

Fachhochschule Bingen

Programmieren

Datentypen / Operatoren / Anweisungen

Prof. Dr. Maximilian Mengel,
Professur Programmiermethodik,
Grundlagen der Informatik und Multimedia
Gebäude 1, Raum 212
Tel.: 06721-409 152
E-Mail: mengel@fh-bingen.de

Datentypen

- In C unterscheidet man Daten gemäß ihres Datentyps:
 - **char**: ein einzelnes Zeichen
 - **int**: ein ganzer Zahlenwert
 - **float**: eine Gleitpunktzahl
 - **double**: eine Gleitpunktzahl mit doppelter Stellenzahl wie float
- Varianten von **int**: **short int** und **long int**
- Weitere Varianten: **signed** und **unsigned** (für **char** und **int**)

10.10.2003

2

Variablen

- Eine Variable ist ein Datencontainer, der über einen Namen angesprochen wird
 - Jede Variable besitzt einen Datentyp
 - Die Namen bestehen aus Buchstaben (inklusive `_`) & Ziffern
 - Das erste Zeichen muß ein Buchstabe sein
 - Groß- und Kleinbuchstaben werden unterschieden
- Variablen müssen vereinbart werden
 - `int meine_Ganze_Zahl;`
- Variablen können initialisiert werden
 - `char mein_Buchstabe = 'M';`
- Variablen die nicht geändert werden dürfen, werden mit dem Schlüsselwort **const** vereinbart

10.10.2003

3

Konstanten (Literele)

- Auch Konstanten besitzen einen Datentyp. Dieser wird meistens implizit gesetzt:
 - 123 Datentyp **int**
 - 0.5 Datentyp **double**
 - 1e-3 Datentyp **double**
- Die explizite Angabe des Datentyps geschieht über einen angefügten Buchstaben:
 - 123L Datentyp **long int**
 - 0.5F Datentyp **float**
 - 123UL Datentyp **unsigned long int**

10.10.2003

4

Operatoren

■ C unterscheidet verschiedene Operatoren*

- Arithmetische Operatoren
- Logische Verknüpfungen
- Inkrement und Dekrement
- Bit-Manipulationen
- Zuweisungen
- Bedingte Bewertung
- Komma

■ In C hat jede Operation ein Ergebnis !

*Die genannte Liste ist nicht vollständig;beispielsweise folgende Operatoren fehlen: *,.,-,>,&

Arithmetische Operatoren

- + Die Addition von zwei Zahlen
- - Die Subtraktion von zwei Zahlen
- * Die Multiplikation von zwei Zahlen
- / Die Division von zwei Zahlen.
WICHTIG: Beide vom Typ int → ganzzahlige Division wird durchgeführt!
- % Die Modulo-Funktion liefert den ganzzahligen Rest einer Division; die Operanden müssen vom Typ int sein
- - Das negative Vorzeichen
- ⚡ Vorrangregel: Vorzeichen - vor *, / und % vor + und -

Logische Verknüpfungen

- > Vergleich auf größer
- < Vergleich auf kleiner
- >= Vergleich auf größer oder gleich
- <= Vergleich auf kleiner oder gleich
- == Vergleich auf Gleichheit
- != Vergleich auf Ungleichheit
- && Logische und Verknüpfung
- || Logische oder Verknüpfung
- ! Logische Negation

⚡ Vorrangregel: >,<,>= und <= vor == und != vor && vor ||

Inkrement und Dekrement

- ++ Unitäres Inkrement (als Präfix oder Postfix)
- -- Unitäres Dekrement (als Präfix oder Postfix)

Beispiel:

```
i=5;
++i;
x = i++;
--x;
y= x--;
```

Resultat: i Wert 7; x Wert 4; y Wert 5 !

Bitoperationen

- | | |
|------|--------------------------------------|
| ■ & | ■ Bitweise und Verknüpfung |
| ■ | ■ Bitweise oder Verknüpfung |
| ■ ^ | ■ Bitweise exklusiv-oder Verknüpfung |
| ■ << | ■ Bit-Shift nach links |
| ■ >> | ■ Bit-Shift nach rechts |
| ■ ~ | ■ Einer-Komplement einer Zahl |

Zuweisungen

- | | |
|------|---------------------------------|
| ■ = | ■ Die „einfache“ Zuweisung |
| ■ += | ■ Addition plus Zuweisung |
| ■ -= | ■ Subtraktion plus Zuweisung |
| ■ *= | ■ Multiplikation plus Zuweisung |
| ■ /= | ■ Division plus Zuweisung |
| ■ &= | ■ Bitweises und plus Zuweisung |
| ■ = | ■ Bitweises oder plus Zuweisung |

Beispiel: `y = 3+(x = 7);`
 `y += x;`

Resultat: x Wert 7; y Wert 17 !

Bedingte Bewertung

- ⚡ Die Bedingte Bewertung hat drei Operatoren
 `<Bed-Bew> ::= <Ausdruck>?<Ausdruck>:<Ausdruck>`
- | | | |
|-------|-------|--|
| ■ ? | ■ : | ■ Der erste Ausdruck wird ausgewertet, ist er wahr, so wird der zweite Ausdruck ausgewertet und dieser stellt das Ergebnis dar, andernfalls wird der dritte Ausdruck ausgewertet und dieser stellt das Ergebnis da |
|-------|-------|--|

Beispiel:

`y = (x>=0)?sqrt(x):sqrt(-x);`

Das Komma

- | | |
|-----|---|
| ■ , | ■ Der Komma Operator trennt zwei Ausdrücke voneinander. Das Ergebnis des gesamten Ausdrucks ist gleich den zweiten Ausdruck |
|-----|---|

Beispiel:

`i=2, j=3, z=4;`
`y=(a=i*j, b=5, a*b*z);`

Resultate:

i Wert 2; j Wert 3; z Wert 4
a Wert 6; b Wert 5; y Wert 120 !

Allgemeines

- In C hat jede Operation ein Ergebnis
 - `a = 5` hat das Ergebnis 5
- Ausdrücke können z.T. sehr elegant geschrieben werden
 - `z = (a > b) ? a : b; /* z = max(a, b) */`
- Ausdrücke werden nur solange ausgewertet bis das Ergebnis fest steht; er werden nur notwendige Teile ausgewertet
 - `y = (x>0)?sqrt(x):sqrt(-x);`
 - `if (x>0 && y/x<0) /*keine Div. durch 0 möglich!*/`

Vorrangregeln*

Operator
<code>! ~ ++ -- -</code>
<code>* / %</code>
<code>+ -</code>
<code><< >></code>
<code>< <= > >=</code>
<code>== !=</code>
<code>&</code>
<code>^</code>
<code> </code>
<code>&&</code>
<code> </code>
<code>? :</code>
<code>= += -= ...</code>
<code>,</code>

*nicht vollständig