EINI LW

Einführung in die Informatik für Wirtschaftsmathematiker

Vorlesung 2 SWS WS 11/12

Dr. Lars Hildebrand
Fakultät für Informatik – Technische Universität Dortmund
lars.hildebrand@udo.edu
http://ls1-www.cs.uni-dortmund.de

technische universität dortmund

Dr. Lars Hildebrand

- **Rechensysteme**
  - Was macht ein Rechensystem aus?
  - Schichten eines Computer Systems
  - Wozu werden Rechensysteme benutzt?

- **Datendarstellung**
  - Grundbegriffe
  - Programme, Texte, Graphiken
  - Logische, bzw. boolesche Werte
  - Natürliche Zahlen, ganze Zahlen, Gleitpunktzahlen
  - Daten vs. Informationen

Was macht ein Rechensystem aus ?

Dr. Lars Hildebrand

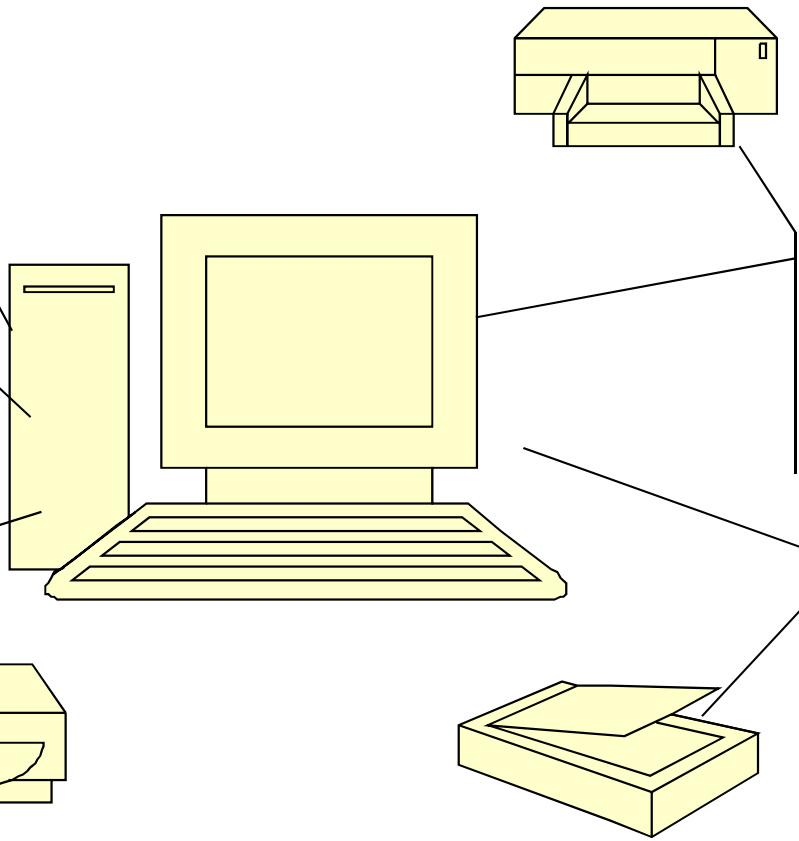
Hardware

- physikalische Komponenten,
- die spezielle Leistungen erbringen,
- Funktionen verfügbar machen

CPU, Prozessor führt Befehle aus Maschinensprache aus

Verbindungssystem (Bus) übermittelt Daten zwischen Komponenten

Speichermedien halten Daten (auch auf Dauer)



Ausgabemedien machen Daten sichtbar

Eingabemedien erlauben Dateneingabe

Was macht ein Rechensystem aus?

Dr. Lars Hildebrand

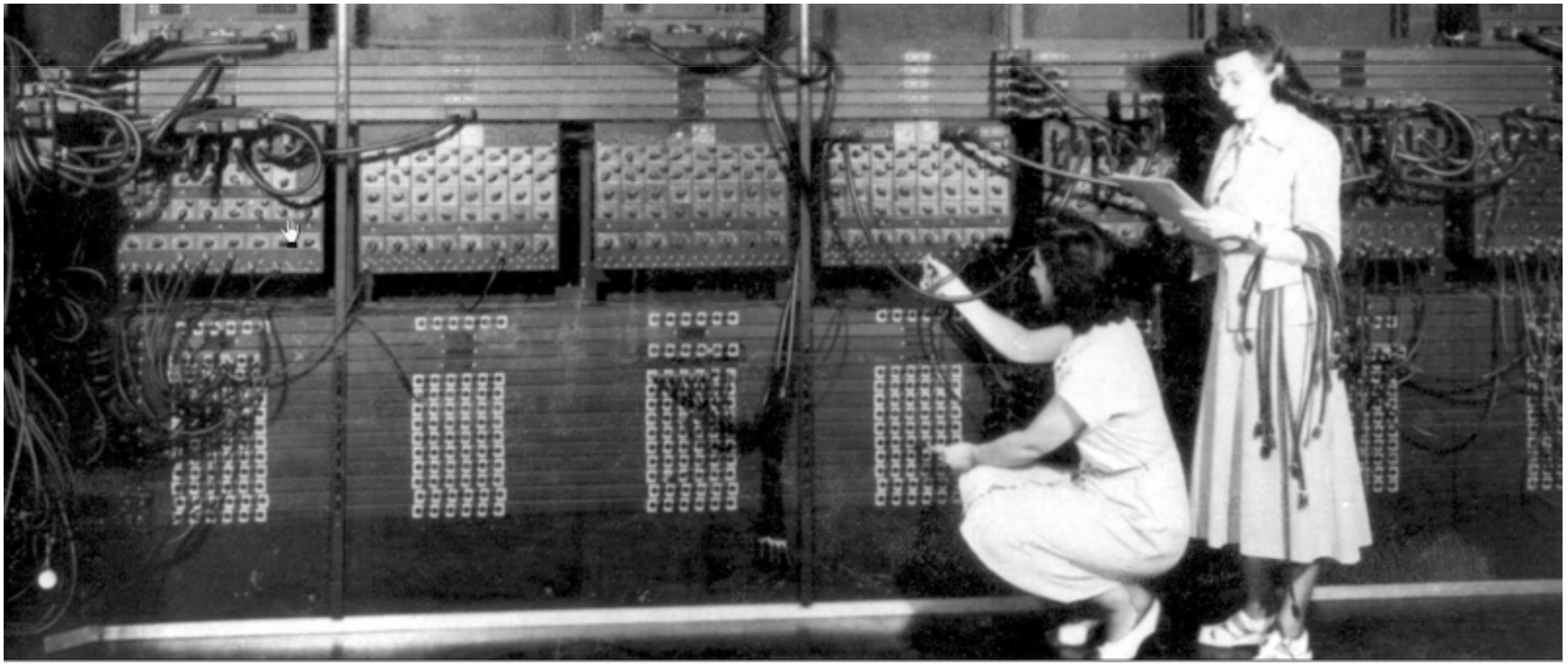
Wofür wird es denn verwendet?

- Von Privatanwendern:
  - Textverarbeitung
  - Tabellenkalkulation
  - Elektronische Post (Email)
  - Internet surfen, Informationsbeschaffung, ...

- Von Firmen:
  - zusätzlich zu den vorherigen Punkten,
  - Verwaltung von Firmendaten und Arbeitsvorgängen, Produktionsplanung und -steuerung, Buchhaltung, ...
    - Datenbankapplikationen,
    - Enterprise Ressource Systeme wie SAP R/3
  - Steuerung automatisierter Fertigungsanlagen

Warum geht das?

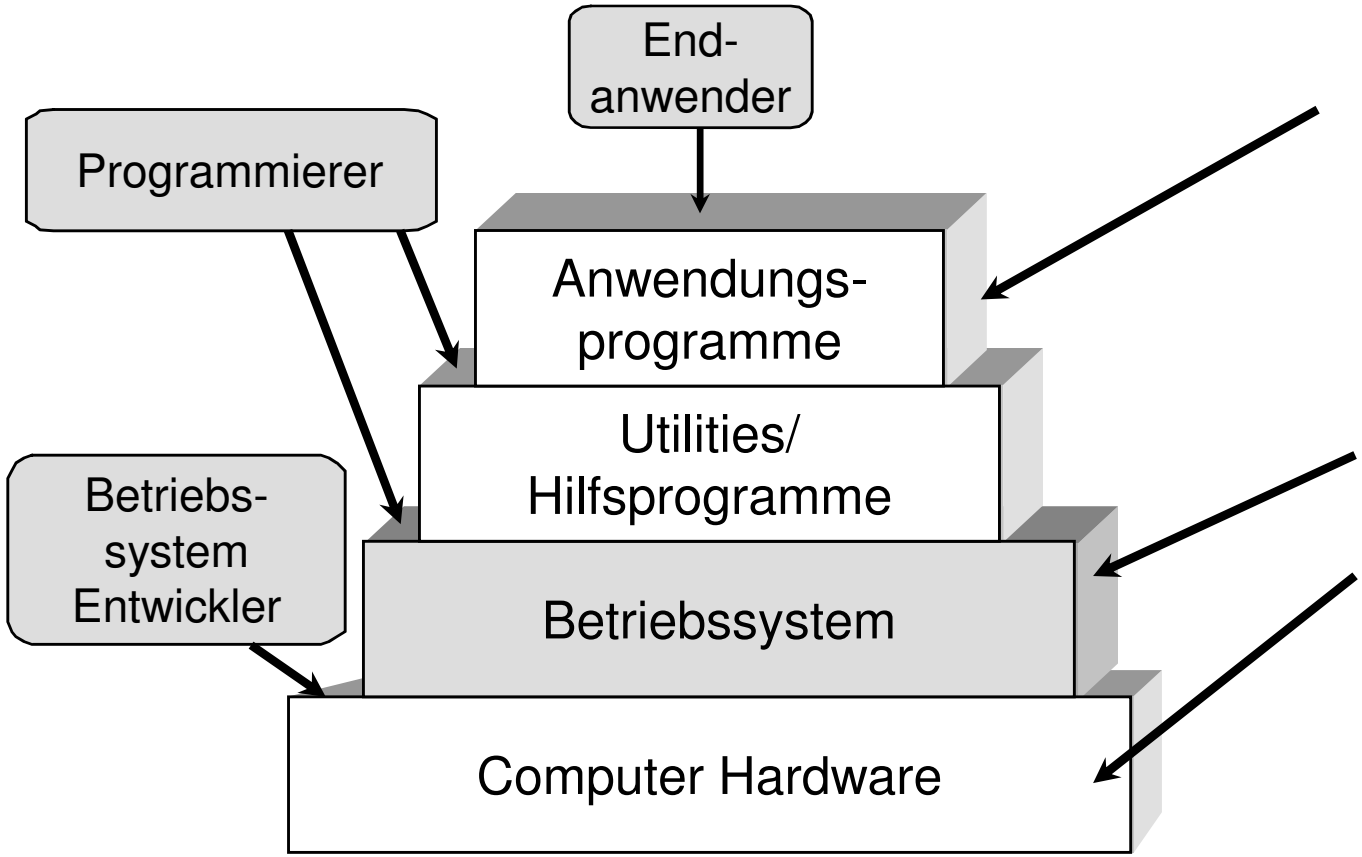
- Rechensysteme sind flexibel einsetzbar,
- ihre Fähigkeiten lassen sich an die jeweiligen Anforderungen anpassen durch
  - Programmierung
  - Softwareentwicklung



Quelle: TomsHardware.com

Schichten (Sichten) eines Rechensystems

Dr. Lars Hildebrand



Beispiel:
Powerpointpräsentation
Powerpoint nutzt
- Dateisystem,
- logische E/A Geräte

Powerpointprozess wird vom BS verwaltet, mittels CPU ausgeführt, Dateisystem, E/A Geräte werden unter Nutzung realer Hardware verfügbar gemacht.

Das Betriebssystem vermittelt zwischen Anwendungsprogrammen und realer Hardware. Z.B. Powerpoint hat eine „Save“ Funktion zum Speichern einer Präsentation in einer Datei. Das Betriebssystem stellt hierzu elementare Operationen zum Lesen / Schreiben von Dateien zur Verfügung und verwaltet Dateien. Es realisiert eine virtuelle Maschine.

Virtuelle Maschine

Dr. Lars Hildebrand

Eine virtuelle Maschine ist eine Menge von Funktionen /Diensten, die sich real nutzen lassen, jedoch durch Software hergestellt werden.

- häufig verwendetes Konzept der Informatik zur Beherrschung von Komplexität durch **Zerlegen in beherrschbare Teilprobleme** und **Kopplung der Teillösungen** mittels Schnittstellen (Teile&Herrsche), wobei eine Teillösung (VM) dann in unterschiedlichen Kontexten genutzt werden kann.

- z.B. Betriebssystem: erzeugt eine **Schnittstelle** festen Funktionsumfanges (=>VM), so dass Anwendungsprogramme auf dieser VM trotz **unterschiedlicher HW Plattformen** ablaufen können. Eine Leistung ist z.B., die real begrenzte Menge Hauptspeicher für das Anwendungsprogramm als quasi beliebig groß erscheinen zu lassen.

- z.B. Kommunikationsprotokolle: erzeugen eine Schnittstelle zur **Übermittlung** von Daten. Z.B. TCP/IP bewirken zuverlässige Verbindung zwischen Kommunikationspartnern, obwohl in Wirklichkeit einzelne Pakete über Zwischenstationen und mit unsicheren Verbindungen durch ein Netzwerk transferiert werden.

Wozu wird ein Rechensystem genutzt ?

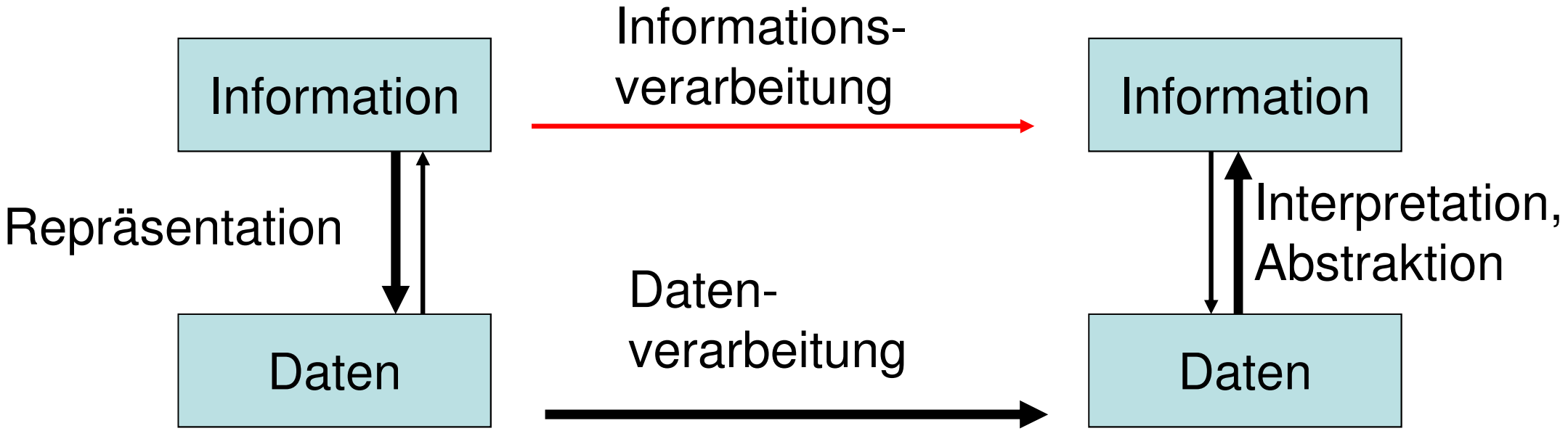
Dr. Lars Hildebrand

Wunsch:

Informationsverarbeitung

Wirklichkeit:

Datenverarbeitung



Anmerkung 1:

Die grundsätzliche Crux der Informatik besteht darin, dass ein System ohne eigenes Verstehen, ohne eigene Erkenntnis geschaffen wird, das dennoch ein sinnvolles Verhalten zeigen soll.

Anmerkung 2:

Repräsentation von Informationen durch Daten kann auf unterschiedlichen Ebenen erfolgen, wir werden uns im direkten Anschluss mit der elementarsten Ebene, nämlich durch Werte 0 und 1, befassen.

Wo stehen wir jetzt ?

Dr. Lars Hildebrand

- Was ist Informatik ?
- Was macht ein Rechensystem aus ?
- Daten vs. Informationen führt zur Frage: Wie werden Daten in einem Rechner dargestellt ?
  - Buchstaben, Zeichenketten, Texte, ...
  - Grafiken
  - Algebren
    - Boolesche Algebra mit Operationen AND, OR, NOT
    - Natürliche Zahlen, ganze Zahlen, reellwertige Zahlen mit Operationen Addition, Subtraktion, Multiplikation, Division, Modulo, ...
    - Achtung: Genauigkeit der Darstellung und damit auch von Berechnungen ist begrenzt!
      Wertebereiche für Zahlen sind beschränkt!

- Wir wollen uns letztlich jedoch mit dem Entwurf von Algorithmen und Programmen, der Programmierung von Rechensystemen und zugehörigen Programmiersprachen befassen.

technische universität dortmund

Dr. Lars Hildebrand

Gliederung

- Grundbegriffe für Datendarstellungen
- Datendarstellung Überblick
- Texte
- Programme, Graphiken
- Logische, bzw. boolesche Werte
- Natürliche Zahlen
  - Umrechnung: Dezimal in Binär
- Ganze Zahlen
  - Zweierkomplement
  - Überprüfung der Zulässigkeit von Resultaten
- Gleitpunktzahlen
- Daten vs. Informationen

technische universität dortmund

Dr. Lars Hildebrand

Bits

- kleinstmögliche Einheit der Information(sdarstellung)
- erlaubt Antwort auf eine Frage mit nur zwei Antwortmöglichkeiten,
  - z.b. {ja,nein}, {wahr,falsch}, {schwarz,weiß}, {links,rechts},
  - meist durch {0,1} codiert

- technisch umgesetzt durch
  - Ladungen: 0 = ungeladen, 1 = geladen
  - Spannungen: 0 = 0 Volt, 1=5 Volt
  - Magnetisierung: 0 = unmagnetisiert, 1 = magnetisiert

Wir gehen im folgenden von {0,1} als verfügbar aus.

technische universität dortmund

Dr. Lars Hildebrand

Bitfolgen

- basieren auf Sequenzen $\{0,1\}^n$, $n \in \mathbb{N}$
- erlauben Codierung von Mengen, z.B.

  000 = Süd 001 = West 010 = Nord 011 = Ost

  100 = Südost 101 = Nordwest 110 = Nordost 111 = Südwest

Es gibt genau 2^N mögliche Bitfolgen der Länge N.

Hexadezimalzahlen

- Bitfolgen werden schnell unübersichtlich, daher Blöcke aus 4 Bits als „Ziffer“

  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

  entspricht Zahlendarstellung zur Basis 16.

technische universität dortmund

Dr. Lars Hildebrand

Byte

- Rechner behandeln nicht einzelne Bits, kleinste betrachtete Bitfolge ist das Byte = 8 Bits,
- gröbere Granularität kommt nur als Vielfaches von 8 Bit vor,
  z.B. 16 Bit, 32 Bit, 64 Bit Rechner
- 1 Byte erlaubt 2^8=256 Werte zu unterscheiden, z.B. zur Codierung von Buchstaben
- Übliche Abkürzungen: B = Byte, b = bit

$$1\text{K} = 1024 = 2^{10}\ (\text{K} = \text{kilo})$$
$$1\text{M} = 1024 \cdot 1024 = 2^{20}\ (\text{M} = \text{mega})$$
$$1\text{G} = 1024 \cdot 1024 \cdot 1024 = 2^{30}\ (\text{G} = \text{giga})$$
$$1\text{T} = 1024 \cdot 1024 \cdot 1024 \cdot 1024 = 2^{40}\ (\text{T} = \text{tera})$$
$$1\text{P} = 1024 \cdot 1024 \cdot 1024 \cdot 1024 \cdot 1024 = 2^{50}\ (\text{P} = \text{Peta})$$
$$1\text{E} = 1024 \cdot 1024 \cdot 1024 \cdot 1024 \cdot 1024 \cdot 1024 = 2^{60}\ (\text{E} = \text{exa})$$

für Längen, Zeiten:

$$1\text{m} = 10^{-3}\ (\text{m} = \text{milli})$$
$$1\mu = 10^{-6}\ (\mu = \text{mikro})$$
$$1\text{n} = 10^{-9}\ (\text{n} = \text{nano})$$
$$1\text{p} = 10^{-12}\ (\text{p} = \text{pico})$$
$$1\text{f} = 10^{-15}\ (\text{f} = \text{femto})$$

Datendarstellung: Überblick

Dr. Lars Hildebrand

- Texte
- Programme
- Bilder
- Zahlen, Algebren
  - Boolesche Algebra, Wahrheitswerte
  - Natürliche Zahlen
  - Ganze Zahlen
  - Reellwertige Zahlen

- Anmerkung:
  - Bei der Darstellung von Zahlen werden wir erkennen, dass nicht alle angenehmen, aus der Mathematik vertrauten Eigenschaften von Zahlen auf einem Rechner erhalten bleiben.

Datendarstellung: Texte

technische universität dortmund

Dr. Lars Hildebrand

- Texte: Zeichenfolgen aus Buchstaben und Satzzeichen
  - Darstellung mittels Bitfolgen
  - Codierung jedes Buchstabens / Zeichens durch Bitfolge

- **ASCII** = American Standard Code for Information Interchange
  - 7 bit (= max 128 Zeichen), Tabelle mit Nummerierung aller Zeichen
  - z.B. „a“ hat Nummer 97, „A“ hat Nummer 65, „?“ hat Nummer 63
  - Klein- und Großbuchstaben nach Alphabet durchnummeriert
  - übliche Erweiterung auf PCs: 8bit, weitere Sonderzeichen, z.B. Umlaute
  - Erweiterung in Europa: Latin-1, (nach Norm ISO 8859-1)

- **Unicode**, z.B. von Java verwendet
  - 16 bit (= max 65536 Zeichen)
  - siehe http://www.unicode.org
  - als Obermenge weltweit geläufiger Zeichensätze

Datendarstellung: Texte

Dr. Lars Hildebrand

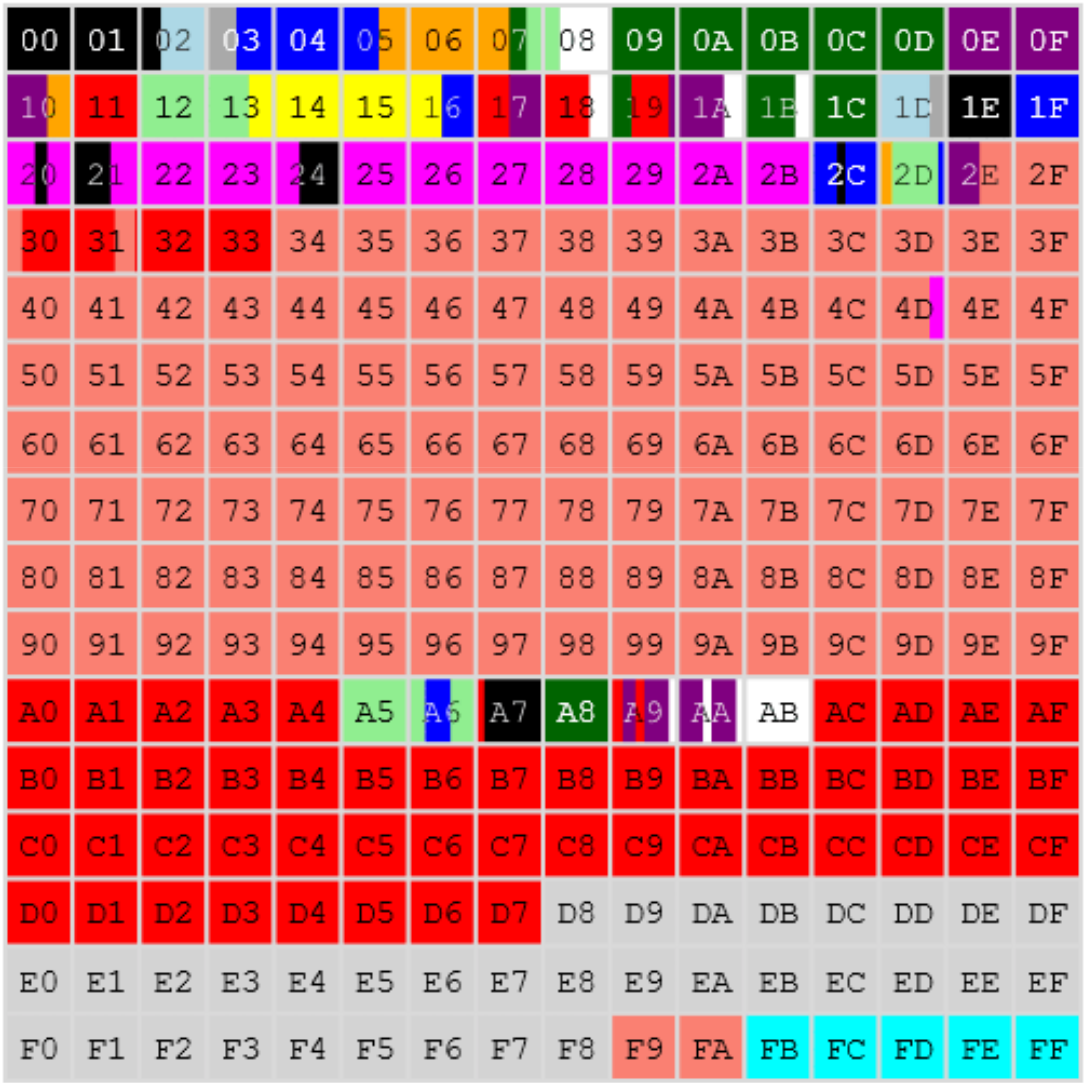
- Unicode Tabelle: Latin 1

0020	0 0030	@ 0040	P 0050	` 0060	p 0070	00A0	° 00B0	À 00C0	Ð 00D0	à 00E0	ð 00F0
! 0021	1 0031	A 0041	Q 0051	a 0061	q 0071	¡ 00A1	± 00B1	Á 00C1	Ñ 00D1	á 00E1	ñ 00F1
" 0022	2 0032	B 0042	R 0052	b 0062	r 0072	¢ 00A2	² 00B2	Â 00C2	Ò 00D2	â 00E2	ò 00F2
# 0023	3 0033	C 0043	S 0053	c 0063	s 0073	£ 00A3	³ 00B3	Ã 00C3	Ó 00D3	ã 00E3	ó 00F3
$ 0024	4 0034	D 0044	T 0054	d 0064	t 0074	¤ 00A4	´ 00B4	Ä 00C4	Ô 00D4	ä 00E4	ô 00F4
% 0025	5 0035	E 0045	U 0055	e 0065	u 0075	¥ 00A5	µ 00B5	Å 00C5	Õ 00D5	å 00E5	õ 00F5
& 0026	6 0036	F 0046	V 0056	f 0066	v 0076	¦ 00A6	¶ 00B6	Æ 00C6	Ö 00D6	æ 00E6	ö 00F6
' 0027	7 0037	G 0047	W 0057	g 0067	w 0077	§ 00A7	· 00B7	Ç 00C7	× 00D7	ç 00E7	÷ 00F7
(0028	8 0038	H 0048	X 0058	h 0068	x 0078	¨ 00A8	¸ 00B8	È 00C8	Ø 00D8	è 00E8	ø 00F8
) 0029	9 0039	I 0049	Y 0059	i 0069	y 0079	© 00A9	¹ 00B9	É 00C9	Ù 00D9	é 00E9	ù 00F9
* 002A	: 003A	J 004A	Z 005A	j 006A	z 007A	ª 00AA	º 00BA	Ê 00CA	Ú 00DA	ê 00EA	ú 00FA
+ 002B	; 003B	K 004B	[005B	k 006B	{ 007B	« 00AB	» 00BB	Ë 00CB	Û 00DB	ë 00EB	û 00FB
, 002C	< 003C	L 004C	\ 005C	l 006C	| 007C	¬ 00AC	¼ 00BC	Ì 00CC	Ü 00DC	ì 00EC	ü 00FC
- 002D	= 003D	M 004D	] 005D	m 006D	} 007D	00AD	½ 00BD	Í 00CD	Ý 00DD	í 00ED	ý 00FD
. 002E	> 003E	N 004E	^ 005E	n 006E	~ 007E	® 00AE	¾ 00BE	Î 00CE	Þ 00DE	î 00EE	þ 00FE
/ 002F	? 003F	O 004F	_ 005F	o 006F	007F	¯ 00AF	¿ 00BF	Ï 00CF	ß 00DF	ï 00EF	ÿ 00FF

Datendarstellung: Texte

Dr. Lars Hildebrand

- Grundlegender mehrsprachiger Codebereich der Unicode Tabelle



technische universität dortmund
Dr. Lars Hildebrand

Programme

- Ein Programm wird zunächst meist als Text (Quelltext) erzeugt und wie normaler Text repräsentiert.
- Übersetzungsprogramme (Compiler) erzeugen daraus Programmcode in Maschinensprache.
- In jeder Form müssen alle Anteile eines Programms durch Bitfolgen codiert werden.

```
public class HelloWorld {
    /**
     * @param args
     */
    public static void main(String
        // TODO Auto-generated met
        System.out.println("Hallo
    }
}
```

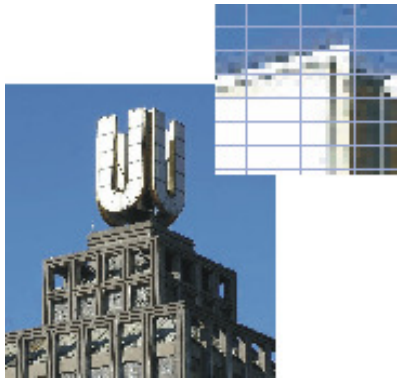
```
// Bytecode stream: 03 3b 84 00 01 1a 05 68 3b a7 ff f9
// Disassembly:
iconst_0      // 03
istore_0      // 3b
iinc 0, 1     // 84 00 01
iload_0       // 1a
iconst_2      // 05
imul          // 68
istore_0      // 3b
goto -7       // a7 ff f9
```

```
:1000000075812F12019912025212025890004D125E
:10001000027B90005B120285750200744D12022D66
:1000200012006912032712000E304F9D8FED9FC7408
:100030000E12022DE59012037A20B304050280020D
:1000400015028502A030B2D37502FF80CE4449501C
:10005000382050312868657829 3A00414455776541
```

technische universität dortmund
Dr. Lars Hildebrand

Grafiken, Bilder

- Rastergrafik
  - Grafik wird aus einer Folge einzelner Rasterpunkte dargestellt
  - Einzelner Rasterpunkt durch 1 Bit, 1 oder mehrere Bytes (wg. Farbe) codiert



- Vektorgrafik
  - Grafik wird aus Linien zusammengesetzt,
  - für die Anfangs- / Endpunkte etc. codiert werden müssen

technische universität dortmund

Dr. Lars Hildebrand

- **Boolesche Algebra**:
  - **Trägermenge** = {false,true} (oft auch als {0,1}) mit
  - **Operationen** (Auswahl)
    - Und-Verknüpfung: AND,
    - Oder-Verknüpfung: OR,
    - Negation: NOT,
    - Exklusives Oder: XOR

- **Darstellung in Rechnern**
  - erfordert meist 1 Byte (mindestens) als kleinste behandelbare Dateneinheit
    - 1 Bit wäre im Prinzip ausreichend, jedoch im Rechner ist ein einzelnes Bit nur als Element innerhalb eines Bytes und über das zugehörige Byte adressierbar
    - Bitfelder dagegen lassen sich mit Platzverbrauch 1 Bit je Boolescher Variable verwalten, wobei das Bitfeld insgesamt jedoch eine Größe in ganzen Bytes haben muss.

technische universität dortmund

Dr. Lars Hildebrand

Satz: Jede natürliche Zahl n besitzt zur Basis $p \geq 2$ ($p \in N$) eine eindeutige m-stellige p-adische Darstellung der Form

$$n = \sum_{i=0}^{m-1} \alpha_i \cdot p^i \quad \text{mit } 0 \leq \alpha_i < p \text{ und } m \geq \log_p n$$

Bemerkungen:

- Ziffern dürfen Basiswert p nicht erreichen!
- Für uns üblich: Dezimalzahlen p=10 und m nach Bedarf

$$2003_{10} = 2 \cdot 10^3 + 0 \cdot 10^2 + 0 \cdot 10^1 + 3 \cdot 10^0$$

Darstellung von natürlichen Zahlen: Beispiel

Dr. Lars Hildebrand

Bemerkungen:

- Ziffern dürfen Basiswert p nicht erreichen!
- Im Rechner üblich: Binärzahlen p=2, m=16, 32 oder 64

$$1110_2 = 1 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0 = 8_{10} + 4_{10} + 2_{10} = 14_{10}$$

- Gelegentlich zur Dokumentation von Zahlenwerten / Adressen:
  - Hexadezimal: p=16
  - Oktal: p=8

Umrechnung: Dezimal in Binär

Dr. Lars Hildebrand

Umrechung von Dezimalzahlen in Binärzahlen durch ganzahlige Division und Modulo Operation, d.h.

$$n = \sum_{i=0}^{m-1} \alpha_i \cdot p^i = \mathrm{p} \cdot (\sum_{i=1}^{m-1} \alpha_i \cdot p^{i-1}) + \alpha_0$$

$$\alpha_0 = n \bmod p \qquad \sum_{i=1}^{m-1} \alpha_i \cdot p^{i-1} = n \div p$$

$$\alpha_i = (n \div p^i) \bmod p$$

- also fortgesetztes Dividieren,
- der Rest r (mathematisch formal: modulo) liefert die Ziffernfolge,
- z.B. $4711_{10} = 1001001100\mathbf{11}_2$
  - 4711 / 2 = 2355 mit Rest 1 -> „**rechteste**“ 1 in der Binärdarstellung
  - 2355 / 2 = 1177 mit Rest 1 -> „**vorletzte**“ 1 in der Binärdarstellung
- Anmerkung: Geläufige Rechenoperationen sind für p-adische Zahlendarstellung unabhängig von p gültig
- Aber Achtung: Bisher nur natürliche Zahlen! Neg. ganze Zahlen machen Ärger!

Umrechnung: Dezimal in Binär - Beispiel

technische universität dortmund

Dr. Lars Hildebrand

Beispiel 2011_{10} =

1005	R 1
502	R 1
251	R 0
125	R 1
62	R 1
31	R 0
15	R 1
7	R 1
3	R 1
1	R 1
0	R 1

Datendarstellung: Ganze Zahlen

Dr. Lars Hildebrand

- Vorzeichenbetragsdarstellung (VB-Zahlen)
  - Standardverfahren unserer Schulmathematik
  - Vorzeichen + oder – (3. + 4. Zeichen in der Kodierung)
  - Anzahl der Stellen muss bekannt sein
  - Unhandlich bei automatisierter Arithmetik

Datendarstellung: Ganze Zahlen

- 2er-Komplement
  - Vermeidet Vorzeichen
  - Anzahl der Stellen muss nicht bekannt sein
  - Erzeugung: Alle Stellen invertieren und 1 addieren
  - Berechnung: höchstwertiges Bit hat negativen Wert

technische universität dortmund

Dr. Lars Hildebrand

Standardformate

- wie bereits angedeutet, realisieren Rechner Zahlendarstellungen nur für bestimmte Wertebereiche (festes m),
- diese Wertebereiche dienen als Datentyp für Variable, analog zu $X \in N$ wird vereinbart: int x
- in einer Programmiersprache wie Delphi (Turbo-Pascal) oder Java werden folgende Bereiche angeboten

Bereich	Größe	Delphi	Java
-128,...,127	8 bit	shortint	byte
-32768,...,32767	16 bit	integer	short
$-2^{31},...,2^{31}-1$	32 bit	longint	int
$-2^{63},...,2^{63}-1$	64 bit		long
0,...,255	8 bit	byte	
0,...,65535	16	word	

Datendarstellung: Festpunktzahlen

- Bisher natürliche und ganze Zahlen (2-er Komplement)

- Bei gebrochenen Zahlen zeigen sich Schwierigkeiten bei der Genauigkeit der Darstellung $x = \sum_{i=0}^{m-1} \alpha_i \cdot 2^i + \sum_{i=1}^{n-1} \alpha_i \cdot 2^{-i}$
  - Hinweise auf Schwierigkeiten gibt es bereits im Dezimalsystem
    - π, keine endliche Darstellung im Dezimalsystem
    - Periodische, gebrochene Dezimalzahl bei 1/3
  - bei Binärzahlen tritt Problem auch bei Zahlen mit endlicher Dezimaldarstellung auf:
    - dezimal 0.1 wird binär zu 0.0001100110011001100110011...
  - wiederum stehen natürlich nur eine endliche und feste Anzahl Bits zur Verfügung ...

technische universität dortmund

Dr. Lars Hildebrand

Gleitpunktzahlen in Programmiersprachen

Bereich	Bytes	Stellen	Delphi	Java
+- 2,9 E -39 .. 1,7 E 38	6	11-12	real	
+- 1,5 E -45 .. 3,4 E 38	4	7-8	single	float
+- 5,0 E -324 .. 1,7 E 308	8	15-16	double	double
+- 3,4 E -4932 .. 1,1 E 4932	10	19-20	extended	

- Die Notation mit „E“ wie Exponent bedeutet:

$$3{,}1415E2 = 3{,}1415 * 10^2 = 314{,}15$$

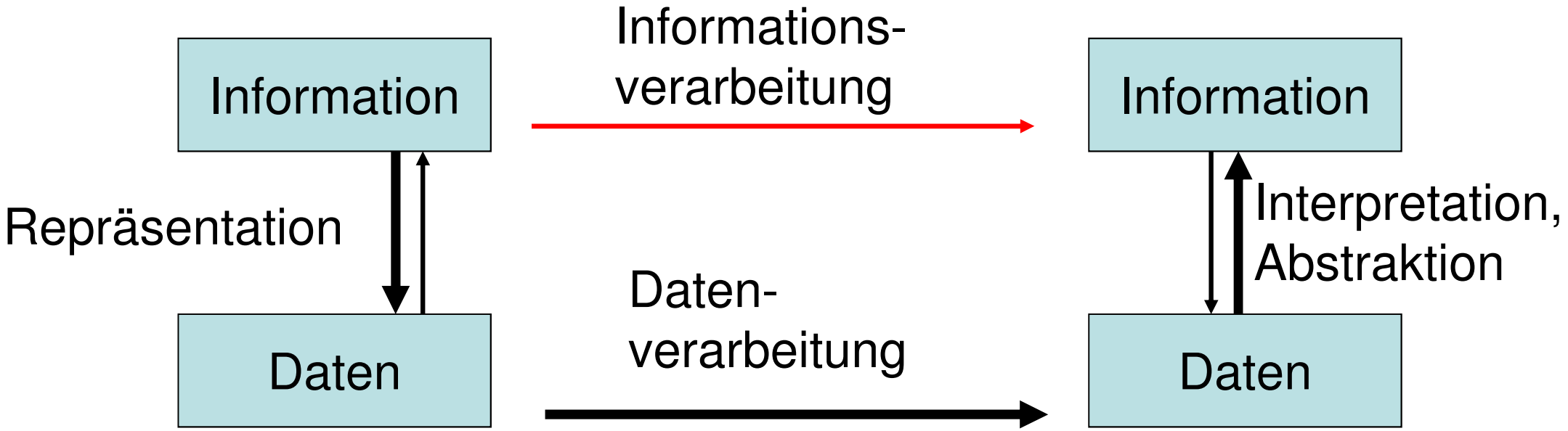
und entstammt Norm IEEE 754, ist in Programmiersprachen üblich.

Daten vs. Informationen

technische universität dortmund
Dr. Lars Hildebrand

Wunsch:
Informationsverarbeitung

Wirklichkeit:
Datenverarbeitung



Bisher betrachtet:

- Behandlung von einfachen mathematischen Objekten, nämlich
- Zahlen (natürliche, ganze Zahlen, reellwertige Zahlen)

=> Repräsentation und Interpretation wesentlich, um Rechensystem mit seinen Fähigkeiten zur Datenverarbeitung für die Informationsverarbeitung sinnvoll nutzen zu können

Wo stehen wir jetzt ?

Dr. Lars Hildebrand

- Was ist Informatik ?
- Was macht ein Rechensystem aus ?
- Daten vs. Informationen führt zur Frage: Wie werden Daten in einem Rechner dargestellt ?
  - Buchstaben, Zeichenketten, Texte, ...
  - Grafiken
  - Algebren
    - Boolesche Algebra mit Operationen AND, OR, NOT
    - Natürliche Zahlen, ganze Zahlen, reellwertige Zahlen mit Operationen Addition, Subtraktion, Multiplikation, Division, Modulo, ...
    - Achtung: Genauigkeit der Darstellung und damit auch von Berechnungen ist begrenzt!
      Wertebereiche für Zahlen sind beschränkt!

- Wir wollen uns letztlich jedoch mit dem Entwurf von Algorithmen und Programmen, der Programmierung von Rechensystemen und zugehörigen Programmiersprachen befassen.

technische universität dortmund

Dr. Lars Hildebrand

Vielen Dank für Ihre Aufmerksamkeit!

Nächste Termine

- Nächste Vorlesung 28.10.2011, 08:30
- Beginn Praktikum 24.10.2011