Einführung in die Informatik
für Nebenfachstudenten

Wolfgang Menzel

Raum:	F 207
Telefon:	5494-2435
E-Mail:	menzel@informatik.uni-hamburg.de
Sprechstunde:	Die 14 - 15 Uhr

Jan W. Amtrup (Übungen)

Raum:	F 211
Telefon:	5494-2519
E-Mail:	amtrup@informatik.uni-hamburg.de
Sprechstunde:	Die 14 - 16 Uhr

Jane Isigkeit (Sekretariat)

Raum:	F 205
Telefon:	5494-2433
E-Mail:	jane@nats5.informatik.uni-hamburg.de

Zeiten/Räume

Vorlesung	Dienstag	8.30 - 10.00	B-201	Wolfgang Menzel
Übungen	Dienstag	10.15 - 11.45	D-010/C-104	Jan W. Amtrup
		12.15 - 13.45	D-010/C-104	Jan W. Amtrup

Weshalb NF-Vorlesung?

- ☐ Lernen, was man mit Computern so machen kann ("Computerführerschein")
- ☐ Lernen, wie man mit Computern Texte/Grafiken verarbeiten kann
- ☐ handwerkliches Programmieren lernen in einer gängigen Programmiersprache für meinen PC
- ☐ internen Aufbau und Funktion von Computern kennenlernen
- ☐ Spezialkenntnisse über meinen PC vertiefen
- ☐ alle Tricks des Hacker-1×1 lernen, z.B. mit meinem PC in NASA Rechner eindringen
- ☐ neue Computerspiele kennenlernen

©Horst Oberquelle

Weshalb NF-Vorlesung?

- grundlegende Informatikkonzepte und typische Denkweisen dieser wissenschaftlichen Disziplin unter Verwendung einer didaktisch geeigneten Programmiersprache exemplarisch kennenlernen

---

- Kennenlernen von typischen Werkzeugen zur Informationsverarbeitung und Verstehen der dafür fundamentalen Konzepte

NF I $\Longrightarrow$ NF II $\Longrightarrow$ NF IIIa $\Longrightarrow$ NF IIIb

Einführung in die Informatik

- ▷ Informatik im Überblick
- ▷ Reguläre Ausdrücke
- ▷ Relationale Datenbanken
- ▷ Tabellenkalkulation
- ▷ Deduktive Datenbanken
- ▷ Logische Programmierung

Informatik im Überblick

- ▷ Grundbegriffe
- ▷ Historischer Abriß
- ▷ Aufgabenstellung Informationsverarbeitung
- ▷ Architektur informationsverarbeitender Systeme
- ▷ Mensch und Computer

Grundbegriffe

- L'informatique (frz.) = <u>infor</u>mation + auto<u>matique</u>

> **Informatik** befaßt sich mit der Entwicklung formaler, maschinell durchführbarer Verfahren zur Lösung von Informationsverarbeitungs-Problemen

> **Informatik** ist die Wissenschaft, Technik und Anwendung der maschinellen Verarbeitung und Übermittlung von Informationen

- Als Wissenschaft umfaßt die Informatik
  - Theorie
  - Methodik
  - Analyse und Konstruktion
  - Anwendungen
  - Auswirkungen des Einsatzes

  informationsverarbeitender technischer Systeme.

Aufgabenbereiche der Informatik

- Analyse von Problemen der Informationsverarbeitung
- Entwurf von komplexen Verfahren und Systemen der Informationsverarbeitung
- Auffinden von Algorithmen, Datenstrukturen, Kommunikationsformen und Organisationsmodellen für bestimmte Aufgaben der Informationsverarbeitung
- präzise Formulierung von Algorithmen und Kommunikationsvorgängen
- formale Definition von Datenstrukturen und Organisationsmodellen
- technische Konstruktion von Datenverarbeitungs- und Kommunikationssystemen
- präzise Beschreibung von Struktur, Wirkungsweise und Fähigkeiten von Datenverarbeitungs- und Kommunikationssystemen
- Implementierung von Algorithmen auf Datenverarbeitungssystemen
- Prüfung und Bewertung von Datenstrukturen, Algorithmen und kompletten Systemen zur Datenverarbeitung und Kommunikation
- Analyse von Grenzen und Wirkungen der automatischen Informationsverarbeitung

Historischer Abriß (1)

3500 bis 3000 v. Chr.	Babylonier und Ägypter kannten bereits die ersten vollständigen Zahlensysteme
ca. 300 v. Chr.	Die Römer gestalten den Rechentisch zu handlichen Geräten um: Abakus
6. bis 8. Jhdt.	Der schottische Lord NAPIER erfindet Rechenstäbchen und stellt komplette Logarithmentafeln vor
1622	Der englische Mathematiker OUGHTRED entwirft den logarithmischen Rechenschieber
1623	Der Tübinger Astronom SCHICKARD konstruiert die erste Rechenmaschine mit Zahnradgetriebe und automatischem Zehnerübertrag
1642	Der französische Mathematiker PASCAL stellt eine Rechenmaschine für 8-stellige Addition vor, die wie ein Kilometerzähler arbeitet
1673	LEIBNITZ entwickelt eine mechanische Vier-Spezies-Rechenmaschine mit Staffelwalze

Historischer Abriß (2)

1812 Der englische Mathematiker BABBAGE entwirft eine Rechenmaschine mit externer Programmsteuerung über Lochkarten zur Überprüfung mathematischer Tabellen

1832 BABBAGE äußert die Idee der Teilung geistiger Arbeit:

"Wir haben bereits erwähnt, daß die Teilung der Arbeit, ..., sich mit gleichem Erfolg und gleicher Zeitersparnis auf geistige Operationen anwenden lasse. ... daß die Teilung der Arbeit, bei mechanischen wie bei geistigen Arbeiten darin besteht, daß man genau so viel Geschicklichkeit und Kenntnis zu erkaufen und anzuwenden braucht, als zu diesem Prozeß nötig ist."

$\longrightarrow$ Taylorismus

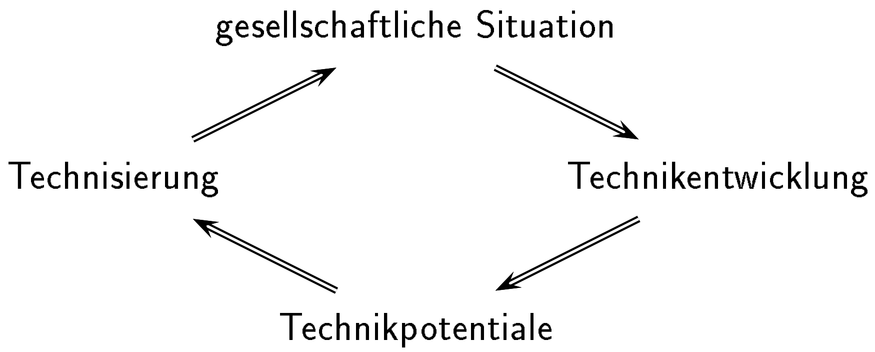
1833 BABBAGE konstruiert als erster einen programmgesteuerten Rechenautomaten ("Analytical Engine"), der wegen fehlender technischer Möglichkeiten allerdings niemals fertig wird.

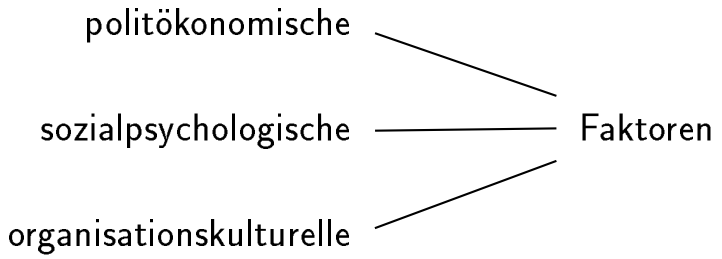
1886 Der amerikanische Ingenier HOLLERITH konstruiert eine elektromechanische Lochkartenmaschine, die zur 11. amerikanischen Volkszählung eingesetzt wird.

Historischer Abriß (3)

1941	ZUSE baut den ersten funktionsfähigen programmgesteuerten Rechenautomaten (Z3).
1944	Erster amerikanischer programmgesteuerter Rechenautomat (Mark I) geht in Betrieb.
1945	ZUSE's Z4 ist funktionstüchtig.
1946	In den USA wird die erste vollelektronische Großrechenanlage ENIAC fertiggestellt
1955	Halbleiter-Transistoren lösen die Elektronenröhren ab: Die zweite Computergeneration beginnt.
1962	Miniaturisierte Transistoren führen zu höheren Rechengeschwindigkeiten und geringerem Platzbedarf: Dritte Computergeneration.
1968	Integrierte Schaltkreise
1978	Hochintegrierte Schaltkreise

Soziale Triebkräfte





Triebkräfte

- Planung in Staat, Verwaltung und Unternehmen
  - Zahlensysteme
  - Buchhaltung
  - Rechenkunst
  - Rechenhilfsmittel

    SCHICKARD: Mathematiker und Astronom

    PASCAL: Sohn eines Steuerbeamten

    BABBAGE: Sterbetafeln, Astronomische Tafeln für die Kriegsmarine

    HOLLERITH: Volkszählung
- Rechenbüros für ingenieurtechnische Anwendungen
  - Ingenieure entwickeln Rechenschemata und numerische Mathematik
  - Rechnerinnen führen sie mit mechanischen Rechenmaschinen für die Grundrechenarten aus

    ZUSE: Codierung von Rechenplänen auf Lochstreifen und automatische Abarbeitung
    - Flatterberechnungen für Flugzeuge
    - Flügelvermessung für fliegende Bomben
- Militärische Aufklärung

  TURING: Entschlüsselung des Funkcodes der deutschen U-Boote (ENIGMA)
- Weitere militärische Schlüsselanwendungen
  - Kernwaffen-Entwicklung
  - Ballistische Berechnungen
  - Steuerung von Waffensystemen

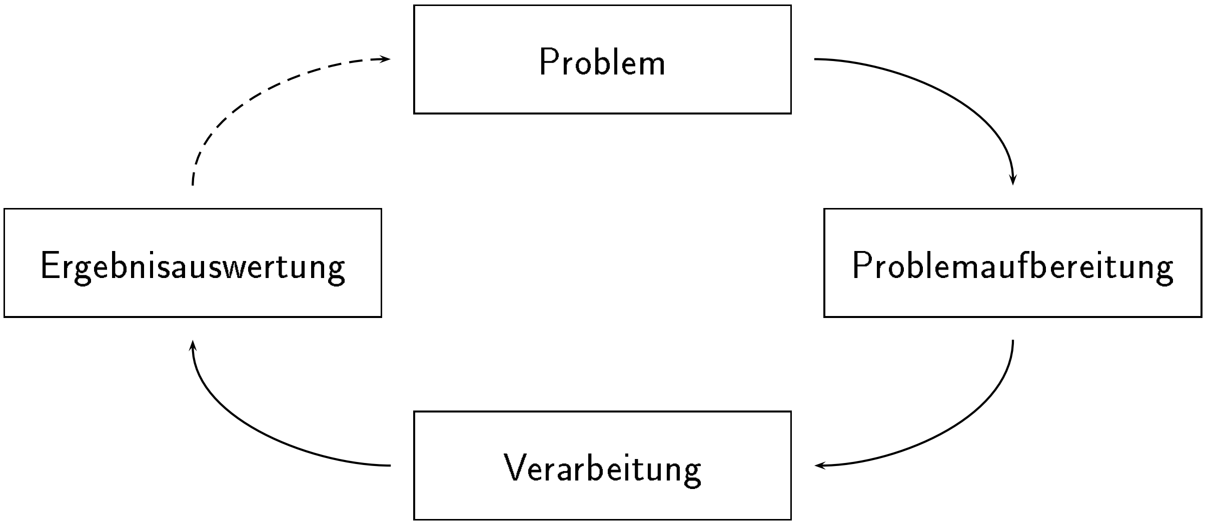
Geschwindigkeiten

US-Census	500 Personen, 7 Jahre
Hollerith	43 Maschinen und Personen, 4 Wochen
Z3	15 ... 20 Additionen/s
	0.2 ... 0.25 Multiplikationen/s
Mark I	3 Additionen/s
	0.33 Multiplikationen/s
ENIAC	360 Multiplikationen/s
2. Generation	1 300 Additionen/s
3. Generation	160 000 Additionen/s

Entwicklungstendenzen

- weitere Verbreitung der Computertechnik in allen Lebensbereichen
- billigere, kleinere und leistungsfähigere Hardwarekomponenten
- weitere Dezentralisierung
- stärkere Vernetzung
- stärkerer Anteil von eingebetteten Systemen
- Erhöhung des Benutzungskomforts
- Erhöhung der Zuverlässigkeit

Aufgabenstellung Informationsverarbeitung



- Warenpreise addieren (Ladenkasse, Inventur, ...)
- Was für ein Wochentag ist der 31.12.1999?
- Wenn 1.5 Hühner in 1.5 Tagen 1.5 Eier legen, wie viele Eier legt ein Huhn an einem Tag?
- Wie komme ich am schnellsten vom Campus zur Informatik?
- Welcher Zug ist in einer gegebenen Schachstellung am vorteilhaftesten?
- Wie wird das Wetter am Wochenende sein?
- Welche Maßnahmen muß man zur Verringerung der Arbeitslosigkeit ergreifen?
- Welches Buch ist als Einführung in die Informatik am besten geeignet?

Beispiel: Buchhaltung

- Ein- und Ausgänge von Waren und Geld werden auf Formularen erfaßt

	Kunde	Ware	Preis
Einkauf	Obst AG	100 kg Äpfel	100 DM

	Kunde	Ware	Preis
Verkauf	Schlemmermarkt	100 kg Äpfel	300 DM

	Kunde	Preis
Kassenbeleg	Schlemmermarkt	300 DM

- Kundenkonten mit Guthaben bzw. Verbindlichkeiten werden geführt

Kunde	Obst AG				
Datum	Einzahlung	Auszahlung	Einkauf	Verkauf	Kontostand
17.10.1996			100 DM		100 DM

Kunde	Schlemmermarkt				
Datum	Einzahlung	Auszahlung	Einkauf	Verkauf	Kontostand
18.10.1996				300 DM	-300 DM
27.10.1996	300 DM				0 DM

- Zahlungsanweisungen bzw. Rechnungslegungen werden veranlaßt
- regelmäßige Überprüfung der Verbindlichkeiten

Buchhaltung

- Arbeitsaufgaben
  - Klassifizieren von Formulararten (Einkauf/Verkauf/Zahlung?)
  - Übertragen von Werten (Lesen/Schreiben)
  - Aktualisieren des Kontostandes (Addieren/Subtrahieren)
  - Ausstellen von Zahlungsanweisungen/Rechnungen
  - Festellen von offenen Zahlungsverpflichtungen (Zahlenvergleich)
  - Ausstellung von Mahnungen
- Beobachtungen
  - Die Bearbeitungsabläufe sind unabhängig von der Warenart
  - Die Grundfähigkeiten sind nicht problemtypisch
  - Die Anforderungen an die Umgebung sind klar definiert (Formulare als "Schnittstelle")
  - Jeder Akteur mit den erforderlichen Fähigkeiten kann die anfallenden Vorgänge erledigen und wird ein identisches Ergebnis erzielen.
  - Es gibt alternative Ablaufpläne die für das gleiche Problem zum gleichen Ergebnis führen.

$\rightarrow$ Abstraktion

Information - Repräsentation

Nachrichten/Signale: Physikalische Phänomene der realen Welt

- ☐ Zustände von Nachrichtenträgern (→ Speicherung)
- ☐ zeitliche Veränderung von Zuständen (→ Übertragung)

Träger: physikalisches Substrat

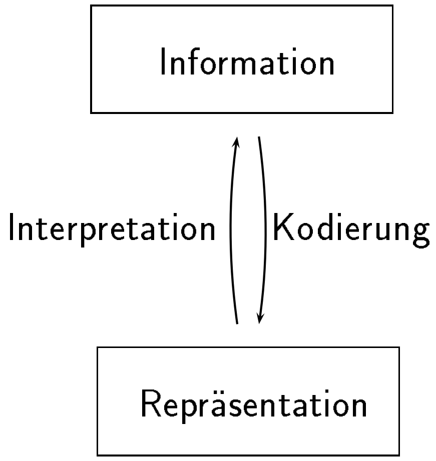
Information: Kenntnis über bestimmte Sachverhalte und Vorgänge in der wahrgenommenen Realität

- ☐ Information ist personen- und interessengebunden
- ☐ Information ist eine abstrakte Kategorie, sie kann nur in Form konkreter *Repräsentationen* ausgetauscht (und verarbeitet) werden

Information - Repräsentation

Daten: abstrakte Nachrichten ohne Berücksichtigung der konkreten physikalischen Realisierung

Datum: Paar aus Repräsentation und seiner Interpretation (Bedeutung)



- ☐ digitale (diskrete) Repräsentationen
- ☐ analoge (kontinuierliche) Repräsentationen

Datenverarbeitung: Manipulation von Repräsentationen in Übereinstimmung mit der Bedeutung dieser Repräsentationen ("bedeutungstreu")

Datenträger: Medium, auf dem Daten (besser: Repräsentationen von Daten) aufbewahrt werden können (Papier, Lochkarten, Disketten, CD-ROMs, ...)

- ☐ nur kontinuierliche Träger sind für kontinuierliche Repräsentationen verwendbar (Papier, Tonband, ...)

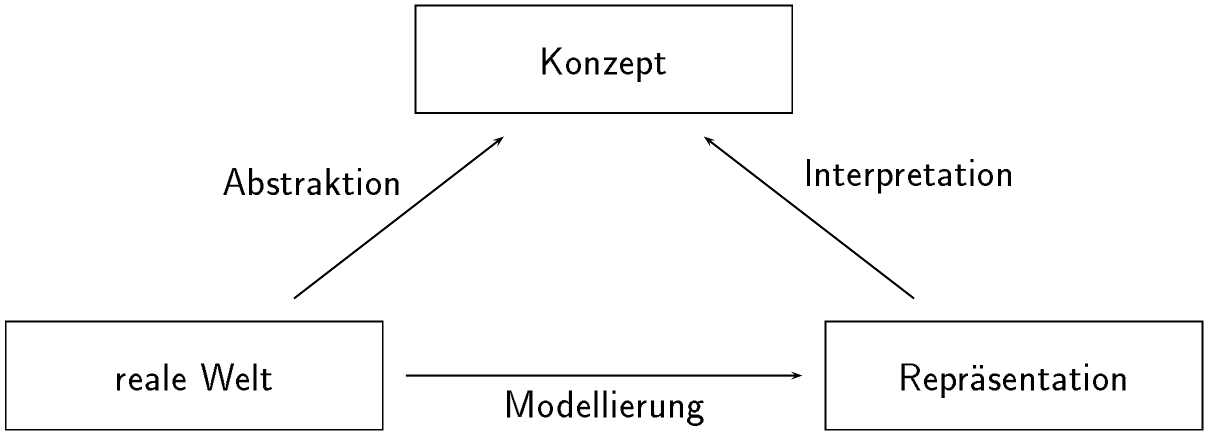
Information - Repräsentation

Semantik einer Repräsentation: Paar aus einer Interpretationsvorschrift und der durch sie erzeugten Information

- ☐ Repräsentationen ohne Interpretationsvorschrift sind sinnlos.
- ☐ Es kann verschiedene Interpretationen derselben Repräsentation geben (Ambiguität).
- ☐ Es kann verschiedene Repräsentationen für dieselbe Information geben (Synonymie).
- ☐ Interpretationsvorschriften sind nicht darstellbar! Definition nur durch soziale Übereinkunft.
  → Quelle zahlreicher Mißverständnisse und "Systemfehler"

Programme: Repräsentationen werden auch für dynamische Abläufe (Algorithmen) benötigt. Unterscheidung zwischen Daten und Programmen ist aber relativ (vgl. HTML).

Modellbildung



Abstraktion: Hervorheben wesentlicher, Unterdrücken unwesentlicher Eigenschaften

Modellbildung: Beschreibung eines Ausschnittes der realen Welt

Interpretation: Zuordnen einer Bedeutung zu einer Repräsentation

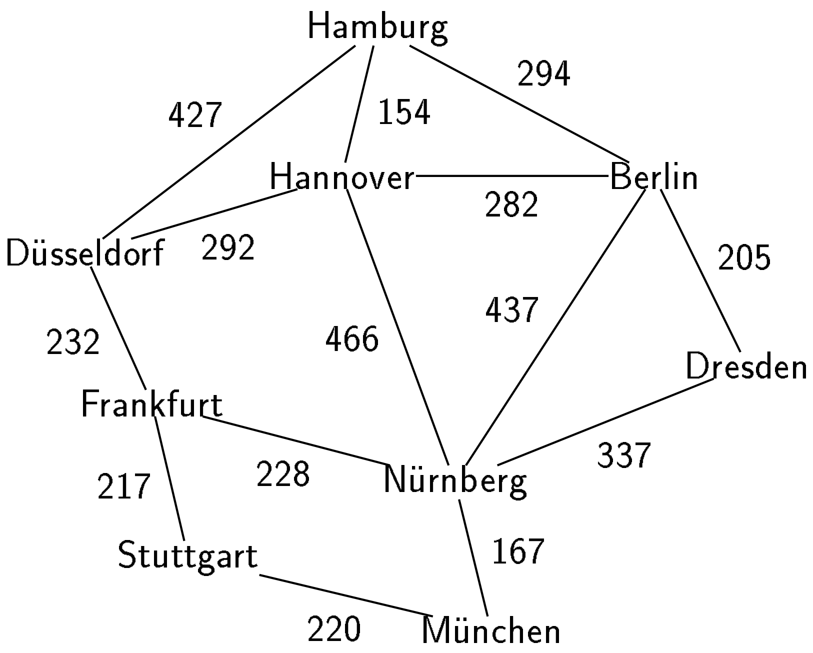
Beispiel: Reiseplanung

- Abstraktion von konkreten Eigenschaften
  - Fahrbahnbeschaffenheit
  - Kurvenverlauf
  - Steigungen
  - maximale Durchfahrtshöhen
  - maximale Brückentragkraft
  - Höchstgeschwindigkeiten
  - Verkehrsdichte usw.

  Konzentration auf Entfernung

- Konzept: Teilstrecke(Ort-2, Ort-1, Entfernung)

- Repräsentation A:



Beispiel: Reiseplanung

- Repräsentation B:

	B	DD	D	F	HH	H	M	N	S
B		205			294	282		437	
DD	205							337	
D				232	427	292			
F			232					228	217
HH	294		427			154			
H	282		292		154			466	
M								167	220
N	437	337		228		466	167		
S				217			220		

- Repräsentation C:

```
connect(hh,b,294).    connect(hh,h,154).
connect(hh,d,427).    connect(h,b,282).
connect(h,d,292).     connect(h,n,466).
connect(d,f,232).     connect(f,n,228).
connect(f,s,217).     connect(s,m,220).
connect(n,m,167).     connect(b,n,437).
connect(b,dd,205).    connect(n,dd,337).
```

Repräsentationen

- Praktisch alle Repräsentationen sind beschränkt, d.h. sie gestatten nicht die Darstellung aller möglichen Informationen.
  - endliche Genauigkeit ($\pi \approx 3.14159265358979323846 26\ldots$)
  - endliche Zahlenbereiche (kleine ganze Zahlen: -32768 . . . 32767)

> → Kann man trotzt der Beschränktheit einer Repräsentation noch adäquat mit ihr arbeiten?

- Unterschiedliche Repräsentationen sind unterschiedlich gut an bestimmte Verarbeitungs-/Übertragungsaufgaben angepaßt.
  - Arithmetik mit römischen bzw. arabischen Zahlen
  - Mathematische Repräsentationen für physikalische Gesetze

$$s = \frac{a}{2}t^2 + v_0 t + s_0$$

  - Chemische Strukturformeln

```
        H     H
        |     |
  H  —  C  —  C  —  OH
        |     |
        H     H
```

> → Wie findet man eine geeignete Repräsentation?

Standardrepräsentationen

Zeichen: Element aus einer vereinbarten Menge von verschiedenen Elementen

Ziffern: {1,2,3,4,5,6,7,8,9,0}

Buchstaben: {A,B,C,D, ...,Z}

Morsezeichen: {·,–}

Binärzeichen: {O,L}

Alphabet: Endliche Menge von Zeichen.

Wort: endliche Zeichenfolge über einem Alphabet, einschließlich der leeren Folge

- Es gibt unendlich viele verschiedene Worte über einem Alphabet

Hierarchische Repräsentationen: Worte über einem Alphabet $\mathcal{A}$ können die Zeichen eines Alphabets $\mathcal{B}$ sein

ASCII-Kode

- American Standard Code for Information Interchange
- $128 = 2^7$ Zeichen (nicht alle druckbar)
- Ursprung: Fernschreibtechnik

0	nul	48	0	65	A	97	a
7	bel	49	1	66	B	98	b
9	ht	50	2	67	C	99	c
10	lf	...		...		...	
13	cr	57	9	90	Z	122	z
27	esc	58	:	91	[	123	{
32	␣	59	;	92	\	124	|
33	!	60	<	93	]	...	
...		...		...		127	del

- Extended ASCII: 8-bit Code

Zahlensysteme

Dezimalzahlen: Zahlen zur Basis 10

Worte über den Ziffern 0 ... 9

$$3785 = \begin{array}{r} 3 * 10^3 \\ +7 * 10^2 \\ +8 * 10^1 \\ +5 * 10^0 \end{array} = \begin{array}{rr} 3* & 1000 \\ +7* & 100 \\ +8* & 10 \\ +5* & 1 \end{array}$$

Ziffernposition bestimmt Potenz der Basis

Verallgemeinerung:

R	Zahlenrepräsentation
B	beliebige Basis
W	numerischer Wert einer Zahl
N	Anzahl von Ziffern in der Zahl
Z	numerischer Wert der Ziffer an einer bestimmten Position

Interpretationsvorschrift:

$$W_B(R) = Z_{N-1} * B^{N-1} + Z_{N-2} * B^{N-2} + \ldots + Z_1 * B^1 + Z_0 * B^0$$

Zahlensysteme

Oktalzahlen: Zahlen zur Basis 8
Worte über den Ziffern 0 . . . 7

$$7311_8 = \begin{array}{r} 7*8^3 \\ +3*8^2 \\ +1*8^1 \\ +1*8^0 \end{array} = \begin{array}{rr} 7* & 512 \\ +3* & 64 \\ +1* & 8 \\ +1* & 1 \end{array} = 3785_{10}$$

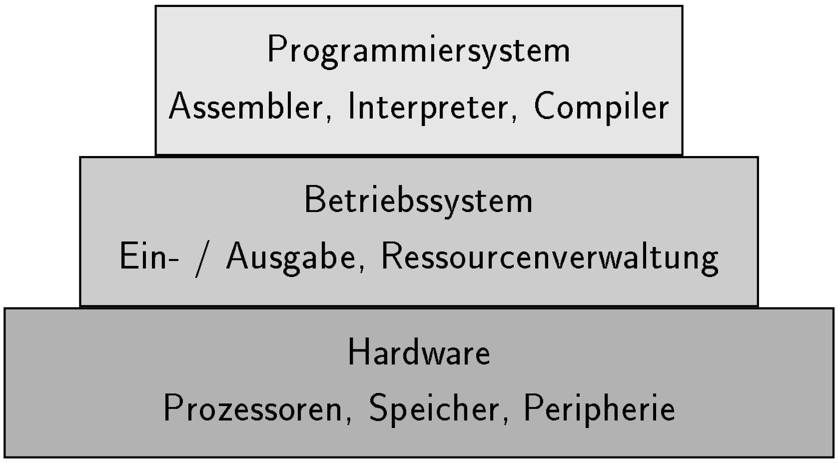
Hexadezimalzahlen: Zahlen zur Basis 16
Worte über den Ziffern 0, . . . , 9, A, B, C, D, E, F

$$\mathrm{DC9}_{16} = \begin{array}{r} 13*16^2 \\ +12*16^1 \\ +9*16^0 \end{array} = \begin{array}{rr} 13* & 256 \\ +12* & 16 \\ +9* & 1 \end{array} = 3785_{10}$$

Dualzahlen: Zahlen zur Basis 2
Worte über den Ziffern 0, 1 (auch O, L)

Architektur informationsverarbeitender Systeme

- Schichtenarchitektur (stark vereinfacht)



- Warum Schichtenarchitektur?
  - Verstecken (Abschirmen, Verschatten) von technischen Details der darunterliegenden Ebenen (auch: Transparenz)
  - Realisierung abstrakter Maschinenmodelle
  - Bewahren der universellen Ausdrucksmächtigkeit

Hardware

- Basis: Binäre Repräsentationen

  → extrem nutzerunfreundliche Repräsentation

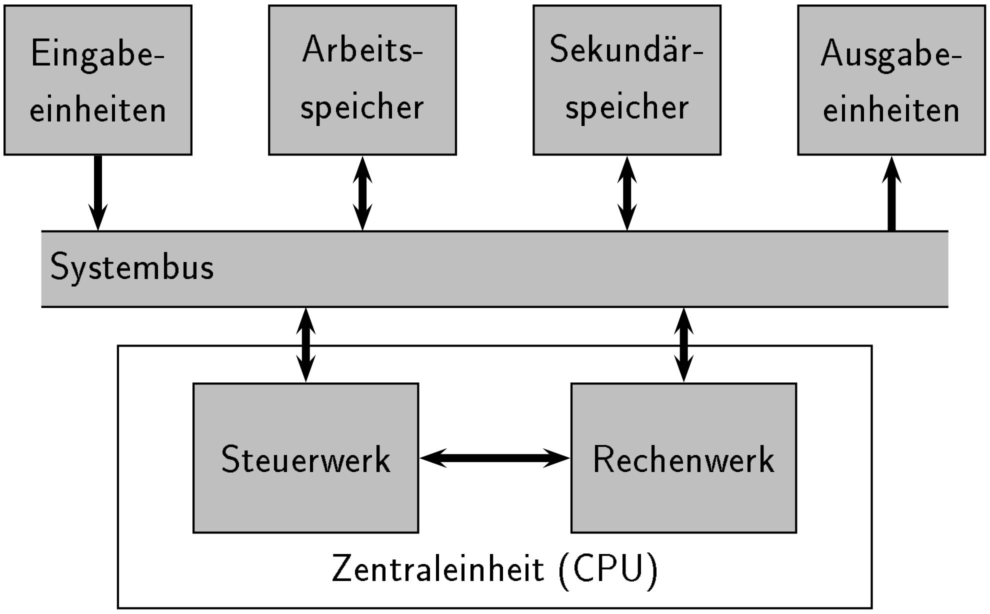
- *Alle* Objekte sind binär repräsentiert
  - Buchstaben
  - Zahlen (ganze Z., rationale Z., komplexe Z., Gleitkommazahlen)
  - Befehle (Maschinenkode, Maschinenprogramm, Binärkode)
    - Operationen
    - Adressen (Speicheradressen, Geräteadressen)

0001 1101	1110 0001	1010 0101 1101 0000
Op-Kode	zus. Inform.	Operandenspezifikation
z.B. LOAD	z.B. Adreß-modus, Register	z.B. Adresse des Operanden

- Interpretation der Bitfolgen liegt in der Hand des Programmierers

  → erstrangige Fehlerquelle

Grundstruktur eines Digitalrechners



Eingabeeinheiten für Daten und Programme: Tastatur, Maus, CD-ROM

Ausgabeeinheiten für Berechnungsresultate: Display, Drucker

Arbeitsspeicher für Daten und Programme: Halbleiter RAM

Sekundärspeicher für Daten und Programme: Festplatten, Disketten, Magnetbänder

Steuerwerk: greift auf Befehle im Arbeitsspeicher zu, analysiert sie und steuert deren Abarbeitung

Rechenwerk: führt arithmetische und logische Grundoperationen aus

Betriebssystem

Ein-/Ausgabeverwaltung

- Behandlung gerätespezifischer Besonderheiten: Kommunikationsanweisungen und Kodierungsvorschriften für Display, Tastatur, Maus, Drucker, Netzanbindung, ...
- Verwaltung der Geräteadressen

Speicherverwaltung

- Behandlung gerätespezifischer Besonderheiten für Festplatten, Wechselplatten, Magnetbänder ...
- Filesystem
- Vergabe von Adressbereichen im Arbeitsspeicher
- Verwaltung virtueller Adressräume

Nutzerinteraktion

- Entgegennahme und Ausführung von Nutzerkommandos

Prozeßverwaltung

- Starten und Beenden von Programmläufen (z.B. geregelter Abbruch)
- Zuordnung von Ressourcen und Nutzern zu Prozessen
- Interprozeßkommunikation

Zugriffsberechtigungen

- Nutzeridentifizierung (Paßwort-Kontrolle)
- Privilegien
- Verbrauchsabrechnung

Programmiersystem

Assemblersprachen

- ☐ Symbolische Befehls- und Adressnotation
- ☐ einfache Speicherverwaltung
  $\rightarrow$ bestimmte (direkte) Adreßfehler werden vermieden
- ☐ rudimentäre Variablendeklaration mit Typangabe
  $\rightarrow$ bestimmte Fehlinterpretationen werden vermieden
- ☐ eineindeutige Zuordnung zu Maschinenbefehlen bleibt erhalten (Ausnahme: Makroexpansion)

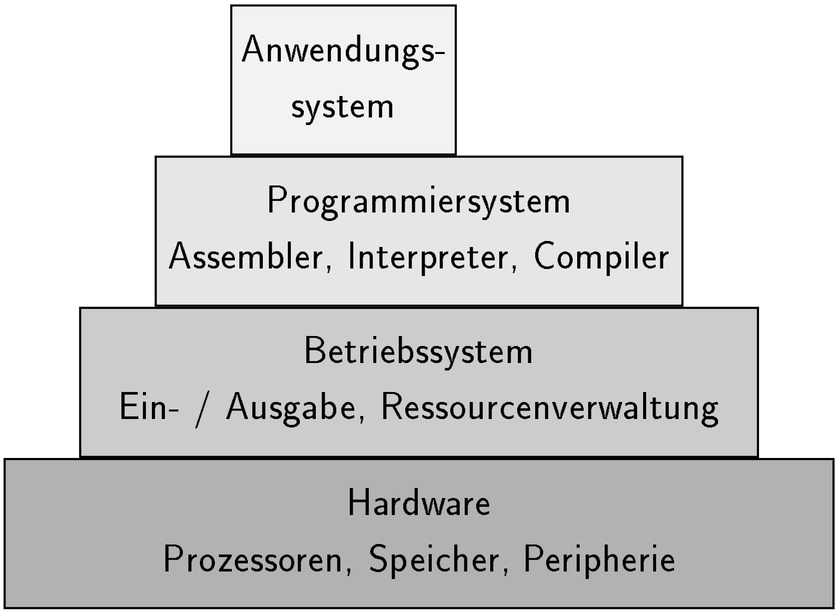
Höhere Programmiersprachen

- ☐ komplexe Befehle (entsprechen mehreren Maschinenbefehlen)
- ☐ Steuerstrukturen (Verzweigung, Zyklen)
- ☐ Variablendeklaration mit reichhaltigem Angebot an Datenstrukturen
- ☐ bequeme IO-Aufrufe
- ☐ Unterprogrammtechnik
- ☐ Verarbeitung durch
  - Interpretation: Quelltextanalyse zur Laufzeit und direkter Befehlsaufruf
  - Compilation: Übersetzung in ein lauffähiges Maschinenprogramm

Systementwicklung

Softwarekrise: Anfang der siebziger Jahren mußten mehrere größere Softwareentwicklungen erfolglos abgebrochen werden.

- ☐ Komplexität der Aufgabenstellung war nicht mehr beherrschbar
- ☐ arbeitsteilige Softwareentwicklung als Schwachstelle



Entwicklungsrichtungen

1. **Softwaretechnik:** Programmiermethodik/Programmierdisziplin

- ☐ Techniken zur Entwicklung von Qualitätssoftware
- ☐ Unterstützung eines guten Programmierstils durch Konstrukte der Programmiersprache
- ☐ Bewahren der Universalität der Programmiersprache

Strukturierte Programmierung

↓

Modularisierung

↓

Abstrakte Datentypen

↓

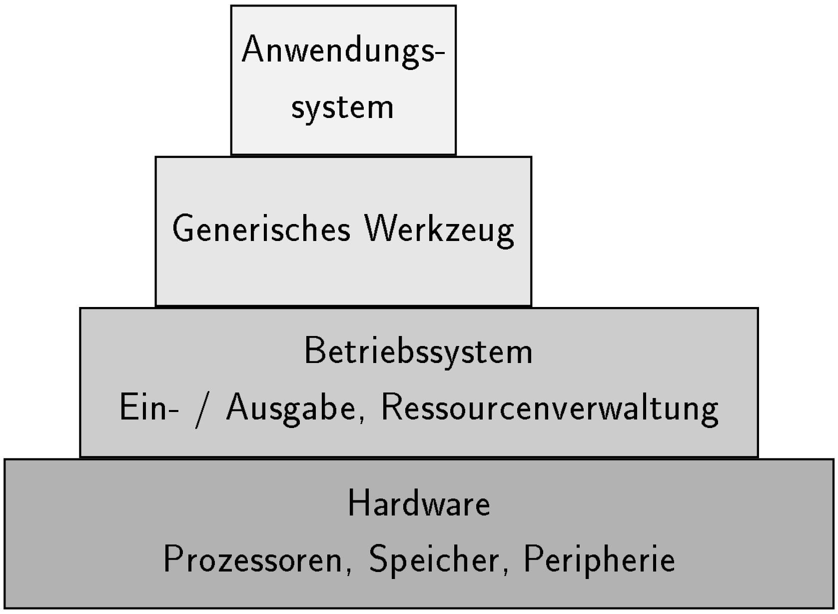
Objektorientierte Programmierung

→ Wie schreibe ich gute Software für eine komplexe Aufgabenstellung?

Entwicklungsrichtungen

2. Systeminformatik: Entwicklung generischer Software-Werkzeuge

- ☐ spezielle Werkzeuge für spezielle Aufgabenklassen
- ☐ Aufgabe des Universalitätsanspruchs
- ☐ vereinfachtes Maschinenmodell
  - ∘ einfache Erlernbarkeit
  - ∘ effiziente Abarbeitung



→ Welches Software-Werkzeug ist für meine Aufgabenstellung das am besten geeignete?

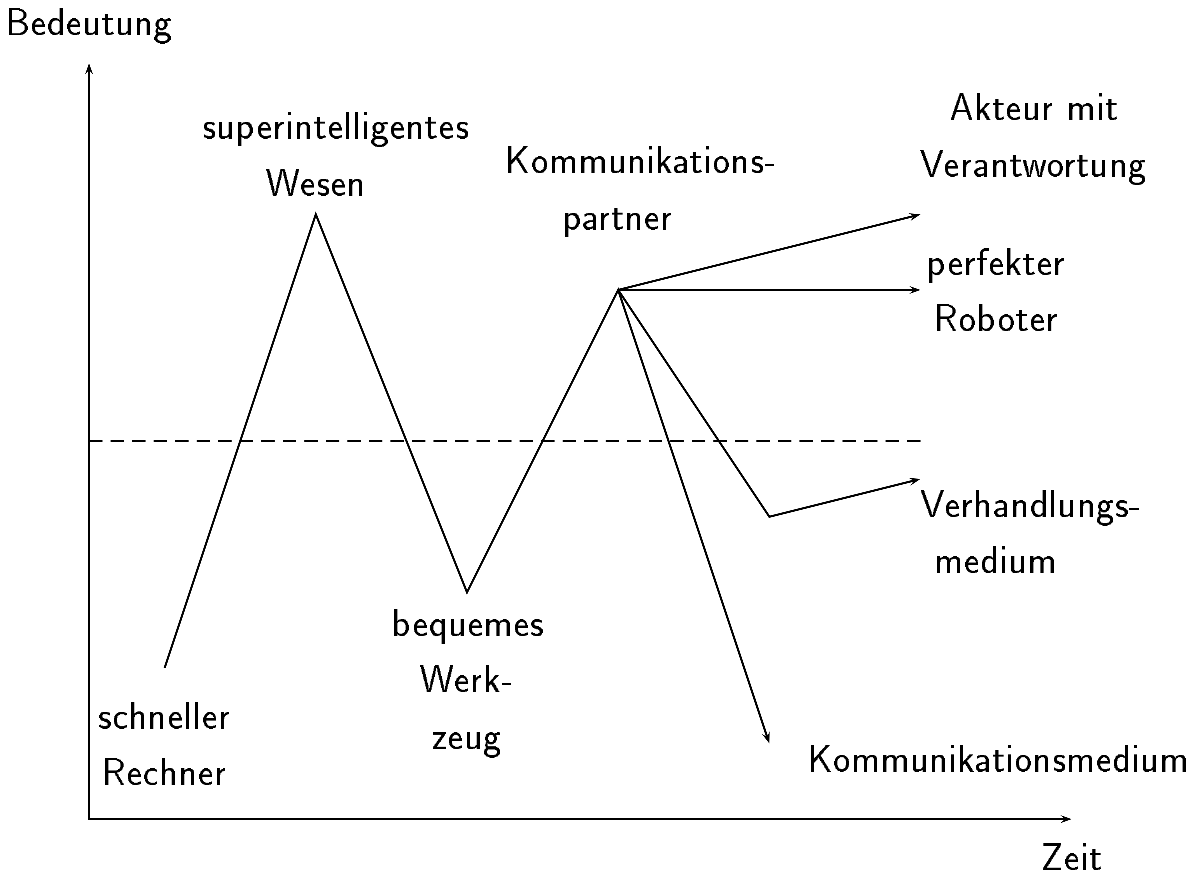
Generische Softwarewerkzeuge

- Texteditoren
- Textverarbeitungssysteme
- Grafikeditoren
- Datenbanken
- Tabellenkalkulation
- Compiler-Generatoren
- Formelmanipulationssysteme
- Statistiksoftware
- Simulationssoftware
- Hypertextsysteme
- ...
- Integrierte Systeme

Mensch und Computer

- Was können Computer (im Prinzip)?
  $\longrightarrow$ Berechenbarkeit, Formalisierbarkeit
- Was können Menschen?
  - Ist Denken berechenbar?
  - Ist alles Wissen formalisierbar?
  - Ist alles Wissen kommunizierbar?
- Was können Computer besser?
- Was können Menschen besser?
- Was soll dem Menschen vorbehalten bleiben?
- Was ist eine optimale Arbeitsteilung?

Unter- und Überschätzungen



C.A.Petri, 1983

Ziel der Vorlesung

Historische Entwicklung:

↓

Entwicklung höherer Programmiersprachen

↙ ↘

systematische Softwareentwicklung

Entwicklung von Softwarewerkzeugen

traditionelle Sicht: Informatik = Programmierung von Rechenmaschinen
Einführung in die Informatik = Einführung in die Programmierung

Entwicklungstendenz: Schaffung von Spezialwerkzeugen für spezielle Aufgaben

- ☐ Vorteil: einfache Erlernbarkeit, Effizienz

Experiment: Durchlauf durch verschiedene Formalismen/Werkzeuge

Ziel der Vorlesung

Ziel: Kennenlernen verschiedener Formalismen

→ Auswahl geeigneter Werkzeuge zur Lösung praktischer Aufgaben

→ Vermittlung von Grundbegriffen der Informatik

→ Vermittlung eines Überblicks über die Informatik

Vorteil: Verbindung von formalen Grundlagen mit Anwendungen

Problem: unterschiedliche Notationssysteme (syntaktische Spezifikationen)

→ nicht einfach

→ intensive Mitarbeit erforderlich

Typische Fragestellungen:

- ☐ Für welche Aufgabenstellungen ist das Werkzeug gut geeignet?
- ☐ Für welche Aufgabenstellungen ist das Werkzeug notfalls geeignet?
- ☐ Welche Informationsstruktur (Formalismus) liegt dem Werkzeug zugrunde?
- ☐ Welche Repräsentationen für die Informationsstruktur sind gebräuchlich?
- ☐ Welche Eigenschaften hat die Informationsstruktur?
- ☐ Für welche Aufgabenstellungen ist das Werkzeug nicht mehr geeignet?
- ☐ Kann man mit dem Formalismus rechnen?
- ☐ Welche Erweiterungen am Formalismus sind möglich/wünschenswert?