

# WARTESCHLANGEN (QUEUES)

## Informell:

(Zwischen-) Speicher nach dem FIFO-Prinzip: First-In-First-Out-Listen

## Anwendung:

Pufferung zwischen "produzierendem" und "konsumierendem" Prozeß,  
→ Betriebssysteme

## als ADT:

Sei  $A$  eine gegebene Menge von Datenelementen. Es muß sichergestellt sein, daß der Elementtyp  $A$  die Operationen implementiert, die im ADT Elem definiert sind (s. Spezifikation des ADT List( $A$ )).

QUEUE( $A$ ) besteht aus:

- Menge der Listen  $L = \langle a_1, a_2, \dots, a_n \rangle$  (endl. Folgen) über  $A$ ;
- den folgenden Operationen (wir verwenden die Operationen des ADT List( $A$ ) für die Spezifikation der Queue-Operationen):

## ADT Queue( $A$ )

Sei  $q$  von Typ Queue( $A$ ):

*Queue()*: Konstruktor-Methode; wie bei allgemeinen Listen;

*clear()*, *boolean isEmpty()*: wie bei allgemeinen Listen;

*enqueue(Elem e)*: "hinten anstellen" ( $\rightarrow$  push());

(V) -

(N)  $q.enqueue(e) \equiv q.append(e)$

*Elem front()*: liefere "vorderstes" (als nächstes zu "konsumierendes") Element der Schlange  $q$  ( $\rightarrow$  top());

(V)  $\neg q.isEmpty()$

(N) Ist  $it = q.getIterator()$ , dann gilt  
 $q.front() = e$  mit  $it.first()$ ;  
 $e = q.retrieve(it)$

*dequeue()*: "abfertigen" ( $\rightarrow$  pop());

(V)  $\neg q.isEmpty()$

(N) Ist  $it = q.getIterator()$ , dann gilt  
 $q.dequeue(e) \equiv it.first()$ ;  
 $q = q.delete(it)$

**Anmerkung:** front() und dequeue() sind hier orthogonal zueinander eingeführt.

Die folgenden Operationen stellen sicher, daß Queue( $A$ ) selbst Implementierung des ADT Elem sein kann. Die Spezifikationen entsprechen denen der Operationen auf allgemeinen Listen.

|                                  |                         |
|----------------------------------|-------------------------|
| <i>Queue copy()</i> :            | Kopieroperation         |
| <i>boolean equals(Queue q)</i> : | Test auf "Gleichheit"   |
| <i>boolean less(Queue q)</i> :   | Test auf "Kleiner"      |
| <i>String toString()</i> :       | Konvertierungsoperation |