

1 LinkedList

1.1 Einleitung

Die `LinkedList` in Java 8 ist eine Umsetzung der `List` und `Deque` Schnittstellen als doppelt verknüpfte Liste. Sie implementiert alle optionalen Listenoperationen, und erlaubt alle Elemente (einschließlich `null`).

1.2 Methodenbeschreibung

- `add([index],e)` - fügt das spezifizierte Element am Ende der Liste / an der spezifizierten Position an
- `addAll([index], c)` - fügt die spezifizierte Collection am Ende der Liste / an der spezifizierten Position an
- `addFirst(e)` - fügt das spezifizierte Element am Anfang der Liste ein
- `addLast(e)` - fügt das spezifizierte Element am Ende der Liste ein
- `clear()` - entfernt alle Elemente aus der Liste
- `clone()` - erstellt eine flache Kopie der Liste
- `contains(o)` - gibt `true` zurück, wenn das spezifizierte Element in der Liste enthalten ist
- `element()` - gibt den Kopf der Liste zurück, ohne ihn zu entfernen
- `get(index)` - gibt das Element an der angegebenen Position zurück
- `getFirst()` - gibt das erste Element der Liste zurück
- `getLast()` - gibt das letzte Element der Liste zurück
- `remove([index])` - gibt das erste/spezifizierte Element in der Liste zurück und entfernt es
- `remove(o)` - entfernt das erste Auftreten des angegebenen Elements, sofern es vorhanden ist
- `size()` - gibt die Menge der Elemente in dieser Liste zurück
- `toArray()` - gibt ein Array mit allen Elementen der Liste in korrekter Reihenfolge zurück

1.3 Effizienzanalyse

Graph einfügen und Effizienz ausrechnen. Für einzelne Probleme anwenden.

1.4 Schlussfolgerung

hier die resultierenden Stärken und Schwächen rausschreiben.