

# Mosbach – Java 2

VL 03

# Interfaces

- Interface
  - technisch: Sammlung von (public) Methoden deklarationen  
(-köpfen  
-signaturen)
  - interface
    - semantisch Verwendung
      - contract Vertrag
      - Schnittstelle
- Implementierungsklasse
  - class . . . implements interface

# Interfaces

4

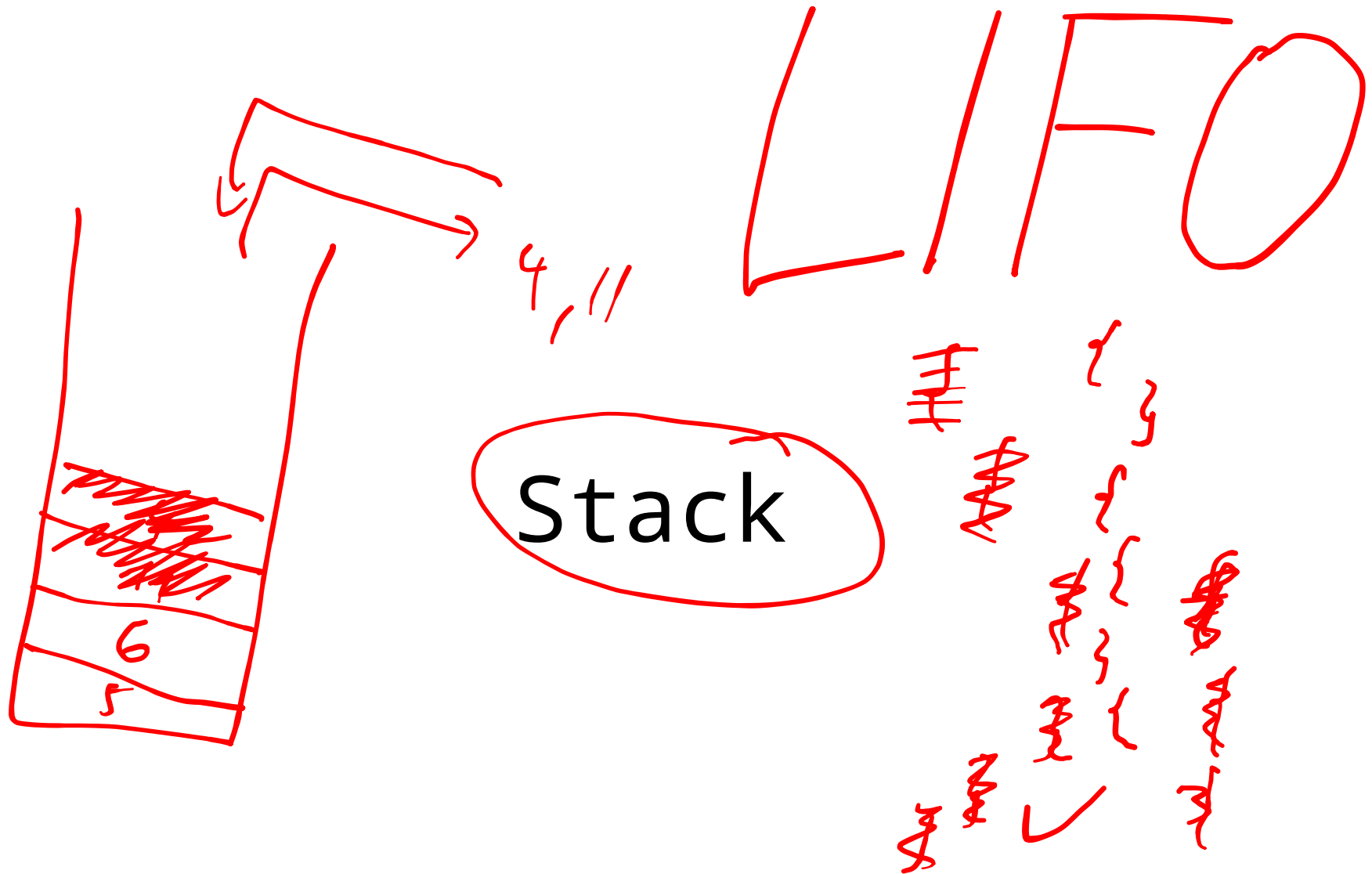
② StackImpl stack = new StackImpl();

- Always implement against the interface. (Warum?)

① Stack stack = new StackImpl();

stack  
stack = new StackImplImproved();  
stack ; "stack"






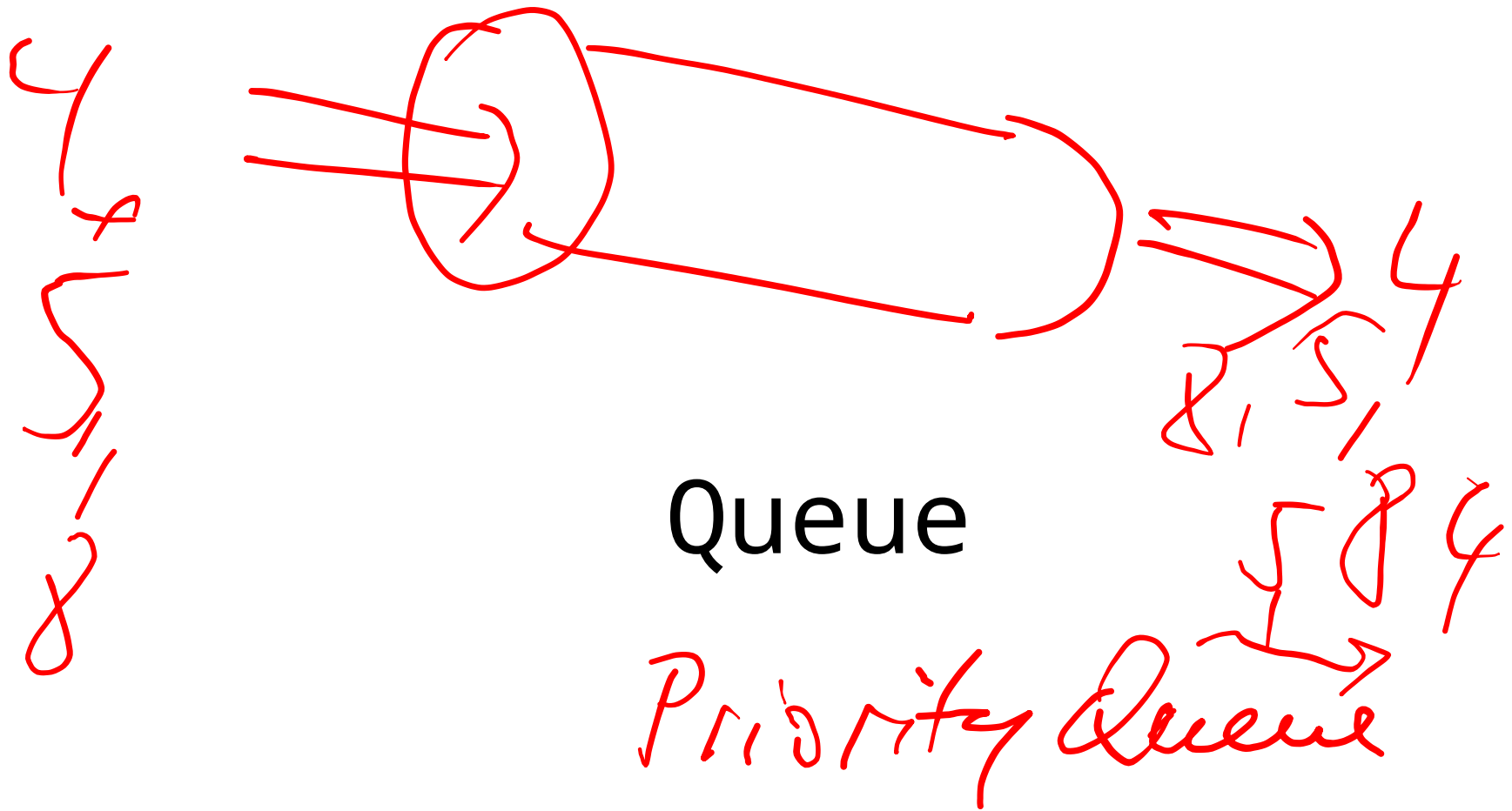
- Interface Stack

push( )  
pop()  
peek()  
clear()  
isEmpty()

- Implementierungsmöglichkeiten

ArrayList  
Array  
Dyn. Array ( )



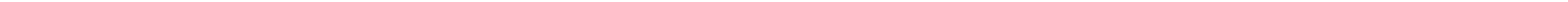




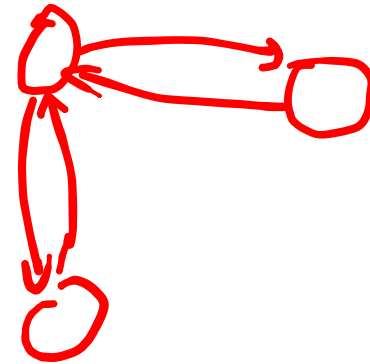
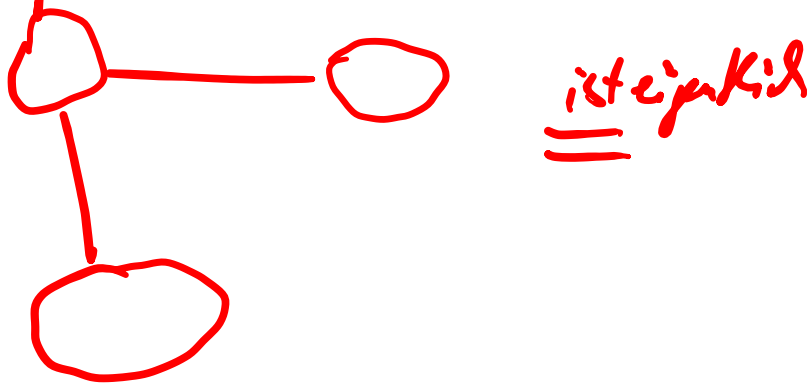
- Schreibe das interface
- Implementiere mit einer Klasse  
Deiner Wahl
- Teste

- Verallgemeinere Stack/Queue zu Storage

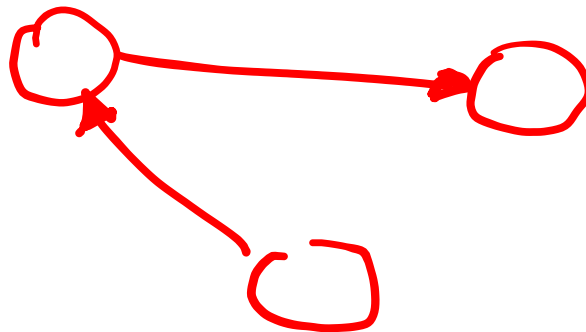
# LinkedLists



Graph  $G = (K_n, K_a)$



Gerichteter Graph  $G = (K_n, K_a)$

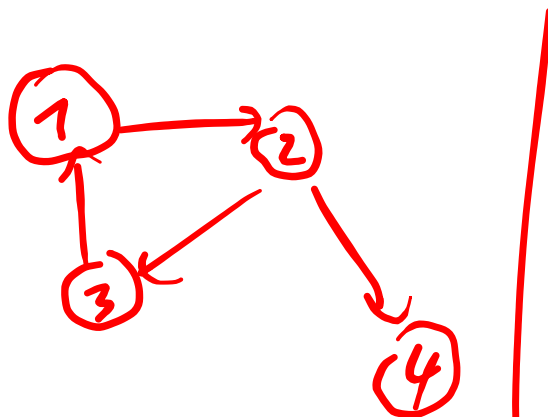


Baum = ein gerichteter Graph mit

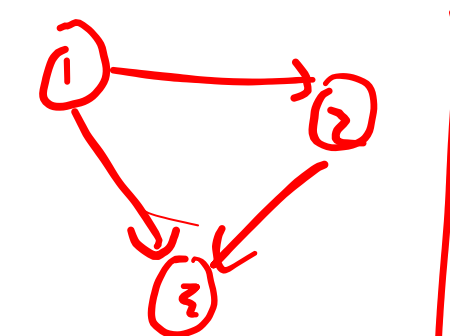
a) kein Zyklus

b) zusammenhängend

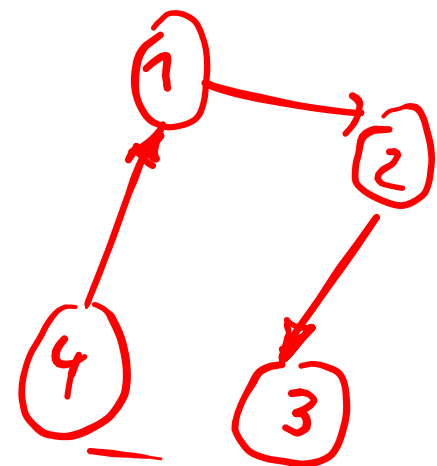
c) jeder Knoten max 1 Vater



I kein  
Zyklus



II kein  
3 hat 2  
Väter



III Baum

- LinkedLists
  - ein geeigneter Sortieralgorithmus
  - Anschauen aller Slides
- Aufgaben bei Zeit

- interface
  - Implementierung
  - Implement against the interface!
- Queue
  - Stack
  - LinkedList,  
DoubledLinkedList