

Anhang C

Reservierte Wörter und Rangfolge von Operatoren

C.1 Reservierte Wörter in C

Reserviertes Wort	Art/Bedeutung
auto	Speicherklasse (gültig, wenn nicht explizit static angegeben)
break	verlässt aktuelle Schleife oder <code>switch</code> -Anweisung
case	markiert einen Fall in der <code>switch</code> -Anweisung
char	Datentyp für Zeichen
const	deklariert schreibgeschützte/s Variable / Funktionsargument
continue	beendet aktuellen Schleifendurchlauf
default	stets eintretender Fall in der <code>switch</code> -Anweisung
do	Wiederholungsanweisung
double	Gleitpunktdatentyp doppelter Genauigkeit
else	leitet alternativen Block in der <code>if</code> -Anweisung ein
enum	Aufzählungsdentyp
extern	Speicherklasse
float	Gleitpunktdatentyp einfacher Genauigkeit
for	Wiederholungsanweisung
goto	Sprunganweisung
if	bedingte Anweisung
int	ganzzahliger Datentyp
long	erweiterter ganzzahliger Datentyp
register	Speicherklasse
return	liefert Funktionswert zurück
short	Typmodifizierer
signed	Typmodifizierer (Standardwert)
sizeof	Operator, bestimmt Größe eines Datenobjekts in Bytes
static	Speicherklasse
struct	deklariert eine Struktur
switch	Fallunterscheidung
typedef	definiert neuen Datentyp

Reserviertes Wort	Art/Bedeutung
union	deklariert eine Union
unsigned	Typmodifizierer
void	leerer Datentyp
volatile	Speicherklasse
while	Wiederholungsanweisung

C.2 Rangfolge und Assoziativität der C-Operatoren

Die Tabelle zur Rangfolge ist folgendermaßen zu verstehen: Operatoren einer niedrigeren Stufe haben Vorrang vor jenen höherer Stufenzahl. Innerhalb einer Stufe bestimmt die Assoziativität, wie der Ausdruck verarbeitet ist, sofern nicht durch Klammern etwas anderes angegeben wird.

Operator	Name/Bedeutung	Assoziativität
<i>Stufe 1</i> () [] -> .	Klammern Array-Element Zeiger auf Strukturelement Struktur- oder Unionelement	<i>linksassoziativ</i>
<i>Stufe 2</i> ! ~ ++ -- - + & * sizeof (Datentyp)	Logische Negation Einerkomplement Inkrement Dekrement Unäres Minus Unäres Plus Adresse Dereferenzierung Größe in Bytes Typumwandlung (<i>Cast</i>)	<i>rechtsassoziativ</i>
<i>Stufe 3</i> * / %	Multiplikation Division Rest einer Division	<i>linksassoziativ</i>
<i>Stufe 4</i> + -	Addition Subtraktion	<i>linksassoziativ</i>
<i>Stufe 5</i> << >>	bitweises Linksschieben bitweises Rechtsschieben	<i>linksassoziativ</i>

Operator	Bedeutung/Name	Assoziativität
<i>Stufe 6</i> < <= > >=	kleiner als kleiner oder gleich größer als größer oder gleich	<i>linksassoziativ</i>
<i>Stufe 7</i> == !=	gleich ungleich	<i>linksassoziativ</i>
<i>Stufe 8</i> &	Bitweises UND	<i>linksassoziativ</i>
<i>Stufe 9</i> ^	Bitweises EXKLUSIV-ODER	<i>linksassoziativ</i>
<i>Stufe 10</i> 	Bitweises ODER	<i>linksassoziativ</i>
<i>Stufe 11</i> &&	Logisches UND	<i>linksassoziativ</i>
<i>Stufe 12</i> 	Logisches ODER	<i>linksassoziativ</i>
<i>Stufe 13</i> ?:	bedingter Ausdruck	<i>rechtsassoziativ</i>
<i>Stufe 14</i> = *= /= %= += -= <<= >>= &= ^= =	Zuweisung arith. Zuweisungsoperatoren bitweise Zuweisungsoperatoren	<i>rechtsassoziativ</i>
<i>Stufe 15</i> ,	Komma-Operator	<i>linksassoziativ</i>