1. Wörter der Prelude-Implementierung in cspForth

Glossary: **prelude** in the context **forth**

prelude (xt --)
Assigns the word with the xt from TOS as a prelude to the next created word. No prelude will be assigned for xt = 0 .

Note: Execution tokens are (and must be) cell-aligned in csp4th.

see also: (prelude)

Glossary: **(prelude)** in the context **forth**

(prelude) (-- a)
A variable holding a word class or the execution token of a prelude to be assigned to the next created word.

The default is the default word class from Reva 6.0. It was called 'forth there and is now called 'default in csp4th. The other word class names from Reva 6.0, 'inline , 'macro and 'notail are unchanged.

To distinguish word classes from preludes, bit 0 of a prelude's xt must be set true. The word **prelude** does this for you.

Note: Execution tokens are (and must be) cell-aligned in csp4th.

(This variable should only be used for debugging purposes!)

see also: prelude

Note for KS: Word classes are specific for Reva Forth / cspForth and don't have anything to do with OOP with cspForth.

Glossary: **prelude?** in the context **forth**

prelude? (-- ?)
Returns a true flag, while a prelude is executed by the Forth text interpreter (the outer interpreter). Returns false otherwise.

Note: It's used by the oop module.

see also: prelude

Glossary: **(header)** in the context **forth**

(header) (a u --)

Create a word header in the dictionary using the string "a,u" as it's name.

Note for KS: A prelude is assigned to the new word, if an execution token was assigned to the variable (prelude). Thereafter the variable is reset to its default value (no prelude).
A words prelude is executed by the outer interpreter before the interpret- or compile-time semantics of the word is applied.

see also: header

2. Wörter für die Kontext-Umschaltung in cspForth

Glossary: **v.context** in the context **forth**

v.context (-- a)
A variable pointing to the top of the vocabulary search order.

see also: context e.context

Glossary: **e.context** in the context **forth**

e.context (-- a)
A variable pointing to the top of the extra search order, i.e. class or interface search order.

see also: context v.context

Glossary: **context** in the context **forth**

context (-- a)
A variable pointing to the top of the current search order. 'context @' returns the context identifier 'cid' of the first wordlist in the search order.

Note for KS: : context (-- a) (esm) @ if e.context else v.context then ;

see also: current

Glossary: **(esm)** in the context **forth**

(esm) (-- a)
A variable. True if the system is in the extra search mode (interface or in the class search mode). False otherwise.

see also: (ecm) (osm) e.context v.context s.context

3. OOP-Implementierung in cspForth

siehe cspForth-sn03-090322.tar.gz

./csp4th-sn03-090322/README-sn0x.pdf
./csp4th-sn03-090322/share/oop.4th

Zwei im cspForth-Kern definierte Variable, die von der OOP-Erweiterung verwendet werden:

Glossary: **(osm)** in the context **forth**

(osm) (-- a)
A variable. True if the system is in object search mode. false otherwise. Should not be changed by an application programmer.

see also: (esm) (ecm)

Glossary: **(ecm)** in the context **forth**

(ecm) (-- a)
A variable. True if the system is in the extra compile mode (class or interface compile mode). False otherwise.

see also: (esm) (osm) e.context v.context s.context