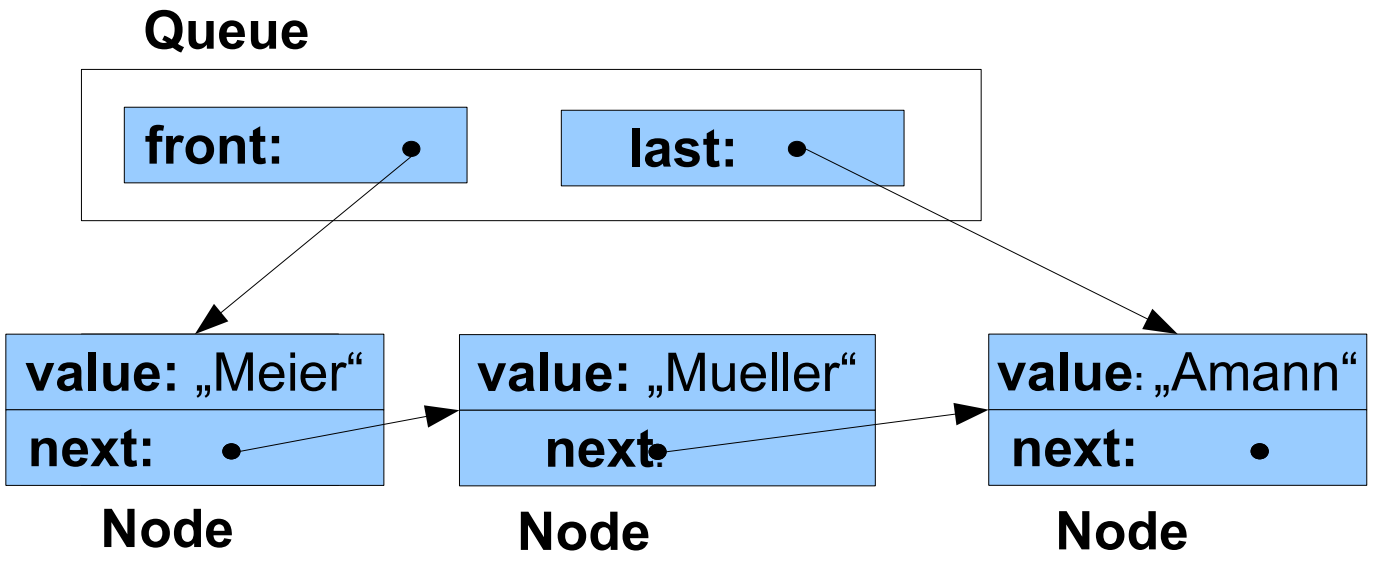
Lineare Datenstrukturen

„Wir schauen unter die Motorhaube.“

Lineare Datenstrukturen

- Wie funktioniert die Datenverwaltung innerhalb von Stack, Queue, List?
- Wie kriegt man es hin, dass diese Datenstrukturen „wachsen“ können?
- Wie baut man eine eigene lineare Datenstruktur, z.B.: „Ring“ ?

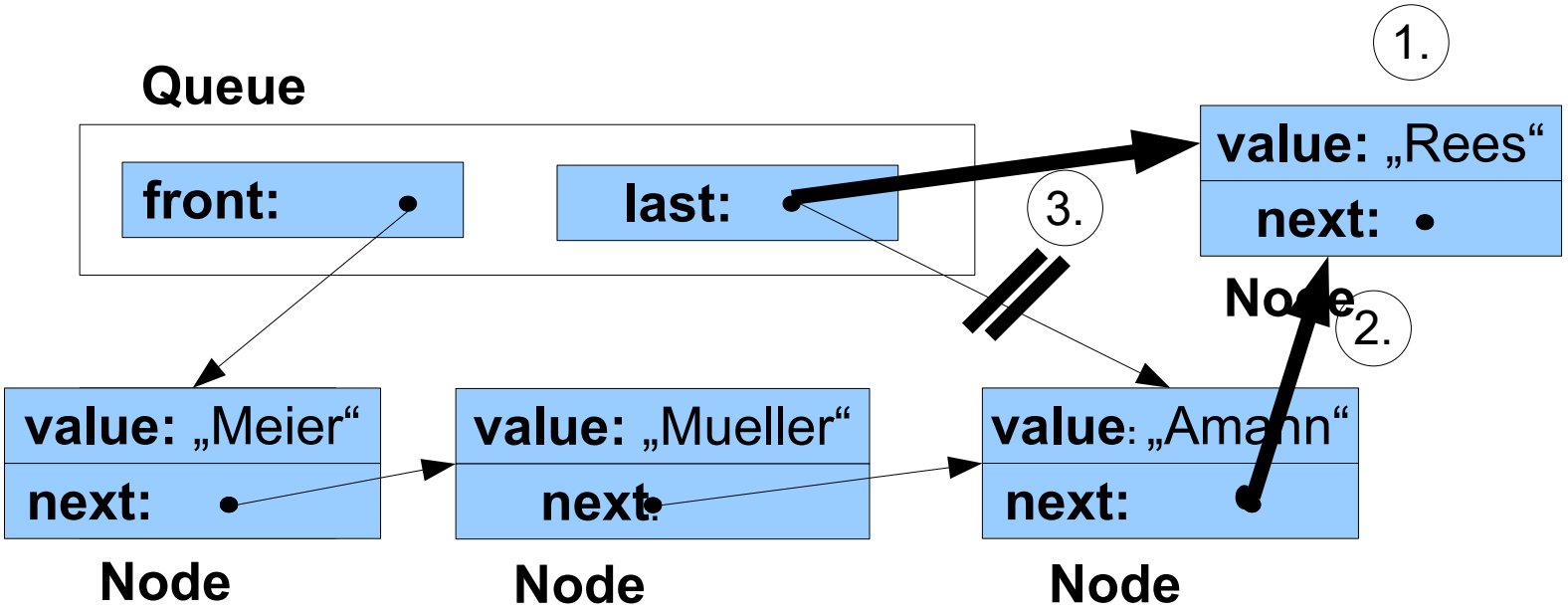
Queue: Objektdiagramm



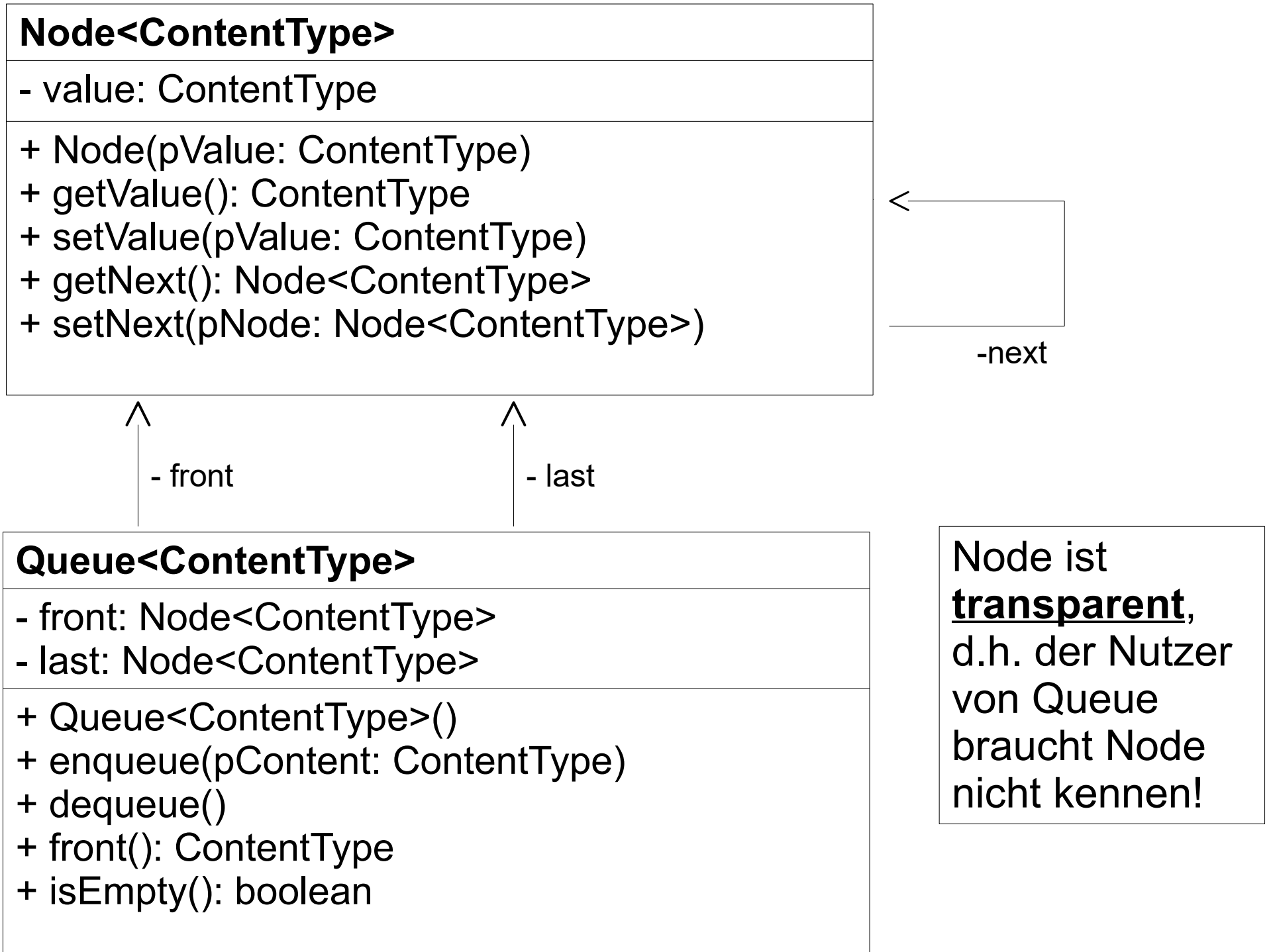
Darstellungsaufgaben

- Stelle dar, wie sich die Objektstruktur des Queues bei den folgenden Methoden ändert:
  - enqueue(“Rees”); *3 Schritte*
  - dequeue(); *1 Schritt*
- Betrachte für die Methoden enqueue und dequeue die folgenden Sonderfälle:
  - enqueue(“Rees”): *Sonderfall: Der Queue ist vorher leer.*
  - dequeue(): *Sonderfall 1: Der Queue enthält nur 1 Element*
    Sonderfall 2: Der Queue ist leer

enqueue


Queue
front:
last:
1.
value: „Rees“
next:
Node
2.
3.
value: „Meier“
next:
Node
value: „Mueller“
next:
Node
value: „Amann“
next:
Node

Implementationsdiagramm



Node ist **transparent**, d.h. der Nutzer von Queue braucht Node nicht kennen!

Java-Quelltext von Queue

```
public class Queue<ContentType>{
   private Node<ContentType> front;
   private Node<ContentType> last;
   …
   public void enqueue(ContentType pContent){
      //TODO
```

```
public void enqueue(ContentType pContent) {
1.  Node<ContentType> newNode =
                    new Node<>(pContent);

    //Sonderfall

    if (this.isEmpty()) {

        front = newNode;

        last = newNode;

        return;

    }

    //Standardfall

2.  last.setNext(newNode);
3.  last = newNode;

}
```