

Programmiermethoden in der Mathematik WS 02/03: Datentypen in C++

Beschreibung	Datentyp in C++	Byte	darstellbarer Zahlenbereich (bei Gleitpunktzahlen $x_{min} \dots x_{max}$)		Literal- ^{**} konstante
ganze Zahl mit Vorzeichen	kurz	(signed) short (int)	2	$-32768 \dots 32767$	SHRT_MIN...SHRT_MAX
	normal	(signed) int	4	$-2\,147\,483\,648 \dots 2\,147\,483\,647$	INT_MIN...INT_MAX
	lang	(signed) long (int)	4*	$-2\,147\,483\,648 \dots 2\,147\,483\,647$	LONG_MIN...LONG_MAX
	extra lang	(signed) long long (int)	8*	$-9.2 \cdot 10^{18} \dots 9.2 \cdot 10^{18}$	LLONG_MIN...LLONG_MAX
ganze Zahl ohne Vorzeichen	kurz	unsigned short (int)	2	$0 \dots 65535$	0...USHRT_MAX
	normal	unsigned (int)	4	$0 \dots 4\,294\,967\,295$	0...UINT_MAX
	lang	unsigned long (int)	4*	$0 \dots 4\,294\,967\,295$	0...ULONG_MAX
	extra lang	unsigned long long (int)	8*	$0 \dots \approx 1.8 \cdot 10^{19}$	0...ULLONG_MAX
Gleitpunktzahl	einfach genau	float	4	$1.2 \cdot 10^{-38} \dots 3.4 \cdot 10^{38}$	FLT_MIN...FLT_MAX
	doppelt genau	double	8	$2.2 \cdot 10^{-308} \dots 1.8 \cdot 10^{308}$	DBL_MIN...DBL_MAX
	dreifach genau	long double	12(Linux) 8(AIX)	$3.4 \cdot 10^{-4932} \dots 1.2 \cdot 10^{4932}$	LDBL_MIN...LDBL_MAX
Zeichen (Character)	mit Vorzeichen	(signed) char	1	entspricht $-128 \dots 127$	'a'
	ohne Vorzeichen	unsigned char	1	entspricht $0 \dots 255$	'a'
logischer Wert		bool	1	true oder false	true
leer (nur als Ein- o. Rückgabewert, z.B. bei main)		void	-	-	-

Bemerkungen:

- Ausdrücke in Klammern sind optional, also ist z.B. `short` eine Abkürzung für `signed short int`.
- Die Werte mit * sind plattformabhängig, d.h. sie können auf verschiedenen Betriebssystemen bzw. Compilern variieren. Angegeben sind die Werte, die auf Linux/AIX mit `g++` sowie auf AIX mit dem IBM-eigenen `x1C` Compiler gelten.
- Die Konstanten in der vorletzten Spalte sind in den Header-Dateien `limits.h` oder `climits` (ganze Zahlen) und `float.h` oder `cfloats` (Gleitkommazahlen) definiert.
- Mit `sizeof()` kann man die Länge einer Variablen und eines Datentyps (außer `void`) feststellen, z.B. liefert `sizeof(x)` die Länge der Variablen `x` (in Byte), und `sizeof(int)` die Länge des Datentyps `int` auf der aktuellen Maschine, also z.B. im Unix-Pool den Wert 4.
- ^{**}Eine Literalkonstante ist eine Konstante, die in einem Rechenausdruck auftaucht, z.B. `j=2*k;`.