

- The ADT Queue represents a container in which access to the elements is restricted to the two ends of the container, called *front* and *rear*.
 - When a new element is added (pushed), it has to be added to the *rear* of the queue.
 - When an element is removed (popped), it will be the one at the *front* of the queue.
- Because of this restricted access, the queue is said to have a **FIFO** policy: First In First Out.
- The domain of the ADT Queue:

$$\mathcal{Q} = \{q \mid q \text{ is a queue with elements of type } TElem\}$$
- **init(q)**
 - **descr:** creates a new empty queue
 - **pre:** True
 - **post:** $q \in \mathcal{Q}$, q is an empty queue
- **destroy(q)**
 - **descr:** destroys a queue
 - **pre:** $q \in \mathcal{Q}$
 - **post:** q was destroyed
- **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
- **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

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

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

- **Note:** queues cannot be iterated, so they do not have an *iterator* operation!
- What data structures can be used to implement a Queue?
 - Static Array - for a fixed capacity Queue
 - In this case an *isFull* operation can be added, and *push* can also throw an exception if the Queue is full.
 - Dynamic Array
 - other data structures (will be discussed later)