Fig 1. Depicts List Specification extended with CircShift and Count, and GetByIndex

sort List Import ts INTEGER

Defines a list where elements are added at the end and removed from the front. The operations are Create, which brings an empty list into existence. Cons, which creates a new list with an added member, Length, which evaluates the list size, Head, which evaluates the front element of the list, Tail, which creates a list by removing the head from its input list, GetByIndex which gets an element from the list, based on its index, CircShift, which circularly shifts list by sending the first element on the end of the list, and Count, which counts a number of occurrences of given element in a list. Undefined represents an undefined value of type Elem.

```
Create -> List
Cons (List, Elem) -> List
Head (List) -> Elem
Length (List) -> Integer
Tail (List) -> List
GetByIndex (List, n) -> Elem
CircShift (List, n) -> List
Count (List, n) -> Integer
Head (Create) = undefined exception (empty list)
Head (Cons (L, v)) = if L = Create then v else Head (L)
Length (Create) = 0
Length (Cons (L, v)) = Length (L) + 1
Tail (Create) = Create
Tail (Cons (L, v)) = if L = Create then Create else Cons (Tail (L), v)
_GetByIndex (L, 1) = Head (L)
\_GetByIndex(L, n) = \_GetByIndex(Tail(L), n-1)
GetByIndex (L, n) = if n < 0 \mid \mid n > Length \mid then error else _GetByIndex(L, n)
CircShift(L, 0) = L
CircShift (L, n) = CircShift (Cons(Tail (L), Head (L)), n-1)
Count (L, n) = if Length (L) = 0 then 0 else _Count (L, n)
Count (L, n) = if GetByIndex(CircShift(L, 1), cont) = n then
                      num<- num+1 and cont <- cont + 1
```