

## 4.2 Nicht-lineare Strukturen

### Die Klasse **BinaryTree<ContentType>**

Mithilfe der generischen Klasse **BinaryTree** können beliebig viele Objekte vom Typ **ContentType** in einem Binärbaum verwaltet werden. Ein Objekt der Klasse stellt entweder einen leeren Baum dar oder verwaltet ein Inhaltsobjekt sowie einen linken und einen rechten Teilbaum, die ebenfalls Objekte der generischen Klasse **BinaryTree** sind.

### Dokumentation der Klasse **BinaryTree<ContentType>**

**Konstruktor** **BinaryTree<ContentType>()**

Nach dem Aufruf des Konstruktors existiert ein leerer Binärbaum.

**Konstruktor** **BinaryTree<ContentType>(ContentType pContent)**

Wenn der Parameter `pContent` ungleich `null` ist, existiert nach dem Aufruf des Konstruktors der Binärbaum und hat `pContent` als Inhaltsobjekt und zwei leere Teilbäume. Falls der Parameter `null` ist, wird ein leerer Binärbaum erzeugt.

**Konstruktor** **BinaryTree<ContentType>(ContentType pContent, BinaryTree<ContentType> pLeftTree, BinaryTree<ContentType> pRightTree)**

Wenn der Parameter `pContent` ungleich `null` ist, wird ein Binärbaum mit `pContent` als Inhaltsobjekt und den beiden Teilbäumen `pLeftTree` und `pRightTree` erzeugt. Sind `pLeftTree` oder `pRightTree` gleich `null`, wird der entsprechende Teilbaum als leerer Binärbaum eingefügt. Wenn der Parameter `pContent` gleich `null` ist, wird ein leerer Binärbaum erzeugt.

**Anfrage** **boolean isEmpty()**

Diese Anfrage liefert den Wahrheitswert `true`, wenn der Binärbaum leer ist, sonst liefert sie den Wert `false`.

**Auftrag** **void setContent(ContentType pContent)**

Wenn der Binärbaum leer ist, wird der Parameter `pContent` als Inhaltsobjekt sowie ein leerer linker und rechter Teilbaum eingefügt. Ist der Binärbaum nicht leer, wird das Inhaltsobjekt durch `pContent` ersetzt. Die Teilbäume werden nicht geändert. Wenn `pContent` `null` ist, bleibt der Binärbaum unverändert.

**Anfrage** **ContentType getContent()**

Diese Anfrage liefert das Inhaltsobjekt des Binärbaums. Wenn der Binärbaum leer ist, wird `null` zurückgegeben.

- Auftrag**      **void setLeftTree(BinaryTree<ContentType> pTree)**  
Wenn der Binärbaum leer ist, wird pTree nicht angehängt.  
Andernfalls erhält der Binärbaum den übergebenen Baum als linken Teilbaum. Falls der Parameter null ist, ändert sich nichts.
- Auftrag**      **void setRightTree(BinaryTree<ContentType> pTree)**  
Wenn der Binärbaum leer ist, wird pTree nicht angehängt.  
Andernfalls erhält der Binärbaum den übergebenen Baum als rechten Teilbaum. Falls der Parameter null ist, ändert sich nichts.
- Anfrage**      **BinaryTree<ContentType> getLeftTree()**  
Diese Anfrage liefert den linken Teilbaum des Binärbaumes. Der Binärbaum ändert sich nicht. Wenn der Binärbaum leer ist, wird null zurückgegeben.
- Anfrage**      **BinaryTree<ContentType> getRightTree()**  
Diese Anfrage liefert den rechten Teilbaum des Binärbaumes. Der Binärbaum ändert sich nicht. Wenn der Binärbaum leer ist, wird null zurückgegeben.