Domain: $\mathcal{B}$ = {b | b is a Bag with elements of the type TElem}

Interface (set of operations):

init(b)

pre : true

post: b ∈ $\mathcal{B}$, b is an empty Bag

add(b, e)

pre: b ∈ $\mathcal{B}$, e ∈ TElem

post: b' ∈ $\mathcal{B}$, b' = b U {e} (Telem *e* is added to the Bag)

remove(b, e)

pre: b ∈ $\mathcal{B}$, e ∈ TElem

post: b' ∈ $\mathcal{B}$, b' = b \ {e} (one ocurrence of e was removed from the Bag).

$$\text{remove} \leftarrow \begin{cases} \text{true, if an element was removed } (\text{size}(b') < size(b)) \\ \text{false, if e was not present in b } (\text{size}(b') = \text{size}(b)) \end{cases}$$

search(b, e)

pre: b ∈ $\mathcal{B}$, e ∈ TElem

$$\text{post: } search \leftarrow \begin{cases} true, if\ e \in \mathbf{B} \\ false, otherwise \end{cases}$$

size(b)

pre: b ∈ $\mathcal{B}$

post: size ← $the\ number\ of\ elements\ from\ b$

nrOccurrences(b, e)

pre: b ∈ $\mathcal{B}$, e ∈ Telem

post: $nrOccurrences \leftarrow the\ number\ of\ occurrences\ of\ e\ in\ b$

destroy(b)

pre: b ∈ $\mathcal{B}$

post: b was destroyed

iterator(b, i)

pre: b ∈ $\mathcal{B}$

post: i ∈ $\mathcal{I}$, i is an iterator over b

ADT Iterator

- Has access to the interior structure (representation) of the Bag and it has a current element from the Bag.

Domain: $\mathcal{I}$ = {i | i is an iterator over b ∈ $\mathcal{B}$ }

Interface:

init(i, b)

pre: b ∈ $\mathcal{B}$

post: i ∈ $\mathcal{I}$, i is an iterator over b. i refers to the first element of b, or it is invalid if b is empty

valid(i)

pre: i ∈ $\mathcal{I}$

post: $valid \leftarrow \begin{cases} true, if\ the\ current\ element\ from\ i\ is\ a\ valid\ one \\ false, otherwise \end{cases}$

first(i)

pre: i ∈ $\mathcal{I}$

post: i' ∈ $\mathcal{I}$, the current element from i' refers to the first element from the bag or i is invalid if the bag is empty

next(i)

pre: i ∈ $\mathcal{I}$, valid(i)

post: i' ∈ $\mathcal{I}$, the current element from i' refers to the next element from the bag b.

throws: exception if i is not valid

getCurrent(i, e)

pre: i ∈ $\mathcal{I}$, valid(i)

post: e ∈ TElem, e is the current element from i

throws: exception if i is not valid

SORTED BAG

- These were the operations in the interface of the ADT Bag:
  - init(b)
  - add(b, e)
  - remove(b, e)
  - search(b, e)
  - nrOfOccurrences(b, e)
  - size(b)
  - iterator(b, it)
  - destroy

- Domain of Sorted Bag:
  - $\mathcal{SB} =$
    $\{\textbf{sb} | \textit{sb is a sorted bag that uses a relation to order the elements}\}$

- init (sb, rel)
  - **descr:** creates a new, empty sorted bag, where the elements will be ordered based on a relation
  - **pre:** *rel* $\in$ *Relation*
  - **post:** *sb* $\in \mathcal{SB}$, *sb* is an empty sorted bag which uses the relation *rel*