



HOCHSCHULE ZITTAU-GÖRLIZ

FAKULTÄT FÜR INFORMATIK UND ELEKTROTECHNIK

Deque Dokumentation

Autoren:
Pascal OTTO
Michael ŠVANCAR

21. Dezember 2014

1 Deque

1.1 Einleitung

Deque gehört zum `java.util` Paket. Es implementiert das `Queue`-Interface. Elemente lassen sich im Gegensatz zu `Queue` an beiden Enden einfügen und entfernen.

1.2 Methodenbeschreibung

Tabelle 1: Deque Methoden

Rückgabotyp	Method Beschreibung
boolean	<code>add(E e)</code> Fügt ein spezifisches Element am Ende der Deque hinzu.
void	<code>addFirst(E e)</code> Fügt ein spezifisches Element am Anfang der Deque hinzu, falls die Kapazität ausreicht
void	<code>addLast(E e)</code> Fügt ein spezifisches Element am Ende der Deque hinzu, falls die Kapazität ausreicht
boolean	<code>contains(Object o)</code> Gibt true zurück wenn die Deque das spezifische Element beinhaltet
Iterator	<code>descendingIterator()</code> Gibt einen Iterator über die Elemente in der Deque in umgekehrter sequenzieller Reihenfolge zurück
Iterator	<code>iterator()</code> Gibt einen Iterator über die Elemente in der Deque zurück
E	<code>element()</code> Gibt das erste Element des Deques zurück, ohne es zu löschen
E	<code>getFirst()</code> Gibt das erste Element des Deques zurück, ohne es zu löschen

Tabelle 2: Deque Methoden

Rückgabotyp	Method Beschreibung
E	getLast() Gibt das letzte Element des Deques zurück, ohne es zu löschen
boolean	offer(E e) Fügt ein spezifisches Element am Ende der Deque hinzu und gibt bei Erfolg <i>true</i> zurück.
boolean	offerFirst(E e) Fügt ein spezifisches Element am Anfang des Deques hinzu.
boolean	offerLast(E e) Fügt ein spezifisches Element am Ende des Deques hinzu.
E	peekFirst() Gibt das erste Element des Deques zurück, ohne es zu löschen. Falls die Deque leer ist, gibt <i>null</i> zurück
E	peekLast() Gibt das letzte Element des Deques zurück, ohne es zu löschen. Falls die Deque leer ist, gibt <i>null</i> zurück
E	pollFirst() Gibt das erste Element zurück und löscht es aus der Deque. Falls die Deque leer ist, gibt <i>null</i> zurück
E	pollLast() Gibt das letzte Element zurück und löscht es aus der Deque. Falls die Deque leer ist, gibt <i>null</i> zurück
E	pop() Gibt das erste Element vom Stack, der mittels Deque repräsentiert wird, zurück
void	push(E e) Fügt ein Element dem Stack hinzu.
E	removeFirst() Gibt das erste Element zurück und löscht es aus der Deque
boolean	removeFirstOccurence(Object o) Löscht das erste Vorkommen des spezifische Element von der Deque
boolean	removeLastOccurence(Object o) Löscht das letzte Vorkommen des spezifische Element von der Deque
E	removeLast() Gibt das letzte Element zurück und löscht es aus der Deque
int	size() Gibt die Anzahl von Elementen in der Deque zurück

1.3 Effizienzanalyse

Graph einfügen und Effizienz ausrechnen. Für einzelne Probleme anwenden.

1.4 Schlussfolgerung

hier die resultierenden Stärken und Schwächen rausschreiben.