**Lösung zu Aufgabe [A05] Breitensuche:**

**Klasse** *public List<Integer> getBreadthFirstOrder(Node<Integer> start)*

* Queue als Linked List und eine Arraylist wird erstellt
* Es wird die Start-Node in die Queue eingefügt
* Solange true ist
  + Hole die nächste Node aus der Queue und definiere sie als currentNode
  + Wenn kein Node mehr vorhanden ist, breche die Schleife ab
  + Frage ob die Node ein Linkes Kind Element hat
    - Wenn ja, dann dieses in die Queue einfügen
  + Frage ob die Node ein Rechtes Kind Element hat
    - Wenn ja, dann dieses in die Queue einfügen
  + Die currentNode in die Liste einfügen
* Die Liste mit den Nodes zurückgeben

**Klasse** *public List<Integer> getBreadthFirstOrderForLevel(Node<Integer> start, int level)*

* Queue als Linked List und eine Arraylist wird erstellt
* An currentLevel wird der NodeLevel vom Start Node übergeben
* An currentNode wird die Start Node übergeben
* Kontrollieren ob level und start gleich sind
  + Wenn ja, wird level um 1 erhöht
* Danach wird die currentNode in die Queue eingefügt
* Solange true ist
  + Hole die nächste Node aus der Queue und definiere sie als currentNode
    - Wenn es keine mehr gibt, dann break
  + Wenn das currentNode ein linkes Kind Element hat
    - Wenn das NodeLevel dieses Elements, kleiner oder gleich groß als das des currentNodes
      * Dann füge das Kind Element in die Queue ein
  + Das gleiche Spiel wiederholt sich beim Rechten Kind Element
  + Wenn der NodeLevel vom currentNode dem level aus dem Übergabe Parameter entspricht
    - Füge das currentNode in die Arrayliste ein
* Die Liste wird zurückgegeben