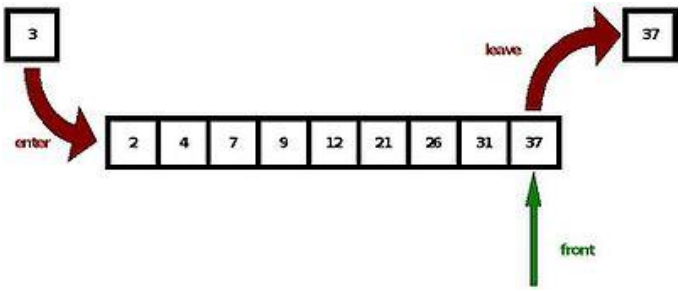
Die Klasse Queue in Java

Deklaration + Veranschaulichung mit GUI



Die generische Klasse Queue <ContentType>

```
public class Queue<ContentType> {

  private class QueueNode {

    private ContentType content = null;
    private QueueNode nextNode = null;

    public QueueNode(ContentType pContent) {
       content = pContent;
       nextNode = null;
    }

    public void setNext(QueueNode pNext) {
       nextNode = pNext;
    }

    public QueueNode getNext() {
       return nextNode;
    }

    public ContentType getContent() {
       return content;
    }
  }
```

Objekte der generischen Klasse **Queue** (Warteschlange) verwalten beliebige Objekte vom Typ **ContentType** nach dem **First-In-First-Out**-Prinzip, d.h., das zuerst abgelegte Objekt wird als erstes wieder entnommen.

*Ein Queue-Element wird durch die intern deklarierte Klasse **QueueNode** festgelegt – es besitzt einen **Inhalts**-Eintrag vom Typ ContentType sowie einen **Verweis** auf das nachfolgende Element (vom Typ QueueNode).*

```
  private QueueNode head;
  private QueueNode tail;

  public Queue() {
     head = null;
     tail = null;
  }

  public boolean isEmpty() {
     return head == null;
  }
```

```
  public void enqueue(ContentType pContent) {
     if (pContent != null) {
           QueueNode newNode = new
           QueueNode(pContent);
       if (this.isEmpty()) {
           head = newNode;
           tail = newNode;
       } else {
           tail.setNext(newNode);
           tail = newNode;
       }
     }
  }
```

```
public void dequeue() {
   if (!this.isEmpty()) {
         head = head.getNext();
         if (this.isEmpty()) {
            //head = null;
            tail = null;
         }
   }
}
```

```
public ContentType front() {
   if (this.isEmpty()) {
      return null;
   } else {
      return head.getContent();
   }
}
}
```

Das Projekt "Queue mit Gui (mit JavaFX)"

```
public class Controller {
   @FXML private TextField tfIsEmpty; … @FXML private Button btFront;
   private Queue<String> q;
   public void btQueue_click() {
      q = new Queue<String>();
      gibAus(); btIsEmpty.setDisable(false); …
   }
   public void btIsEmpty_click() {
      if (q.isEmpty()) tfIsEmpty.setText("ja - leer!");
         else tfIsEmpty.setText("nicht leer!");
   }
   public void btEnqueue_click() {
      q.enqueue(tfEnqueue.getText()); gibAus();
   }
   public void btDequeue_click() {
      q.dequeue(); gibAus();
   }
   public void btFront_click() {
      tfFront.setText("" + q.front());
   }
   public void gibAus() {
      //Ausgabe über Hilfsschlange
      Queue <String> q2 = new Queue <String> ();
      lv.getItems().clear();
      while (!q.isEmpty()){
         //vorderstes Element ausgeben + zwischenspeichern auf Hilfsschlange + löschen
         lv.getItems().add(q.front()); q2.enqueue(q.front()); q.dequeue();
      }
      //Queue rekonstruieren (und Hilfsschlange leeren)
      while (!q2.isEmpty()){
         q.enqueue(q2.front()); q2.dequeue();
      }
   }
}
```