Rechnerorganisation

Mathematische Grundlagen (1)

Boolesche Algebren: BMA, BAA (2,3)

Kombinatorische Schaltungen (4,5)

Automaten (6,7)

Sequentielle Schaltungen (8)

Programmierbare Strukturen (9)

Rechneraufbau und ~funktion (10,11)

Informationskodierung (12,13,**14**)

Hinweis zum Praktikum RO

Praktikum Rechnerorganisation

Name	Vorname	Matrikel-Nr.	Studiengang

Rechnerorganisation	Datum	Ergebnis	Unterschrift des
Versuch 1: Digitale Grundschaltungen			
Versuch 2: Emulator 8086			
Informationen im Internet: http://www.tu-ilmenau.de/iks/lehre/bachelor-studiengae ecture_id=40			

Im Sommersemester (April, Mai) möglich

Zum Abrechnen melden Sie das Praktikum bitte zunächst im Thoska-System an!
Geben Sie **anschließend** die Testatkarte im **Raum Z 1031** ab (Sekretariat IKS).

Diese Testatkarte ist für den Bachelorstudiengang IN ab der PO-Version **2013** gültig.

Für die Studiengänge BT, und II ist diese Testatkarte **nicht** geeignet.

Hinweis zum Praktikum TI

TECHNISCHE UNIVERSITÄT ILMENAU

Testatkarte

Fakultät für Informatik und Automatisierung

Praktikum Technische Informatik

Name ______ Vorname ______ Matrikel-Nr. ______ Studiengang ______

Technische Informatik 1	Datum	Ergebnis	Unterschrift des Verantwortlichen
Versuch 1			
Versuch 2			
Informationen im Internet: http://tu-ilmenau.de/iks/lehre/bachelor-studiengaenge/			

Technische Informatik 2	Datum	Ergebnis	
Versuch 1			
Versuch 2			
Informationen im Internet: http://tu-ilmenau.de/?r-p-ti2			

Im Sommersemester (April, Mai) möglich

ERGEBNIS	Unterschrift des Verantwortlichen	Eintragung im Online-System (Datum)

Zum Abrechnen melden Sie das Praktikum bitte zunächst im Thoska-System an!

Geben Sie dann die Testatkarte im Raum Z 2055 ab (Sekretariat RAuES).

Diese Testatkarte ist für die Bachelorstudiengänge BT und II ab der PO-Version 2013 sowie für den Diplomstudiengang MB gedacht.

Für andere Studiengänge ist diese Testatkarte nicht geeignet.

Datenkodierung

Daten

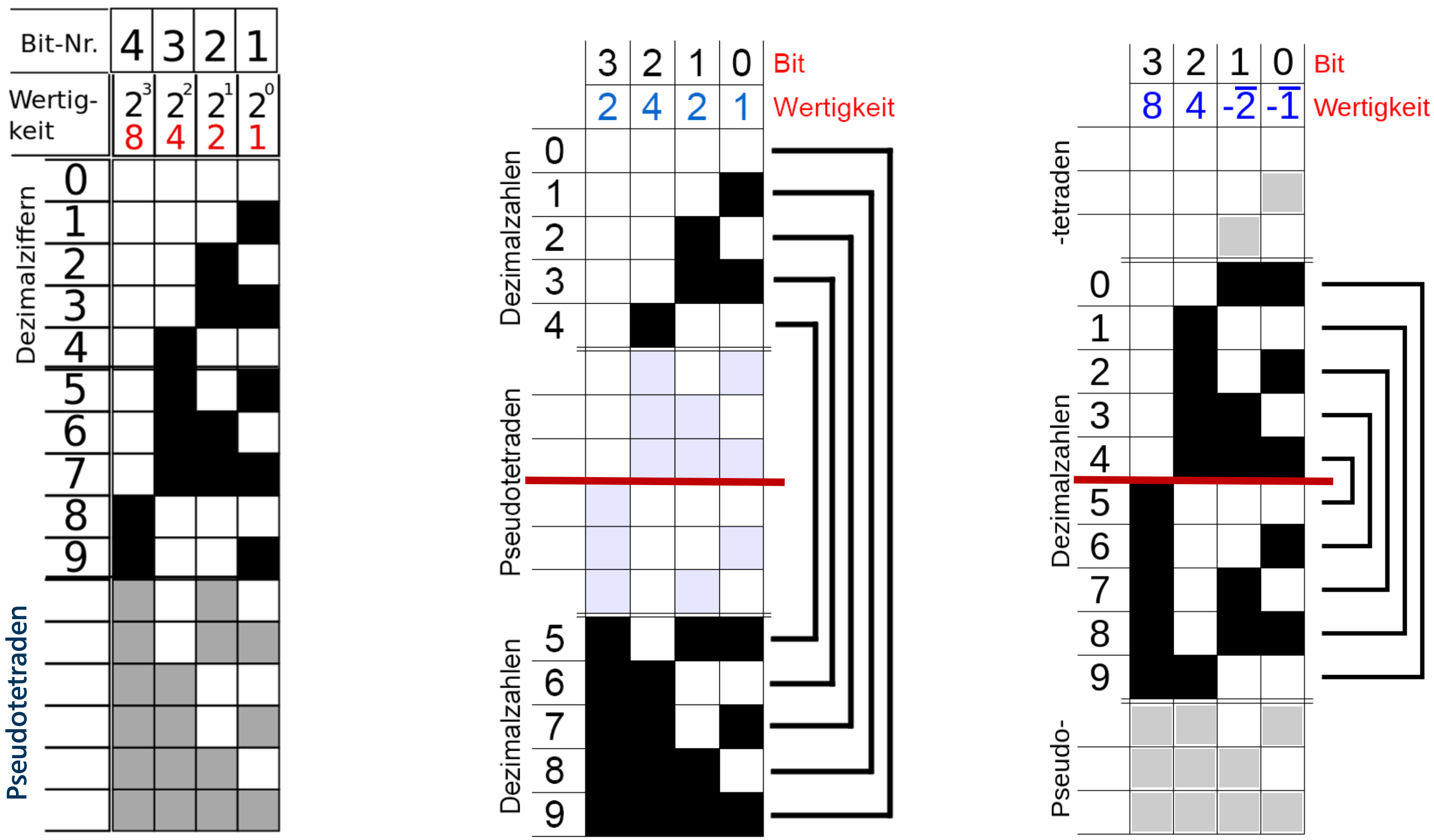
alphanumerische Zeichen

Zahlen

- BCD
- vorzeichenbehaftete Zahlen
- 2K-Zahlen
- Gleitkomma-Zahlen

Zahlenkodierung - BCD

direkter BCD Code | **Aiken Code** | **3xS Code**



Zahlenkodierung - BCD

Operationen

Addition + Korrektur

- bei Addition $\mathcal{K}(z_i + z_i') = \mathcal{K}(z_i) + \mathcal{K}(z_i') + \mathbf{C}$
- bei Subtraktion $\mathcal{K}(z_i - z_i') = \mathcal{K}(z_i) - \mathcal{K}(z_i') - \mathbf{C}$

	BCD	Aiken	3xS
Korrektur erforderlich	bei **ü** oder **p**	nur bei **p**, dann abhängig von **ü**	immer, aber abhängig von **ü**
Konstante C	0110	−0110 bei **ü** +0110 sonst	+0011 bei **ü** −0011 sonst

Datenkodierung

Daten

alphanumerische Zeichen

Zahlen

- BCD
- vorzeichenbehaftete Zahlen
- 2K-Zahlen
- Gleitkomma-Zahlen

Datenkodierung

Daten

alphanumerische Zeichen

Zahlen

- BCD
- vorzeichenbehaftete Zahlen
- 2K-Zahlen
- Gleitkomma-Zahlen

Zahlenkodierung – 2K-Zahlen

Konegative Zahlen (siehe Arbeitsblätter S. 30)

Ergänzung zu 2^n bzw. 2^n-1

- Vb: $-2^{n-1} < z < 2^{n-1}$
- Nb: N_n (üblich: N_8, N_{16}, N_{32})
- KV:

$$\mathcal{K} = (z) = \begin{cases} z_n \leftrightarrow z \geq 0 \\ \overline{z_n} \text{ sonst} \end{cases}$$

z_n ... n-stellige Dualzahl von z

$\overline{z_n}$ ... Komplement der Dualzahl

Komplementbildung

Man unterscheidet Einer- und Zweierkomplemente und dementsprechend als konegative Zahlen 1K- und 2K-Zahlen

- 1K-Zahlen: $\overline{z_n} = 2^n - 1 - z_n$
- 2K-Zahlen: $\overline{z_n} = 2^n - z_n$

$z_{n1} + \overline{z_{n1}} = 2^n$

Zahlenkodierung – 2K-Zahlen

Bildung der 2K-Zahlen

(a) Subtraktion von 2^n: $\overline{z_n} = 2^n - z_n$

(b) 1K-Zahl (Negation) + 1: $\overline{z_n} = 2^n - 1 - z_n + 1$

(c) beginnend von rechts die erste 1 suchen,

diese bleibt stehen,

alle Ziffern links davon invertieren

Zahlenkodierung – 2K-Zahlen

Operationen (siehe Arbeitsblätter S. 30)

$$z_{n1} > z_{n2}\ ,\ z_{n1} + z_{n2} = s_n\ ,\ z_{n1} - z_{n2} = d_n$$

Operation	1k-Zahlen		2K-Zahlen	
	Ergebnis	Korrektur	Ergebnis	Korrektur
$z_{n1} + z_{n2}$	s_n	-	$z_{n1} + z_{n2} = s_n$	-
$z_{n1} + \overline{z_{n2}}$	$d_n + 2^n - 1$	$-2^n + 1$	$z_{n1} - z_{n2} + 2^n = d_n + 2^n$	-2^n
$\overline{z_{n1}} + z_{n2}$	$\overline{d_n}$	-	$-(z_{n1} - z_{n2}) + 2^n = \overline{d_n}$	-
$\overline{z_{n1}} + \overline{z_{n2}}$	$\overline{s_n} + 2^n - 1$	$-2^n + 1$	$-(z_{n1} + z_{n2}) + 2^n + 2^n = \overline{s_n} + 2^n$	-2^n

$$\text{2K:}\quad z_{n1} + \overline{z_{n1}} = 2^n$$

Zusammenfassung Zeichen, Integer

Zahlenbereiche für 1Byte (=8Bit)

ASCII:	0	...	9	
Direkt BCD:	0	...	99	
VZ-Betragszahlen:	-127	...	+127	(+0 -0)
2K-Zahl:	-128	...	+127	
1K-Zahl:	-127	...	+127	(+0 -0)

Zusammenfassung Zeichen, Integer

Binärkode 0011 0110 interpretierbar als

ASCII:	Zeichen	„6“
BCD (direkt):	Zahl	„36“
(3xS):	Zahl	„03“
(Gray):	Zahl	„24“
Vorzeichen-BZ:	pos. Zahl	„54“
2K-Zahl:	pos. Zahl	„54“
1K-Zahl:	pos. Zahl	„54“

Rechnerorganisation

Mathematische Grundlagen (1)

Boolesche Algebren: BMA, BAA (2,3)

Kombinatorische Schaltungen (4,5)

Automaten (6,7)

Sequentielle Schaltungen (8)

Programmierbare Strukturen (9)

Rechneraufbau und ~funktion (10,11)

Informationskodierung (12,13,**14**)

Datenkodierung

Daten

alphanumerische Zeichen

Zahlen

- BCD
- vorzeichenbehaftete Zahlen
- 2K-Zahlen
- Gleitkomma-Zahlen

Gleitkommazahlen (GK); (Floating point, FP)

$$z = \pm M \times B^{E}$$

M = Mantisse
B = Basis
E = Exponent

Normierung:

gleiche Vorkommastelle : 0 (M = 0,...)

$125 = (0{,}125 \times 10^{3})_{10}$ (dezimal)

$7DH = (0{,}7D \times 10^{2})_{16}$ (hexadezimal)

$= (0{,}1111101 \times 10^{111})_{2}$ (dual)

Gleitkommazahlen (GK); (Floating point, FP)

$$z = \pm M \times B^{E}$$

M = Mantisse
B = Basis
E = Exponent

Normierung:

gleiche Vorkommastelle : 1 (M = 1,...)

$125 = (0{,}125 \times 10^{3})_{10}$ (dezimal)

$7DH = (0{,}7D \times 10^{2})_{16}$ (hexadezimal)

$= (0{,}1111101 \times 10^{111})_{2}$ (dual)

$= (1{,}111101 \times 10^{110})_{2}$ (dual)

Gleitkommazahlen (GK); (Floating point, FP)

$$z = \pm M \times B^{E}$$

M = Mantisse
B = Basis
E = Exponent

Normierung:

gleiche Vorkommastelle : 1 (M = 1,...)

$$= \quad (1{,}111101 \times 10^{110})_2 \quad \text{(dual)}$$

- nur im Dualsystem möglich
- 1 wird nicht abgespeichert
- => Doppelte Genauigkeit !

Gleitkommazahlen (GK); (Floating point, FP)

IEEE – Standard (IEEE 754-1985)

- Vb: Menge rationaler Zahlen z
- Nb: $N_{32}\ldots$short, $N_{64}\ldots$long, $N_{80}\ldots$temporary
- KV: $s\ldots$ sign (Vorzeichen der Zahl)
  $e\ldots$ biased exponent (vorzeichenloser Exponent)
  $f\ldots$ fractional part (gebrochener Anteil)

Ermittlung von

s = Vorzeichen
e = vorzeichenloser Exponent
f = gebrochener Anteil

ANSI/IEEE
Std 754-1985

An American National Standard

IEEE Standard for
Binary Floating-Point Arithmetic

Sponsor
Standards Committee of the
IEEE Computer Society

Approved March 21, 1985
IEEE Standards Board

Approved July 26, 1985
American National Standards Institute

- short real (all systems)
- long real (all systems)
- temporary real (x86-based)

Gleitkommazahlen (GK); (Floating point, FP)

$$z_n = \pm \, M \times 10_2^{\,E}$$

$$= \boxed{1{,}111101} \times 10^{110}$$

IEEE – Standard (IEEE 754-1985)

Gleitkommazahlen (GK); (Floating point, FP)

$$z_n = \pm \mathbf{M} \times 10_2^{\,E}$$

$$= 1,\boxed{111101} \times 10^{110}$$

$$\mathbf{M} = 1,f$$

f ... fractional part

IEEE – Standard (IEEE 754-1985)

Gleitkommazahlen (GK); (Floating point, FP)

$$z_n = \pm M \times 10_2^{E}$$

$$= 1{,}111101 \times 10^{\boxed{110}}$$

$$M = 1{,}f$$

f … fractional part

Exponent - Anpassung: (hier mit 8 Bit für short real)

$e = E + bias$ ⟶ vorzeichenloser (biased) Exponent e

$e = E + 7FH$

$e = 6 + 7FH$

$e = 85H$

IEEE – Standard (IEEE 754-1985)

Gleitkommazahlen (GK); (Floating point, FP)

125 = 7DH = 0111 1101 = $1{,}111101 \times 10^{110}$

f = 111101

e = 85H = 1000 0101

s = 0 , da positive Zahl

Format: short real (Länge: 4 Byte)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
s	e								f																						

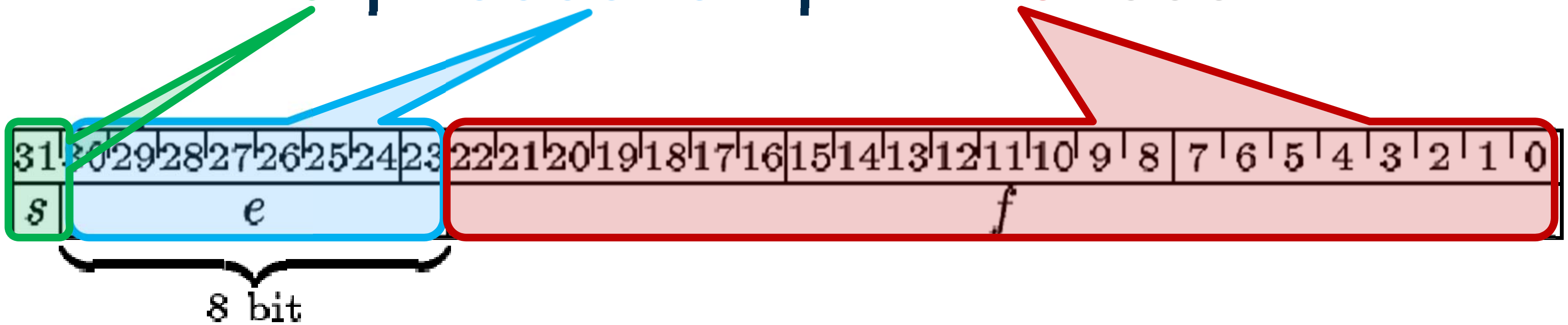
$\underbrace{e}_{8\ \text{bit}}$

IEEE – Standard (IEEE 754-1985)

Gleitkommazahlen (GK); (Floating point, FP)

$$125 = (\,1{,}111101 \times 10^{\,110}\,)_2$$

0 | 10000101 | 111101000 ...

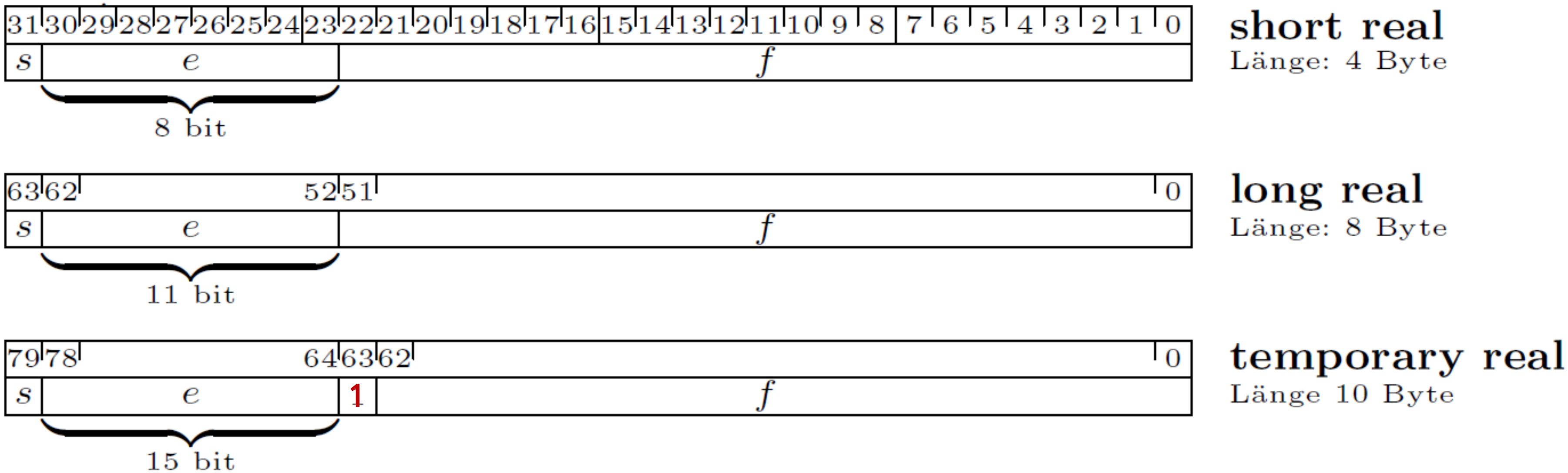


= 0100 0010 1111 1010 0000 0000 0000 0000

= 42 FA 00 00

IEEE – Standard (IEEE 754-1985)

Gleitkommazahlen (GK); (Floating point, FP)



Wertebereiche

Datentyp	Bits	Wertebereich	Genauigkeit
short real	32	$\pm 1,2 * 10^{-38} \ldots \pm 3,4 * 10^{38}$	
long real	64	$\pm 2,2 * 10^{-308} \ldots \pm 1,8 * 10^{308}$	
temporary real	80	$\pm 1,1 * 10^{-4932} \ldots \pm 1,2 * 10^{4932}$	

IEEE – Standard (IEEE 754-1985)

Beispiele

(a) $-335{,}125_{\text{Dez.}}$

⇩

short real:	C3 A7 90 00
long real:	C0 74 F2 00 00 00 00 00
temporary real:	C0 07 A7 90 00 00 00 00 00 00

(b) **short real: 42 B5 60 00** ⇨ $90{,}6875_{\text{Dez.}}$

Gleitkommazahlen (GK); (Floating point, FP)

Sonderformate

s	e	f	Wert
$1/0$	$=00$	$=0$	± 0
$1/0$	$=00$	$\neq 0$	denormalisiert
$1/0$	$=FF$	$=0$	$\pm\infty$
$1/0$	$=FF$	$\neq 0$	NaN (not a number)

IEEE – Standard (IEEE 754-1985)

Gleitkommazahlen (GK); (Floating point, FP)

Operation: Addition

$$125 + 20$$

$125 = 01111101$: $\quad 1{,}111101 \times 10^{110}$

$20 = 00010100$: $\quad + 1{,}01 \times 10^{100}$

Exponenten-Anpassung:

$$\begin{array}{rl} & 1{,}111101 \times 10^{110} \\ + & 0{,}0101 \quad \times 10^{110} \\ \hline & 10{,}010001 \times 10^{110} \end{array}$$

Neukodierung:

$$1{,}0010001 \times 10^{111} = 10010001 = 145$$

Gleitkommazahlen (GK); (Floating point, FP)

Applet zur rechnerinternen Zahlenverarbeitung

www.tu-ilmenau.de/iks -> Lehre - Rechnerorganisation

Gleitkommazahlen nach IEEE-754 | 2K-Zahlen

Dezimal → Binär | Binär → Dezimal

Bitte eine Gleitkommazahl eingeben:

[ ] Long real (64 Bit) [Umrechnen]

Berechne IEEE-754 Darstellung von

Umwandlung erfolgt in 6 Schritten:

1. Bestimmung von des Vorzeichenbits s:
2. Umwandlung des Betrags |z| in eine Dualzahl:
3. Normierung auf M * 2^E:
4. Abspaltung des Nachkommaanteils f:
5. Verschiebung des Exponenten in den positiven Bereich. Berechnung von e=E+bias:
6. Formatanpassung:

   s e f

Gleitkommazahlen (GK); (Floating point, FP)

Applet zur rechnerinternen Zahlenverarbeitung

https://www.h-schmidt.net/FloatConverter/IEEE754de.html

Tools & Thoughts

IEEE-754 Konverter für Fließkommazahlen

Translations: en

Diese Webseite dient zur Umrechnung zwischen der Dezimaldarstellung von Zahlen (z.B. "1.02") und den von modernen CPUs verwendeten IEEE 754-Fließkommazahlen. Aus Platzgründen werden nur Zahlen mit einfacher Genauigkeit (32 Bit) dargestellt. Diese Seite soll dem besseren Verständnis von Fließkommazahlen dienen.

IEEE 754 Converter (JavaScript), V0.21

	Sign	Exponent	Mantissa
Value:	-1	2^2	1.65625
Encoded as:	1	129	5505024
Binary:	☑	☑ ☐ ☐ ☐ ☐ ☐ ☐ ☑	☑ ☐ ☑ ☐ ☑ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐ ☐

Decimal representation	-6.625	+1
Value actually stored in float:	-6.625	
Error due to conversion:		-1
Binary Representation	11000000110101000000000000000000	
Hexadecimal Representation	0xc0d40000	

Datentypen höherer Sprachen - PASCAL

Entsprechend Maschinendatentypen

Pascal-Typ	abgebildet auf Maschinentyp	Länge
BYTE	ordinal	1 byte
WORD	ordinal	2 byte
INTEGER	integer	2 byte
SHORTINT	integer	1 byte
LONGINT	integer	4 byte
SINGLE	short real	4 byte
DOUBLE	long real	8 byte
EXTENDED	temporary real	10 byte
CHAR	ASCII	1 byte
^datentyp	pointer	4 byte

Datentypen höherer Sprachen - JAVA

Datentyp	Größe	Wrapper-Klasse	Wertebereich	Beschreibung
boolean	JVM-Spezifisch	java.lang.Boolean	true / false (0/1)	Boolescher Wahrheitswert
char	16 bit	java.lang.Character	Buchstaben, Zeichen	Unicode-Zeichen (UTF-16)
byte	8 bit	java.lang.Byte	-128 ... 127	Zweierkomplement-Wert
short	16 bit	java.lang.Short	-32.768 ... 32.767	Zweierkomplement-Wert
int	32 bit	java.lang.Integer	-2.147.483.648 ... 2.147.483.647	Zweierkomplement-Wert
long	64 bit	java.lang.Long	-9.223.372.036.854.775.808 ... 9.223.372.036.854.775.807	Zweierkomplement-Wert
float	32 bit	java.lang.Float	+/-1,4E-45 ... +/-3,4E+38	Gleitkommazahl (IEEE 754)
double	64 bit	java.lang.Double	+/-4,9E-324 ... +/-1,7E+308	Gleitkommazahl doppelter Genauigkeit (754)

Das war‘s für heute

Viel Spaß beim Wiederholen!

Bis nächsten Donnerstag 15.00 ...