



Staatliches Gymnasium  
"Karl Theodor Liebe "  
Gera

Fachbereich Informatik

---

# Informatik

## Iteration und Rekursion

F. Möckel

26. November 2020



**Inhaltsverzeichnis**

## Interne Darstellung von Gleitkommazahlen / UG/SST

### Definition

#### Gleitkommazahlen:

Eine Gleitkommazahl (auch Gleitpunktzahl oder Fließkommazahl; englisch floating point number) ist eine approximative Darstellung einer reellen Zahl.

Die Menge der Gleitkommazahlen ist eine Teilmenge der rationalen Zahlen. Zusammen mit den auf ihnen definierten Operationen (Gleitkommaarithmetik) bilden die Gleitkommazahlen eine endliche Arithmetik, die vor allem im Hinblick auf numerische Berechnungen mit Computern entwickelt wurde.<sup>1</sup>

#### Gleitpunkt-Typen in Java:

Typ	Wertebereich	Speicherplatz
float	$-3.4 \cdot 10^{-38}$ bis $+3.4 \cdot 10^{+38}$ (ergibt ungefähr 6 bis 7 signifikante Dezimalziffern Genauigkeit)	4 Bytes (32 Bits)
double	$-1.7 \cdot 10^{-308}$ bis $+1.7^{+308}$ (ergibt ungefähr 15 signifikante Dezimalziffern Genauigkeit)	8 Bytes (64 Bits)

### DIN - Vorschrift

Einzig international definierte Zahlendarstellung in Programmiersprachen:

Die Norm IEEE 754 (ANSI/IEEE Std 754-1985; IEC-60559:1989 – International version) definiert Standarddarstellungen für binäre Gleitkommazahlen in Computern und legt genaue Verfahren für die Durchführung mathematischer Operationen, insbesondere für Rundungen, fest. Der genaue Name der Norm ist englisch IEEE Standard for Binary Floating-Point Arithmetic for microprocessor systems (ANSI/IEEE Std 754-1985).

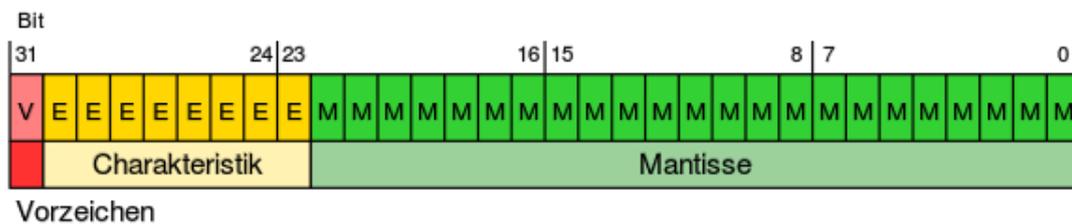
Die Darstellung einer Gleitkommazahl

$$x = s \cdot m \cdot b^e$$

besteht aus:

- Vorzeichen  $s$  (1 Bit)
- Mantisse  $m$  ( $p$  Bits)
- Basis  $b$  (bei normalisierten Gleitkommazahlen nach IEEE 754 ist  $b=2$ )
- Exponent  $e$  ( $r$  Bits)

<sup>1</sup>[https://de.wikipedia.org/wiki/IEEE\\_754](https://de.wikipedia.org/wiki/IEEE_754)



Das Vorzeichen  $s = (-1)^S$  wird in einem Bit  $S$  gespeichert, sodass  $S = 0$  positive Zahlen und  $S = 1$  negative Zahlen markiert.

Der Exponent  $e$  wird als nichtnegative Binärzahl  $E$  ( $E$  wird manchmal auch als Charakteristik oder biased exponent bezeichnet) gespeichert, indem man den festen Biaswert  $B$  addiert:  $E = e + B$ . Der Biaswert (engl: Verzerrung) berechnet sich durch  $2^{r-1} - 1$ . Der Biaswert  $B$  dient also dazu, dass negative Exponenten durch eine vorzeichenlose Zahl (die Charakteristik  $E$ ) gespeichert werden können, unter Verzicht auf alternative Kodierungen wie z. B. das Zweierkomplement.

Schließlich ist die Mantisse  $1 \leq m < 2$  ein Wert, der sich aus den  $p$  Mantissenbits mit dem Wert  $M$  als  $m = 1 + M/2^p$  berechnet. Einfacher ausgedrückt denkt man sich an das Mantissenbitmuster  $M$  links eine "1, anhängt:  $m = 1\{, \}M$ .

- $s = (-1)^S$
- $e = E - B$
- $m = 1, M = 1 + M/2^p$

Dieses Verfahren ist möglich, weil durch Normalisierung (s. u.) die Bedingung  $1 \leq m < 2$  für alle darstellbaren Zahlen immer eingehalten werden kann. Da dann die Mantisse immer links mit 1. beginnt, braucht dieses Bit nicht mehr gespeichert zu werden. Damit gewinnt man ein zusätzliches Bit Genauigkeit.

### Berechnung Dezimalzahl $\rightarrow$ IEEE754-Gleitkommazahl

Die Zahl  $18,4_{10}$  soll in eine Gleitkommazahl umgewandelt werden, dabei nutzen wir den (Java float) Single IEEE-Standard.

1. Umwandlung der Dezimalzahl in eine duale Festkommazahl ohne Vorzeichen

18,4

$$\begin{aligned}
 18/2 &= 9 \text{ Rest } 0 \text{ (Least - Significant - Bit)} \\
 9/2 &= 4 \text{ Rest } 1 \\
 4/2 &= 2 \text{ Rest } 0 \\
 2/2 &= 1 \text{ Rest } 0 \\
 1/2 &= 0 \text{ Rest } 1 \text{ (Most - Significant - Bit)} \\
 &= 10010
 \end{aligned}$$

Berechnung Nachkommastellen

$$\begin{aligned}
 0,4 \cdot 2 &= 0,8 \quad - 0 \text{ (Most - Significant - Bit)} \\
 0,8 \cdot 2 &= 1,6 \quad - 1 \\
 0,6 \cdot 2 &= 1,2 \quad - 1 \\
 0,2 \cdot 2 &= 0,4 \quad - 0 \\
 0,4 \cdot 2 &= 0,8 \quad - 0 \\
 0,8 \cdot 2 &= 1,6 \quad - 1 \text{ (Least - Significant - Bit)} \\
 &\dots \\
 &= 0,0110011001100110011\dots \\
 &\Downarrow \\
 18,4 &= 10010,011001100110011\dots
 \end{aligned}$$

2. Normalisieren und Bestimmen des Exponenten

Ermittlung Bias  $2^{r-1} - 1$

Bias =  $1x \text{ '0' } + (r-1)x \text{ '1'}$ , für  $r=8$ : 01111111

$$\begin{aligned}
 10010,011001100\dots &\cdot 2^{(01111111-01111111)} = && // 2 \text{ hoch } 127 - 127 = 0 \\
 1001,0011001100\dots &\cdot 2^{(10000000-01111111)} = && // 2 \text{ hoch } 128 - 127 = 1 \\
 100,10011001100\dots &\cdot 2^{(10000001-01111111)} = && // 2 \text{ hoch } 129 - 127 = 1 \\
 10,010011001100\dots &\cdot 2^{(10000010-01111111)} = && // 2 \text{ hoch } 130 - 127 = 1 \\
 1,0010011001100\dots &\cdot 2^{(10000011-01111111)} \Downarrow && // 2 \text{ hoch } 131 - 127 = 1
 \end{aligned}$$

oder

127 + 4 Kommastellen nach links = 131  $\rightarrow$  10000011

Mantisse: 1,0010011001100...

Exponent mit Bias: 10000011

## 3. Vorzeichen-Bit bestimmen

positiv  $\rightarrow$  0

## 4. Die Gleitkommazahl bilden

1 Bit Vorzeichen + 8 Bit Exponent + 23 Bit Mantisse

0 10000011 00100110011001100110011  $\rightarrow$  die Vorkomma-Eins wird als Hidden Bit weggelassen;  
da dort immer eine 1 steht, braucht man diese nicht zu speichern.

**Berechnung IEEE754-Gleitkommazahl  $\rightarrow$  Dezimalzahl**

Nun soll die Gleitkommazahl von oben wieder in eine Dezimalzahl zurück gewandelt werden, gegeben ist also folgende IEEE754-Zahl:

0 10000011 00100110011001100110011

## 1. Berechnung des Exponenten

Umwandeln des Exponenten in eine Dezimalzahl:

10000011  $\rightarrow$  131

Da dies aber der Biased Exponent ist, der zuvor um den Bias verschoben wurde, wird nun der Bias wieder abgezogen:

$131 - 127 = 4$  ist also der Exponent

## 2. Berechnung der Mantisse

Da es sich um eine normalisierte Zahl handelt, wissen wir, dass sie eine 1 vor dem Komma hat:

1,00100110011001100110011

Nun muss das Komma um 4 Stellen nach rechts verschoben werden:

10010,0110011001100110011

## 3. Umwandlung in eine Dezimalzahl

## Inhaltsverzeichnis

Vorkommastellen:

$$10010_2 = 18_{10}$$

Nachkommastellen:

Um den Wert der Nachkommazahl zu erhalten, muss man denselben Prozess durchführen wie bei ganzen Zahlen, nur in umgekehrter Richtung. Also von links nach rechts. Dabei muss der Exponent negativ sein und mit einer 1 beginnen.

In der Form

$$0 \cdot 2^{-1} + 1 \cdot 2^{-2} + 1 \cdot 2^{-3} + 0 \cdot 2^{-4} + 0 \cdot 2^{-5} + 1 \cdot 2^{-6} + 1 \cdot 2^{-7} + 0 \cdot 2^{-8} + 0 \cdot 2^{-9} + 1 \cdot 2^{-10} + 1 \cdot 2^{-11} + 0 \cdot 2^{-12} + 0 \cdot 2^{-13} + 1 \cdot 2^{-14} + 1 \cdot 2^{-15} + 0 \cdot 2^{-16} + 0 \cdot 2^{-17} + 1 \cdot 2^{-18} + 1 \cdot 2^{-19}$$

Weil die 0,4 im binären eine periodische Darstellung hat, kann sie nicht mehr genau umgewandelt werden

$$0,0110011001100110011_2 \approx 0.39999961853_{10}$$