

# Durchlaufordnungen in Binärbäumen

Oft ist es notwendig, *alle* Knoten eines Baumes der Reihe nach zu besuchen. Solche Anwendungen sind zum Beispiel die Bestimmung der Anzahl Knoten eines Baumes oder seiner Höhe oder das Ausgeben aller Knoten des Baumes.

Bei linearen Datenstrukturen, wie zum Beispiel der verketteten Liste, können deren Elemente vom vordersten bis zum hintersten Element der Reihe nach durchlaufen werden.

Bei Bäumen sind verschiedene **Durchlaufordnungen** möglich. Die Aktion beim Besuchen eines Knotens ist dabei für verschiedene Anwendungen unterschiedlich und reicht vom einfachen Ausgeben des Knotens bis zu komplizierten Berechnungen für diesen Knoten.

Das Durchlaufen eines Baumes wird auch als **Traversierung** bezeichnet.

**Durchlaufordnung** Die Reihenfolge, in der die Knoten eines Baumes durchlaufen werden.

Für das Durchlaufen von **Binärbäumen** sind drei Reihenfolgen von besonderer Bedeutung:

die **Preorder-** (Hauptreihenfolge), die **Postorder-** (Nebenreihenfolge) und die **Inorder-Reihenfolge** (symmetrische Reihenfolge).

Die Bezeichnungen **Preorder**, **Postorder** und **Inorder** verdeutlichen, ob ein Knoten *vor*, *nach* oder *zwischen* seinen Teilbäumen besucht wird.

Die Reihenfolgen können auf einfache Weise **rekursiv** formuliert werden.

## Preorder-Reihenfolge (Hauptreihenfolge)

Bei dieser Reihenfolge wird ein Knoten vor seinen Kindern durchlaufen.

Bei der **Preorder-Reihenfolge** (Hauptreihenfolge) wird ein Knoten jeweils vor seinem linken und rechten Teilbaum durchlaufen.

Die **Preorder-Reihenfolge** wird also zuerst die **Wurzel** des Baumes besuchen. Danach wird *rekursiv* der **linke** und dann der **rechte Teilbaum** durchlaufen.

Das Durchlaufen eines Binärbaumes mit Wurzel  $v$  in der Preorder-Reihenfolge erfolgt in den folgenden Schritten:

1. Besuche den Knoten  $v$ .
2. Durchlaufe den linken Teilbaum des Knotens  $v$  in der Preorder-Reihenfolge.
3. Durchlaufe den rechten Teilbaum des Knotens  $v$  in der Preorder-Reihenfolge.

Beim Durchlaufen eines Baumes in der Preorder-Reihenfolge entsteht eine lineare Ordnung der Knoten, bei der die Elternknoten immer *vor* ihren Kindern besucht werden.

Der folgende Pseudocode zeigt den rekursiven Algorithmus **preorder()** für diese Durchlaufordnung.

```
1: algorithm preorder( $v$ )
2: { Durchläuft alle Knoten des Binärbaumes mit Wurzel  $v$  in der Preorder-
   Reihenfolge }
3: if  $v \neq \text{null}$  then
4:   Besuche den Knoten  $v$  { *}
5:   preorder( left( $v$ ) )
6:   preorder( right( $v$ ) )
7: end if
8: end algorithm
```

Bei der mit { \*} markierten Stelle im Algorithmus findet die eigentliche Bearbeitung des besuchten Knotens statt. Z.B. kann hier der Inhalt des Knotens ausgegeben werden. Die Preorder-Reihenfolge ist dann nützlich, wenn die Berechnung für einen Knoten vor der Berechnung seiner Kinder stattfinden muss.

## Postorder-Reihenfolge (Nebenreihenfolge)

Bei dieser Reihenfolge wird ein Knoten nach seinen Kindern durchlaufen.

Das Durchlaufen eines Binärbaumes in der **Postorder-Reihenfolge** (Nebenreihenfolge) erfolgt ähnlich zur Preorder-Reihenfolge.

Bei dieser Durchlaufordnung wird jedoch ein Knoten  $v$  *nach* seinem linken und rechten Teilbaum besucht.

1. Durchlaufe den linken Teilbaum des Knotens  $v$  in der Postorder-Reihenfolge.
2. Durchlaufe den rechten Teilbaum des Knotens  $v$  in der Postorder-Reihenfolge.
3. Besuche den Knoten  $v$ .

Die Postorder-Reihenfolge liefert beim Durchlaufen eines Baumes eine lineare Ordnung der Knoten, bei dem die Elternknoten immer *nach* ihren Kindern besucht werden.

Das Durchlaufen in der Posterorder-Reihenfolge ist für verschiedene Anwendungen nützlich. Zum Beispiel wenn für jeden Knoten  $v$  eine Eigenschaft berechnet werden soll, für deren Bestimmung aber zuerst die Eigenschaften der Kinder von  $v$  berechnet werden müssen.

## Inorder-Reihenfolge (Symmetrische Reihenfolge)

Die Preorder- und die Postorder-Reihenfolgen sind für alle Bäume anwendbar, die Inorder-Reihenfolge ist jedoch nur für Binärbäume definiert.

Bei dieser Reihenfolge wird ein Knoten zwischen seinen Kindern durchlaufen.

Bei der **Inorder-Reihenfolge** (symmetrischen Reihenfolge) wird ein Knoten  $v$  *zwischen* dem Durchlaufen seines linken und rechten Teilbaumes besucht.

1. Durchlaufe den linken Teilbaum des Knotens  $v$  in der Inorder-Reihenfolge.
2. Besuche den Knoten  $v$ .
3. Durchlaufe den rechten Teilbaum des Knotens  $v$  in der Inorder-Reihenfolge.

Beim Durchlaufen eines Binärbaumes in der **Inorder-Reihenfolge** ergibt sich eine lineare Ordnung der Knoten. In dieser Ordnung erscheint ein Knoten immer *zwischen* seinen Kindern. Die Knoten des Binärbaumes werden quasi von links nach rechts besucht.