Einführung in die Programmierung

by André Karge

Übung - Listen & Stacks & Queues

letzte Woche

- Packages
- Recursion

diese Woche

- Besprechung Übungsblatt 6
- Listen
- Stack
- Queue
- Besprechung Übungsblatt 7

Übungsblatt 6

Übungsblatt 6

Aufgaben 1-3

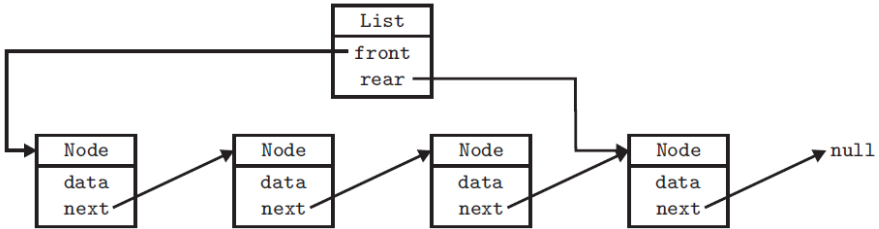
Code Beispiel

Listen

Listen

Beschreibung

- Arbeit mit Daten unbestimmter Länge / unterschiedlicher Länge
- Bessere Operationen im Vergleich zu Arrays (insert, delete, push, pop, ...)
- Ist ein Datenobjekt, welches Referenzen speichert



Listen

Beispiel

```
public class Node { // Container für Integer Listenelement
  private int data; // Inhalt eines Listenelements
  private Node next; // Referenz auf das nächste Element in der Liste
  //...
}
public class List {
  private Node front; // Referenz auf das erste Element in der Liste
  private Node rear; // Referenz auf das letzte Element in der Liste
  private int nodeCount; // Übersicht, wie viele Nodes wir haben
  public List() {
    this.front = null;
    this.rear = null;
    this.nodeCount = 0;
  }
}
```

Listen

Übungsaufgabe (15 min)

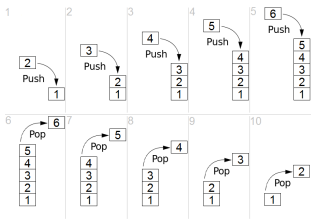
1. Schreiben Sie ihre eigene Klasse für eine einfach verkettete Liste für Integer (eine *List*-Klasse und eine *Node*-Klasse)
2. Achten sie auf korrekte Sichtbarkeit (public vs. private) der Attribute und Methoden
3. Schreiben Sie jeweils eine Methode um Nodes am Ende, zu einem bestimmten Index und am Anfang der Liste einzufügen
4. Schreiben Sie eine Methode, die ein Element anhand eines übergebenen Index' zurück gibt
5. Testen Sie die Liste in einer geeigneten main-Methode

Stack

Stack

Beschreibung

- Liste mit begrenzter Funktion
- wir haben in *Stack* immer nur Zugriff *Last Element*
- Hinzufügen und Entfernen arbeitet immer nur mit dem letzten Element
- LIFO Prinzip (Last In First Out)



https://techwelkin.com/differences-between-stack-and-queue

Stack

Beispiel

```
public class Stack {
  private Node top; // Link zum obersten Stack Element
  private int size;
  public Stack() {
    this.top = null;
    this.size = 0;
  }
  public void push(Node n) {
    if (this.top == null) {
      this.top = n; // Referenz speichern
    }
    else {
      n.setPrev(this.top); // Stackinterner Link zum vorher aktuellsten Element
      this.top = n; // Stackinterner Link zum aktuellsten Element
    }
    this.size++;
  }
  //...
}
```

Stack

Übungsaufgabe (15 min)

1. Kopieren Sie ihre *List*-Klasse aus der vorherigen Aufgabe
2. Ändern Sie die Klasse so, dass sie einen Stack abbildet
3. Testen Sie auch hier die Funktionen in einer main-Methode

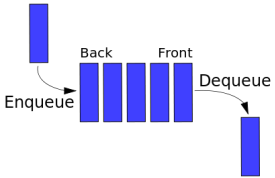
Hinweis: die Funktion pop() entfernt ein Element und die Funktion push() fügt ein Element hinzu

Queue

Queue

Beschreibung

- Wieder eine spezielle Form einer Liste
- *push()* und *pop()* haben hier andere Funktionen
- *enqueue()*: fügt ein Element von links der Queue hinzu
- *dequeue()*: entfernt ein Element am rechten Ende der Queue
- Funktionsweise: FIFO (First In First Out)



https://techwelkin.com/differences-between-stack-and-queue

Queue

Übungsaufgabe (15 min)

1. Kopieren Sie ihre *List*-Klasse aus der ersten Aufgabe
2. Ändern Sie die Klasse so, dass sie eine Queue abbildet
3. Testen Sie auch hier die Funktionen in einer main-Methode

Fragen?