Datentypen

Eine Variable ist ein Platz im Arbeitsspeicher, der mit einem Namen versehen ist. Damit der Compiler weiß, wie viel Speicherplatz für die einzelnen Variablen im Arbeitsspeicher reserviert werden soll und was die abgelegten Daten bedeuten, muss der Datentyp der Variablen angegeben werden.

Diese Übersicht beschreibt die wichtigsten Datentypen in C++. Einige weitere werden wir im Lauf der Zeit kennenlernen.

Ganze Zahlen

Datentyp	Größe (Visual Studio, 32 Bit)	Wertebereich (Visual Studio, 32 Bit)
`short` oder `short int`	16 Bit	-2^{15} bis $+2^{15}-1$
`int`	32 Bit	-2^{31} bis $+2^{31}-1$
`long` oder `long int`	32 Bit	-2^{31} bis $+2^{31}-1$
`long long` oder `long long int`	64 Bit	-2^{63} bis $+2^{63}-1$

Meistens verwendet man `int`, weil dieser Typ für den Prozessor am einfachsten zu verarbeiten ist.

Die Größenangaben gelten für den C++-Compiler von Microsoft, wenn er 32-Bit-Programme erzeugt. Allgemein ist `short` höchstens so groß wie `int` und `long` mindestens so groß. `long long` hat mindestens 64 Bit.

Den Ganzzahltypen kann man das Wort `unsigned` voranstellen. Dann speichern sie nur positive Zahlen. `unsigned int` (oder kurz nur `unsigned`) hat zum Beispiel bei 32 Bit einen Zahlenbereich von 0 bis $2^{32}-1$.

Reelle Zahlen (Gleitkommazahlen)

Datentyp	Größe	Gültige Stellen	Zahlenbereich
`float`	32 Bit	7	$-3{,}4\cdot10^{38}$ bis $-3{,}4\cdot10^{-38}$ und $+3{,}4\cdot10^{-38}$ bis $+3{,}4\cdot10^{38}$
`double`	64 Bit	15	$-1{,}7\cdot10^{308}$ bis $-1{,}7\cdot10^{-308}$ und $+1{,}7\cdot10^{-308}$ bis $+1{,}7\cdot10^{308}$

Auch hier gelten alle Werte für den Microsoft-Compiler. Man verwendet in der Regel `double` und weicht nur auf `float` aus, wenn der Speicherplatz knapp ist.

Wahrheitswerte

Wahrheitswerte speichert man im Typ `bool`. Er kennt nur die Werte `true` und `false`. Früher wurde auch `int` für Wahrheitswerte verwendet. Dann stand 0 für falsch und alle anderen Zahlen bedeuteten wahr.

Einzelne Zeichen

Für Zeichen gibt es den Datentyp `char`. Er speichert in Wirklichkeit kein Zeichen, sondern den Code des Zeichens als ganze Zahl mit 8 Bit. Sie können mit `signed char` oder `unsigned char` festlegen, ob das erste Bit als Vorzeichen interpretiert wird.

Definieren von Variablen

Jede Variable muss in C++ definiert werden, bevor man sie benutzt. Variablennamen können aus Buchstaben (a-z, A-Z), Ziffern (nicht an der 1. Stelle) und dem Unterstrich gebildet werden. Sonderzeichen (z.B. ä,Ö,ü,ß) sind nicht empfehlenswert, weil man sie höchstens mit bestimmten Zeichencodes einsetzen kann.

Beispiele:

```
int       zahl;                  /* Typ    Name    Strichpunkt */
short     kleineZahl;
double    gleitkommazahl;
```

Mit Kommas getrennt kann man mehrere Variablen vom gleichen Typ in einer Zeile definieren:

```
int a, b, c;
```

> Jedes dieser Beispiele stellt eine **Definition** dar: Es wird der Variablenname mit seinem Datentyp festgelegt und Speicherplatz reserviert.
>
> In der Praxis sagt man dazu oft **Deklaration**. Das ist in C++ streng genommen falsch, denn dieser Fachbegriff steht nur für die Bekanntgabe des Namens und des Datentyps.

Initialisierung

Man kann Variablen gleich bei der Definition einen Wert geben:

```
int       zahl = 1234;
short     kleineZahl = -34;
double    gleitkommazahl = 1.45;

int a=1, b=17, c=42;
```

Zeichen muss man in Hochkommas setzen, sonst sind sie nicht von Variablennamen zu unterscheiden.

```
char z = 'A';
```

Bei einer Initialisierung steht rechts vom =-Zeichen fast immer ein fester Wert im Quellcode. So einen Wert nennt man **Literal**. `1234` ist zum Beispiel ein Ganzzahl-Literal und `'A'` ein Zeichen-Literal.

Besonderheiten bei Zeichen

Einige Zeichen können nicht auf dem Bildschirm dargestellt oder mit der Tastatur erzeugt werden. Im C++-Quellcode verwendet man für sie eine besondere Schreibweise:

Bedeutung	Steuerzeichen, ASCII-Code	Schreibweise
Klingelzeichen	BELL, 7	'\a'
Backspace	BS, 8	'\b'
Horizontal tab, Tabulator	HT, 9	'\t'
Linefeed, Zeilenvorschub	LF, 10	'\n'
Carriage return, Wagenrücklauf	CR, 13	'\r'

Weil das Hochkomma, das Anführungszeichen und der Backslash \ selbst eine Bedeutung haben, wird ihnen ein Backslash voran gestellt. Man sagt, sie werden maskiert:

Hochkomma	'	'\''
Anführungszeichen	"	'\"'
Backslash	\	'\\'

Beispiel:

```
// a bekommt als Wert einen einzelnen Backslash:
char a = '\\';
```