHE HOCHSCHULE REGENSBURG
INFORMATIK UND MATHEMATIK

UML: Komponentendiagramme





Komponente: Softwarebaustein, der über klar definierte Schnittstellen Verhalten bereitstellt

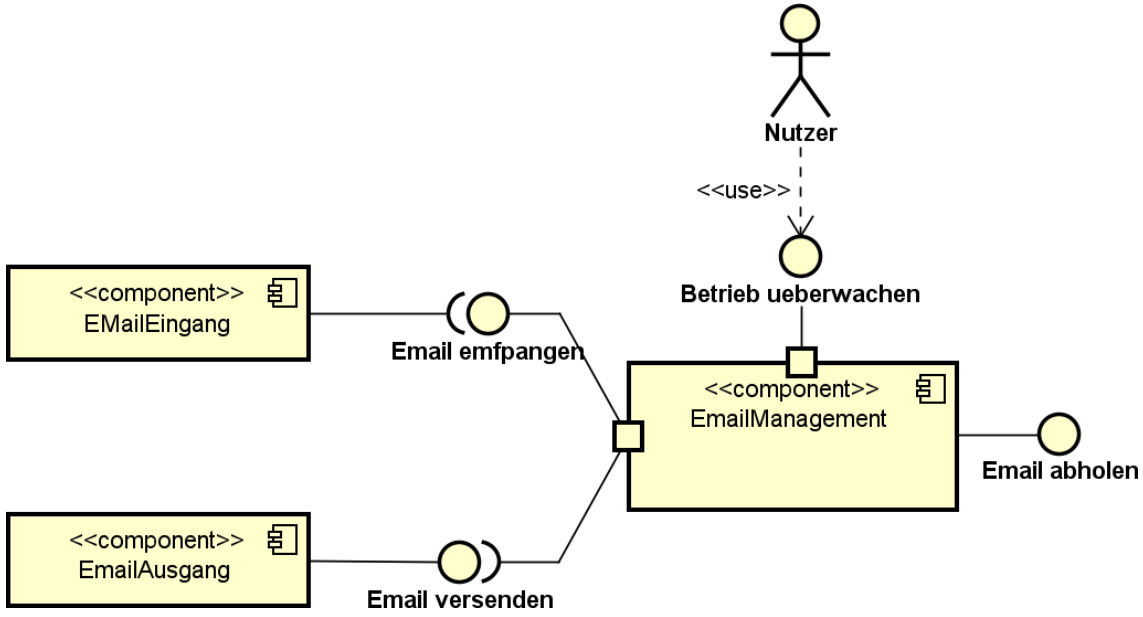
Komponentendiagramm

Wozu benötige ich das Komponentendiagramm?

- Man bekommt die Möglichkeit, die Struktur eines Systems zur Laufzeit darzustellen
- Darstellung orientiert sich an gewählter Komponentenstruktur

- **Es beantwortet die Frage:**

  „Wie ist mein System strukturiert und wie werden diese Strukturen erzeugt?“



[Quelle: Wikipedia]

Komponentendiagramm: Überblick

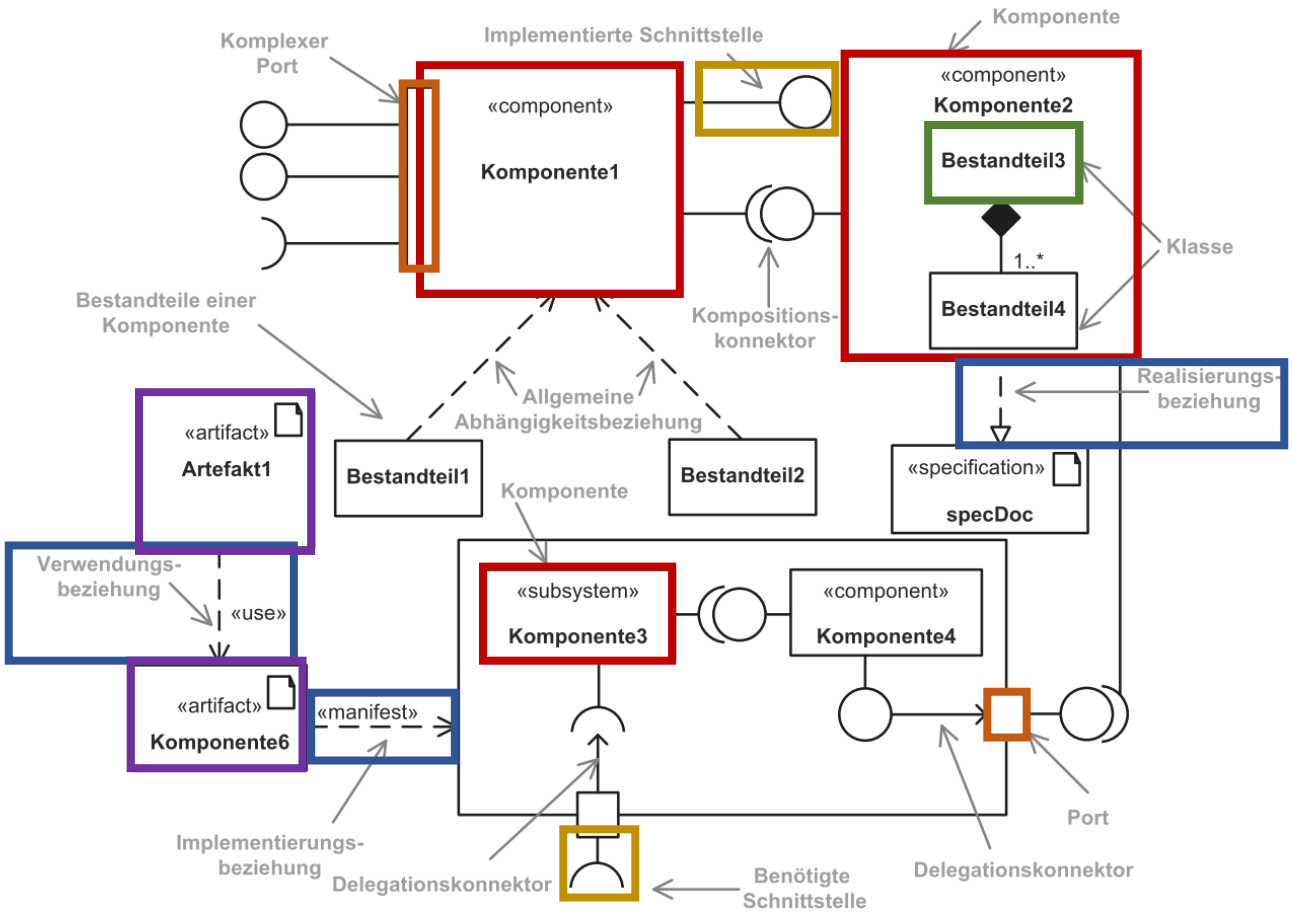
Das Komponentendiagramm:

- Stellt verschiedene Bestandteile eines Systems als Komponenten dar
- Zeigt, wie Komponenten zur Laufzeit organisiert sind und welche Abhängigkeiten sich daraus ergeben

Komponenten:

- Stellen ein abgegrenztes und
- Über klar definierte Schnittstellen zugreifbares Verhalten bereit
- Konkrete Realisierung einer Komponente kann gegen andere Komponenten, die über dieselben Schnittstellen verfügen, ausgetauscht werden, ohne Änderungen am System vorzunehmen.

Komponentendiagramm: Elemente

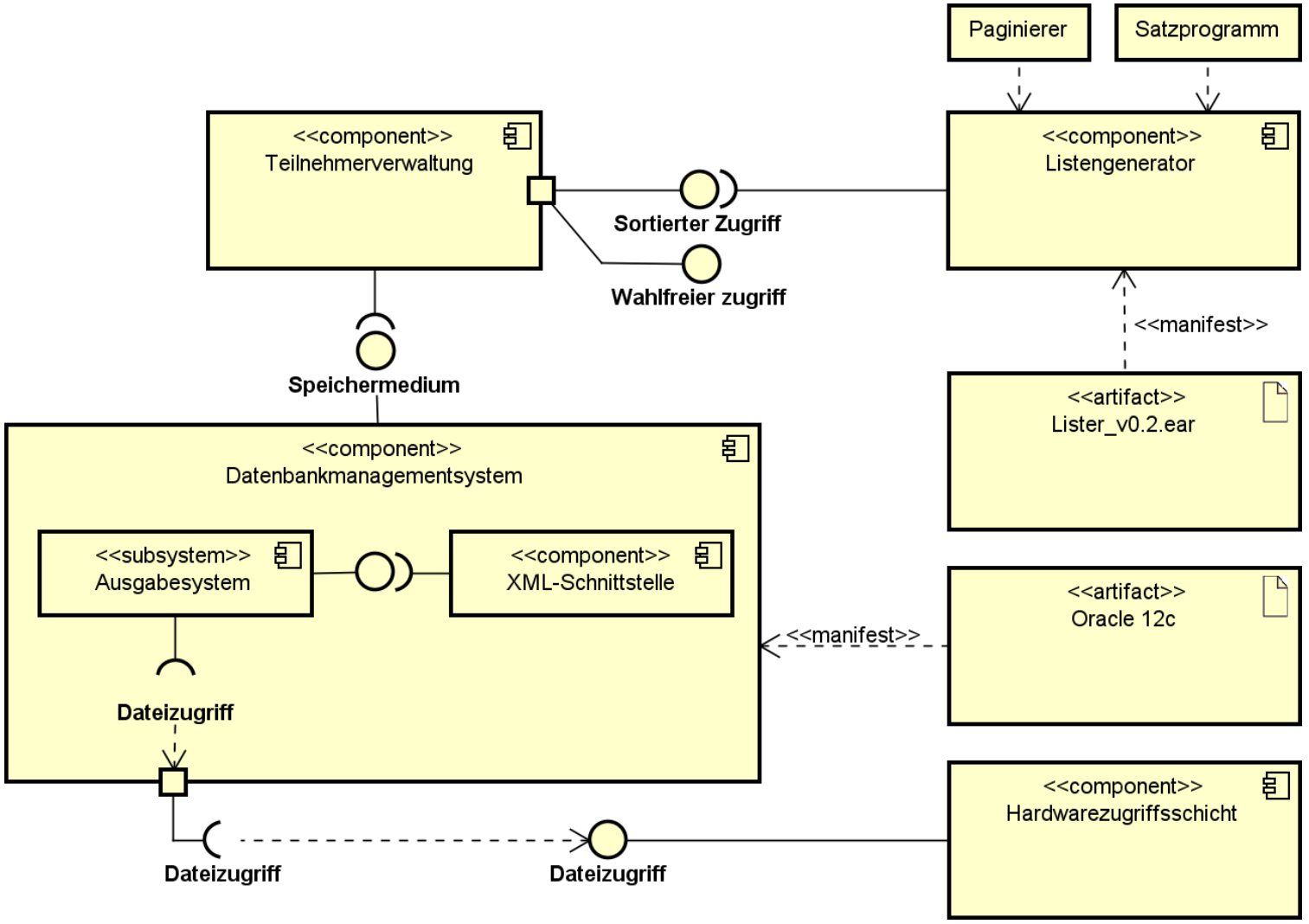


Elemente:

- Komponente
- Schnittstelle
- Realisierungs-, Implementierungs-, Verwendungsbeziehung
- Klasse
- Artefakt
- Port

[Quelle: UML 2 glasklar]

Komponentendiagramm: Anwendungsbeispiel (Teilnehmerverwaltung)



Komponentendiagramm: Elemente

- **Komponentendiagramm definiert selbst nur die Elemente:**
  - Komponente
  - Artefakt
  - Einige zusätzliche Stereotype
  - Daneben sind Notationselemente anderer Diagrammtypen enthalten (Schnittstellen, Ports, Klassen, etc.)

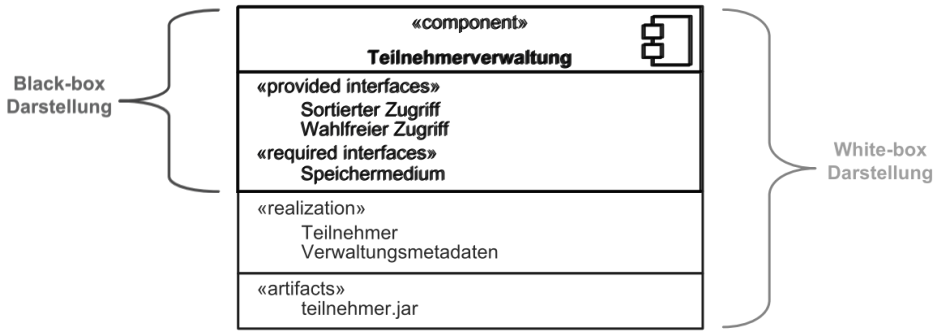
Komponentendiagramm: Elemente

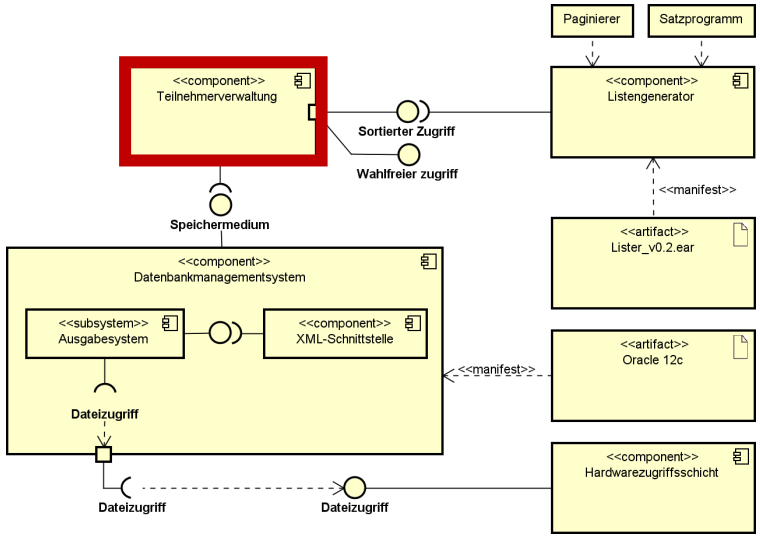
Komponente:

- Repräsentiert einen modularen Teil eines Systems
- Dargestellt durch ein mit <<component>> versehenem Klassensymbol
- Zusätzlich kann oben rechts ein Komponentensymbol stehen (s. mittlere Abb.)
- Für Einheiten großer Systeme wird auch Stereotyp <<subsystem>> genutzt

Komponentendiagramm: Elemente

Komponente: detaillierte Darstellung





- Black-Box-Darstellung:
  - Wenn nur die verwendeten Schnittstellen gezeigt werden
  - Interne Realisierung der Komponente wird nicht gezeigt
- White-Box-Darstellung:
  - Zusätzlich zu verwendeten Schnittstellen noch Angaben über internen Komponenten-Aufbau (Schlüsselwort: <<realization>>) und Artefakte (<<artifacts>>)

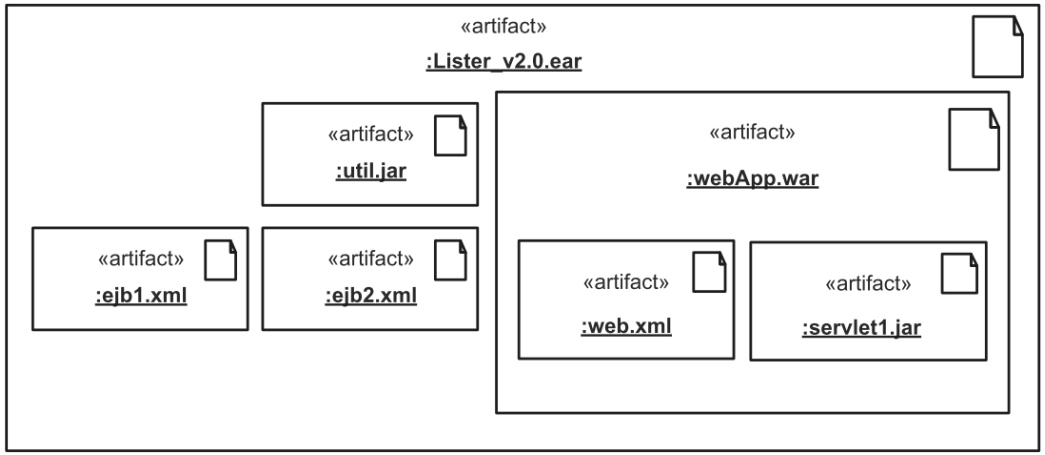
Komponentendiagramm: Elemente

Artefakt:

- An artifact is the specification of a physical piece of information that is used or produced by a software development process, or by deployment and operation of a system.
- Schlagen Brücke von der “logisch” geprägten Designphase zur Realisierungsphase

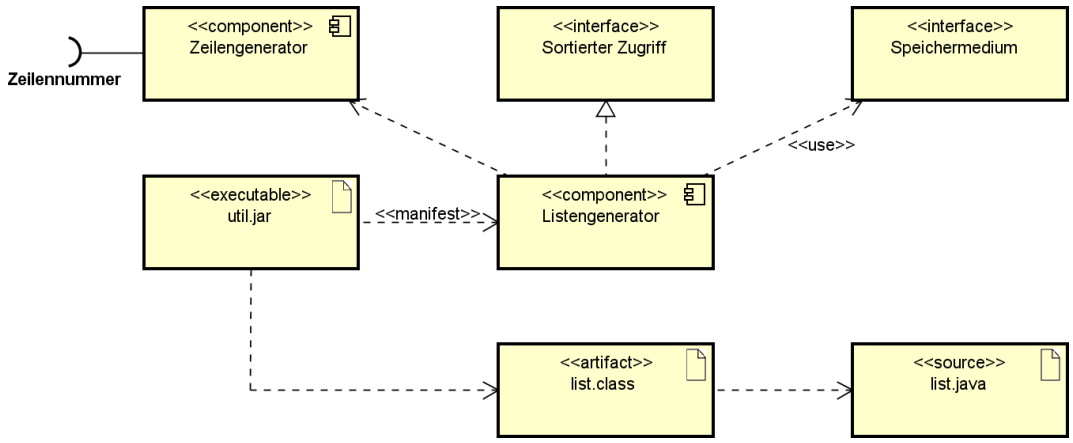
Beispiele:

- Realisierung einer Komponente
- Quelldateien
- Skripte
- Binäre Ausführungsdateien
- Datenbanktabellen
- Emailnachricht, etc.



Komponentendiagramm: Elemente

Abhängigkeiten zwischen Komponenten und Artefakten:



Abhängigkeiten:

<<manifest>>: Verbindet mind. 1 Artefakt mit 1 Komponente

Bsp.: util.jar realisiert Listengenerator

<<use>>: Verbindet Komponente mit den von ihr benötigten/verwendeten Schnittstellen

Realisierungsbeziehung:

Bsp.: Listengenerator implementiert Schnittstelle "Sortierter Zugriff"

- Zusätzliche Stereotype können Charakter von Artefakten konkretisieren
  - <<executable>>
  - <<source>>
  - <<file>>, <<document>>, <<library>>, …

Komponentendiagramm: Elemente

Abhängigkeiten zwischen Komponenten und Artefakten:

- Das Schlüsselwort «manifest» verbindet mindestens ein Artefakt mit der realisierenden Komponente. Eine Komponente kann dabei von mehr als einem Artefakt gleichzeitig manifestiert werden.
  Im Beispiel wird die Komponente Listengenerator durch das ausführbare Artefakt util.jar manifestiert (realisiert)

- Das Schlüsselwort «use» verbindet eine Komponente mit den von ihr benötigten und verwendeten Schnittstellen.
  Im Beispiel verwendet die Komponente Listengenerator die Schnittstelle Speichermedium

- Die Realisierungsbeziehung verbindet eine Komponente mit den durch sie umgesetzten und angebotenen Schnittstellen.
  Im Beispiel implementiert die Komponente Listengenerator die Schnittstelle „Sortierter Zugriff“