

Deklaration von Variablen

Im Wesentlichen beschäftigt sich die Programmierung mit der Verarbeitung unterschiedlichster Daten. Um beispielsweise Werte einzugeben (Eingabe), Berechnungen durchzuführen (Verarbeitung) und die Ergebnisse auszugeben (Ausgabe), benötigen wir Speicher für diese Daten. Bevor wir den Speicherplatz benutzen können müssen wir im Programm einen symbolischen Namen für diesen Speicherbereich (=Bezeichner) und einen Datentyp festlegen (= Deklaration einer Variablen).

Bezeichner

Bezeichner sind Namen von beliebiger Länge für Klassen, Objekte, Funktionen, Variablen, benutzerdefinierte Datentypen usw.

Regeln für die Bildung von Bezeichnern:

Bezeichner können die Buchstaben a bis z, A bis Z (ohne sprachspezifische Besonderheiten wie Ä, ä, ö, ß usw.), den Unterstrich „_“ und die Ziffern 0 bis 9 enthalten. Es gibt jedoch zwei Einschränkungen:

Das erste Zeichen muss

- ein Buchstabe oder
- ein Unterstrich

sein.

Länge von Bezeichnern:

Nach dem ANSI-Standard sind die ersten 31 Zeichen zu unterscheiden.

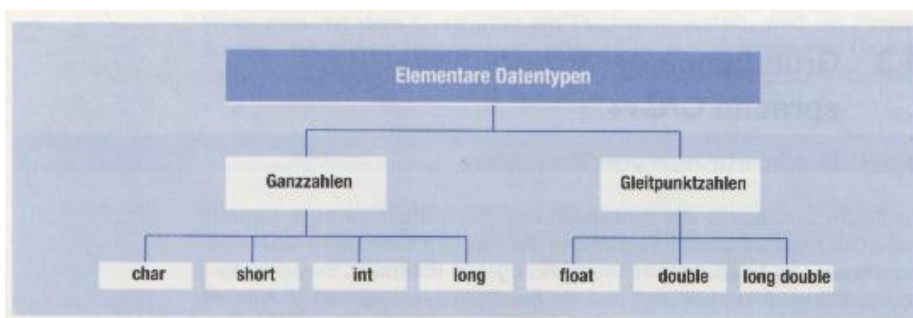
Groß- und Kleinschreibung von Bezeichnern:

In C/C++ wird zwischen Groß- und Kleinschreibung unterschieden, so dass *Sum*, *sum* und *suM* unterschiedliche Bezeichner sind.

Der Bezeichner darf nicht mit einem Schlüsselwort von C++ übereinstimmen.

Elementare Datentypen

Nicht nur der Name einer Variablen ist von Bedeutung, sondern man muss bei der Deklaration einer Variablen auch den Datentyp festlegen. Der Datentyp einer Variablen entscheidet darüber, wie viel Platz das Datenobjekt im Speicher belegt und in welchem Format die Variable gespeichert wird. Daraus ergibt sich der Wertebereich (möglicher Inhalt der Variablen) für dieses Datenobjekt.



In C++ sowie in anderen Programmiersprachen unterscheidet man zwei verschiedene Grundtypen, in denen Werte gespeichert werden: Ganzzahlentypen (Integer) oder Gleitpunktzahlentypen (floating point).

Ganzzahlentypen sind zum Beispiel die Zahlen 10, 14 und 512. Gleitpunktzahlentypen dagegen sind beliebige Kommazahlen wie 3.14 oder 1.23e-3. Zunächst werden hier einmal die verschiedenen Typen von C++ und deren Wertebereiche in einer Tabelle dargestellt.

Typ	Größe	Bereich	Beispielanwendungen
unsigned char	8 Bit	0 bis 255	Kleine Zahlen und kompletter PC-Zeichensatz
char	8 Bit	-128 bis 127	Sehr kleine Zahlen und ASCII-Zeichen
short int	16 Bit	-32,768 bis 32,767	Zähler, kleine Zahlen, Schleifensteuerung
unsigned int	32 Bit	0 bis 4,294,967,295	Große Zahlen und Schleifen
int	32 Bit	-2,147,483,648 bis 2,147,483,647	Zähler, große Zahlen, Schleifensteuerung
unsigned long	32 Bit	0 bis 4,294,967,295	Astronomische Distanzen
enum	32 Bit	-2,147,483,648 bis 2,147,483,647	Geordnete Wertemengen
long	32 Bit	-2,147,483,648 bis 2,147,483,647	Sehr große Zahlen, statistische Grundgesamtheiten
float	32 Bit	$1,18 \cdot 10^{-38}$ bis $3,40 \cdot 10^{38}$	Wissenschaftlich, 7-stellige Genauigkeit
double	64 Bit	$2,23 \cdot 10^{-308}$ bis $1,79 \cdot 10^{308}$	Wissenschaftlich, 15-stellige Genauigkeit
long double	80 Bit	$3,37 \cdot 10^{-4932}$ bis $1,18 \cdot 10^{4932}$	Finanzrechnung, 18-stellige Genauigkeit

Bei der Deklaration von Variablen muss man bedenken, welche Informationen in einem solchen Datenelement abgespeichert werden müssen. Hat man beispielsweise einen Datentyp gewählt, bei dem der Wertebereich nicht ausreicht, um entsprechend große Zahlen aufzunehmen, kommt es zwar bei der Kompilierung zu keinen Fehlermeldungen, aber bei gewissen Datenkonstellationen ergeben sich Laufzeitfehler. Um die zu vermeiden, könnte man generell die Datentypen wählen, die den größtmöglichen Wertebereich zur Verfügung stellen (in bzw. long double) und damit den meisten Speicherplatz beanspruchen. Dies mag für die meisten Variablen richtig sein. Bei Datenobjekten einer Datenbank, in der möglicherweise hunderttausende von Datensätzen erfasst werden, führt das jedoch zu einer enormen Verschwendung von Speicherplatz und außerdem verringert sich die Laufzeiteffizienz des Programms.

```
Beispiele:      int Zahl;           // Deklaration
                Zahl = 5;          // Initialisierung
                double Summe, Betrag;
                int Zaehler = 0;    // Definition
                unsigned char Zeichen;
```

Die Variable Zaehler wird bei der Deklaration mit dem Wert 0 initialisiert. Die Initialisierung von Variablen ist sehr wichtig, da (lokale) Variablen bei Start eines Programms nicht automatisch auf einen Initialwert gesetzt werden. Bei Gleitkommazahlen müssen Sie beachten, dass das Dezimalkomma durch die amerikanische Schreibweise (Dezimalpunkt) zu ersetzen ist.