

3. TUTORIUM

Einführung in die Strukturierte Programmierung



Zahlensysteme

Zahlensysteme

- |
- ||
- |||
- ||||
- |||||
- |||||
- |||||
- ...

Zahlensysteme



- Idee: Stellenwertsystem
 - ▣ Macht der Kombination

Basiswechsel



- Ziffer · Stellenwert aufsummieren
- Division durch neue Basis
 - Reste (Modulo) ergeben die Ziffern in der neuen Basis
- Spezialfall: Eine Basis Potenz der anderen
 - In Gruppen einteilen (z.B.: Hexadezimal ↔ Binär)
- Ansonsten: “Zwischenschritt” über Dezimalsystem

Basiswechsel

- Umrechnung zwischen den Zahlensystemen anhand von handverlesenen Beispielen
 - ▣ 0xBEEF → [DEZIMAL]
 - ▣ Tiefer Denker:
101010b → [DEZIMAL]
 - ▣ Frag doch den Inder:
0650 → [DEZIMAL]

Zahlenrepräsentation

- What else:
110010101111110b → [HEXADEZIMAL]
- I'm loving it:
195948557 → [HEXADEZIMAL]
- The matrix has you:
1984 → [BINÄR]
- Always:
0xC0CA (C01A) → [BINÄR]
- Der Endgegner:
007 + 0xBÉDEAD → [DEZIMAL]
- Referenz: <http://www.arndt-bruenner.de/mathe/scripts/Zahlensysteme.htm>



Bit-Operationen

Bit-Operationen

- \sim bitweises NOT
- $|$ bitweises OR
- $\&$ bitweises AND
- \wedge bitweises XOR

Logische- vs. Bit-Operatoren

- `!2` vs. `~2` ?
- `(1 & 2)` vs. `(1 && 2)` ?

Bit-Operationen

- Einzelne Bits auslesen:

```
    01001011 Information
AND 00001000 Bitmaske (1 Bit gesetzt)
-----
= 00001000 Ergebnis (TRUE, also gesetzt)
```

```
    01000011 Information
AND 00001000 Bitmaske (1 Bit gesetzt)
-----
= 00000000 Ergebnis (FALSE, also ungesetzt)
```

Bit-Operationen

- Einzelnes Bit setzen:

```
      01000011 Information
OR   00001000 Bitmaske
-----
      = 01001011 Ergebnis (FALSE, also ungesetzt)
```

- Einzelnes Bit löschen:

```
NOT  00001000 Bitmaske
-----
      = 11110111 invertierte Bitmaske

      01001011 Information
AND  11110111 invertierte Bitmaske
-----
      = 01000011 Ergebnis
```

Bit-Operationen

- Bits toggeln (umschalten):

```
      01000111 Information
XOR  00001100 Bitmaske
-----
      = 01001011 Ergebnis
```

- Bitmasken zusammenfassen

```
      00000100 Bitmaske I
OR   00001000 Bitmaske II
-----
      = 00001100 Ergebnis
```

Bit-Operationen

- Bitshifts

```
00000100 Wert (4)
<< (left)  -----
00001000 Ergebnis (4 * 2 = 8)
```

```
00000100 Wert (4)
>> (right) -----
00000010 Ergebnis (4 / 2 = 2)
```

- Achtung: Beim Verschieben der Bits über die Grenzen des Datentypes gehen diese Informationen verloren

Bis zum nächsten Mal