ADT Queue - Interface I

- The domain of the ADT Queue:
  $\mathcal{Q} = \{q | q \textit{ is a queue with elements of type } TElem\}$

- The interface of the ADT Queue contains the following operations:

ADT Queue - Interface II

- init(q)
  - **descr:** creates a new empty queue
  - **pre:** True
  - **post:** $q \in \mathcal{Q}$, q is an empty queue

ADT Queue - Interface III

- destroy(q)
  - **descr:** destroys a queue
  - **pre:** $q \in \mathcal{Q}$
  - **post:** q was destroyed

ADT Queue - Interface IV

- push(q, e)
  - **descr:** pushes (adds) a new element to the rear of the queue
  - **pre:** $q \in \mathcal{Q}$, e is a *TElem*
  - **post:** $q' \in \mathcal{Q}$, $q' = q \oplus e$, e is the element at the rear of the queue

ADT Queue - Interface V

- pop(q)
  - **descr:** pops (removes) the element from the front of the queue
  - **pre:** $q \in \mathcal{Q}$, q is not empty
  - **post:** $pop \leftarrow e$, e is a *TElem*, e is the element at the front of q, $q' \in \mathcal{Q}$, $q' = q \ominus e$
  - **throws:** an *underflow* exception if the queue is empty

ADT Queue - Interface VI

- top(q)
  - **descr:** returns the element from the front of the queue (but it does not change the queue)
  - **pre:** $q \in \mathcal{Q}$, q is not empty
  - **post:** $top \leftarrow e$, e is a *TElem*, e is the element from the front of q
  - **throws:** an *underflow* exception if the queue is empty

ADT Queue - Interface VII

- isEmpty(s)
  - **descr:** checks if the queue is empty (has no elements)
  - **pre:** $q \in \mathcal{Q}$
  - **post:**

$$isEmpty \leftarrow \begin{cases} true, \ if \ q \ has \ no \ elements \\ false, \ otherwise \end{cases}$$

ADT Queue - Interface VIII

- **Note:** queues cannot be iterated, so they do not have an *iterator* operation!