



# **Programmierung 1**

## **Übungsskript**

Falk Jonatan Strube

Übung von Prof. Dr.-Ing. Beck

11. November 2015

## Inhaltsverzeichnis

|          |                              |          |
|----------|------------------------------|----------|
| <b>1</b> | <b>Eingebaute Datentypen</b> | <b>1</b> |
| 1.1      | Zahlentypen . . . . .        | 1        |
| <b>2</b> | <b>Ausdrücke</b>             | <b>2</b> |

# 1 Eingebaute Datentypen

## 1.1 Zahlentypen

Zahl: 1 0 8  
8 · 10<sup>0</sup> 0 · 10<sup>1</sup> 1 · 10<sup>2</sup>

⇒ 10er-System (Decimal)

Zahl:  $0110 \underset{\leftarrow usw.}{1} \underset{1 \cdot 2^2}{0} \underset{0 \cdot 2^1}{0} \underset{0 \cdot 2^0}{0} = 0 + 0 + 4 + 8 + 0 + 32 + 64 + 0 = 108$

⇒ 2er-System (Binär)

Zahl:  $001|101|100 = 108$   
 $\quad \quad \quad 1 \quad \quad 5 \quad \quad 4$

⇒ 8er-System (Octal)

Zahl:  $0110|1100 = 108$

⇒ 16er-System (Hexa)

Unterschied:  $108_{/10}$ ,  $01101100_{/2}$ ,  $154_{/8}$  (in C gekennzeichnet durch  $0154 \rightarrow$  Octalzahl) und  $6C_{/16}$  (in C gekennzeichnet durch  $0x6C$ )

## Veranschaulichung

|   |              |
|---|--------------|
| $108 : 2 = 54$                                    | $R\emptyset$ |
| $65 : 2 = 27$                                     | $R\emptyset$ |
| $27 : 2 = 13$                                     | $R1$         |
| $13 : 2 = 6$                                      | $R1$         |
| $6 : 2 = 3$                                       | $R\emptyset$ |
| $3 : 2 = 1$                                       | $R1$         |
| $1 : 2 = 0$                                       | $R1$         |
| $\Rightarrow 1101100$ von unten nach oben gelesen |              |

$$\begin{array}{rcl} 108 : 8 & = & 13 & R4 \\ 13 : 8 & = & 1 & R5 \\ 1 : 8 & = & 0 & R1 \\ & \Rightarrow & 154 & \end{array}$$
$$\begin{array}{ll} 108 : 16 = 6 & R12 = RC \\ 6 : 16 = 0 & R6 \\ \Rightarrow 6C & \end{array}$$

Beispielzahl 0x12AB

Speicherblock:

|   |   |   |   |               |
|---|---|---|---|---------------|
| 1 | 2 | A | B | big-endian    |
| B | A | 1 | 2 |               |
| A | B | 1 | 2 | little-endian |

Letzte Version ist die, die heutzutage meistens (Intel) verwendet wird: Das niederwertigste Byte liegt auf der niedrigsten Adresse.

**2er Komplement** positive Zahl: 0110 1100

Negation: 1001 0011

+1

Komplement: 1001 0100 = -108 = 0x94

|   |                |                |                |                |                |   |   |   |      |
|---|----------------|----------------|----------------|----------------|----------------|---|---|---|------|
|   | 0              | 1              | 1              | 0              | 0              | 1 | 1 | 0 | +108 |
| 1 | 1 <sub>1</sub> | 0 <sub>1</sub> | 0 <sub>1</sub> | 1 <sub>1</sub> | 0 <sub>1</sub> | 1 | 0 | 0 | -108 |
| 1 | 0              | 0              | 0              | 0              | 0              | 0 | 0 | 0 |      |

## 2 Ausdrücke

Simple Sort

```
#include <stdio.h>

int data[] = {7,3,9,2,5};

int main(){
    int ige, iro; // entsprechende Pfeile unter den Zahlen auf Papier
    for (irt=0; irt<(5-1); irt++){
        for (ige = irt+1, ige<5, ige++){
            if (data[ige] < data[irt]){
                int tmp = data [ige];
                data[ige] = data[irt];

                data[irt] = tmp;
                // tauschen alternativ (ohne Zwischenspeichern): (^= ist XOR)
                // data[irt]^=data[ige];
                // data[ige]^=data[irt];
                // data[irt]^=data[ige];
            }
        }
    }
    for (irt=0; irt<5; irt++){
        printf("%d ", data[irt]);
    }
    printf("\n");
    return 0;
}
```

Alphabetische Sortierung

```
#include <stdio.h>

#define N 10

//[10]: länge der Zeichenkette
char data[][10] = {"Max", "Huckebein", "Bolte", "Lempel", "Maecke",
    "Helene", "Antonius", "Schlich", "Moritz", "Boeck"};

int main(){
    int ige, iro, ibl; // entsprechende Pfeile unter den Zahlen auf Papier
```

```
for (irt=0; irt<(N-1); irt+=1){
    for (ige = irt+1, ige<N, ige++){
        for (ibl = 0; data[irt][ibl] == data[ige][ibl] &&
            data[irt][ibl]!=0; ibl++){
            ;
        }
        if(data[irt][ibl] > data[ige][ibl]){
            char tmp;
            for (ibl = 0; ibl<N ; ibl++){
                tmp = data[irt][ibl];
                data[irt][ibl] = data[ige][ibl];
                data[ige][ibl] = tmp;
            }
        }
    }
}
for (irt=0; irt<5; irt+=1){
    printf("%d ", data[irt]);
}
printf("\n");
return 0;
}
```