

# Wirtschaftsinformatik 1. Semester

**Informatik:** Beschäftigt sich mit der Automatisierung durch Computer

**Wirtschaftsinformatik:** Integriert BWL in die Informatik (dazu Teile Recht und Ingenieurwissenschaften)

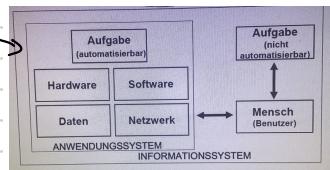
**Anwendungssystem:** Führen Vollautomatisiert Aufgaben durch, dabei kommunizieren sie mit Nutzern, Partnerystemmen oder Maschinen



**Anwendungssoftware:** Das zur Anwendung gebrauchte Programm (ohne Hardware etc.)

**Informationssysteme:** Software die Informationen verwaltet und ihre Zielsetzung ist es:  
(soziotechnisch)  
↓      ↓  
Sozial    technik  
• richtige Informationen  
• im richtigen Umfang  
• in der richtigen Form  
• zur richtigen Zeit  
• am richtigen Ort ... zur Verfügung an Stellen

Komponente eines Informationssystems:



**Soziotechnisches Design:** Beschreibt ein Benutzer-freundliches Design

**IT-Infrastruktur:** Alle Materiellen und Immateriellen Güter, die den Betrieb von Software ermöglichen  
↳ Hardware, Bauliche Einrichtungen, Netzwerktechnik, Tastatur etc., Strom

**IT-Architektur:** Komponenten: Prozesse, Anwendungssoftware, Betriebssysteme, Hardware, Datenbanksysteme, Netzwerk, Speicher

Von Strategie zur Architektur

↳ (Geschäfts)strategien, → IT-Strategien → IT-Architekturen  
Geschäftsprozesse  
IT-Market (innovativer)

• Eig nur das auswendig lernen, Komponente eines Informationssystems und die von Informationssystemen sind besonders wichtig

# Cloud-Computing

**Definition:** Bezeichnet das dynamisch an den Bedarf angepasste Anbieten, Nutzen und abrechnen von IT-Dienstleistungen über ein Netz. Angebot und Nutzung dieser Dienstleistungen erfolgen dabei über definierte technische Schnittstellen und Protokolle. Die Dienstleistungen umfassen das komplette Spektrum der Informatik und beruheln u.a. Infrastruktur, Plattformen, Software.

**NIST Cloud Computing:** Eine weitere Definition von Cloud-Service

**Eigenschaften davon:**

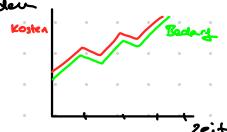
- On-demand self-service = Verfügbar ohne Interaktion mit dem Service provider

- Broad Network Access = Service ist an keinen bestimmten Client gebunden  
Sondern frei zugänglich

- Resource Pooling = Ressourcen liegen in einem Pool, aus dem viele Anwender sich bedienen können. Anwender wissen hier nicht, wo diese Ressourcen sich genau befinden, können den Speicherort aber vertraglich festlegen.

- Rapid Elasticity = Service kann schnell und elastisch zur Verfügung stehen

- Measured Services = Ressourceneinsatz kann überwacht und gemessen werden und entsprechend berechnet auch den Cloud-Anwendern zur Verfügung gestellt werden

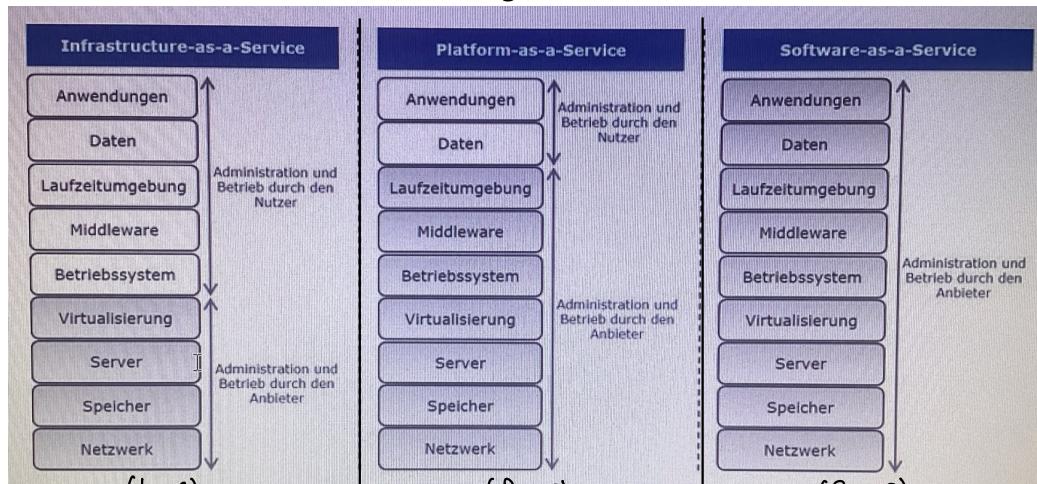


**Zusammengefasst:**

- Flexibel nutzbar

- Anwender zahlt nur, was er benötigt →

## Servicemodelle des Cloud Computings:



Hier haben Kunden die volle Kontrolle über die Infrastruktur und können eigene Anwendungen und Betriebssysteme installieren und konfigurieren. IaaS stellt also nur die grundlegende Infrastruktur bereit.

PaaS bietet eine Entwicklungs- und Bereitstellungsumgebung. Kunden müssen sich hier keine Gedanken über Infrastrukturen machen und können Anwendungen auf einer vom Anbieter gehosteten Plattform ausführen. I.o.R bietet der CSP eigene Werkzeuge an

Hier werden Anwendungen über das Internet bereitgestellt und geliefert. Kunden mieten die Anwendung und laden über Internet diese herunterladen zu ihrem Zugriff. Der Anbieter ist verantwortlich für Wartung, Sicherung der Daten etc.

# Informationsmanagement

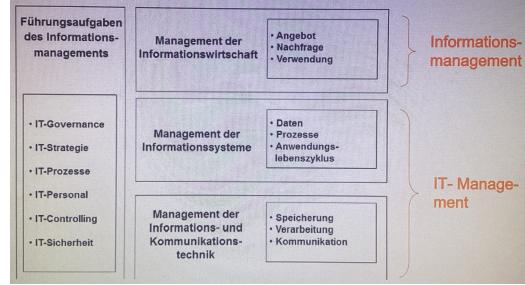
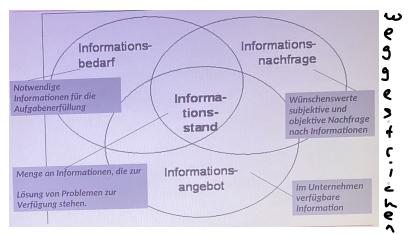
**Definition:** Ist die effektive und effiziente Bewirtschaftung des Produktionsfaktors Information in Organisationen.

Die Planung und Gestaltung der Informationsverarbeitung in Unternehmen erfolgt aus Sicht des Informationsmanagements mit dem Ziel der Optimierung der Informationsversorgung und -nutzung in allen Unternehmensbereichen.

Informationsmanagement ≠ IT-Management

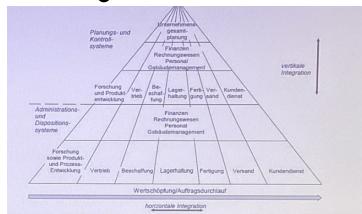
Was genau ist Informationsmanagement?:

Wozu?:



Zielsetzung: Informationen ... • Korrekt  
• im richtigen Umfang  
• in richtiger Form  
• zur richtigen Zeit  
am richtigen Ort ... zur Verfügung zu stellen

Informationsfluss im Unternehmen:



Informationsprozess: Informationsgewinnung

unterstützen die betrieblichen Geschäftsprozesse, Materialprozesse, etc.



Informationseingabe



Informationsverarbeitung ↔ Informationsspeicherung



Informationsausgabe

# Wissensmanagement

**Definition:** Bezeichnet den gesamten Prozess der systematischen Herleitung, Strukturierung, Darstellung, Verteilung, Suche und Speicherung von Wissen. (Problem von Wissen: Hat nicht jeder)

**Wissensmanagementsysteme:** Bieten eine IT-Unterstützung zur Nutzung des im Unternehmen und außerhalb des Unternehmens vorhandenes Wissens.

Beispiele: Datenbanken, Internet, Intranet, Suchmaschinen...

## Arten von Informationssystemen

Man unterscheidet Informationssysteme in operative und analytische Systeme

**Operative Informationssysteme:** Unterstützen den Benutzer bei betriebswirtschaftlichen Routinearbeiten. Geschäftsprozesse mit gleichförmigen Abläufen und einer großen Anzahl zu verarbeitender Daten sollen damit effizienter werden. Im Vordergrund stehen Rationalisierungseffekte und Kosteneinsparung

**Analytische Informationssysteme:** Durch sie erhalten Manager bei der Entscheidungsvorbereitung geeignete Hilfestellung. Im Vordergrund stehen Analysen, Berechnungen, Kennzahlen sowie grafische Aufbereitungen der Verarbeitungsergebnisse

## ERP-Systeme und Branchensoftware

**Definition:** Enterprise - Resource - Planning - Systeme sind integrierte betriebswirtschaftliche Standardsoftware, die sämtliche oder wesentliche Teile der Geschäftsprozesse innerhalb des Unternehmens unterstützen, wie z.B. Produktion, Vertrieb...

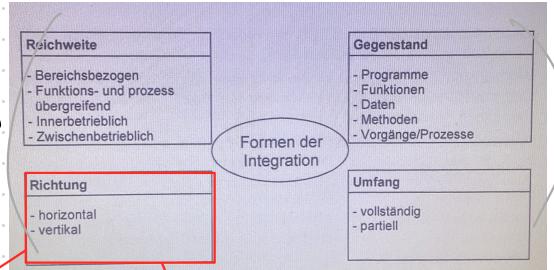
Im Vordergrund steht die Integration und das Zusammenwirken der verschiedenen Aufgaben über eine gemeinsame Datenbank, wodurch Datenredundanzen vermieden und Abteilungsübergreifende Geschäftsprozesse unterstützt werden.

# Funktionen und Aufgaben:

Aufgaben	Funktionen
Administration Datenhaltung für Geschäftsvorfälle	Fertigung - Bestellsteuerung - Materialbedarfsplanung - Einkauf - Produktionsplanung
Disposition - Automatisierung von Routinevorgängen	Vertrieb - Anforderungsangang - Rechnungsabrechnung - Dokumentenverarbeitung
Information - Kennzahlendarstellung	Rechnungswesen Forderungen und Verbindlichkeiten Buchhaltung Abschlusserstellung und Überwachung
Analysse - Auswertungen, Zeitreihenmodelle	Finanzwesen - Liquiditätsmanagement - Finanzplanung
	Personalwesen - Lohn- und Gehaltsberechnung - Zuschläge und Prämien

## Die Integration von ERP-Systemen:

Der Name Enterprise - Resource - Planning ist eigentlich falsch gewählt, da vielmehr die integrative Abläufung von Geschäftsprozessen im Mittelpunkt von ERP-Systemen steht!

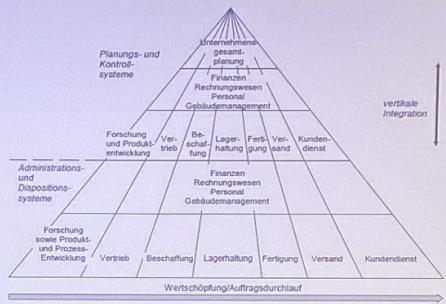


Der Begriff der Integration

### Horizontale und vertikale Integration (1/2)

Horizontale Integration
<ul style="list-style-type: none"> <li>Verschiedene Abteilungen auf einer Hierarchieebene</li> <li>Optimierung von Material- und Informationsflüssen</li> <li>Abteilungs- und funktionsübergreifende Abläufe</li> </ul>
Vertikale Integration
<ul style="list-style-type: none"> <li>Integration über verschiedene Hierarchieebenen</li> <li>Umsetzung von analytischen Aufgaben</li> </ul>
Integrationsgegenstände
<ul style="list-style-type: none"> <li>Daten</li> <li>Funktionen</li> <li>Vorgänge und Methoden</li> </ul>

### Horizontale und vertikale Integration (2/2)

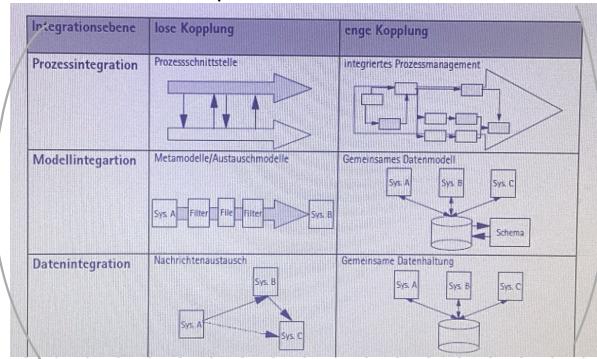
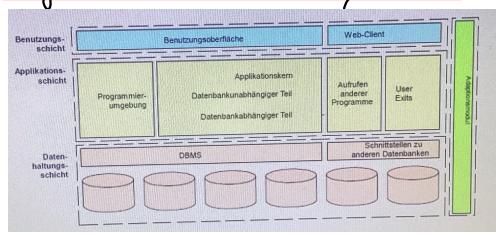


© Prof. Dr.-Ing. Norbert Gronau, Universität Potsdam

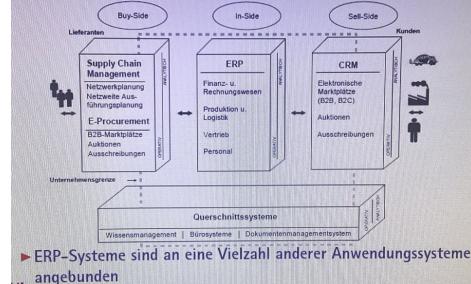
Mertens 2007, S.

## Formen der Integration zwischen Informationssystemen :

### Aufbau eines ERP-Systems



# Aufgabeverteilung betrieblicher Informationssysteme



## Vor und Nachteile einer standardisierten IT

### Nachteile

#### Standardisierung



- Verlust von Wettbewerbsvorteilen durch Vereinheitlichung
- Fördert Stabilität und Weiterentwicklung
- Individuelle Anpassungen sind schwierig umzusetzen

### Vorteile

#### Individualisierung



- Lange Spezifikations- und Entwicklungsphasen
- Hohe Kosten
- Kein Funktionsübergang und keine Anforderungslücken
- Abhängigkeit vom Entwickler

Die Einführung von ERP-Systemen kann mit 2 Verfahren gemacht werden.

- 1: Big Bang: Alles auf einmal einführen
- 2: Sukzessiv: Nach und nach Module einführen

## Make or Buy Entscheidung

(Bezogen auf Standardsoftware oder Eigenentwicklung)

### Warum Standardsoftware?

#### Vorteile

#### Nachteile

- | Vorteile   | Nachteile   |
|--|---|
| <ul style="list-style-type: none"><li>• Umfassendes Leistungssprofil</li><li>• Hohes Reifegrad</li><li>• Hoher Integrationsgrad</li><li>• Gesäärleistung</li><li>• Internationalität</li></ul> | <ul style="list-style-type: none"><li>• Funktionsübergang und Anforderungslücken</li><li>• Releasewechsel vorgegeben</li><li>• Unternehmensfreundliche Terminologie</li></ul> |

### Zeitvorteile durch Standardsoftware

#### Vorgehen bei Individualentwicklung

Problem-erkenntnis	Anforderungs-spezifikation	Entwicklung/Test	Anpassung

#### Vorgehen bei Einführung von Standardsoftware

Problem-erkenntnis	Anforderungs-spezifikation	Alternativen-auswahl	Anpassung

► Standardsoftware ist in ihrer grundlegenden Form sofort verfügbar!

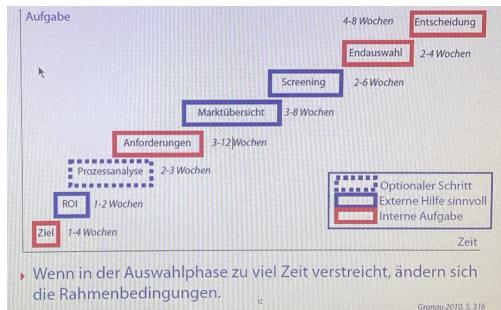
## Flexibilität durch Standardsoftware

### Vorteile

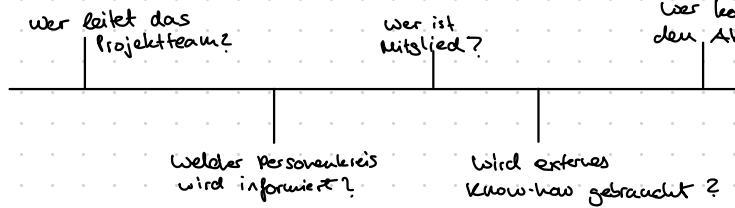
### Nachteile

- |   |  |
|---|--|
| <ul style="list-style-type: none"> <li>• Parameterisierung statt Programmierung</li> <li>• Programmveränderung und Ergänzung möglich</li> <li>• Unabhängigkeit von einzelnen IV-Mitarbeitern</li> </ul> | <ul style="list-style-type: none"> <li>• Systembeherrschbarkeit, Anpassung</li> <li>• Verlust der Release-Fähigkeit</li> <li>• Abhängig von Hersteller</li> <li>• Verlust von Entwicklungs-Know-how</li> </ul> |
|---|--|

## Vorgehen und Dauer der Auswahlphase von Anwendungssystemen

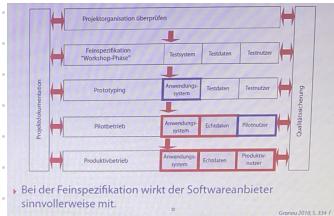


Zur diesem Entscheid braucht man ein Projektteam, dieses wird folgendermaßen gebildet:



Beim Entscheid kann es auch zu Fehlern kommen:

Ist es nur entschieden folgt dieser Ablauf:  
Einführung:



## Fehler bei der Auswahl von Standardsoftware

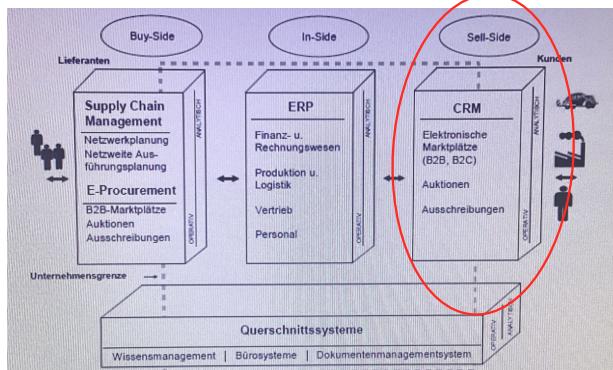


## Zwischenbetriebliche Informationssysteme

Hier gibt es:

1. CRM-Systeme (Customer Relationship Management Syst.)
2. SCM-Systeme (Supply Chain Management Systeme)

# CRM Systeme



## Wesentliche Punkte von CRM

1. **Kundenorientiert**  
Ziel: Ausrichtung von Geschäftsprozessen nach dem Kundenverhalten

2. **Einsatz von Informations und Kommunikationstechnologien**  
Zweck: Gewinnung von Daten über das Kundenverhalten, um sich darauf richten zu können

## CRM-Systeme (Software):

- Speziell auf das Kundenmanagement zugeschnittene Software
- Datenbankanwendung, die eine strukturierte und geöffnete automatisierte Erfassung sämtlicher Kundendaten ermöglicht
- Unterstützt die Arbeit von Vertriebsmitarbeitern
- In größeren Unternehmen: Nutzung der Daten des CRM-Systems in einem Data-Warehouse für eine weitgehende Auswertung mittels Data-Mining

## Software:

- Proprietäre Software: Software auf dem Markt, basierend auf Standardsoftware
- Heute: Zuwachs von cloudbasiert, SaaS- und Open-Source Lösungen
- CRM-Lösungen für besondere Anforderungen sind meist individuelle Lösungen

## Kosten:

- Beschaffung der Software
- Pflege der Daten
- Customizing

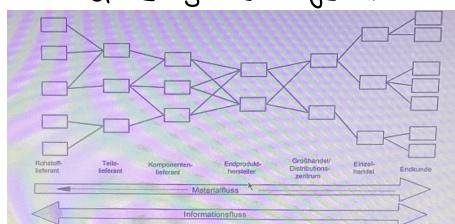
## Integration CRM → ERP:

Zur Vermeidung von Datenredundanzen, aufßerdem wird die Datenqualität besser, da CRM-Systeme so auf ERP-Daten zugreifen kann.

## SCM - Systeme

**Definition SCM:** Die Planung, Steuerung und Kontrolle von unternehmens-übergreifenden Wertschöpfungssystemen mit Netzwerkstruktur, dessen Institutionen und Prozesse über Alter, Finanz und Informationsflüssen in Beziehung stehen

↓  
Steuerung der Lieferkette

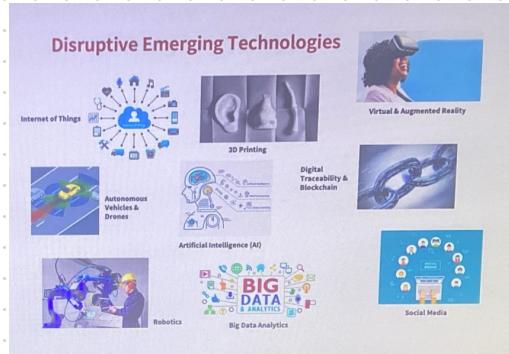


# Digitale Transformationen (Industrie 4.0, neue Geschäftsmodelle etc.)

**Definition:**

- Übertragung von analogen Informationen in Digitale
- Übertragung von Menschenanfängen in Computeranfängen

**Industrie 4.0:** Innovative, digitale Unternehmensführung und Steuerung



# Informatik

## Wirtschaftsinformatik

Ziel →

maschinelle (rechnergestützte) Verarbeitung / Übermittlung von Daten

Entwurf, Entwicklung, Nutzung von IT in Wirtschaft und Verwaltung (Rechnungswesen)

simulierte Vollautomation, Ersatz da schneller, günstiger, sicherer, Brückenfunktion

## Anwendungssystem (AS)

System, das va Aufgaben durchführt (inkl. Hardware)

## Anwendungsofware

Programm

## Informationssystem

computergestütztes Anwendungssystem, sozio-technisch, Ziel: richtige Information

Mensch -> Maschine

## Komponenten

(Aufgabe, Hardware, Software)

Aufgabe

AS -> Mensch

Mensch ↔ Technik  
Organisation

## IT-Infrastruktur

### Sichten

alle Güter für ein AS

Systemsoftware  
Hardware  
Basisinfrastruktur  
Technische Sicht

Virtuelle  
Maschine

+ Institution + Person  
Sicht Informationsmanagement

→ IT-Markt

neue  
Technologien  
neue  
Anwendungen  
Veränderung  
der Konkurrenz

## Architektur

Geschäftsstrategie / -prozesse → It-Strategie → IT-Architektur (System, Application, Service)

muss unterstützen

## Cloud Computing

### Charakteristik

Modell, über ein Netz auf Rechnerressourcen zugreifen

On-Demand Self-Service

Broad Network Service

Resource Pooling

Rapid Elasticity

Measured Services

⇒ flexibel, kostengünstiger

### Servicemodell

IaaS PaaS SaaS

## Informationsmanagement

ziel:

effektive und effiziente Bewirtschaftung von Informationen

Optimierung

Unterschied It-Management = Informationssysteme + Informations- und Kommunikationstechnik

## Informationsprozesse

unterstützt Geschäftsprozesse

Informationsgewinnung → Informationseingabe → Informationsverarbeitung ↔ Informationsspeicherung

↓  
Informationsausgabe

## Wissensmanagement

Def: zweckorientierte Vernetzung von Info

Problem: Wissen schwer übertragbar

Prozess: Wissenstreppe nach North

Unterstützung: Intranet, Datenbanken

## operative und analytische Informationssysteme

↳ routine

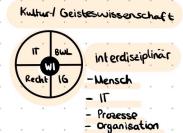
↳ Berechnungen,  
Kennzahlen

## ERP enterprise resource planning

Benutzerschicht  
Applikationsschicht  
Datenhaltungsschicht

standard individual

Make or Buy?



# Systemanalyse

Wie wird Software entwickelt?

(im Bereich der Analyse und des Entwurfs werden UML (Unified modeling language) Diagramme angewandt.)

Davon gibt es verschiedene Arten: Strukturdiagramme und Verhaltensdiagramme

wir beschäftigen uns mit:

- Klassendiagramme
- Objektdiagramme

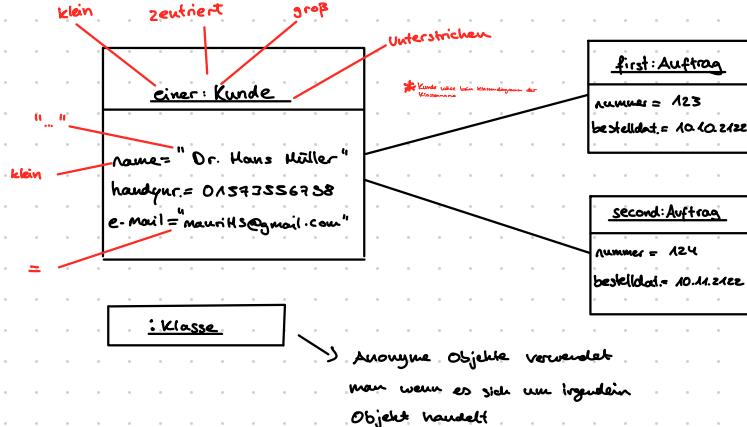
- Use-case Diagramme
- Aktivitätsdiagramme

## Objektdiagramme

**Definition Objektidentität:** jedes Objekt besitzt eine Identität, die es von allen anderen Objekten unterscheidet. Selbst wenn zwei Objekte aufgrund derselben Attributwerte bestehen, haben sie eine unterschiedliche Identität. Im Speicher wird die Identität durch unterschiedliche Adressen realisiert.

**Definition Objektdiagramm:** UML-Diagramm, das Objekte und Objektbeziehungen darstellt. Sie werden verwendet, um einen Ausschnitt des Systems zu einem bestimmten Zeitpunkt zu modellieren.

Aufbau:

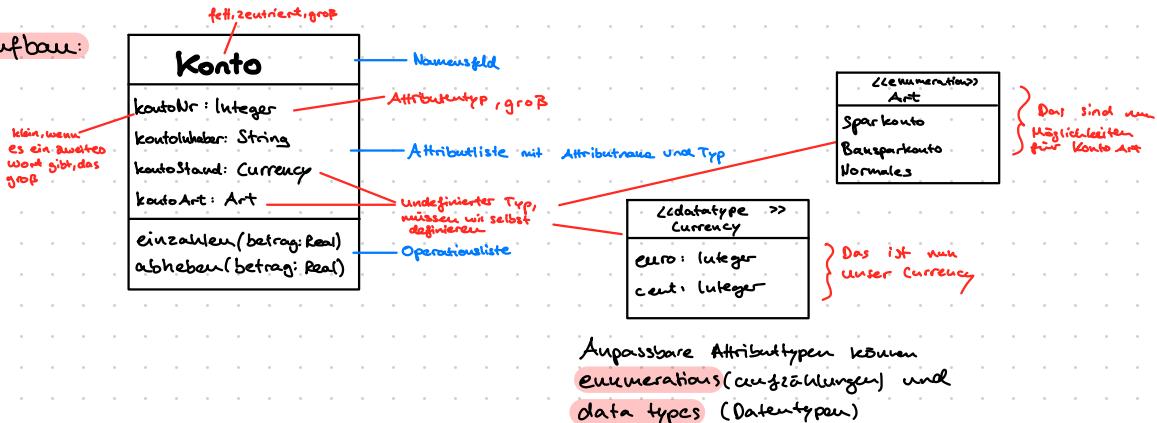


- Die Representation dieser Daten nach außen sollte verborgen sein. Ist dies der Fall, dann realisiert ein Objekt das Geheimnisprinzip. Es ist durch operationen verborgen

# Klassendiagramme

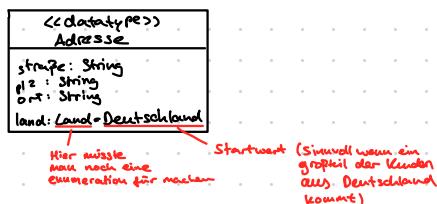
**Definition:** UML-Diagramm, das die Klassen mit Attributen und Operationen, die Generalisierungstrukturen und die Assoziationen zwischen Klassen darstellt

**Aufbau:**

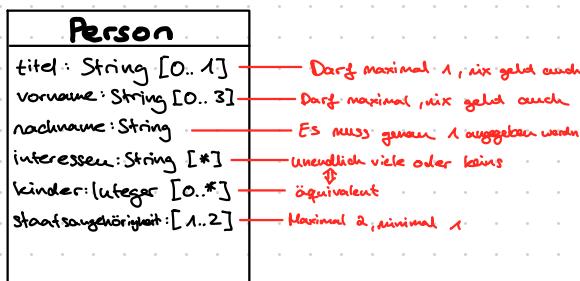


Attribute können auch einen **Anfangswert** an einem neu festgelegten Objekts fest.

**Beispiel :**



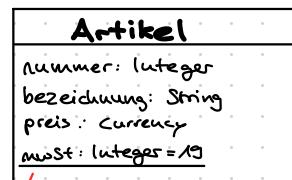
**Multiplicität von Attributen:**



haben. Dieser legt den Startwert von

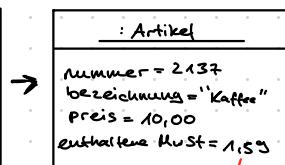
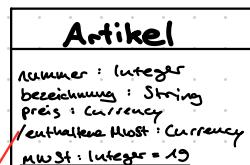
**Klassenattribute**: Attribut, das auf alle anderen Attribute trifft.

**Beispiel:**



- Unterstrichen**: Hier würde es dafür stehen, dass beispielsweise ein Geschäft auf alle Angebote 18% MwSt erhält.
- Wird nicht ins Objektdiagramm übernommen**.

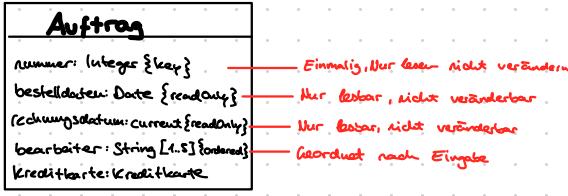
**Abgeleitetes Attribut**: Kann andere Attributwerte berechnen. Ändert sich ein Wert, so ändert sich das des abgeleiteten Attributs.



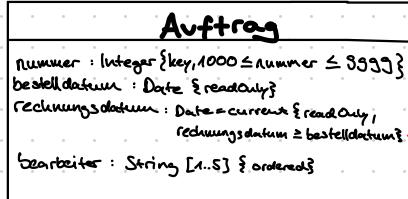
**→ bezeichnung von abgeleiteten Attributien**

**abgeleitetes Attribut**, verändert sich mit **preis** (10,00 | 1,53)

**Eigenschaftswerte:**



## Einschränkungen:

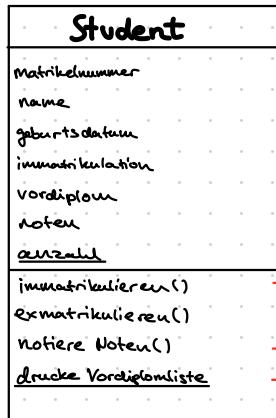


Nummer muss zwischen 1000 und 9999 sein

Rechnungsdatum muss größer (später) als das Bestelldatum sein

- Operationen:**
1. Objektoperationen = Beziehen sich auf ein einzelnes Objekt
  2. Konstruktorenoperationen = Erzeugt ein neues Objekt und führt Datenerfassung durch
  3. Klassenoperationen = Kann nicht auf ein einzelnes Objekt angewendet werden

**Beispiel:**



Erhöht eine Operation indirekt etwas mit, macht es das nicht zur Klassenoperation, es kann auch eine Konstruktorenoperation sein.

— Konstruktorenoperation → erzeugt neues Objekt

— Objektoperation → Bezieht sich nur auf Noten

— Klassenoperation → Bezieht sich auf mehrere Objektattribute

**Pakete:** Gleiche Dinge können in Pakete gepackt und benannt werden  
(Alle enumerations etc.)

**Kommunikation:** Kommen an Element um diese zu erklären

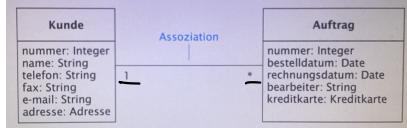
Beispiel:



## Klassenbeziehungen

**1. Assoziationen:** Modelliert Objektbeziehungen zwischen Objekten einer oder mehrerer Klassen.

**Beispiel:**



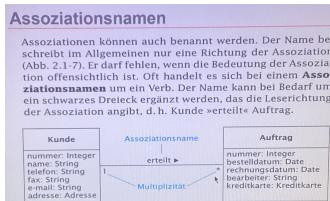
Ein Kunde kann beliebig viele Aufträge aufgeben

## Multipizität von Assoziationen

1	1	genau 1 Objekt
0..1	0 oder 1 Objekt	
*	0 bis viele Objekte	
3..*	3 bis viele Objekte	
0..2	0 bis 2 Objekte	
2	genau 2 Objekte	

# Assoziationen haben Namen oder Rollennamen

## Assoziationsnamen:

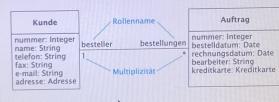


## Rollennamen:

### Rollennamen

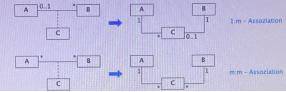
#### Rollenname (role name)

Beschreibt, welche Bedeutung ein Objekt in einer Assoziation besitzt.

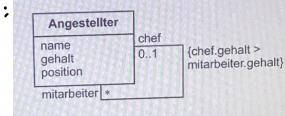


Assoziation mit Rollennamen.

## Auflösen einer Assoziationsklasse



Assoziationen können auch **mit sich selbst** verbinden sein, wenn der Chef zum Beispiel immer ein Angestellter ist:



## Navigierbarkeit von Assoziationen

Assoziationen sind zunächst unspezifiziert. Jedoch er möglichst soll es unidirektionale ( $\rightarrow$ ) und bidirektionale Verbindungen ( $\leftrightarrow$ ) festzulegen

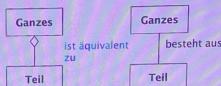
**Beispiel bidirektional:** Person A ist mit Person B befreundet, also ist Person B auch mit Person A befreundet

**Beispiel unidirektional:** Ein Lehrer A bestimmt über Schüler A über Schüler A nicht über Lehrer A

## Weitere Arten von Assoziationen:

### Aggregation

Eine **Aggregation** (shared aggregation) liegt vor, wenn zwischen den Objekten der beteiligten Klassen (kurz: den beteiligten Klassen) eine Rangordnung gilt, die sich durch »ist Teil von« bzw. »besteht aus« beschreiben lässt (Abb. 2.3-5). Man spricht auch vom Ganzen und seinen Teilen (whole part). Das Ganze wird auch als Aggregatklasse, das Teil als Teilklasse bezeichnet.



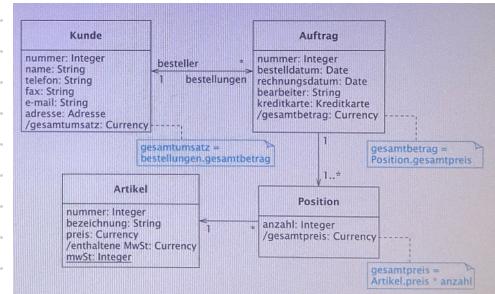
Aggregation – eine besondere Form der Assoziation

### Komposition

Eine **Komposition** (composition, composite aggregation) ist eine starke Form der Aggregation. Auch hier muss eine whole-part-Beziehung vorliegen. Darüber hinaus muss gelten:

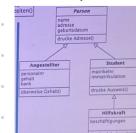
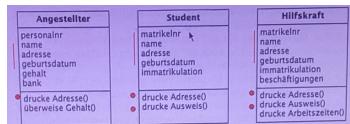
- Jedes Objekt der Teilklasse kann – zu einem Zeitpunkt – nur Komponente eines einzigen Objekts der Aggregatklasse sein, d. h. bei der Aggregatklasse eingeschlossene Multipizität darf nicht größer als eins sein. Ein Teil darf jedoch in einer anderen Ganzen enthalten werden.
- In der Abb. 2.3-6 kann jede Teil zu einem Zeitpunkt nur in einem Dateiordnern enthalten sein, aber zwischen verschiedenen Ordnern hin und hergeschoben werden.
- Das Gesamte ist für die Erzeugung und Verwaltung seiner Teile zuständig.

- Die Lebensdauer des Teils ist die Lebensdauer des Ganzen gehoben. Das heisst, der Teil wird erst dann gelöscht, wenn der entsprechende Dateiordnern gelöscht werden. Beim Löschen eines Dateiordnern werden alle darin enthaltenen Dateien gelöscht.
- Gilt, dass die Funktionalität des Ganzen auch für seine Teile. Wird beispielsweise ein Dateiorndner kopiert, dann werden auch alle darin enthaltenen Dateien kopiert.



### Generalisierung:

## Zusammenfassung von Klassen



# Use-Case Diagramme

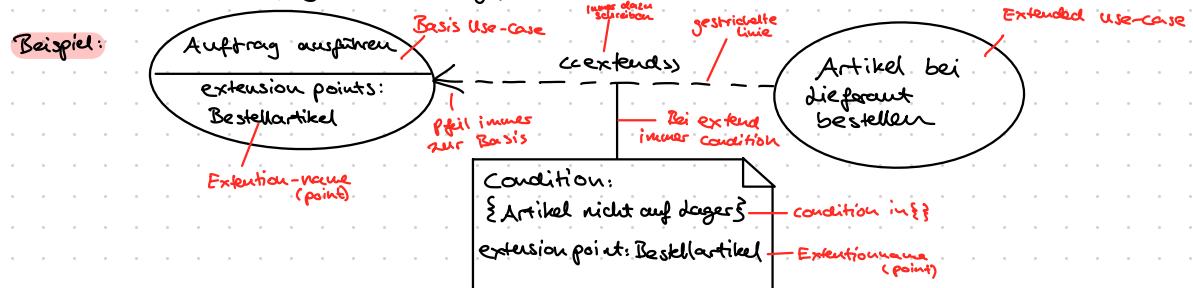
**Definition Use-Case:** Funktionalität des Softwaresystems, die ein Akteur ausführen muss, um ein gewünschtes Ergebnis zu erhalten oder ein Ziel zu erreichen.

**Definition Use-Case Diagramm:** UML-Diagramm, das die Beziehungen zwischen Akteuren und Use-Cases in einem Softwaresystem modelliert.

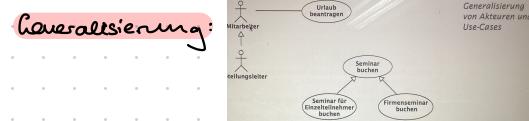
**Akteure:** Rolle, die ein Benutzer spielt. Akteure befinden sich immer außerhalb vom System.



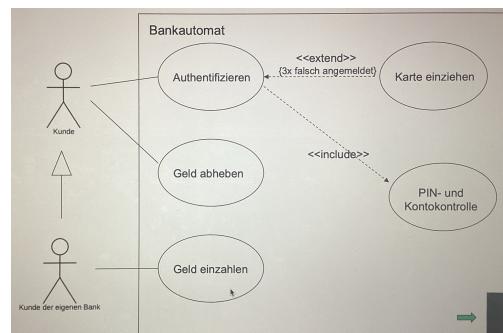
**extend Beziehung:** Erweitert einen Use-Case A um einen optionalen Use-Case B, der unter einer gewissen Bedingung (condition) ausgeführt wird.



**Include-Beziehung:** "Ein Teil von" Beziehung. Um Use-Case A auszuführen wird hier Use-Case B benötigt. Die Beziehung wird verwendet, wenn 2 oder mehr Use-Cases ein gemeinsames Verhalten haben. → andersrum als bei extend



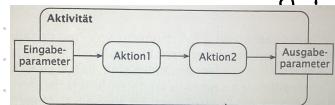
Include und extend = YouTube gucken



Beispiel

# Aktivitätsdiagramme

**Definition:** UML Diagramm, das eine Aktivität durch ein großes Rechteck mit abgerundeten Ecken modelliert.



**Zusammenführung:** Führt Kontrollflüsse, die durch Entscheidungen entstanden sind zurück.

**Entscheidung:**

**Splitting:** Parallelie Ausführung →



**Start und Endknoten**

● → Start der Aktivität

→ ● Ende der Aktivität

→ × Ende des aktuellen Pfades im Diagramm

**Synchronisation:** Führt Splitting wieder zusammen nachdem beide gesplittete Aktivitäten ausgeführt wurden

**Beispiel:**

