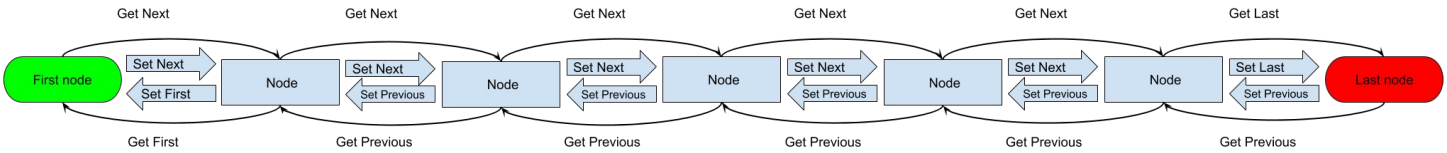
Double Linked List



Link zum Coden auf Github(auf GitHub aktuellste Version der Beschreibung):
https://github.com/pprugger/ALD-Gruppe5/tree/master/src/A03_DoubleLinkedList

Beschreibung einzelner Methoden:

public void add(T a)

1. überprüfung ob der first null.
2. wenn ja:
    a.) Erstelle neuen Node
    b.) Setze nächsten und vorherigen Node auf null
    c.) last und first sind an der gleichen Position
3. wenn nein:
    a.) Erstelle neue node
    b.) setze previous aus den aktuellen last
    c.) der next vom aktuellen last ist die neue Node
    d.) Der neue Last ist nun der neue Node

public void reset, public void reset to last

current auf first, bzw. last zeigen lassen

getlast

liefert last wenn last ungleich null

sonst null

next

1.check ob current schon gesetzt ist. wenn nicht: liefert null zurück

3. wenn ja current = current.next

4. liefert den Wert von current vor dem Weitersetzen zurück

previous

1.check ob current schon gesetzt ist. wenn nicht: liefert null zurück

2.wenn ja current = der vorherige vom current

3. Liefert den Wert von Current vor dem zurücksetzen zurück

moveNext

1. check ob current schon gesetzt ist. wenn nicht: return
2. check ob es ein Nachfolgeelement gibt. wenn nicht: return
3. current weitersetzen

movePrevious

1. check ob current schon gesetzt ist. wenn nicht: return
2. check ob es ein Vorgängerelement gibt. wenn nicht: return
3. current auf den Vorgänger setzen

getCurrent

1. wenn current ungleich null: return Wert von current
2. sonst: Wirf CurrentNotSetException

get

1. temporären Node für Startpunkt und counter mit Wert 1 initialisieren
2. durch die Liste bis zum gesuchten Element iterieren, abbrechen wenn Ende der Liste erreicht
3. Wert des gesuchten Nodes returnen

remove

1. temporären Node für Startpunkt und counter mit Wert 1 initialisieren
2. Überprüfen ob temp null ist, dann ist die Liste leer

3. Überprüfen ob first gleich last ist, dann ist nur mehr ein Element in der Liste und alle Nodes werden null gesetzt
4. Durch die Liste zum gesuchten Element interieren
   Sonderfälle:
5. Temp ist first: Setze first zum nächsten Element und den Vorgängerzeiger auf null
6. Temp ist last: Setze den Last auf Vorgängerelement und den Nachfolgezeiger auf null
7. Setze die Zeiger des Elements
8. Überprüfe ob Temp gleich current, wenn ja setze current gleich null

removeCurrent

1. Überprüfe ob current gesetzt ist, wenn nicht return
2. Überprüfe ob current gleich first und first gleich last ist, wenn ja dann ist nur mehr ein Element in der Liste und alle Nodes werden auf null gesetzt
   Sonderfälle:
3. Current ist first: Setze first ein Element weiter, setze den Vorgängerzeiger auf null
   Setze current eine Position weiter
4. Current ist last: Setze last ein Element zurück, setze den Nachfolgezeiger auf null
   Setze current eine Position zurück
5. Setze die Zeiger des Elements
6. Wenn es ein Nachfolgeelement gibt, setze current auf das Nachfolgeelement
   Sonst setze current auf das Vorgängerelement

insertAfterCurrentAndMove

1. Initialisiere einen neuen Node und einen temporären Node
2. Wenn Current gleich null return
3. Setze das temporäre Element auf das Nachfolgeelement von current
4. Setze die Zeiger des neuen Elements
5. Wenn das Nachfolgeelement des temporären Elements ungleich null ist, setze den Zeiger auf den Vorgänger
6. Setze das Nachfolgeelement von current auf das neue Element
7. Setze current auf das neue Element