

Zahlenbasen Umwandlung – in Theorie und Praxis

Alexander Hermann

25. April 2016

Inhaltsverzeichnis

| | | |
|----------|--|-----------|
| 1 | Einleitung | 9 |
| 1.1 | Zahlendarstellung | 9 |
| 1.1.1 | Beispiel dazu | 10 |
| 1.2 | Hier mögliche Zahlen | 10 |
| 2 | Umwandlungen | 11 |
| 2.1 | Umwandlung von Zahlen der Basis b in das Zahlensystem der Basis 10 | 11 |
| 2.1.1 | Allgemeine Formel zur Wandlung von Basis b zu Basis 10 | 11 |
| 2.1.2 | Berechnungsablauf | 11 |
| 2.2 | Umwandlung von Zahlen der Basis 10 in das Zahlensystem der Basis b | 13 |
| 2.2.1 | Allgemeine Formel zur Wandlung von Basis 10 zu Basis b | 13 |
| 2.2.2 | Berechnungsablauf | 14 |
| A | Beispiele | 17 |
| A.1 | Umwandlung ins Dezimalsystem | 17 |
| A.1.1 | Beispiel der Zahlenbasis $b_1 = 2$ | 17 |
| A.1.2 | Beispiel der Zahlenbasis $b_1 = 8$ | 18 |
| A.1.3 | Beispiel der Zahlenbasis $b_1 = 16$ | 19 |
| A.2 | Umrechnung vom Dezimalsystem in andere Zahlensysteme | 20 |
| A.2.1 | Beispiel der Zahlenbasis $b_2 = 2$ | 20 |
| A.2.2 | Beispiel der Zahlenbasis $b_2 = 3$ | 22 |
| A.2.3 | Beispiel der Zahlenbasis $b_2 = 8$ | 23 |
| B | Umsetzung in Programmiersprachen | 25 |
| B.1 | PHP-Codierung | 25 |

Listings

| | | |
|-----|---|----|
| B.1 | PHP Interface der Zahlenbasis | 25 |
| B.2 | PHP Implementierung der Zahlenbasis | 26 |

Abbildungsverzeichnis

| | | |
|-----|--|----|
| 2.1 | Struktogramm Umwandlung in das Dezimalsystem | 12 |
| 2.2 | Struktogramm Berechnung des Character-Werts | 13 |
| 2.3 | Struktogramm Berechnung des Zwischenergebnisses | 14 |
| 2.4 | Struktogramm Umwandlung vom Dezimalsystem | 15 |
| 2.5 | Struktogramm Umwandlung einer Zahl in einen Character-Wert | 16 |

Kapitel 1

Einleitung

Zahlen in verschiedenen Zahlenbasen werden im Wesentlichen für eine Vereinfachung zur menschlichen Kommunikation bzw. zur Umschreibung mit maschinellen Automatisierungen verwendet. Bei der Ausführung von Software auf reiner Hardware-Ebene läuft alles letztendlich rein binär¹ ab.

Da wir Menschen es gewohnt sind im Dezimalsystem² zu rechnen – was möglicherweise daran liegt, dass der Mensch zehn Finger hat – und auch dafür ausgebildet wurden, ist es im Allgemeinen einfacher, auf dieser Basis zu rechnen.

1.1 Zahlendarstellung

In vielen Programmiersprachen werden die Zahlensysteme **binär**, **octal**, **dezimal** und **hexadezimal** im Programmiercode zur Vereinfachung bzw. zur korrekten Interpretation durch den Compiler unterschiedlich eingegeben.

- **binär:** 0b101110
- **octal:** 0c576302
- **dezimal:** 964
- **hexadezimal:** 0xAFFE09

Da hier aber generell alle möglichen Zahlensysteme verwendet werden, bzw. die verallgemeinerte Form der Umrechnung erklärt werden soll, werden im

¹auf der Basis $b = 2$

²Basis $b = 10$

Folgenden Zahlen eines bestimmten Zahlensystems der Basis b wie folgt dargestellt:

$$bxv$$

Wobei b für die entsprechende Basis steht, x zur Markierung immer als **x** verwendet wird und v der Wert im entsprechenden Zahlensystem ist.

1.1.1 Beispiel dazu

Ähnlich wie oben:

- **binär:** 2x101110
- **octal:** 8x576302
- **dezimal:** 10x964
- **hexadezimal:** 16xAFFE09

Wäre das alles, wäre es wohl kaum nötig eine zusätzliche Darstellung zu verwenden. Aber an eher seltenen Zahlensystemen, ist eine generalisierte Darstellung dann doch vorteilhaft:

- **ternär:** 3x211201
- **quinär:** 5x402314
- **tridezimal** 13x5A9C0B3
- **oktovigesimal** 28xNOR70KRANK
- **hexatridezimal:** 36xGIRAFFE0Z6A

1.2 Hier mögliche Zahlen

Es gibt auch durchaus Umwandlungsmethoden, um \mathbb{R} eale Zahlen umzurechnen. Hier wird aber nur mit Natürlichen Zahlen gearbeitet.

Kapitel 2

Umwandlungen

2.1 Umwandlung von Zahlen der Basis b in das Zahlensystem der Basis 10

2.1.1 Allgemeine Formel zur Wandlung von Basis b zu Basis 10

Diese Formel ist für Zahlenbasen der Basis $b = 2$ bis Basis $b = 36$ ¹ mit den Ziffern 0 bis 9 und den Buchstaben A bis Z möglich. Die Formel setzt sich zusammen aus der Basis b , der Stellenposition² s und dem angezeigten Wert w . Wenn der Wert ein Buchstabe ist, ist der Wert gleich Buchstabenstelle bu_s im Alphabet $+9$ $w = bu_s + 9$ ansonsten der Zahlenwert $w = w$. Die Anzahl der maximalen Zeichen ist der Basiswert.³ Im „normalen“, dezimalen Zahlensystem von 0 bis 9 ist die 10 bereits *zweistellig*.

Merke: *das erste Zeichen ist immer 0!*

$$x_s = b^{s-1} * w \quad (2.1)$$

Die Ergebnisse der einzelnen Stellen werden summiert.

2.1.2 Berechnungsablauf

Der Berechnungsablauf kann wie in dem, in Abbildung 2.1 dargestellten Struktogramm dargestellt werden. Die meisten Programmiersprachen haben vordefinierte Funktionen zur Längenberechnung von `string`-Variablen; ebenso gibt es Funktionen um an bestimmten Stellen eines Strings einzelne Zeichen

¹Basis 36 bei ASCII; bei UTF-8 auch größer

²von rechts nach links

³Deswegen ist 2 auch die kleinstmögliche Zahlenbasis, weil bei nur einem Zeichen kein Unterschied mehr möglich ist.

2.2. UMWANDLUNG VON ZAHLEN DER BASIS 10 IN DAS ZAHLENSYSTEM DER BASIS B_{13}

| Berechnung des Character-Werts | |
|---|--|
| Parameter: | |
| c | {Eine char Variable, die den auszuwertenden Character angibt.} |
| lokale Variablen: | |
| result | {Eine int Variable, die das Endergebnis beinhaltet.} |
| z | {Eine int Variable als Zwischenwert.} |
| Lese den ASCII oder UTF-8 Wert des Parameters c aus und schreibe es in das Zwischenergebnis z . | |
| result = z + 9 – erste Buchstabenposition | |
| result zurückgeben | |

Abbildung 2.2: Struktogramm Berechnung des Character-Werts

abzurufen. Damit entfällt die genauere Beschreibung der Längenabfrage und der Stellenabfrage. Was hier noch fehlt, ist die Berechnung des Character-Werts, dies wird im Struktogramm in Abbildung 2.2 dargestellt, falls es sich **nicht** um eine Zahl handelt, so wie die Berechnung des Zwischenergebnisses, welches im Struktogramm in Abbildung 2.3 dargestellt wird.

2.2 Umwandlung von Zahlen der Basis 10 in das Zahlensystem der Basis b

Eine der Anleitungen fand ich im Web⁴. Die umzurechnende Zahl z wird durch die Basis b geteilt; der Quotient q wird zur erneuten Rechnung verwendet; der jeweilige Rest r wird mit 10 hoch dem Rechenschritt s multipliziert; der erste Rechenschritt ist $s = 0$. Es wird so häufig gerechnet, bis der Quotient 0 ist.

2.2.1 Allgemeine Formel zur Wandlung von Basis 10 zu Basis b

In dieser Formel wird der Quotient des vorherigen Rechenschritts als das Zwischenergebnis s_n bezeichnet, wobei n die Nummer des Rechenschritts ist. Die Zählung der Rechenschritte fängt mit 0 an. Also ist für die erste Stelle

⁴<http://www.arndt-bruenner.de> [Brü15]

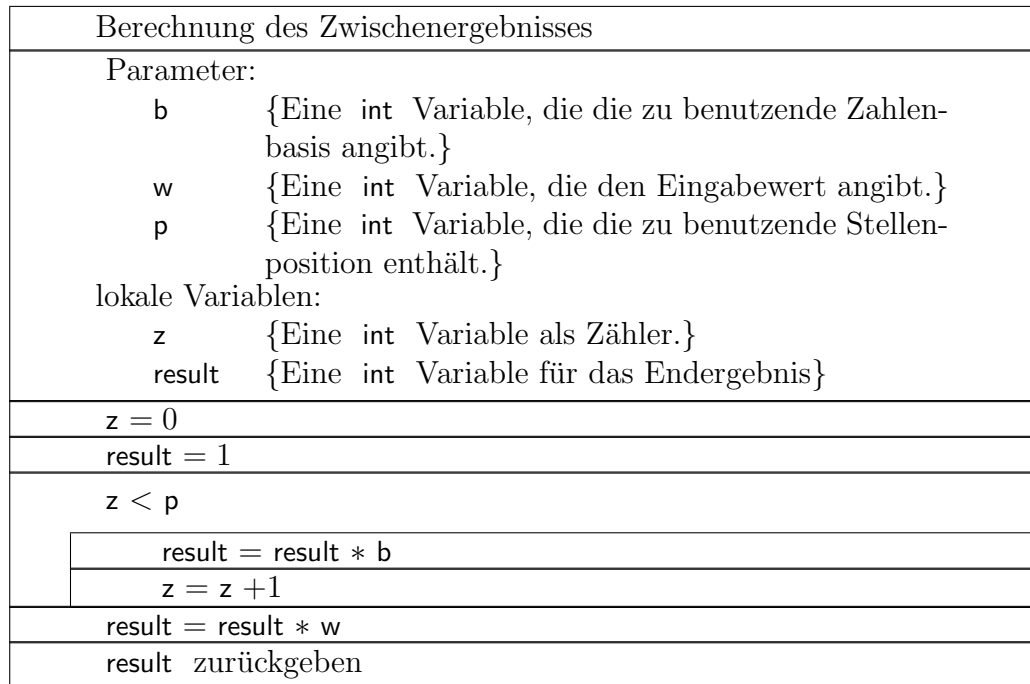


Abbildung 2.3: Struktogramm Berechnung des Zwischenergebnisses

s_0 die umzuwandelnde Zahl z der Basis b zu verwenden.

$$\frac{s_n}{b} = q_n; r_n \quad (2.2)$$

2.2.2 Berechnungsablauf

Nach dem ersten Rechenschritt (wenn der Quotient $q \neq 0$ ist), $n \geq 1$ gilt:

$$s_n = q_{n-1} \quad (2.3)$$

Das Gesamtergebnis g ergibt sich wie folgt, wenn die Zielbasis $b < 10$ ist:

$$g = r_0 * 10^0 + r_1 * 10^1 \cdots + r_n * 10^n \quad (2.4)$$

Wenn die Zielbasis $b > 10$ ist, müssen die einzelnen Zeichendarstellungen der Reste r_n rückwärts in eine Zeichenfolge zusammengesetzt werden. Das Ganze ist auch im Struktogramm in Abbildung 2.4 dargestellt. Die Umwandlung einer Zahl > 9 läuft ähnlich wie im Struktogramm in Abbildung 2.2; nur umgekehrt. Siehe dazu das Struktogramm in Abbildung 2.5

2.2. UMWANDLUNG VON ZAHLEN DER BASIS 10 IN DAS ZAHLENSYSTEM DER BASIS B15

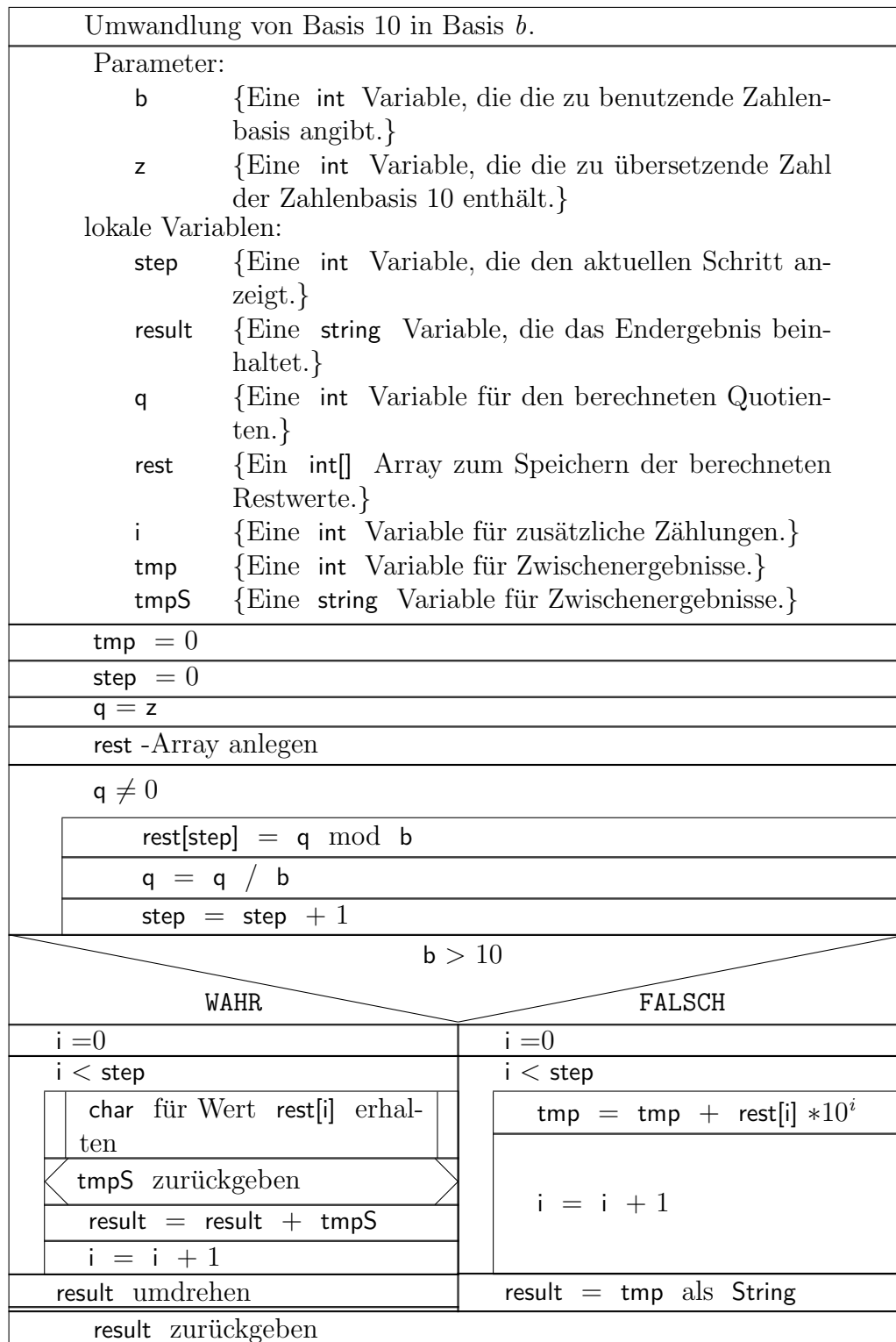


Abbildung 2.4: Struktogramm Umwandlung vom Dezimalsystem

| Berechnung eines Character-Werts | |
|--|---|
| Parameter: | |
| <code>x</code> | {Eine <code>int</code> Variable, die den auszuwertenden wert angibt.} |
| lokale Variablen: | |
| <code>result</code> | {Eine <code>char</code> Variable, die das Endergebnis beinhaltet.} |
| <code>z</code> | {Eine <code>int</code> Variable als Zwischenwert.} |
| Lese den ASCII oder UTF-8 Wert des Buchstabens A aus und schreibe es in das Zwischenergebnis <code>z</code> . | |
| <code>result = x - 9 + z</code> | |
| <code>result</code> zurückgeben | |

Abbildung 2.5: Struktogramm Umwandlung einer Zahl in einen Character-Wert

Anhang A

Beispiele

A.1 Umwandlung ins Dezimalsystem

Beispiele der Umrechnung von der Zahlenbasis $b_1 = x$ in die Zahlenbasis $b_2 = 10$.

A.1.1 Beispiel der Zahlenbasis $b_1 = 2$

Im Binärsystem gibt es die zwei Zeichen 0 und 1.

Beispiel 1

Die Binärzahl $2x100$ wird wie folgt nach Formel 2.1 umgerechnet: von rechts nach links:

1. An Stelle $s = 1$:

$$x_1 = 2^0 * 0 = 1 * 0 = 0$$

2. An Stelle $s = 2$:

$$x_2 = 2^1 * 0 = 2 * 0 = 0$$

3. An Stelle $s = 3$:

$$x_3 = 2^2 * 1 = 4 * 1 = 4$$

Die Summierung von x_1 bis x_3 ist:

$$0 + 0 + 4 = 4$$

Beispiel 2

Die Binärzahl $2x110101$ wird wie folgt nach Formel 2.1 umgerechnet: von rechts nach links:

1. An Stelle $s = 1$:

$$x_1 = 2^0 * 1 = 1 * 1 = 1$$

2. An Stelle $s = 2$:

$$x_2 = 2^1 * 0 = 2 * 0 = 0$$

3. An Stelle $s = 3$:

$$x_3 = 2^2 * 1 = 4 * 1 = 4$$

4. An Stelle $s = 4$:

$$x_4 = 2^3 * 0 = 8 * 0 = 0$$

5. An Stelle $s = 5$:

$$x_5 = 2^4 * 1 = 16 * 1 = 16$$

6. An Stelle $s = 6$:

$$x_6 = 2^5 * 1 = 32 * 1 = 32$$

.

Die Summierung von x_1 bis x_6 ist:

$$1 + 0 + 4 + 0 + 16 + 32 = 53$$

A.1.2 Beispiel der Zahlenbasis $b_1 = 8$

Im Oktalsystem gibt es acht Zeichen von 0 bis 7.

Beispiel 1

Die Oktalzahl $8x70$ wird wie folgt nach Formel 2.1 umgerechnet: von rechts nach links:

1. An Stelle $s = 1$:

$$x_1 = 8^0 * 0 = 1 * 0 = 0$$

2. An Stelle $s = 2$:

$$x_2 = 8^1 * 7 = 8 * 7 = 56$$

Die Summierung von x_1 bis x_2 ist:

$$0 + 56 = 56$$

A.1.3 Beispiel der Zahlenbasis $b_1 = 16$

Im Hexadezimalsystem gibt es sechzehn Zeichen von 0 bis F .

Beispiel 1

Die Hexadezimalzahl $16xD4$ wird wie folgt nach Formel 2.1 von rechts nach links umgerechnet:

1. An Stelle $s = 1$:

$$x_1 = 16^0 * 4 = 1 * 4 = 4$$

2. An Stelle $s = 2$:

$$x_2 = 16^1 * (4 + 9) = 16 * 13 = 208$$

Die Summierung von x_1 bis x_2 ist:

$$4 + 208 = 212$$

Beispiel 2

Die Hexadezimalzahl $16xAFFE$ wird wie folgt nach Formel 2.1 von rechts nach links umgerechnet:

1. An Stelle $s = 1$:

$$x_1 = 16^0 * (5 + 9) = 1 * 14 = 14$$

2. An Stelle $s = 2$:

$$x_2 = 16^1 * (6 + 9) = 16 * 15 = 240$$

3. An Stelle $s = 3$:

$$x_3 = 16^2 * (6 + 9) = 256 * 15 = 3840$$

4. An Stelle $s = 4$:

$$x_4 = 16^3 * (1 + 9) = 4096 * 10 = 40960$$

Die Summierung von x_1 bis x_4 ist:

$$14 + 240 + 3840 + 40960 = 45054$$

A.2 Umrechnung vom Dezimalsystem in andere Zahlensysteme

Beispiele der Umrechnung von der Zahlenbasis $b_1 = 10$ in die Zahlenbasis $b_2 = x$.

A.2.1 Beispiel der Zahlenbasis $b_2 = 2$

Die Zahlenbasis nennt sich **Binär**.

Beispiel 1

Die Dezimalzahl $z = 13$ wird wie folgt nach Formel 2.2 umgerechnet:

1. An Stelle 1: $n = 0$:

$$\frac{s_0 = z}{2} = \frac{13}{2} = q_0 = 6; r_0 = 1$$

2. An Stelle 2: $n = 1$:

$$\frac{s_1 = q_0}{2} = \frac{6}{2} = q_1 = 3; r_1 = 0$$

3. An Stelle 3: $n = 2$:

$$\frac{s_2 = q_1}{2} = \frac{3}{2} = q_2 = 1; r_2 = 1$$

4. An Stelle 4: $n = 3$:

$$\frac{s_3 = q_2}{2} = \frac{1}{2} = q_3 = 0; r_3 = 1$$

5. Gesamtergebnis g der Basis $b = 2$:

$$g = r_0 * 10^0 + r_1 * 10^1 + r_2 * 10^2 + r_3 * 10^3$$

$$g = 1 * 1 + 0 * 10 + 1 * 100 + 1 * 1000 = 1101$$

$$10x13 = 2x1101$$

Beispiel 2

Die Dezimalzahl $z = 141$ wird wie folgt nach Formel 2.2 umgerechnet:

1. An Stelle 1: $n = 0$:

$$\frac{s_0 = z}{2} = \frac{141}{2} = q_0 = 70; r_0 = 1$$

2. An Stelle 2: $n = 1$:

$$\frac{s_1 = q_0}{2} = \frac{70}{2} = q_1 = 35; r_1 = 0$$

3. An Stelle 3: $n = 2$:

$$\frac{s_2 = q_1}{2} = \frac{35}{2} = q_2 = 17; r_2 = 1$$

4. An Stelle 4: $n = 3$:

$$\frac{s_3 = q_2}{2} = \frac{17}{2} = q_3 = 8; r_3 = 1$$

5. An Stelle 5: $n = 4$:

$$\frac{s_4 = q_3}{2} = \frac{8}{2} = q_4 = 4; r_4 = 0$$

6. An Stelle 6: $n = 5$:

$$\frac{s_5 = q_4}{2} = \frac{4}{2} = q_5 = 2; r_5 = 0$$

7. An Stelle 7: $n = 6$:

$$\frac{s_6 = q_5}{2} = \frac{2}{2} = q_6 = 1; r_5 = 0$$

8. An Stelle 8: $n = 7$:

$$\frac{s_6 = q_6}{2} = \frac{1}{2} = q_6 = 0; r_6 = 1$$

9. Gesamtergebnis g der Basis $b = 2$:

$$\begin{aligned} g &= r_0 * 10^0 + r_1 * 10^1 + r_2 * 10^2 + r_3 * 10^3 + r_4 * 10^4 \\ &\quad + r_5 * 10^5 + r_6 * 10^6 \\ g &= 1 * 1 + 0 * 10 + 1 * 100 + 1 * 1000 + 0 * 10000 \\ &\quad + 0 * 100000 + 0 * 1000000 + 1 * 10000000 \\ &= 10001101 \end{aligned}$$

$$10x141 = 2x10001101$$

A.2.2 Beispiel der Zahlenbasis $b_2 = 3$

Die Zahlenbasis nennt sich **Ternär**.

Beispiel 1

Die Dezimalzahl $z = 13$ wird wie folgt nach Formel 2.2 umgerechnet:

1. An Stelle 1: $n = 0$:

$$\frac{s_0 = z}{3} = \frac{13}{3} = q_0 = 4; r_0 = 1$$

2. An Stelle 2: $n = 1$:

$$\frac{s_1 = q_0}{3} = \frac{4}{3} = q_1 = 1; r_1 = 1$$

3. An Stelle 3: $n = 2$:

$$\frac{s_2 = q_1}{3} = \frac{1}{3} = q_2 = 0; r_2 = 1$$

4. Gesamtergebnis g der Basis $b = 3$:

$$g = r_0 * 10^0 + r_1 * 10^1 + r_2 * 10^2$$

$$g = 1 * 1 + 1 * 10 + 1 * 100 = 111$$

$$10x13 = 3x111$$

Beispiel 2

Die Dezimalzahl $z = 141$ wird wie folgt nach Formel 2.2 umgerechnet:

1. An Stelle 1: $n = 0$:

$$\frac{s_0 = z}{3} = \frac{141}{3} = q_0 = 47; r_0 = 0$$

2. An Stelle 2: $n = 1$:

$$\frac{s_1 = q_0}{3} = \frac{47}{3} = q_1 = 15; r_1 = 2$$

3. An Stelle 3: $n = 2$:

$$\frac{s_2 = q_1}{3} = \frac{15}{3} = q_2 = 5; r_2 = 0$$

4. An Stelle 4: $n = 3$:

$$\frac{s_3 = q_2}{3} = \frac{5}{3} = q_3 = 1; r_3 = 2$$

5. An Stelle 5: $n = 4$:

$$\frac{s_4 = q_3}{3} = \frac{1}{3} = q_4 = 0; r_4 = 1$$

6. Gesamtergebnis g der Basis $b = 3$:

$$g = r_0 * 10^0 + r_1 * 10^1 + r_2 * 10^2 + r_3 * 10^3 + r_4 * 10^4$$

$$\begin{aligned} g &= 0 * 1 + 2 * 10 + 0 * 100 + 2 * 1000 + 1 * 10000 \\ &= 12020 \end{aligned}$$

$$10x141 = 3x12020$$

A.2.3 Beispiel der Zahlenbasis $b_2 = 8$

Die Zahlenbasis nennt sich **Oktal**.

Beispiel 1

Die Dezimalzahl $z = 13$ wird wie folgt nach Formel 2.2 umgerechnet:

1. An Stelle 1: $n = 0$:

$$\frac{s_0 = z}{8} = \frac{13}{8} = q_0 = 1; r_0 = 5$$

2. An Stelle 2: $n = 1$:

$$\frac{s_1 = q_0}{8} = \frac{1}{8} = q_1 = 0; r_1 = 1$$

3. Gesamtergebnis g der Basis $b = 8$:

$$g = r_0 * 10^0 + r_1 * 10^1$$

$$g = 5 * 1 + 1 * 10 = 15$$

$$10x13 = 8x15$$

Beispiel 2

Die Dezimalzahl $z = 141$ wird wie folgt nach Formel 2.2 umgerechnet:

1. An Stelle 1: $n = 0$:

$$\frac{s_0 = z}{8} = \frac{141}{8} = q_0 = 17; r_0 = 5$$

2. An Stelle 2: $n = 1$:

$$\frac{s_1 = q_0}{8} = \frac{17}{8} = q_1 = 2; r_1 = 1$$

3. An Stelle 3: $n = 2$:

$$\frac{s_2 = q_1}{8} = \frac{2}{8} = q_2 = 0; r_2 = 2$$

4. Gesamtergebnis g der Basis $b = 8$:

$$g = r_0 * 10^0 + r_1 * 10^1 + r_2 * 10^2$$

$$\begin{aligned} g &= 5 * 1 + 1 * 10 + 2 * 100 \\ &= 215 \end{aligned}$$

$$10x141 = 8x215$$

Anhang B

Umsetzung in Programmiersprachen

B.1 PHP-Codierung

Angesehen werden kann die Umsetzung in PHP 5.x unter
<http://demo.hermann-bsd.de/zahlensysteme/>¹

Zuerst als abstraktes Interface fuer die Definition von Zahlenbasen

Listing B.1: PHP Interface der Zahlenbasis

```
1 namespace ahbsd\Zahlensysteme
2 {
3     /**
4      * Interface fuer grundlegende Funktionen, der Basis eines
5      * Zahlensystems.
6      *
7      * @author A. Hermann
8      * @copy Copyright &copy; 2016
9      * Alexander Hermann - Beratung, Software, Design
10     * Zahlensysteme
11     *
12     * @version 1.0
13     */
14     interface IBase
15     {
16         /**
17          * Gibt die Bezeichnung zurueck.
18          * @return string
19          */
20         function GetName();
21
22         /**
23          * Gibt das Zahlensystem als Integer zurueck.
24          * @return int Zahlensystem-Basis
```

¹Einiges funktioniert da noch nicht...

```

25         */
26         function GetSystem();

28         /**
29          * Gibt das hoechstmoegliche Zeichen zurueck.
30          * @return char hoechstmoegliches Zeichen
31          */
32         function GetMaxSign();

34         /**
35          * Gibt das Zeichen der Basis fuer den Wert $x zurueck.
36          * @param int $x Wert x
37          * @return char Zeichen der Basis fuer den Wert x
38          */
39         function GetSign($x);
40     }
41 }

```

Implementierung des Interfaces:

Listing B.2: PHP Implementierung der Zahlenbasis

```

1 namespace ahbsd\Zahlensysteme
2 {
3     /**
4      * Basis eines Zahlensystems.
5      *
6      * @author A. Hermann
7      * @copy Copyright &copy; 2016 Alexander Hermann - Beratung, Software,
8      * Design
9      * Zahlensysteme
10     * @version 1.0
11     */
12     /**
13      class Base implements IBase
14      {
15          /**
16           * Konstante, die die ASCII (und UTF-8) Position von 'A' speichert.
17           * @var int
18           */
19           const A_POS_UTF8 = 65;

21         /**
22          * System-Name
23          * @var string
24          */
25         private $systemName;

27         /**
28          * System als Integer-Zahl. Maximale Anzahl an Zeichen.
29          *
30          * @var int
31          */
32         private $systemInt;

34         /**
35          * Hoechstmoegliches Zeichen.
36          *
37          * @var char
38          */

```

```

39     private $maxSign;

41     /**
42      * Konstruktor
43      *
44      * @param int $system Zahlensystem-Basis (Maximale Anzahl an Zeichen)
45      * @param string $name (Optional) Bezeichnung des Zahlensystems
46      */
47     public function __construct($system, $name="")
48     {
49         $this->systemInt=intval($system);
50         $this->systemName=$name;

52         if ($name=="")
53         {
54             $this->systemName = sprintf("Basis %i\${s}", intval($system));
55         }
56         $tmp = A_POS_UTF8 - 11 + intval($system);

58         if ($system <= 10)
59         {
60             $this->maxSign = $system - 1;
61         }
62         else
63         {
64             $this->maxSign = mb_convert_encoding('&#' . $tmp . ';' , 'UTF-8' ,
65                 'HTML-ENTITIES');
66         }
67     }

68     /**
69      * (non-PHPdoc)
70      * @see \ahbsd\Zahlensysteme\IBase::GetSign()
71      */
72     public function GetSign($x)
73     {
74         $tmp = 65-11+intval($x+1);
75         $result = $x;

77         if(intval($x) >= 10 || intval($x) < 0)
78         {
79             $result = mb_convert_encoding('&#' . $tmp . ';' , 'UTF-8' , 'HTML-
80                 ENTITIES');
81         }

82         return $result;
83     }

84     /**
85      * (non-PHPdoc)
86      * @see \ahbsd\Zahlensysteme\IBase::GetName()
87      */
88     public function GetName()
89     {
90         return $this->systemName;
91     }

92     /**
93      * (non-PHPdoc)
94      * @see \ahbsd\Zahlensysteme\IBase::GetSystem()
95      */
96     public function GetSystem()

```

```

99     {
100         return $this->systemInt;
101     }

102
103     /**
104      * (non-PHPdoc)
105      * @see \ahbsd\Zahlensysteme\IBase::GetMaxSign()
106      */
107     public function GetMaxSign()
108     {
109         return $this->maxSign;
110     }

111
112     /**
113      * Statische Funktion zur Umwandlung einer Zahl aus dem Dezimalsystem
114      * in eine
115      * Zahl des Zahlensystems bX.
116      *
117      * @param int $b10 Umzuwandelnde Zahl aus dem Dezimalsystem.
118      * @param int $bX Zahlensystem in das b10 umgewandelt werden soll.
119      * @param bool $rechenweg (Optional) Gibt an, ob der Rechenweg
120      * angezeigt werden soll oder nicht; ohne Angabe standardmaessig
121      * FALSE.
122      * @return string Ergebnis in Basis bX
123      */
124     public static function Base10toBaseX($b10, $bX, $rechenweg=false)
125     {
126         $targetBase = new Base(intval($bX));
127         $result = array();
128         $restArray = array();
129         $rOut = "";

130
131         $quotient = intval($b10);
132         $rest = 0;
133         $cnt = 0;

134
135         if ($rechenweg)
136         {
137             echo "\n<!-- start Rechenweg --><pre>\n";
138             echo "Rechenweg:\n";
139         }

140
141         while (intval($quotient) != 0)
142         {
143             $rest = $quotient % $bX;
144             if ($rechenweg) echo "$quotient : $bX = " . intval($quotient /
145                 $bX) . " Rest $rest [" . $targetBase->GetSign($rest) . "]\n";
146             $restArray[] = $rest;
147             $quotient = intval($quotient / $bX);
148         }
149         if ($rechenweg) echo "-----\n";
150         $cnt = count($restArray);

151
152         for($i=0;$i < $cnt; $i++)
153         {
154             $result[$cnt - ($i + 1)] = $targetBase->GetSign($restArray[$i]);
155         }

156
157         for($i=0; $i < $cnt; $i++)
158         {
159             $rOut .= $result[$i];
160         }

```

```

158     if ($rechenweg)
159     {
160         printf("Das Ergebnis der Umwandlung von %1\$s der Basis 10 in die
161             %2\$s ist '%3\$s'\n", $b10, $targetBase->GetName(), $rOut);
162         echo "</pre><!-- ende Rechenweg -->\n\n";
163     }
164     return $rOut;
165 }

166 /**
167  *
168  * @param string $bxVal Der Wert der Basis $bX
169  * @param int $bX Die Quell Basis.
170  * @param bool $rechenweg Gibt an, ob der Rechenweg ausgegeben werden
171  *     soll,
172  *     oder nicht.
173  * @return int Der Wert $bxVal umgerechnet in Basis 10.
174  */
175 public static function BaseXtoBase10($bxVal, $bX, $rechenweg=false)
176 {
177     $sourceBase = new Base(intval($bX));
178     $step = strlen($bxVal) - 1;
179     $result = 0;
180     $z = 0;
181     $curCarCorrect = false;
182     $intW = 0;

183     if ($rechenweg)
184     {
185         echo "\n<!-- start Rechenweg --><pre>\n";
186         echo "Rechenweg:\n";
187     }

188     while ($step >= 0) {
189         $charW = $bxVal[$step];
190         $tmpIntVal = intval($charW, 10);

191         $curCarCorrect = (',' . $tmpIntVal . ','==$charW);

192         if ($rechenweg) printf("%3\$s) Zeichen '%1\$s' an Stelle %2\$s "
193             , $charW, $step, $z+1);

194         if ($curCarCorrect) {
195             $intW = $tmpIntVal;
196         }
197         else {
198             // CharWert Umrechnung
199             $tmp2 = ord($charW);
200             $intW = intval($tmp2) - 65 + 10; // A_POS_UTF8;

201             if ($rechenweg) printf("= (int) %1\$s", $intW);
202         }

203         if ($rechenweg) echo "\n";

204         $tmpR = 1;

205         for ($i = 0; $i < $z; $i++) {
206             $tmpR = $tmpR * $bX;
207         }

```

```
217         if ($rechenweg) printf("%1\$s^%2\$s=%3\$s\n%3\$s * %4\$s = ",
                                $bX, $z, $tmpR, $intW);

219         $tmpR = $tmpR * $intW;

221         if ($rechenweg) echo $tmpR . "\n\n";

223         $step--;
224         $result += $tmpR;
225         $z++;
226     }

228     if ($rechenweg)
229     {
230         printf("Das Ergebnis der Umwandlung von '%1\$s' der %2\$s in die
                Basis 10 ist %3\$s\n", $bxVal, $sourceBase->GetName(),
                $result);
231         echo "</pre><!-- ende Rechenweg -->\n\n";
232     }

234     return $result;
235 }
236 }
237 }
```

Literaturverzeichnis

[Brü15] BRÜNNER, ARNDT: *Umrechnung von Zahlensystemen*, 12 2015.

Index

binär, 9, 10
Binärsystem, 9

dezimal, 9–11

Hardware, 9
hexadezimal, 9, 10
hexatridezimal, 10

octal, 9, 10
oktovigesimal, 10

quinär, 10

Software, 9

ternär, 10
tridezimal, 10

Zahlen, 9, 10
 -basen, 9, 11
 -system, 10, 11
 -systeme, 9
 -wert, 11
Ziffern, 11