

Arithmetik#

Table of Contents

- [Arithmetik](#)
- [Stacknotation](#)
- [VolksForth Arithmetik](#)
- [VolksForth Logik Worte](#)
- [VolksForth 32bit Worte](#)
- [Stack Operations](#)

Stacknotation#

Im folgenden werden hauptsächlich Worte in ihrer Einzelfunktion beschrieben. In dieser Form der Beschreibung, die Sie bereits kennengelernt haben, wird die Wirkung eines Wortes auf den Stack in Klammern angegeben und zwar in folgender Form:

(vorher -- nachher)

vorher - Werte auf dem Stack vor Ausführung des Wortes nachher - Werte auf dem Stack nach Ausführung des Wortes

In dieser Notation wird das oberste Element des Stacks (tos) immer ganz rechts geschrieben. Sofern nicht anders angegeben, beziehen sich alle Stacknotationen auf die spätere Ausführung des Wortes. Bei immediate Worten wird auch die Auswirkung des Wortes auf den Stack während der Kompilierung angegeben. Worte werden ferner durch folgende Symbole gekennzeichnet:

c	Dieses Wort kann nur während der Kompilation einer :-Definition benutzt werden.
I	Dieses Wort ist ein immediate Wort, das auch im kompilierenden Zustand ausgeführt wird.
83	Dieses Wort wird im Forth83-Standard definiert und muß auf allen Standardsystemen äquivalent funktionieren.
U	Kennzeichnet eine Uservariable.

Weicht die Aussprache eines Wortes von der natürlichen englischen Aussprache ab, so wird sie in Anführungszeichen angegeben. Gelegentlich folgt auch eine deutsche Übersetzung. Die Namen der Stackparameter folgen, sofern nicht suggestive Bezeichnungen gewählt wurden, dem nachstehendem Schema. Die Bezeichnungen können mit einer nachfolgenden Ziffer versehen sein.

Stacknotation	Zahlentyp	Wertebereich in Dezimal	minimale Feldbreite
flag	logischer Wert	0=false, sonst=true	16bit
true (tf)	logischer Wert	-1 (als Ergebnis)	16bit
false (ff)	logischer Wert	0	16bit
b	Bit	0..1	1bit
char	Zeichen	0..127 (0..255)	7(8bit)
8b	8 beliebige Bits	nicht anwendbar	8bit
16b	16 beliebige Bits	nicht anwendbar	16bit
n	Zahl, bewertete Bits	-32768 .. 32767	16bit

+n	positive Zahl	0 .. 32767	16bit
u	vorzeichenlosen Zahl	0 .. 65535	16bit
w	Zahl, n oder u	-32768..65535	16bit
addr	Adresse, wie u	0..65535	16bit
32b	32 beliebige Bits	nicht anwendbar	32bit
d	doppelt genaue Zahl	-2,147,483,648..2,147,483,647	32bit
+d	positive doppelt genaue Zahl	0..2,147,483,647	32bit
ud	vorzeichenlose, doppelt genaue Zahl	0..4,294,967,295	32bit
sys	systemabhängige Werte	nicht anwendbar	nicht anwendbar

VolksForth Arithmetik#

-1 (-- -1)
0 (-- 0)
1 (-- 1)
2 (-- 2)
3 (-- 3)
4 (-- 4)

Oft benutzte Zahlenwerte wurden zu Konstanten gemacht. Definiert in der Form :

n Constant n

Dadurch wird Speicherplatz eingespart und die Ausführungszeit verkürzt.

- [1+](#)
- [1-](#)
- [2+](#)
- [2-](#)
- [2*](#)
- [2/](#)
- [3+](#)
- [abs](#)
- [not](#)
- [negate](#)
- [even](#)
- [max](#)
- [min](#)
- [+](#)
- [-](#)
- [*](#)
- [/](#)
- [mod](#)
- [/mod](#)
- [*/](#)
- [*/mod](#)
- [u/mod](#)
- [umax](#)
- [umin](#)

VolksForth Logik Worte#

- [true](#)
- [false](#)
- [0=](#)
- [0<>](#)
- [0<](#)
- [0>](#)
- [=](#)
- [<](#)
- [>](#)
- [u<](#)
- [u>](#)
- [and](#)
- [or](#)
- [xor](#)
- [uwithin](#)
- [case?](#)

VolksForth 32bit Worte#

- [extend](#)
- [dabs](#)
- [dnegate](#)
- [d+](#)
- [d-](#)
- [d*](#)
- [d=](#)
- [d<](#)
- [d0=](#)
- [m*](#)
- [um*](#)
- [m/mod](#)
- [ud/mod](#)
- [um/mod](#)

Stack Operations#

Herkömmliche Programmiersprachen enthalten mehr oder weniger ausgeprägt das Konzept der PROZEDUREN: Für bestimmte Programmfunktionen notwendige Operatoren werden in benannten Programmteilen zusammengefasst, um diese Programmfunktionen an mehreren Stellen innerhalb eines Programmes über ihren Namen aktivieren zu können. Da FORTH ohne jede Einschränkung prozedural ist, macht FORTH auch keinen Unterschied zwischen Prozeduren und Funktionen oder seinen Operatoren. Alles wird als ein WORT bezeichnet.

FORTH als Programmier-SPRACHE besteht also aus Wörtern. Somit können FORTH-Wörter sein:

1. Datenbereiche
2. Algorithmen (Befehle)
3. Programme

Um Prozeduren sinnvoll benutzen zu können, kennen die meisten Sprachen auch PARAMETER: Dies sind Daten, die einer Prozedur bei ihrem Aufruf zur Bearbeitung übergeben werden. Daten, die ausschließlich innerhalb einer Prozedur benötigt werden, heißen LOKAL zu dieser Prozedur; im

Gegensatz dazu nennt man Daten, die außerhalb von bestimmten Prozeduren zur Verfügung stehen und auf die von allen Prozeduren aus mit allen Operatoren zugegriffen werden kann, GLOBAL.

Die erste Möglichkeit der Parameterübergabe zwischen Prozeduren ist die Vereinbarung von benannten GLOBALEN Variablen. Diese globalen Variablen sind für die gesamte Laufzeit des Programmes statisch existent und können von allen Prozeduren manipuliert werden.

Eine andere Möglichkeit der Parameterübergabe besteht im Einrichten eines Speicherbereiches, in dem während des Aufrufes eines Wortes namentlich benannte Parameter dynamisch verwaltet werden. Diesen Mechanismus für benannte lokale Variable stellt Standard-FORTH nicht zur Verfügung, weil die Organisation dieser lokalen Variablen mit einem Verlust sowohl der lokalen Daten nach der Ausführung des Wortes als auch einem Verlust in der Ausführungsgeschwindigkeit des Wortes verbunden sind.

Für das volks4TH wurde in der VD 1/88 eine Implementierung benannter lokaler Variablen vorgestellt.

FORTH benutzt zur gegenseitigen Übergabe von Parametern an Wörter hauptsächlich den STACK, einen bestimmten Speicherbereich, in dem die Wörter ihre Parameter erwarten. Diese Parameter erhalten keine Namen, sondern ihre Interpretation ergibt sich aus der Position innerhalb des Stack-Speicherbereiches. Daraus resultiert die Vielzahl von Operatoren zur Änderung der Stack-Position eines Wertes, für die FORTH berühmt/berüchtigt ist.

Damit deutlich wird, welche und wieviele Parameter ein Wort benötigt, werden allgemein STACK-KOMMENTARE verwendet:

Der öffnenden runden Klammer folgt eine Aufzählung der Parameter. Dabei steht der Parameter, der als oberstes Stack-Element erwartet wird, ganz rechts. Dann folgt ein "--", das die Ausführung des Wortes symbolisieren soll. Anschließend wird der Zustand des Stacks nach der Ausführung des Wortes dargestellt, wobei das oberste Stackelement wieder ganz rechts steht. Die schließende runde Klammer beendet den Stack-Kommentar.

Ein Wort SQRT, das die Quadratwurzel einer Integerzahl liefert, würde in FORTH so benannt und beschrieben:

```
sqrt ( number -- sqrt )
```

Wird dieses neue Wort aufgerufen, so werden alle darin enthaltenen Wörter ausgeführt, eventuell bereitgestellte Parameter bearbeitet und daraus resultierende Ergebnisse auf dem Stack übergeben.

Der Aufruf von Prozeduren erfolgt in FORTH implizit durch die Nennung des Namens, ebenso wie auch die Datenübergabe zwischen Wörtern meist implizit erfolgt.

- [drop](#)
- [2drop](#)
- [dup](#)
- [?dup](#)
- [2dup](#)
- [swap](#)
- [2swap](#)
- [nip](#)
- [over](#)
- [2over](#)
- [under](#)
- [rot](#)

- [-rot](#)
- [roll](#)
- [-roll](#)
- [pick](#)
- [.s](#)
- [clearstack](#)
- [depth](#)
- [s0](#)
- [sp!](#)
- [sp@](#)