ADT List

- **init(l)**
- **element(l, p, e)**
- **position(l, e)**
- **modify(l, p, e)**
- **addFirst(l, e)**
- **addEnd(l, e)**
- **addAfter(l, p, e)**
- **addBefore (l, p, e)**
- **remove(l, p, e)**
- **search(l, e)** => *true / false*
- **isEmpty (l)**
- **size(l)**
- **destroy(l)**
- **iterator(l)** => **it**

Iterator

init, valid, getCurrent, next, first

ADT Sorted MultiMap

- **init (smm, R)**
  pre: R – relation on the set of all possible keys
  post: $smm \in SMM, smm = \phi$
- **destroy(smm)**
  pre: $smm \in SMM$
  post:smm was destroyed (allocated memory was freed)

- **add(smm, k, v) –** can be called put or insert
  pre:
  $smm \in SMM, k \in TKey, v \in TValue$
  post: the pair <k,v> was added into smm

- **remove(smm, k, v)**
  pre:
  $smm \in SMM, k \in TKey, v \in TValue$
  post: if the pair <k,v> is in smm
  remove = true
  smm' = smm without the pair <k,v> (the pair was deleted)
  else *remove = false*

- **search(smm, k, l)**
  pre: $smm \in SMM, k \in TKey$
  post: $l \in L$, *l is the list of values associated with k*
  empty list if k is not in smm

- **iterator(smm)** => **it**

ADT List

- **init(l)**
- **element(l, p, e)**
- **position(l, e)**
- **modify(l, p, e)**
- **addFirst(l, e)**
- **addEnd(l, e)**
- **addAfter(l, p, e)**
- **addBefore (l, p, e)**
- **remove(l, p, e)**
- **search(l, e)** => *true / false*
- **isEmpty (l)**
- **size(l)**
- **destroy(l)**
- **iterator(l)** => **it**

Iterator

init, valid, getCurrent, next, first

ADT Sorted MultiMap

- **init (smm, R)**
  pre: R – relation on the set of all possible keys
  post: $smm \in SMM, smm = \phi$
- **destroy(smm)**
  pre: $smm \in SMM$
  post:smm was destroyed (allocated memory was freed)

- **add(smm, k, v) –** can be called put or insert
  pre:
  $smm \in SMM, k \in TKey, v \in TValue$
  post: the pair <k,v> was added into smm

- **remove(smm, k, v)**
  pre:
  $smm \in SMM, k \in TKey, v \in TValue$
  post: if the pair <k,v> is in smm
  remove = true
  smm' = smm without the pair <k,v> (the pair was deleted)
  else *remove = false*

- **search(smm, k, l)**
  pre: $smm \in SMM, k \in TKey$
  post: $l \in L$, *l is the list of values associated with k*
  empty list if k is not in smm

- **iterator(smm)** => **it**