

## 1 STACK

Erweitern Sie die algebraische Spezifikation des Stacks um 2 weitere Funktionen...

### *Sorten*

*Integer, Boolean, Stack*

### *Signatur*

*new:  $\rightarrow$  Stack*

*push: Integer  $\times$  Stack  $\rightarrow$  Stack*

*pop: Stack  $\rightarrow$  Stack*

*top: Stack  $\rightarrow$  Integer*

*top2: Stack  $\rightarrow$  Integer*

*is\_empty: Stack  $\rightarrow$  Boolean*

*length: Stack  $\rightarrow$  Integer*

### *Axiome (Gleichungen)*

*top(push(e,s)) = e*

*top2(push(e,(push(f,s)))) = f*

*pop(push(e,s)) = s*

*is\_empty(new) = true*

*is\_empty(push(e,s)) = false*

*length(new) = 0*

*length(s) = 1 + length(pop(s))*

### *mit Fehlersituation:*

*top(new) = ERROR*

*top2(e, top(new)) = ERROR*

*top2(new) = ERROR*

*pop(new) = new (oder: ERROR)*

## 2 QUEUE

Erstellen Sie eine formale Spezifikation für eine Queue unter Verwendung der Z-Notation.

*Queue*  
 $queue\_elements : seq\ \mathbb{N}$

*create*  
 $\Delta Queue$   
 $queue\_elements' = \langle \rangle$

*append*  
 $\Delta Queue$   
 $elem? : \mathbb{N}$   
 $queue\_elements' = queue\_elements \frown elem?$

*getSecondLeast2*  
 $\exists Queue$   
 $elem2! : \mathbb{N}$   
 $\#queue\_elements \geq 2$   
 $elem2! = head\ (tail\ queue\_elements)$

*deleteSecondLeast2*  
 $\Delta Queue$   
 $\#queue\_elements \geq 2$   
 $queue\_elements' = head\ queue\_elements \frown tail\ (tail\ queue\_elements)$

*moveFirstToLast*  
 $\Delta Queue$   
 $\#queue\_elements \geq 1$   
 $queue\_elements' = tail\ queue\_elements \frown head\ queue\_elements$

*top2*  
 $top2 - OK \vee top2 - FAIL$

*top2 - OK*  
 $\exists Queue\ elem! : \mathbb{N}$   
 $\#queue\_elements \geq 2$   
 $elem! = head\ (tail\ queue\_elements)$

$top2 - FAIL$
$\exists Queue$
$r! : seq\ CHAR$
$\#queue\_elements \leq 1$
$r! = \text{"Queue has not enough elements"}$

$length$
$\exists Queue$
$n! : \mathbb{N}$
$n! = \#queue\_elements$

## 3 BENZINTANK

*Container**contents* :  $\mathbb{N}$ *capacity* :  $\mathbb{N}$  $contents \leq capacity$ *Indicator**light* : {*off*, *on*}*reading* :  $\mathbb{N}$ *danger\_level* :  $\mathbb{N}$  $light = on \Leftrightarrow reading \leq danger\_level$ *Indicator2**light2* : {*off*, *on*}*reading* :  $\mathbb{N}$ *danger\_level2* :  $\mathbb{N}$  $light = on \Leftrightarrow reading \leq danger\_level2$ *Storage\_tank**Container**Indicator**Indicator2* $reading = contents$  $capacity = 5000$  $danger\_level = 50$  $danger\_level2 = capacity * 0.95$ *half* $half\_OK \vee half\_EMPTY$ *half\_OK* $\Delta Storage\_tank$  $contents/2 > danger\_level$  $contents' = contents/2$ *half\_EMPTY* $\Delta Storage\_tank$  $contents < danger\_level$  $contents' = danger\_level$

$double\_OK \vee double\_OVERFLOW$
------------------------------------

$double\_OK$ $\Delta Storage\_tank$ $contents * 2 < danger\_level$ $contents' = contents * 2$
--

$double\_OVERFLOW$ $\Delta Storage\_tank$ $contents * 2 \geq danger\_level$ $contents' = danger\_level2$
---

## 4 RAUMSTATION COLUMBUS

[ACS\_ID, CMD]

# ACS\_ID  $\geq 110$

$max\_cmds\_per\_acs : \mathbb{N}$ $max\_cmds\_per\_acs \geq 50$
---

$Defined\_ACSS$ $commands : ACS\_ID \rightarrow seq\ CMD$ $enabled : \mathbb{P}\ ACS\_ID$ $\forall id : ACS\_ID \bullet 1 \leq \# commands(id) \leq max\_cmds\_per\_acs$
---

$Pending\_ACSS$ $Defined\_ACSS$ $queue : seq\ commands$ $pending : seq\ commands$ $current\_id : ACS\_ID$ $current\_cmds : seq_1\ CMD$ $queue = \diamond \Rightarrow pending = \diamond$ $queue \neq \diamond \Rightarrow queue = \langle current\_id, current\_cmds \rangle \frown pending$ $\#pending \geq 30$
--