

Fakultät für Verkehrs- und Maschinensysteme (Fakultät V)
Institut für Werkzeugmaschinen und Fabrikbetrieb
Fachgebiet Industrielle Informationstechnik
Prof. Dr.-Ing. R. Stark

Einführung in die Informationstechnik für Ingenieure

Klausurvorbereitung SS 2013



Klausur

Die Klausur setzt sich zu gleichen Teilen:

- aus der VL (theoretische Fragen zum VL-Teil der hier präsentierten Inhalte) und
- der UE (kleine Programmieraufgaben und theoretische Fragen) zusammen.
- Bedingung für die Teilnahme ist die erfolgreiche Absolvierung der Übungen
- Insgesamt 100 Punkte, 50 aus jedem Teil.
- Zum Bestehen sind 40 Punkte notwendig.

Termin der Klausur:

- **19.07.2013 von 12:30 – 14:30 Uhr**
- PTZ-S 001 (großer Hörsaal neben der Kantine)
- **Bitte spätestens um 12:15 da sein und Studentenausweis sowie etwas zum Schreiben mitbringen (keine Bleistifte). Sämtliches benötigtes Papier wird gestellt.**
- **Es wird keine Nachklausur stattfinden!**

Termin der Klausureinsicht:

- Voraussichtlich eine Woche später im VELC
 - Benachrichtigung erfolgt über ISIS.
 - Vorläufige Noten im ISIS-Tool mit den letzten 4 Ziffern der Matrikelnr.
-

- Muss über QISPOS oder in besonderen Fällen über das Prüfungsamt bereits geschehen sein!

- Rechnerinterne Informationsdarstellung
- Rechnerarchitektur
- Betriebssysteme
- Datenbanken
- Algorithmen
- Programmiersprachen
- UML & SysML
- Software Engineering
- Rechnernetze
- Datensicherheit in Rechnernetzen

- Rechnerinterne Informationsdarstellung
- Rechnerarchitektur
- Betriebssysteme
- Datenbanken
- Algorithmen
- Programmiersprachen
- UML & SysML
- Software Engineering
- Rechnernetze
- Datensicherheit in Rechnernetzen

Grundlagen - Begriffe

Information

Information ist eine Mitteilung, die sich aus einer räumlichen oder zeitlichen Folge physikalischer Signale zusammensetzt, die mit bestimmten Wahrscheinlichkeiten auftreten, und die beim Empfänger ein bestimmtes (Denk-)verhalten bewirkt (Duden).

Daten (nach DIN 44300):

Daten sind Zeichen oder kontinuierliche Funktionen, die Informationen aufgrund bekannter oder unterstellter Abmachungen zum Zwecke der Verarbeitung darstellen.

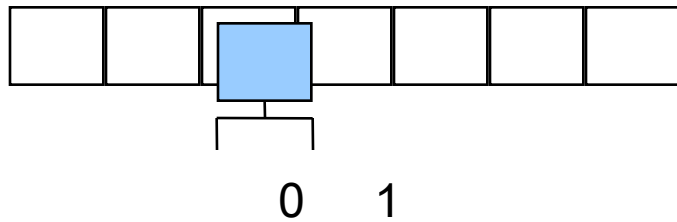
Daten sind zum Beispiel:

- **Zahlen:** 17, 25.3
- **Zeichen:** 'a', 'b'
- **Zeichenketten:** "Schlagbohrer"
- **Datumsangaben:** 11.3.2001

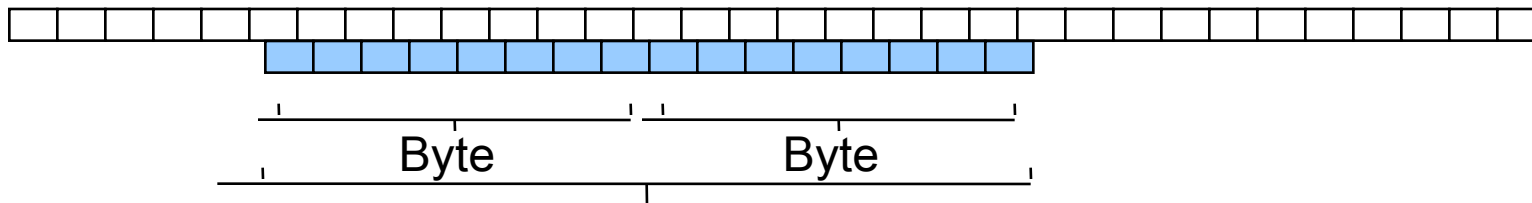
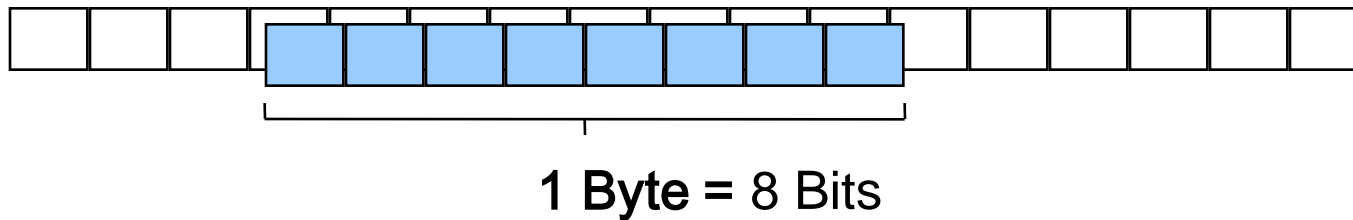
Nachrichten

Eine Nachricht lässt sich als Folge von Zeichen auffassen, die von einem Sender ausgehend in irgendeiner Form einem Empfänger übermittelt wird.

Binärcode



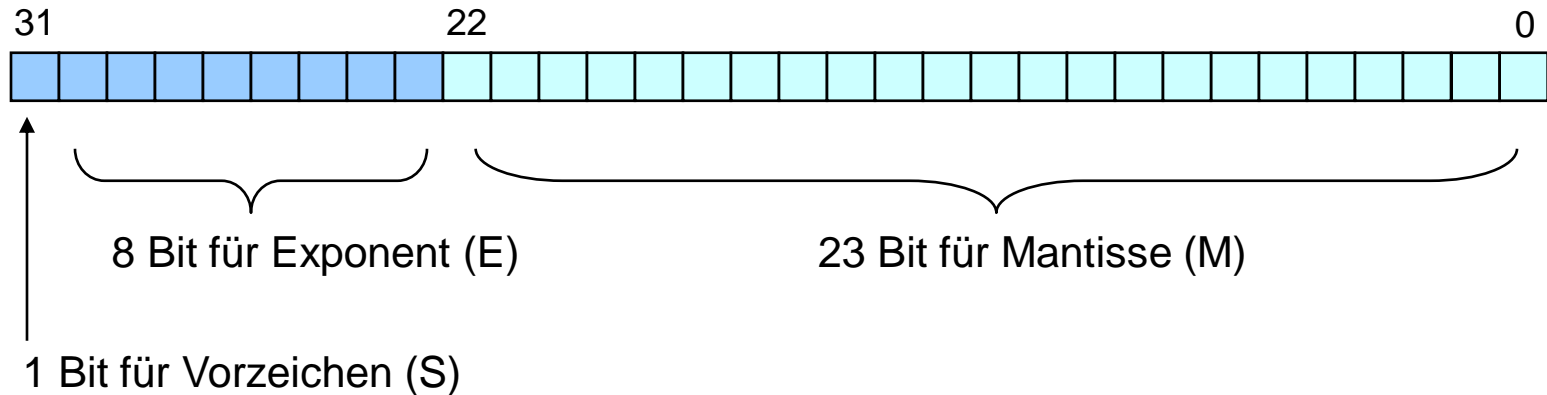
- Ein Bit (Binary Digit) ist die kleinste Darstellungseinheit des Binärcodes.
- Der Wert eines Bits wird als Zustand bezeichnet.
- Mit n Bits lassen sich 2^n verschiedene Zustände darstellen.



Ein **Wort** besteht aus einer logisch zusammengehörigen Folge von Bytes, üblicherweise 2, 4 oder 8.

Im Speicher ist ein Wort die kleinste adressierbare Einheit.

Gleitkommazahlen



- Darstellung einer 32-Bit Gleitkommazahl

$$X = (-1)^S * 2^{E-127} * (1.M)$$

X = Zahl E = Exponent

S = Vorzeichen M = Mantisse

Bias (Offset) des Exponenten: -127

Die Bias-Darstellung des Exponenten erlaubt einen einfacheren Binärvergleich zwischen zwei Gleitkommazahlen, indem die Bits lexikographisch verglichen werden können → <http://www.itf.fh-flensburg.de/lang/informatik/ieee-format.htm>

Eigenschaften der Gleitkommaarithmetik

- **Auslöschung** (Catastrophic cancellation)

Bei der Subtraktion zweier fast gleichgroßer Zahlen kann das Ergebnis falsch sein.

- **Absorbtion**

Addition oder Subtraktion einer betragsmäßig sehr viel kleineren Zahl ändert die größere nicht.

- **Unterlauf** (Underflow)

tritt auf, wenn das Ergebnis einer Rechenoperation betragsmäßig kleiner ist als die kleinste darstellbare Gleitkommazahl. Einige Compiler runden selbstständig nach 0.

- **Rundungsfehler** (Rounding error)

treten auf, wenn eine reelle Zahl im Gleitkommasystem nicht dargestellt werden kann und zur nächsten Gleitkommazahl gerundet werden muss, bei einem System zur Basis 2 etwa entsteht für 0.1 ein periodischer Bruch:

$$0.1 = \frac{1}{10} = \frac{0}{2^1} + \frac{0}{2^2} + \frac{0}{2^3} + \frac{1}{2^4} + \frac{1}{2^5} + \frac{0}{2^6} + \frac{0}{2^7} + \dots$$

Umwandlungen zwischen Zahlensystemen (1)

Beispiel 1: Wandeln Sie die Dezimalzahl 345,625 in eine Dualzahl um!

Lösung: $345,625 - 256 = 89,625$

$$89,625 - 64 = 25,625$$

$$25,625 - 16 = 9,625$$

$$9,625 - 8 = 1,625$$

$$1,625 - 1 = 0,625$$

$$0,625 - 0,5 = 0,125$$

$$0,125 - 0,125 = 0$$

$$1 \cdot 2^8 + 0 \cdot 2^7 + 1 \cdot 2^6 + 0 \cdot 2^5 + 1 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 + 1 \cdot 2^{-1} + 0 \cdot 2^{-2} + 1 \cdot 2^{-3}$$

$$= (101011001,101)_2$$

Umwandlungen zwischen Zahlensystemen (2)

Beispiel 2: □

Umwandlung einer Zahl zur Basis 10 in eine Zahl zur Basis 2.

$$456_{10} \Rightarrow X_2$$

$456_{10} : 2 =$	228	Rest : 0
$228_{10} : 2 =$	114	Rest : 0
$114_{10} : 2 =$	57	Rest : 0
$57_{10} : 2 =$	28	Rest : 1
$28_{10} : 2 =$	14	Rest : 0
$14_{10} : 2 =$	7	Rest : 0
$7_{10} : 2 =$	3	Rest : 1
$3_{10} : 2 =$	1	Rest : 1
$1_{10} : 2 =$	0	Rest : 1

Die umgewandelte Zahl ergibt sich durch Lesen der Restspalte von unten nach oben :

$$456_{10} = 111001000_2$$

Allgemein ausgedrückt muß die umzuwandelnde Zahl im Zahlensystem a wiederholt durch den Basiswert b dividiert werden.

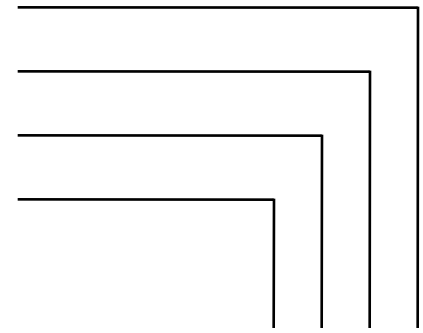
Umwandlungen zwischen Zahlensystemen (3)

Beispiel 3: Wandeln Sie die Dualzahl 1111100111000101 in eine Hexadezimalzahl um!

Lösung: $\underbrace{1111}_F \underbrace{1001}_9 \underbrace{1100}_C \underbrace{0101}_5 = F9C5$

Dual	Hex
0000	= 0
0001	= 1
0010	= 2
0011	= 3
0100	= 4
0101	= 5
0110	= 6
0111	= 7
1000	= 8
1001	= 9
1010	= A
1011	= B
1100	= C
1101	= D
1110	= E
1111	= F

Beispiel 4: Bestimmen Sie die Hexadezimalzahl zu 63941_{10} !

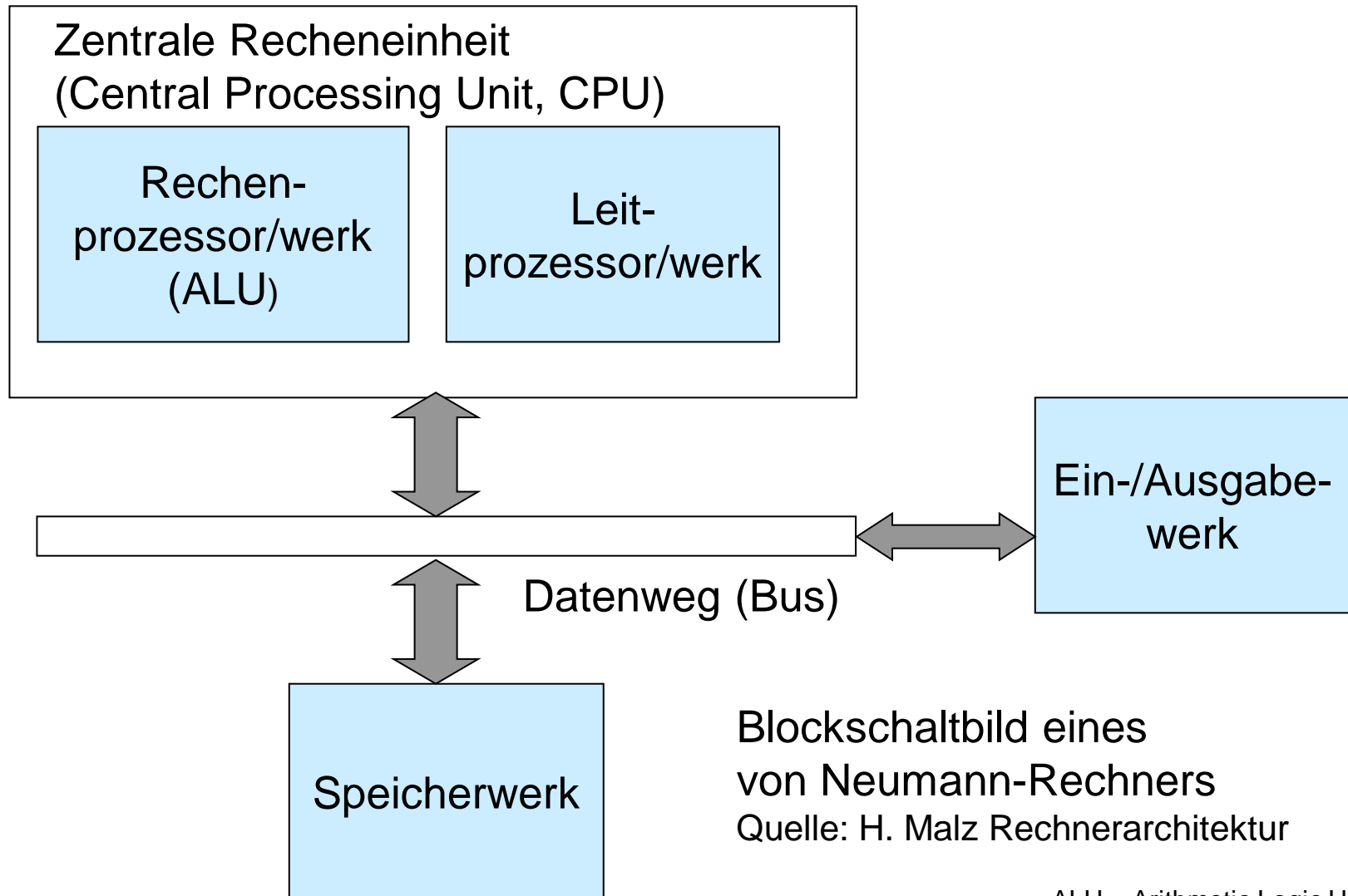
Lösung:
$$\begin{array}{rcll} 63941 : 16 & = & 3996 & \text{Rest } 5 \\ 3996 : 16 & = & 249 & \text{Rest } 12 \\ 249 : 16 & = & 15 & \text{Rest } 9 \\ 15 : 16 & = & 0 & \text{Rest } 15 \end{array}$$

$$63941 = F9C5$$



Zum Organisatorischen

- Rechnerinterne Informationsdarstellung
- **Rechnerarchitektur**
- Betriebssysteme
- Datenbanken
- Algorithmen
- Programmiersprachen
- UML & SysML
- Software Engineering
- Rechnernetze
- Datensicherheit in Rechnernetzen

Allgemeiner schematischer Aufbau eines von Neumann-Rechners



ALU = Arithmetic Logic Unit

Zentrale Recheneinheit (CPU)

- Die CPU besteht aus **Rechen-** und **Leitwerk**.
- Durch das Zusammenspiel von Rechenwerk und Steuerwerk werden Befehle geholt und ausgeführt.
- Im **Rechenwerk** findet die Bearbeitung der Daten statt, arithmetische Operationen, wie z.B. Addition, oder logische Operationen, wie z.B. AND werden ausgeführt. Es besitzt eine feste Verarbeitungsbreite bei 64-Bit-Systemen 64 Bit.
- Im **Leitwerk** werden Maschinenbefehle interpretiert und in Steuerkommandos umgesetzt.

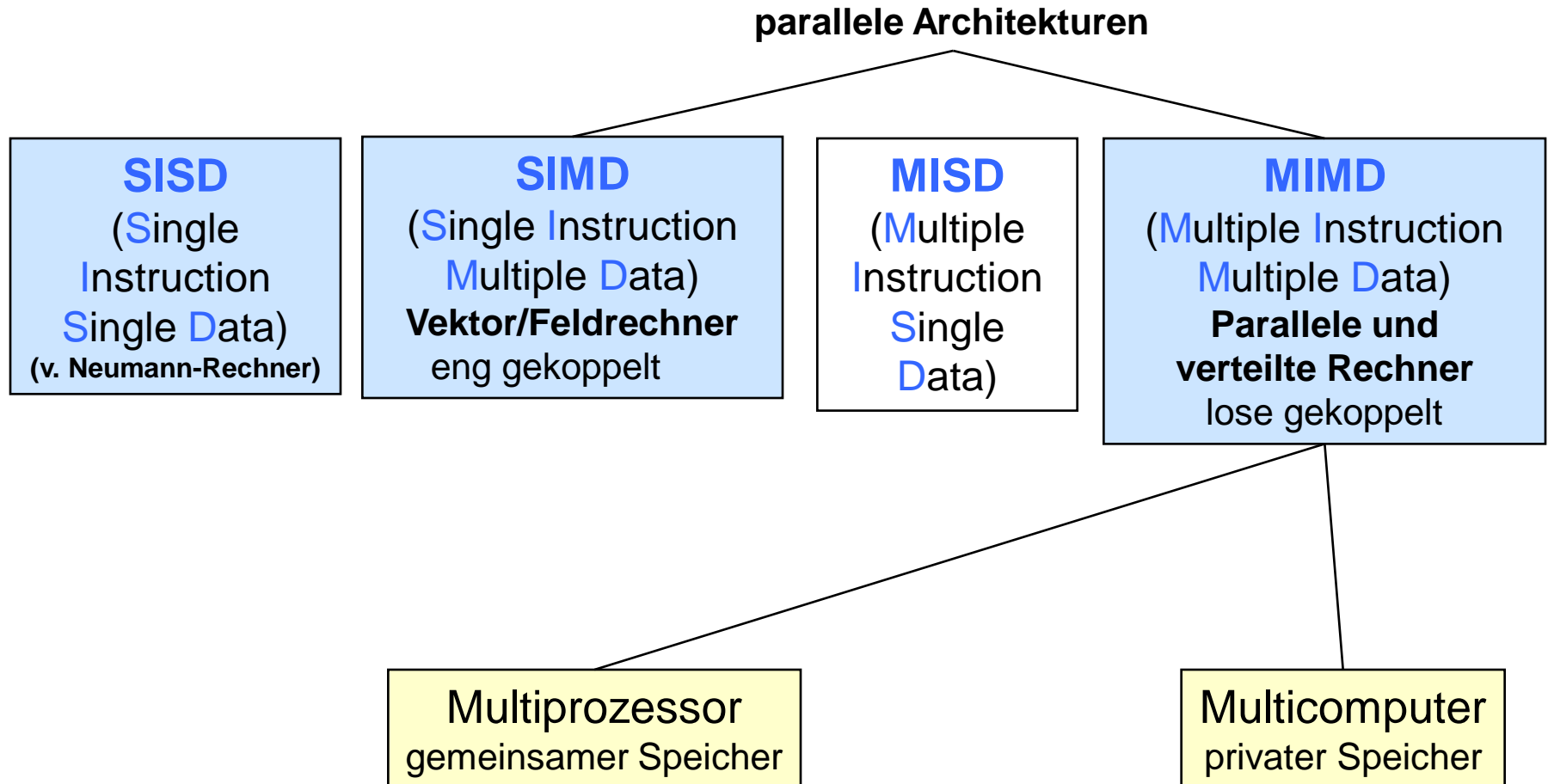
Problem: Von Neumann-Flaschenhals

- Transport
 - der Befehlsadresse vom Prozessor zum Speicher
 - des Befehls vom Speicher zum Prozessor
 - der Datenadresse vom Prozessor zum Speicher
 - der Daten vom Speicher zum Prozessor
- Früher benötigte die Befehlsausführung die meiste Zeit. Heute ist der enorme Verkehr von Befehlen und Daten auf den internen Datenwegen zu einem Engpass geworden.
- Die Kommunikation zwischen CPU und Speicher nennt man deshalb den **Von Neumann-Flaschenhals**.
Lösung: Einführung des Cache-Speichers

Weitere Leistungssteigerungstechniken (i. d. nächsten VL):

- Koprozessoren
- Pipelining

Rechnerklassifikation nach Flynn (1966)



Fehlerarten:

- Permanent (z.B.: Fehler in der Aufzeichnungsschicht)
 - Temporär (z.B.: Staub)
 - Einzelbitfehler (selten)
 - Burst-Fehler (mehrere Bit hintereinander)
 - Fehler in benachbarten Spuren (z.B.: Kratzer, häufig)
 - Statistische Fehlerwahrscheinlichkeit beim Speichern: 10^{-6} bis 10^{-3}
- ⇒ Ohne Fehlerkorrektur ergeben sich große Probleme

Methoden zur Fehlerbehebung

- Standardmethode: Speichern von zusätzlicher Information
- CRC (Cyclic Redundancy Checking)
- Rechteckcode
- Hammingcode

Beispiel: Rechteckcode zur Fehlersicherung

- Codeblöcke werden in Matrixform geordnet
- Jede Zeile erhält ein Paritätsbit
- Durch die Auswertung der Zeilen- und Spaltenparität wird der Fehler gefunden
- Beispiel:

0	0	1	1	0	
1	0	1	0	0	
1	1	0	1	1	
1	0	0	1	0	
1	1	0	1		

Bit-fehler →

Datenbit

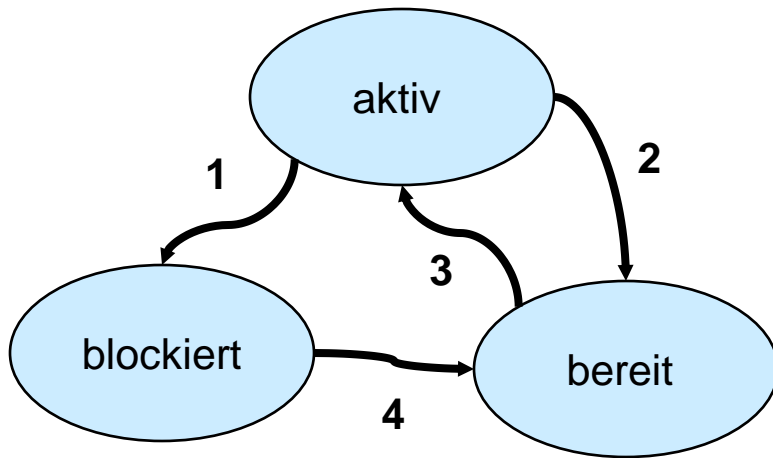
Paritätsbit

- Rechnerinterne Informationsdarstellung
- Rechnerarchitektur
- **Betriebssysteme**
- Datenbanken
- Algorithmen
- Programmiersprachen
- UML & SysML
- Software Engineering
- Rechnernetze
- Datensicherheit in Rechnernetzen

Prozess- und Prozessorverwaltung (2/3)

Zustände eines Prozesses:

- **bereit** (wartet auf die Zuteilung des Prozessors)
- **aktiv** (besitzt den Prozessor und läuft ab)
- **blockiert** (Prozessor wurde entzogen, wartet auf Betriebsmittel)



Vier Übergänge sind zwischen diesen Zuständen möglich:

1. Prozess blockiert, notwendiges Betriebsmittel steht nicht zur Verfügung.
2. Scheduler entzieht den Prozessor.
3. Scheduler teilt Prozessor zu.
4. Benötigtes Betriebsmittel steht zur Verfügung.

Scheduling

```
graph TD; S[Scheduling] --> SO[Strategien ohne Unterbrechung]; S --> SM[Strategien mit Unterbrechung];
```

Strategien ohne Unterbrechung

Dem laufenden Prozess kann der Prozessor nicht entzogen werden.

Vorteil: Geringer Aufwand

Nachteil: Für den Dialogbetrieb und Echtzeitsysteme nicht geeignet.

Strategien mit Unterbrechung

Dem laufenden Prozess kann der Prozessor entzogen werden.

Warten mehrere Prozesse auf den Prozessor, wird die Prozessorzeit auf diese aufgeteilt.

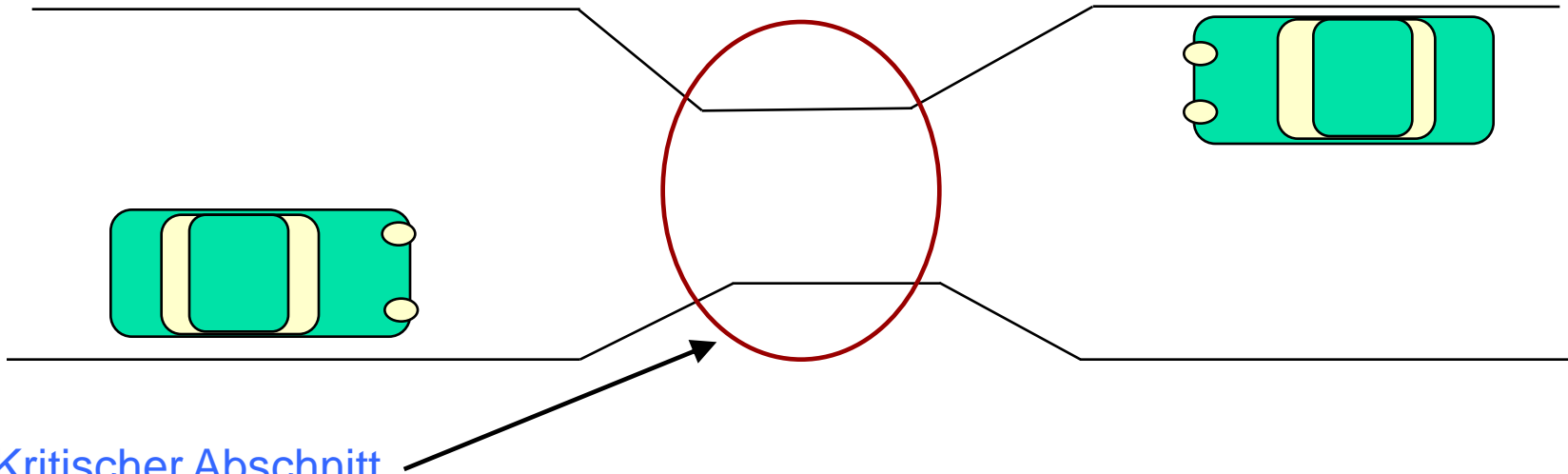
Vorteil: gerechte Betriebsmittelzuteilung

Nachteil: höherer Aufwand (Overhead)

Betriebsmittelbelegung

- Betriebsmittel werden vom BS entzogen und zugeteilt.
- Problem: Zwei Prozesse wollen auf das gleiche Betriebsmittel zugreifen und dieses ist nur exklusiv nutzbar.

Betriebsmittelproblem des Alltags:

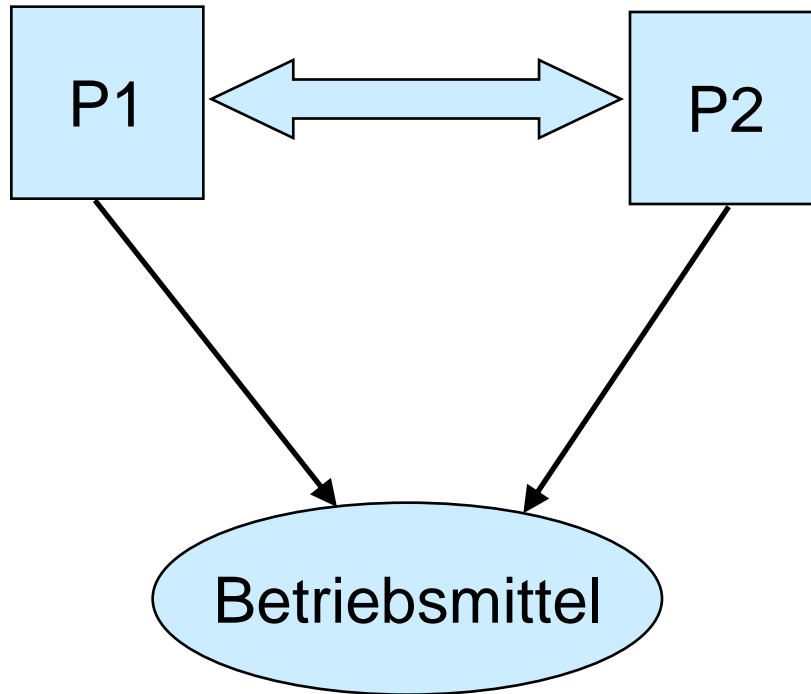


Kritischer Abschnitt

wird der Bereich genannt in dem sich höchstens ein Prozess befinden darf.

Lösung:

Verständigung der Teilnehmer, wechselseitiger Ausschluss



Kommunikation
zwischen den
beteiligten Prozessen
P1 und P2 über
Protokolle

Kollisionsvermeidung
durch Nutzung von
Sperren zum
gegenseitigen
Ausschluss beim
Passieren des
kritischen Abschnitts.

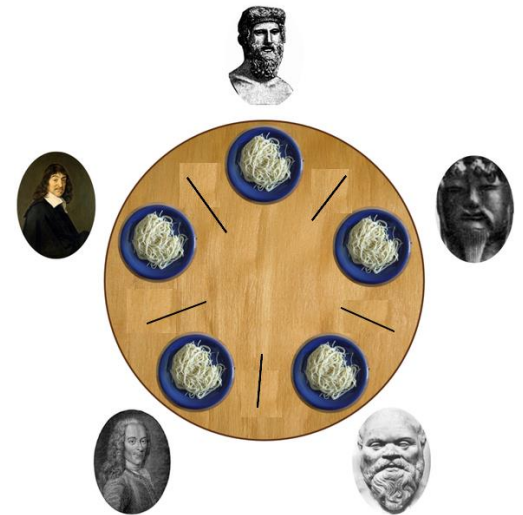
Philosophenproblem von Dijkstra

- Fünf Philosophen sitzen an einem Tisch und denken nach.
- Sie können entweder Denken oder Essen.
- Es gibt zwar für jeden einen Teller, aber nur fünf Stäbchen.
- Zwei benachbarte Philosophen können nicht gleichzeitig speisen.
- Möchte ein Philosoph essen, greift er zuerst nach dem linken, dann nach dem rechten Stäbchen.
- Möchten alle fünf Philosophen essen, greift jeder nach dem linken Stäbchen und wartet auf das rechte...

Problem:

Zwei Prozesse warten auf Betriebsmittel, die der jeweils andere Prozess blockiert.

→ Verklemmung (engl. deadlock)



- Rechnerinterne Informationsdarstellung
- Rechnerarchitektur
- Betriebssysteme
- **Datenbanken**
- Algorithmen
- Programmiersprachen
- UML & SysML
- Software Engineering
- Rechnernetze
- Datensicherheit in Rechnernetzen

Datenbanken...

- bieten logische und physische Datenunabhängigkeit,
- sind anwendungsneutral und damit flexibel,
- ermöglichen die Verwaltung sehr großer Datenmengen,
- sind anpassungsfähig hinsichtlich geänderter Bedürfnisse,
- gewährleisten den Zusammenhalt der Daten (Konsistenz bzw. Integrität),
- schützen die verwalteten Daten,
- bieten Sicherheit der Daten und
- bewältigen dies mit hoher Effizienz.

... ohne, dass der Anwender diese Aktivitäten explizit steuern muss!

Datenbankkonzepte: Relationaler Datenbankansatz

- Datentypen werden als Tabellen dargestellt; Spalten repräsentieren Attribute (Eigenschaften) und Datenobjekte werden in Zeilen abgelegt.
- Jeder Datentyp verfügt über mindestens ein eindeutiges Attribut.
- Verknüpfungen (Relationen) werden auch in Tabellen dargestellt, die eindeutigen Attribute der verknüpften Datenobjekte werden in Spalten eingetragen.

Datentyp: Lehrveranstaltung (LV)

Name	LV-Nummer
Industrielle Informationstechnik	0534 L 1015
Technologien der Virtuellen Produktentwicklung	0534 L 1017

Relation: Studierende besuchen LV

LV-Nummer	Matrikelnummer
0534 L 1015	343334
0534 L 1015	343335
0534 L 1017	343334

Datentyp: Studierende

Name	Alter	Matrikelnummer
Klaus	23	343334
Inge	19	343335

Datenobjekte

Attribute des Datentyps „Studierende“

Eigenschaften der Tabellen

1. Alle Einträge einer Spalte haben denselben Typ.
2. Alle Zeilen sind verschieden. Keine Zeile kommt mehrfach vor.
3. Die Reihenfolge der Zeilen innerhalb der Tabelle ist beliebig.
4. Die Tabellennamen sind innerhalb des Datenbanksystems eindeutig.
5. Die Bedeutung jeder Spalte wird durch einen Namen gekennzeichnet.

Definition:

Eine Transaktion ist die Zusammenfassung einer oder mehrerer elementarer Datenbankoperationen zu einer Einheit.

- Die von einer Transaktion ausgeführten Operationen überführen die Datenbank von einem konsistenten Zustand in einen neuen konsistenten Zustand.
- Eine Transaktion wird immer vollständig ausgeführt oder überhaupt nicht.
- Eine Transaktion muss gekennzeichnet sein.
(z.B.: BEGIN TRANS und END TRANS).

- Rechnerinterne Informationsdarstellung
- Rechnerarchitektur
- Betriebssysteme
- Datenbanken
- **Algorithmen**
- Programmiersprachen
- UML & SysML
- Software Engineering
- Rechnernetze
- Datensicherheit in Rechnernetzen

- Unter dem Begriff „Algorithmus“ versteht man allgemein eine genaue Handlungsvorschrift zur Lösung eines Problems oder einer bestimmten Art von Problemen.

Definition:

Ein Algorithmus ist eine genaue Vorschrift, nach der ein gewisses System von Operationen in einer bestimmten Reihenfolge auszuführen ist und mit der man alle Aufgaben eines gegebenen Typs lösen kann [Bieh00].

- Algorithmen sind eines der zentralen Themen der Informatik und der Mathematik. In Form von Computer-Programmen oder elektrischen Schaltkreisen steuern sie Computer oder andere Maschinen.

Ein Algorithmus ist determiniert, wenn er bei gleichen Parametern und Startwert stets das gleiche Resultat liefert.

- Durch die Eigenschaft der Determiniertheit wird die **eindeutige Abhängigkeit der Ausgabedaten von den Eingabedaten** gewährleistet. Somit ist es möglich **Funktionen** algorithmisch zu beschreiben.
- Ein Algorithmus ist **nicht** determiniert, wenn dessen Ausgabe zum Teil auf dem Zufall beruht.

Parallelität

- Ein Prozessor(-kern) erlaubt nur das **streng sequentielle Ausführen** von Elementaroperationen.
- Mehrere Prozessoren erlauben eventuell das **gleichzeitige Ausführen verschiedener Teile** eines Algorithmus.

sequentiell
<i>Prozessor1:</i> $x = 3;$ $y = 4;$

parallel	
<i>Prozessor 1:</i> $x = 3;$	<i>Prozessor 2:</i> $y = 4;$

Ein Algorithmus heißt parallel, wenn er so in Teilaufgaben aufgeteilt ist, dass diese gleichzeitig durch verschiedene Prozessoren ausgeführt werden können.

- Die Aussagen:
 „Der Algorithmus 1 gehört zur Komplexitätsklasse $O(n)$ “
 oder
 „Das Laufzeitverhalten von Algorithmus 1 entspricht $O(n)$ “

bedeuten folglich, dass der Algorithmus 1 im schlechtesten Fall über die Eingabegröße n ein lineares Laufzeitverhalten zeigen wird.

- Die Laufzeit steigt maximal in linearer Abhängigkeit zur Eingabegröße.

Komplexitätsklassen (1/2)

Komplexitätsklasse	O-Notation	Beispiele (a,b,c):
Konstant	$O(1)$	3,10,23
Logarithmisch	$O(\log n)$	$2 \cdot \log n, 5 + \log(2 + 3n)$
Linear	$O(n)$	$1 + n, n + \log n, 10n$
Logarithmisch-Linear	$O(n \log n)$	$2n \log n, n \log(3 + n)$
Polynomisch	$O(n^k)$	$2n^2 + 3n, 16n^3 + 5$
Exponentiell	$O(a^n)$	$2^{4n}, 3^n + n^3$

Komplexitätsklassen – Beispiel Bubblesort

Für jedes i der Daten von 1 bis n

{

 Für jedes j der Daten von 1 bis $n-i$

 {

 Wenn $j > (j+1)$

$O(1)$

 tausche j gegen $(j+1)$

$O(1)$

} $O(1)$

$n * O(1)$
 $= O(n)$

$(n-1) * O(n)$
 $= O(n^2)$

 }

}



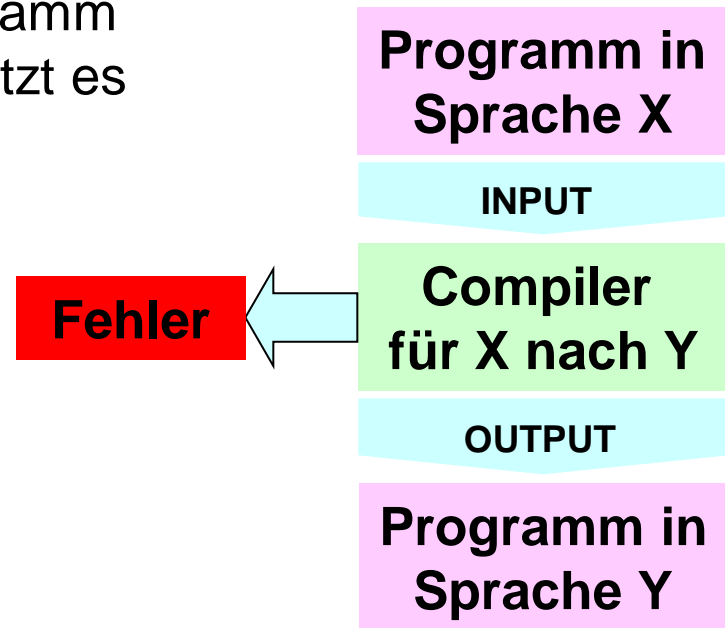
Komplexitätsklasse = $O(n^2)$

- Rechnerinterne Informationsdarstellung
- Rechnerarchitektur
- Betriebssysteme
- Datenbanken
- Algorithmen
- **Programmiersprachen**
- UML & SysML
- Software Engineering
- Rechnernetze
- Datensicherheit in Rechnernetzen

Compiler – Fazit

Fazit: Ein Compiler prüft, ob ein Programm ein Wort der Sprache X ist und übersetzt es in eine Zielsprache Y.

Ein Compiler muss die Semantik des Quellprogramms erhalten!

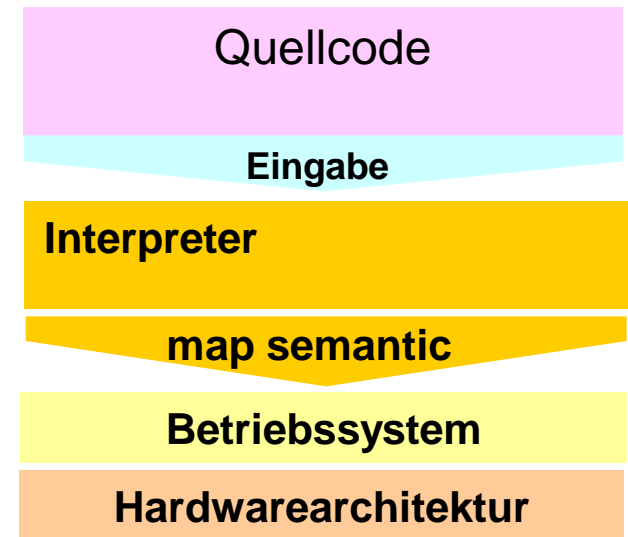


- Input ist Quellcode
- Input wird überprüft und evt. Fehler gemeldet
- Output in Zielsprache wird erzeugt

Interpreter 1/2

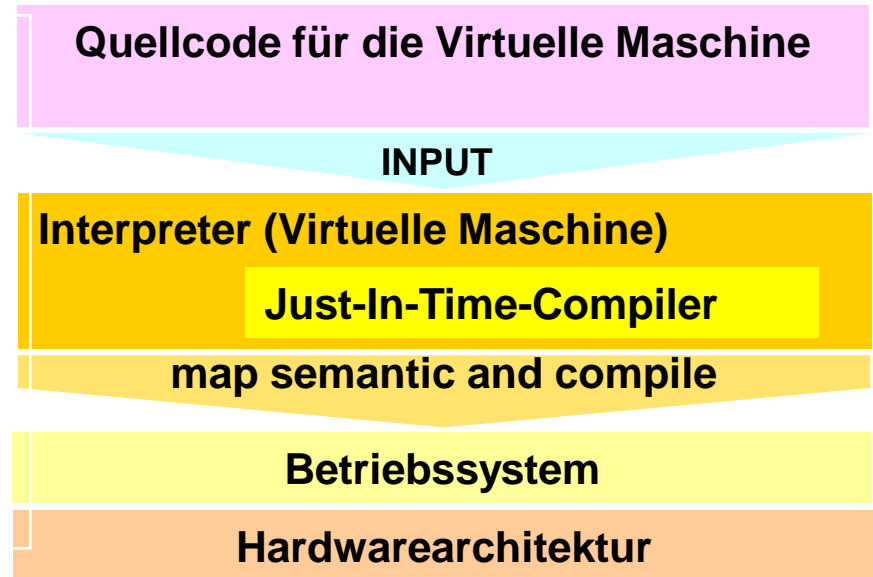
Ein **Interpreter** ist ein Programm, das Quellcode im Gegensatz zu Compilern nicht in eine auf dem System direkt ausführbare Datei umwandelt, sondern den Code einliest, analysiert und ausführt. Die Analyse des Quellcodes erfolgt also zur Laufzeit des Programms.

- Kein Kompilieren notwendig.
- Die Analyse des Quellcodes erfolgt zur Laufzeit.
- **Vorteil:** Unabhängigkeit von Betriebssystem und Hardwarearchitektur
- **Nachteil:** Langsamere Ausführung im Vergleich zu kompilierten Programmen



Interpreter 2/2

- Eine Geschwindigkeitssteigerung lässt sich mit einem **Just-In-Time-Compiler (JIT)** erzielen.
- Der JIT- Compiler übersetzt **Bytecode** zur Laufzeit bei Bedarf in Maschinensprache.
- Der JIT ist Bestandteil einer Virtuellen Maschine (z.B. JVM bei Java oder CLR bei .NET).
- Eine Virtuelle Maschine ist eine virtuelle Laufzeitumgebung für Programme.
- Bytecode ist hardwareunabhängiger Zwischencode.



Vorteile: Plattformunabhängigkeit: das entwickelte Programm läuft auf jeder Hardware, auf der die Virtuelle Maschine läuft. Schneller als Interpreter ohne JIT-Compiler

Nachteile: Langsamer als Programme, die speziell für eine bestimmte Hardware kompiliert wurden.

Objektorientierte Programmierung (OOP)

Vorteile:

- Wiederverwendung von bereits bestehenden Programmteilen und Programmen durch Vererbung und Klassenbibliotheken.
- Bei Wartung des Codes müssen nur kleine Teile angepasst und verändert werden. Voraussetzung dafür sind gute OO Analyse, Entwurf und Implementierung der Software.
- Abstrakte Klassen bilden die Basis für einheitliche Schnittstellen für andere Klassen.
- Entwurfsfehler können vermieden werden, da auf einem hohen, problemorientierten Abstraktionsniveau entworfen wird.

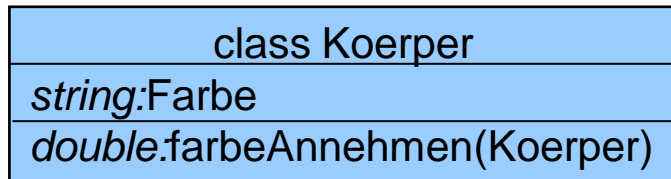
Nachteile:

- Lange Einarbeitungszeit ins objektorientierte Denken
- Wissen über das Formulieren von Algorithmen reicht nicht aus. Die innere Struktur von Klassen und Bibliotheken muss bekannt sein, um an den richtigen Stellen Erweiterungen vornehmen zu können.

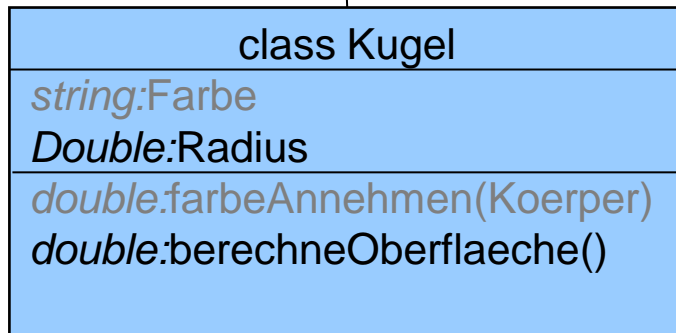
Vererbung in der OOP

Vererbung

- Erstellen neuer Klassen durch Wiederverwendung und Erweiterung der Semantik von bereits existierenden.
- Methoden und Attribute der abgeleiteten Klasse können von der neuen Klasse benutzt werden.
- Bsp.:



Die Klasse Koerper beinhaltet alle Methoden und Attribute für **einen geometrischen Grundkörper**.



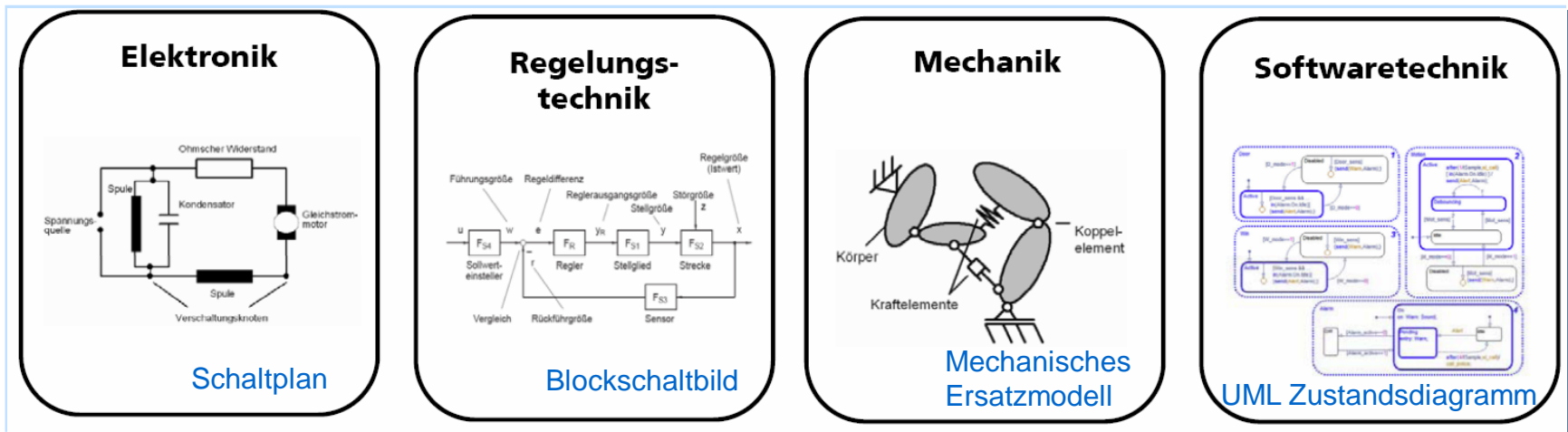
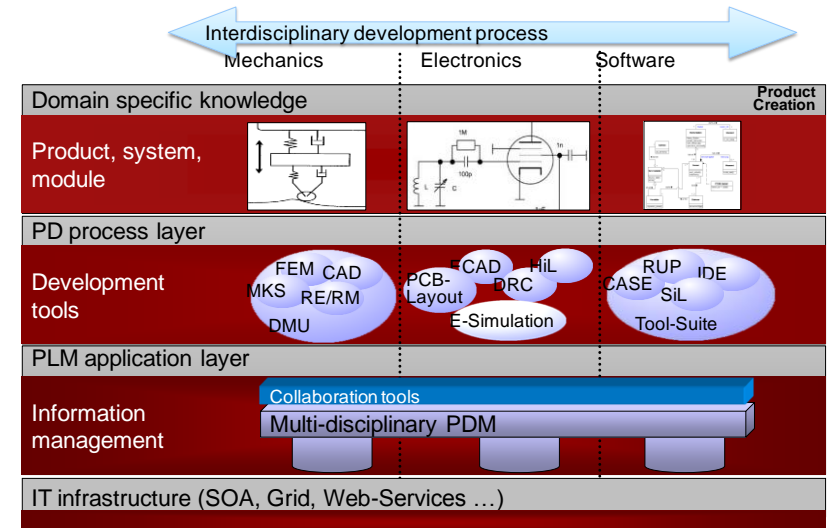
Die Klasse Kugel erbt die Methoden und Attribute der Klasse Koerper **und wird erweitert um das Attribut Radius und Berechnung der Kugelfläche**.

- Rechnerinterne Informationsdarstellung
- Rechnerarchitektur
- Betriebssysteme
- Datenbanken
- Algorithmen
- Programmiersprachen
- **UML & SysML**
- Software Engineering
- Rechnernetze
- Datensicherheit in Rechnernetzen

Motivation für UML/SysML

Domänenspezifische Modelle:

Innerhalb eines interdisziplinären Entwicklungsprozesses besitzen die einzelnen Fachdomänen unterschiedliche Diagrammart und Formalismen, um die Systemkomponenten, die sie beisteuern, zu beschreiben und darzustellen.

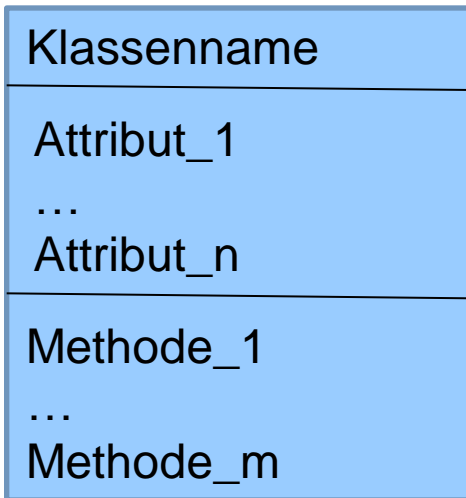


Darstellung von Klassen in Klassendiagrammen (1/3)

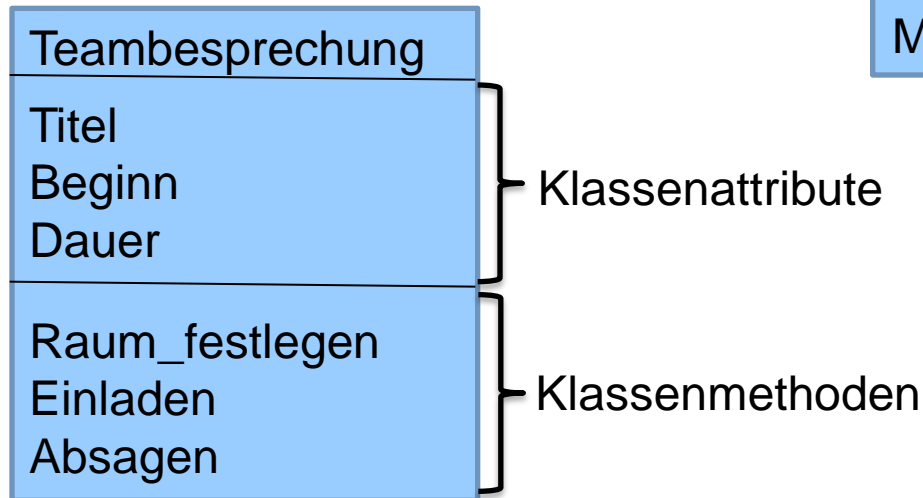
Nach der UML-Notation wird die Klasse durch ein Rechteck dargestellt und besteht aus:

- **Klassenname**: Benennung einer Klasse aus dem Sprachgebrauch des Anwendungsgebietes.
- **Attribut**: Ein Datenwert, den die Objekte dieser Klasse besitzen.
- **Operation/Methode**: Eine Funktion, die Anwendung auf Objekte einer Klasse ermöglicht.

Klassennotation



Beispiel:

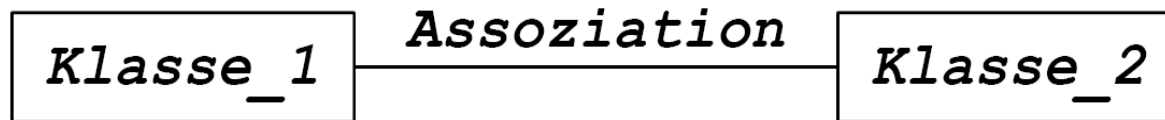


Assoziationen (2/3)

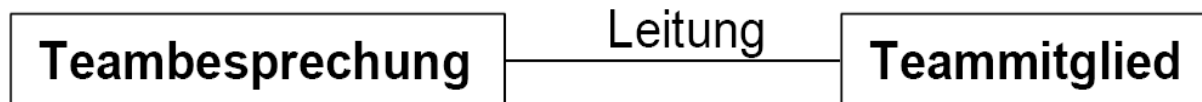
Eine **Assoziation** zwischen zwei Klassen K1 und K2 beschreibt, dass die Instanzen der beiden Klassen in einer fachlich wesentlichen Beziehung zueinander stehen.

- Für jedes Objekt der Klasse K1 gibt es eine individuelle, veränderbare und endliche Menge von Objekten der Klasse K2, mit denen die Assoziation besteht. Analoges gilt für die Objekte von K2.

Notation:



Beispiel:

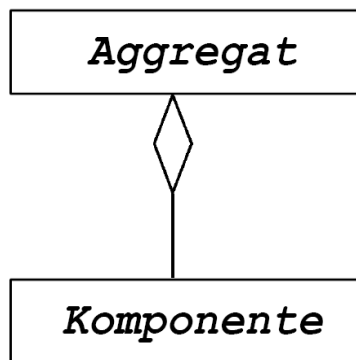


Aggregation und Komposition (3/3)

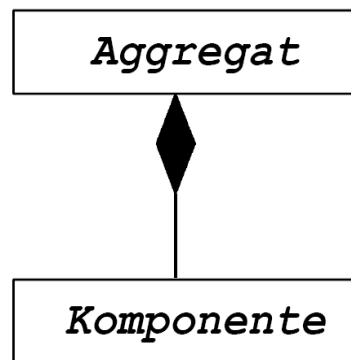
Aggregationen und **Kompositionen** sind spezielle Assoziationen, die "Teile/Ganzes"-Beziehungen und "Hat-eine"-Beziehungen darstellen.

- Bei der Aggregation können die "Teile" des "Ganzen" auch einzeln existieren.
- Bei der Komposition existieren die "Teile" nur, wenn auch das "Ganze" existiert. (z.B. Rechnungspositionen auf einer Rechnung)

Notation:



Aggregation



Komposition

- Rechnerinterne Informationsdarstellung
- Rechnerarchitektur
- Betriebssysteme
- Datenbanken
- Algorithmen
- Programmiersprachen
- UML & SysML
- **Software Engineering**
- Rechnernetze
- Datensicherheit in Rechnernetzen

Zusammenfassung – Probleme bei der SW-Entwicklung

- Beherrschung der Komplexität der Aufgabenstellung
 - Bei der Problemanalyse und dem Entwurf entstehen die meisten Fehler
 - Schwierigkeiten bei der:
 - vollständigen Erfassung der Anforderungen
 - korrekten Spezifikation der Anforderungen
 - Zerlegung der Aufgabe in Teilprobleme
 - Spezifikation der Schnittstellen
 - Einschätzung der Kosten und des Zeitbedarfs
- Pro 1000 Zeilen werden bei der Entwicklung zwischen 50 und 60 Fehler entdeckt.
- Nach der Auslieferung bis zu 4 Fehler pro 1000 Zeilen.

„Code and Fix“:

- Code now, fix later
 - Keine feste Planungsphase
 - Prinzip:
 - Schreibe ein Programm
 - Finde und behebe die Fehler im Programm
 - Nachteile
 - Code ist schlecht strukturiert
 - Fehlerbehebung wird durch mangelnde Tests erschwert
- ➔ Entwicklung von Modellen zur strukturierten Softwareentwicklung

Phasen der Softwareentwicklung

- Anforderungen
- Analyse
- Entwurf (Design)
- Implementierung
- Test
- Inbetriebnahme
- Wartung

Klassische Modelle

- Wasserfallmodell
- V-Modell
- Spiralmodell

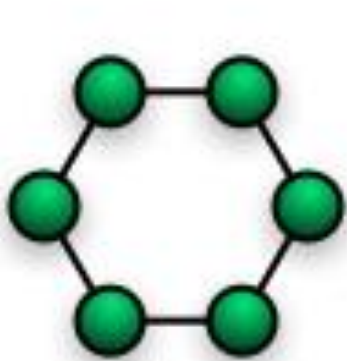
- Neuer Ansatz der Softwareentwicklung, die auf der Entwicklung und der Lieferung von sehr kleinen Funktionalitätszuwächsen basiert.
- Klassische Methoden scheitern oft in der Praxis trotz ständiger Weiterentwicklung.
- Vertreter neuer Methoden formulierten 2001 Gemeinsamkeiten in vier Grundwerten.
- Es entstand das *Manifest Agiler Softwareentwicklung*.

4 Eckpfeiler des Manifests:

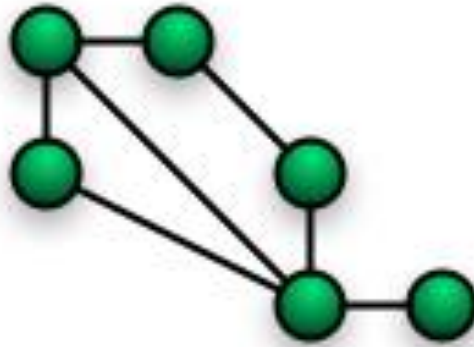
- **Menschen und Zusammenarbeit** sind wichtiger als Prozesse und Werkzeuge.
- **Funktionierende Software** ist wichtiger als umfassende Dokumentation.
- **Zusammenarbeit mit dem Kunden** ist wichtiger als Vertragsverhandlungen.
- **Reagieren auf Veränderung** ist wichtiger als die Einhaltung eines Plans.

- Rechnerinterne Informationsdarstellung
- Rechnerarchitektur
- Betriebssysteme
- Datenbanken
- Algorithmen
- Programmiersprachen
- UML & SysML
- Software Engineering
- **Rechnernetze**
- Datensicherheit in Rechnernetzen

Topologien in Rechnernetzen



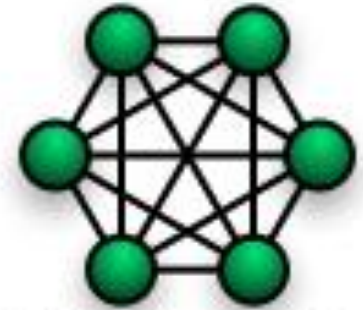
Ring



Mesh



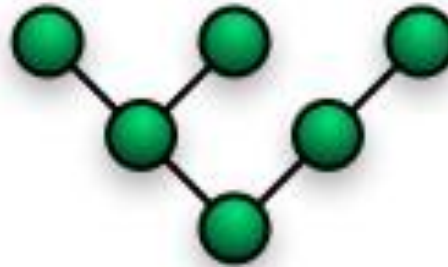
Star



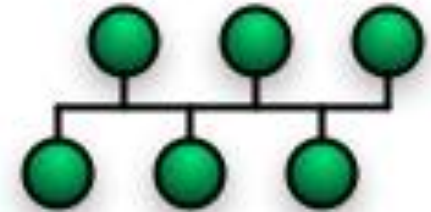
Fully Connected



Line



Tree

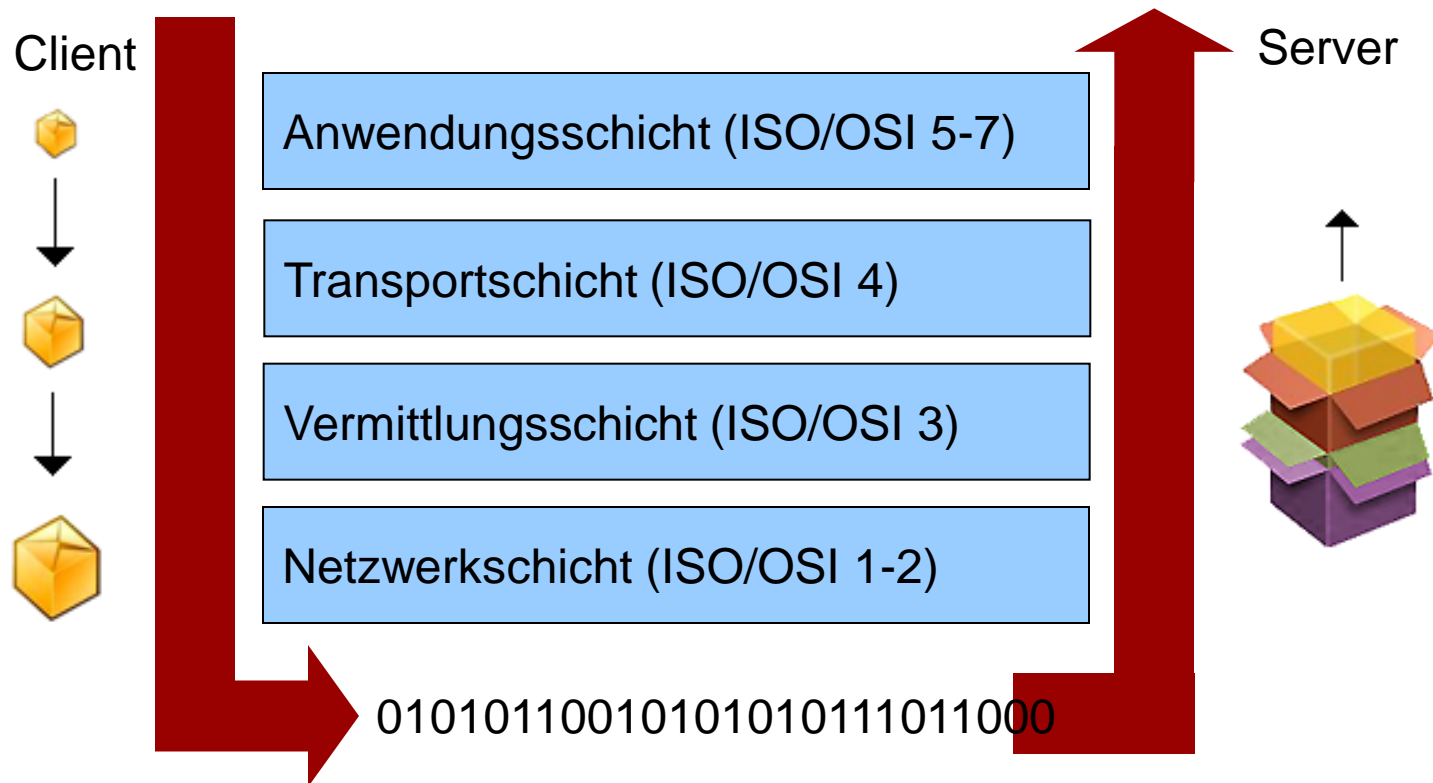


Bus

Quelle: Wikipedia

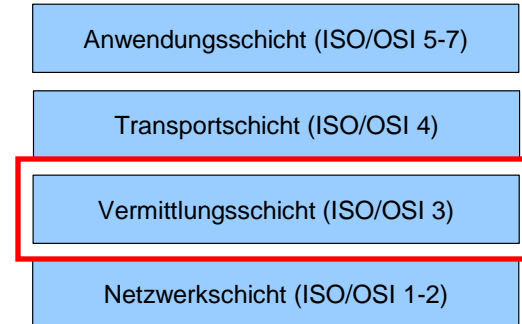
Vereinfachtes ISO/OSI Modell

- ISO: *International Organization for Standardization*
- OSI: *Open Systems Interconnection Reference Model*
- In Theorie 7-Schichten-Modell zur Entwicklung von Kommunikationsprotokollen
- In Praxis wird oftmals in nur 4 Schichten unterschieden



Vermittlungsschicht (ISO/OSI 3)

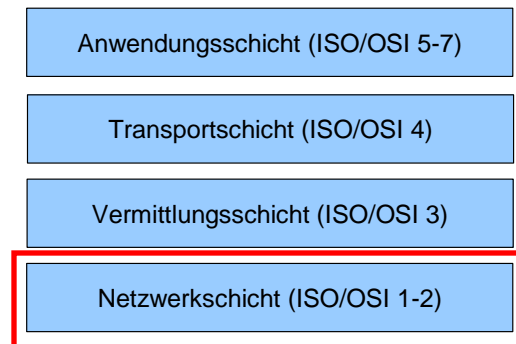
- Die Vermittlung von Nachrichten in einem Netzwerk erfolgt über IP-Adressen
- IP: Internet Protocol
- Eine URL (Uniform Resource Locator) ist ein „Platzhalter“ für IP-Adressen
- Es gibt zwei IP-Formate: IPv4 und IPv6
 - IPv4 ist das aktuell gängige Format, reicht aber nicht mehr aus
 - IPv6 wird irgendwann IPv4 ersetzen



Protokoll	Beispiel-Adresse	Max. Menge von Adressen	Address-Größe
IPv4	192.168.1.123	2^{32} oder ca. 4,3 Milliarden	32 Bit
IPv6	2001:0db8:85a3:08d3:1319:8a2e:0370:7344	2^{128} oder die Ziffer 34 gefolgt von 37 Nullen	128 Bit

Netzwerkschicht (ISO/OSI 1-2)

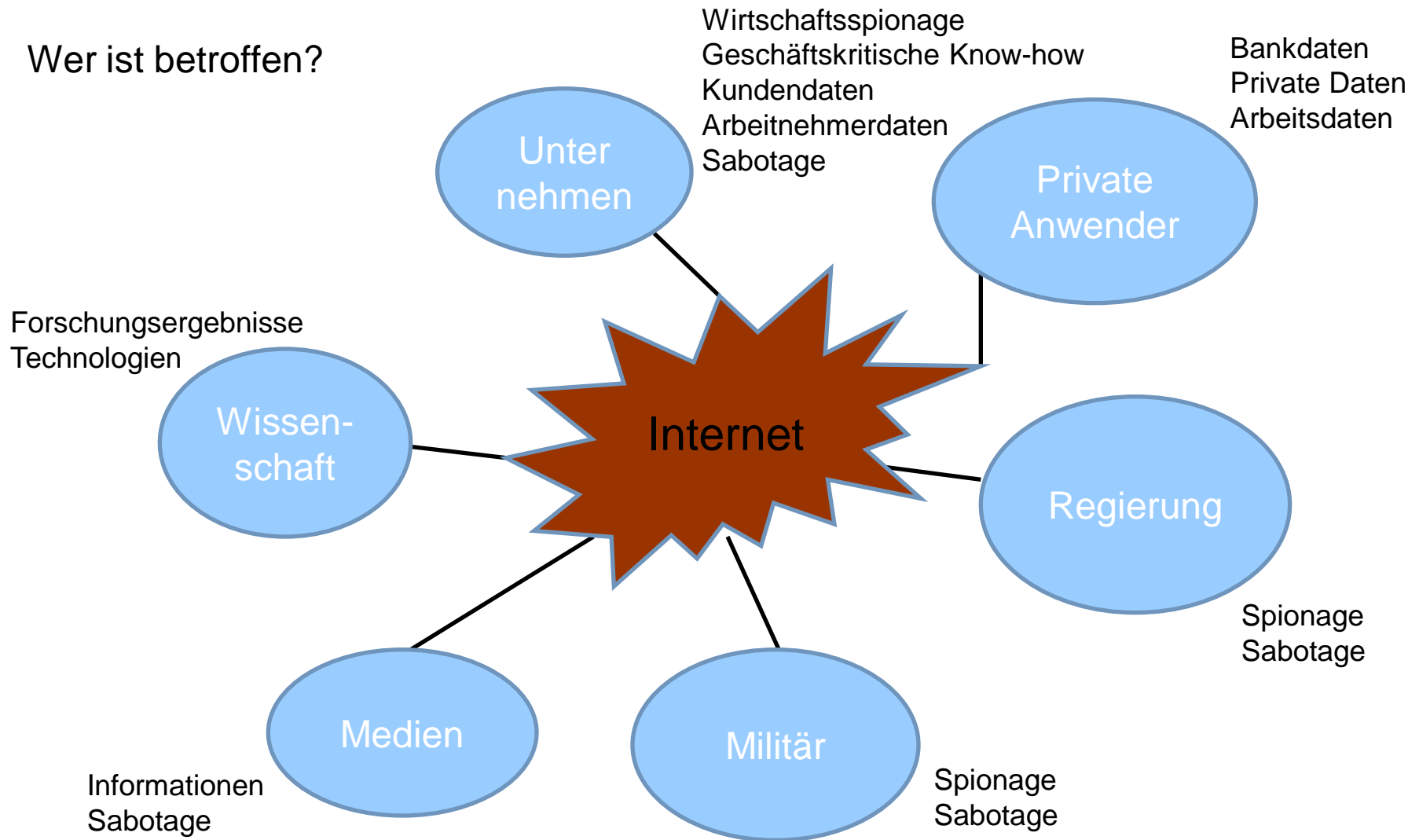
- Die Netzwerkschicht stellt Techniken für die Punkt-zu-Punkt Datenübertragung bereit. Sie ist die Basis des Internetprotokolls.
- Beispiele
 - Ethernet
 - Kabelgebundene Netzwerkübertragung
 - WLAN
 - Kabellose Netzwerkübertragung
 - IPoAC
 - IP over Avian Carriers
 - PPP
 - Verbindung über Wählleitungen (Modem,GSM)
 - PPPoE (PPP over Ethernet)
 - das DSL-Protokoll in Deutschland



- Rechnerinterne Informationsdarstellung
- Rechnerarchitektur
- Betriebssysteme
- Datenbanken
- Algorithmen
- Programmiersprachen
- UML & SysML
- Software Engineering
- Rechnernetze
- Datensicherheit in Rechnernetzen

Angriffe auf Rechnernetze

Wer ist betroffen?



Angriff durch Schadprogramme: Trojanische Pferde

Als Trojanisches Pferd (Trojaner) bezeichnet man ein Programm, das als nützliche Anwendung getarnt ist, im Hintergrund aber ohne Wissen des Anwenders eine andere Funktion erfüllt. (Quelle: Wikipedia)

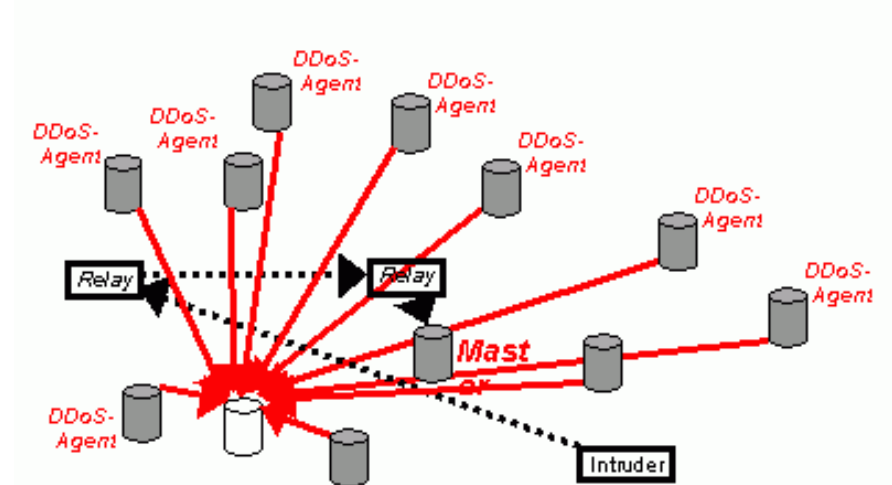
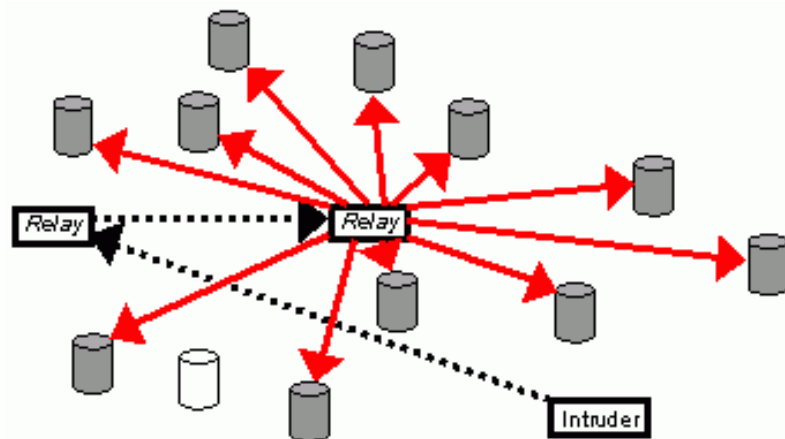
- Besteht aus einem eigenständigen Programm.
- Wird häufig in Anwendungs- und Dienstprogramme eingebaut.
- Im Gegensatz zum Virus keine alleinige Fortpflanzung.
- Muss vom Anwender per Download, Attachment einer Email oder durch persönliche Weitergabe kopiert werden.
- Beispiel: Der Hamburger "Chaos Computer Club" bewies 1997, dass Trojaner mit Hilfe von installierter Homebanking-Software im Hintergrund Überweisungen auf fremde Konten veranlassen konnten.



Angriffsarten – Denial of Service - Netzwerkangriffe

Denial of Service (engl. Versagen\Blockade eines Dienstes) **hat zum Ziel Dienste zu unterbrechen bzw. zu blockieren.**

- Angriff auf Verfügbarkeit
- Überlastung von Servern und Netzwerkverbindungen
- Führt zu temporärem Ausfall aller Netzwerkverbindungen und Dienste
- Bevorzugt werden Trojanische Pferde als Hilfsmittel verwendet.



Die Phasen einer DoS-Attacke
Quelle: Bundesamt für Sicherheit

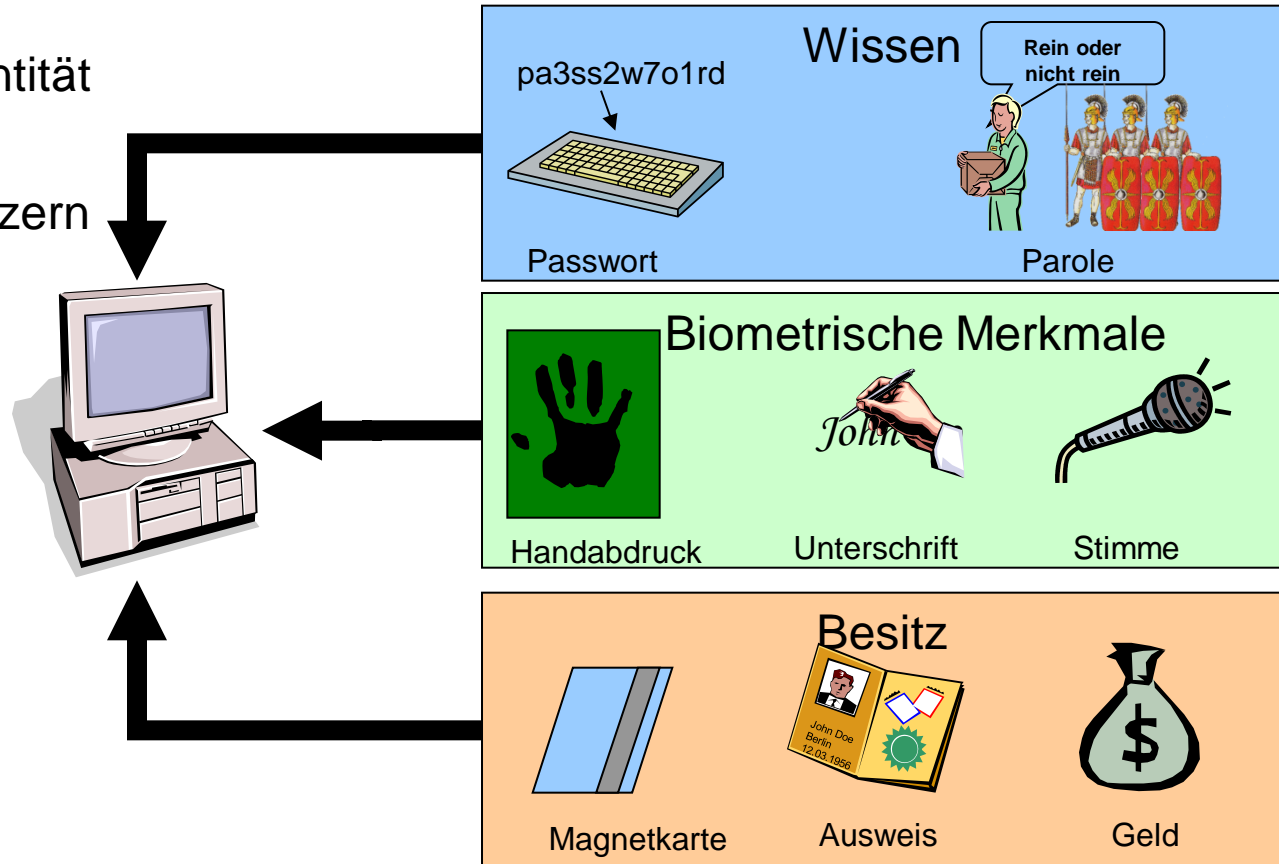
Arten der Authentifikation

- Verifikation der Identität

- Zuordnung von Nutzern zu Rechten für:

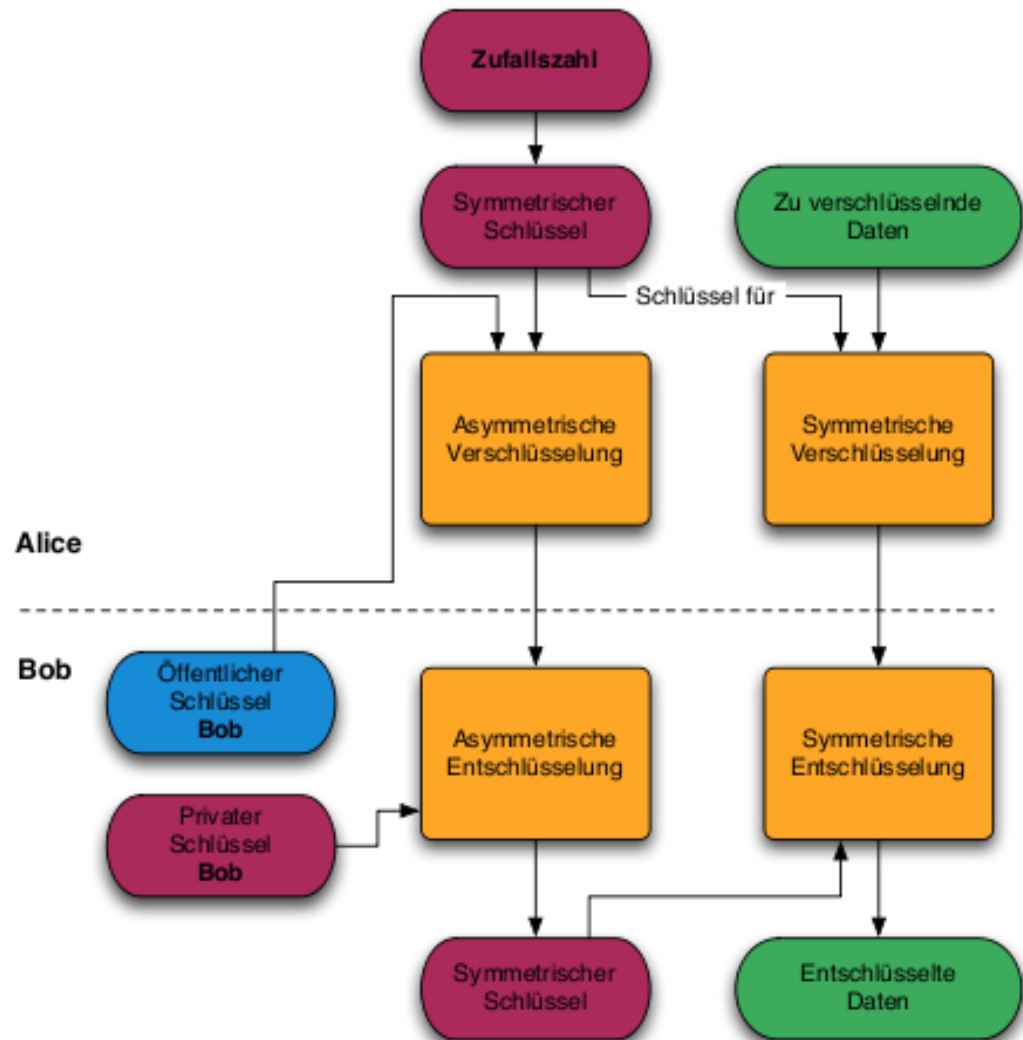
- Datenzugriff
- die Nutzung von Diensten

- Zugriffskontrolle



Hybride Verfahren (1/2)

- Symmetrische Verfahren werden in der Regel nicht alleinstehend eingesetzt, weil das Schlüsselmanagement bei diesen Algorithmen schwer in den Griff zu bekommen ist.
- Stattdessen werden sie zusammen mit anderen kryptographischen Verfahren eingesetzt.
- Sie spielen z.B. in hybriden Verfahren eine große Rolle zur Verschlüsselung der eigentlichen Sitzungsdaten.



Quelle: heise-security

VIEL ERFOLG BEI DER KLAUSUR!