

How To

Benjamin Arnold Felix Hoeborn

26. Februar 2013

Inhaltsverzeichnis

1	Top-Level	1
1.1	Includes	1
1.2	Defines	1
1.3	Klassen	1
1.4	Funktionsdefinition	1
2	Low-Level	1
2.1	Expressions	1
2.1.1	Operationen	1
2.1.2	Funktionsaufrufe	2
2.1.3	Set	2
2.1.4	Terme	2
2.2	Terme	2

1 Top-Level

Diese Sprache hat auf oberster Ebene 4 Bestandteile.

1.1 Includes

`include dateiname ;`

1.2 Defines

Es können globale Variablen definiert werden.

`< type > variablenname ;` für nicht initiierte Variablen.

`< type > variablenname = term ;` für initiierte Variablen.

1.3 Klassen

`klassenname {`

Innerhalb einer Klasse können Definitionen (siehe 1.2) stehen.

Konstrukturen können nach folgendem Muster definiert werden:

```
cons id (< type > parametername1 ( , < type > parametername2 , ... ) )  
{  
  expr 2.1 und/oder define 1.2  
}
```

`}`

1.4 Funktionsdefinition

Funktionen werden wie folgt definiert:

```
< type > parametername1 (< type > parametername2 ( , < type > id , ... ) )  
{  
  expr 2.1 und/oder define 1.2  
}
```

2 Low-Level

2.1 Expressions

2.1.1 Operationen

Operationen haben immer den gleichen Aufbau.

`term 2.2 Operand expr 2.1`

Operanten sind + , - , * , / , % , is , smaller und bigger

2.1.2 Funktionsaufrufe

Es gibt 3 Arten von Funktionsaufrufen:

call **call** *parameter1* (*parameter2* (, *id* , ...))

new **new** *parameter1* (*parameter2* (, *id* , ...))

destroy **destroy** *id*

2.1.3 Set

id = *term* ;

2.1.4 Terme

Für eine *expr* kann auch ein *term*_{2.2} eingesetzt werden.

2.2 Terme

expr (*expr*)

id Zeichenketten mit großen und kleinen Buchstaben sowie Sterne

number ganze Zahlen

null **null**

boolean **true** oder **false**