

Liste

Konstruktor	new Liste (): Liste
Vorher	./.
Nachher	Es wurde eine neue, leere Liste erzeugt. Die Schreib-/Lesemarke steht hinter der Liste, d.h. enthält kein vorhandenes aktuelles Element.
Auftrag	rein (Object neuesElement): void
Vorher	Es existiert eine Liste
Nachher	Das Objekt <i>neuesElement</i> wurde vor dem aktuellen Element (=Element, auf dem die Schreib-/Lesemarke steht) in die Liste eingefügt. <i>neuesElement</i> wurde zum aktuellen Element.
Anfrage	istLeer (): boolean
Vorher	Es existiert eine Liste
Nachher	Die Anfrage hat genau dann <i>true</i> geliefert, wenn die Liste keine Elemente enthielt/enthält.
Auftrag	zeige (): Object
Vorher	Es existiert eine nicht-leere Liste mit vorhandenem aktuellem Element (d.h. die Schreib-/Lesemarke steht auf einem vorhandenen Element -- <i>istLeer</i> und <i>istZuEnde</i> würden beide <i>false</i> liefern)
Nachher	Das aktuell ausgewählte Element wurde angegeben, die Liste nicht verändert.
Auftrag	raus (): Object
Vorher	Nicht-leere Liste mit vorhandenem aktuellem Element (<i>istLeer</i> , <i>istZuEnde</i> beide <i>false</i>)

(Fortsetzung: raus())	
Nachher	Das aktuell ausgewählte Element wurde angegeben und aus der Liste entfernt. Der Nachfolger des entfernten Elements wurde zum aktuellen Element (d.h. die Schreib-/Lesemarke wurde auf den [vorhandenen oder nicht-vorhandenen] Nachfolger des entfernten Elements gesetzt)
Auftrag	anDenAnfang (): void
Vorher	Es existiert eine Liste
Nachher	Das erste Element der Liste wurde zum aktuellen Element (d.h. die Schreib-/Lesemarke wurde auf das vorderste Element gesetzt. Nur bei einer leeren Liste ist dieses aktuelle Element nicht vorhanden [d.h. <i>istZuEnde</i> würde dann <i>true</i> liefern]).
Anfrage	istZuEnde (): boolean
Vorher	Es existiert eine Liste
Nachher	Es wurde genau dann <i>true</i> geliefert, wenn das aktuelle Element nicht existierte (d.h. die Schreib-/Lesemarke hinter dem letzten vorhandenen Listenelement steht)
Auftrag	weiter (): void
Vorher	Nicht-leere Liste mit vorhandenem aktuellem Element (<i>istLeer</i> und <i>istZuEnde</i> beide <i>false</i>)
Nachher	Der Nachfolger des aktuellen Elements wurde aktuell (d.h. die Schreib-/Lesemarke wurde auf den Nachfolger des aktuellen Elements gesetzt -- Das neue aktuelle Element existiert nicht, wenn zuvor das letzte vorhandene Element der Liste aktuell war).