

Elementare Datentypen

Datentyp int (Integer):

Dieser Datentyp kann ganzzahlige Werte aufnehmen. Die Größe dieses Datentyps ist systemabhängig. Bei 16-BIT-Betriebssystemen bzw. Compilern hat dieser Datentyp die Größe von 2 Byte. Bei 32-BIT-Systemen (z.B. Windows XP) hat dieser Datentyp die Größe von 4 Byte. Entsprechend der Byte-Größe hat der int-Typ folgenden Wertebereich:

| | |
|------------|--|
| 2-Byte int | -32768 32767 |
| 4-Byte int | -2147483648..... 2147483647 (long int) |

Die Voreinstellung der Ganzzahlvariablen ist vorzeichenbehaftet. Werden hingegen nur positive Werte benötigt, so muß das explizit angegeben werden mit dem Schlüsselwort **unsigned**.

Datentyp char (Character):

Dieser Datentyp kann ganzzahlige Werte bzw. Zeichen aufnehmen. Die Größe dieses Datentyps beträgt ein Byte. Entsprechend der Byte-Größe hat der char-Typ folgenden Wertebereich:

| | |
|---------------|---------------|
| char | -128127 |
| unsigned char | 0..... 255 |

Mit dem Datentyp char können die gleichen Operationen durchgeführt werden wie mit dem int - Datentyp. Meistens wird der char - Datentyp für die Speicherung eines Zeichens benutzt bzw. für die Speicherung einer Zeichenkette (String).

Datentyp float (Floating point):

Für die Darstellung von Gleitpunktzahlen gibt es den **float** - Datentyp bzw. für doppelt genaue Darstellung den **double** - Datentyp bzw. **long double** -Datentyp.

Der float-Typ ist 4 Byte, der double-Typ 8 Byte und der long-double-Typ 10 Bytes groß. Intern werden die Gleitpunktzahlen in Form von Mantisse und Exponent dargestellt.

Wertebereich der Datentypen:

| | |
|-------------|-------------------------------|
| float | -3.4 E -383,4 E + 38 |
| double | -1.7 E -3081,7 E + 308 |
| long double | -1.1 E -4932.....1,1 E + 4932 |

Dabei steht E für die Exponentialschreibweise: 3.4 E +38 bedeutet $3.4 * 10^{38}$

Datentyp void:

Der Datentyp void steht für "leerer Datentyp". Er kommt beispielsweise dann zur Anwendung, wenn Funktionen keinen Rückgabewert haben. Diese Funktionen werden dann als void-Funktionen geschrieben.

Eingabe über die Tastatur mit cin

Das Objekt **cin** (dazu später mehr) ist in der Lage, Eingaben über die Tastatur in einer Variablen zu speichern (*cin steht für character input und ist ein Objekt der Klasse istream*). Sehr angenehm ist dabei, dass das Objekt automatisch die Datentypen der Variablen erkennt und eine korrekte Zuweisung machen kann.

Beispiele:

```
int x;
char y;
float z;

cin >> x;
cin >> x >> y >> z;    // Einlesen der Variablen x , y und z
```

Zuweisungsoperator und arithmetische Operatoren

Bei der Deklaration von Variablen können den Variablen sofort Werte zugewiesen werden. Ebenso können Variablen im Programm Werte zugewiesen werden. Diese Zuweisung geschieht mit dem **Zuweisungsoperator** =.

Beispiel:

```
int x = 10;
cout << x;    // Es wird 10 ausgegeben
x = 20;
cout << x;    // Es wird 20 ausgegeben
```

Wie aus der Mathematik bekannt, können die Grundrechenarten mit Variablen durchgeführt werden. Die Operatoren sind +, -, *, **und** /.

Beispiel:

```
int x = 10;
int y = 20;
int z;
z = 10 * 20;
cout << z;    // Es wird 200 ausgegeben
```

Aufgaben:

Schreiben Sie C++-Programme, um folgende Aufgabenstellungen umzusetzen:

- Zwei Gleitpunktzahlen sollen eingelesen werden. Anschließend sollen Summe, Produkt und Differenz der beiden Zahlen angezeigt werden.
- Nach Eingabe von Spannung und Widerstand soll die Stromstärke berechnet werden.
- Es ist das Endkapital einer Spareinlage zu berechnen. Dazu soll das Startkapital und der Zinssatz eingelesen werden. Die Anlage dauert immer drei Jahre. Beachten Sie den Zinseszinsseffekt. Anschließend soll das Endkapital ausgegeben werden.