Die Klasse TOrderedTree

Objekte der Klasse *TOrderedTree* verwalten Objekte in einem geordneten Binärbaum für den gilt, dass alle Knoteninhalte im linken Unterbaum kleiner sind als der Wurzelinhalt und alle Knoteninhalte im rechten Teilbaum größer sind als der Wurzelinhalt.
Diese Bedingung gilt auch in allen Unterbäumen. Die Knoteninhalte sind Objekte einer Unterklasse von *TItem*, in der durch Überschreiben der drei Vergleichsmethoden *isLower, isGreater, isEqual* (s. *TItem*) eine eindeutige Ordnungsrelation festgelegt werden muss.

Dokumentation der Methoden Klasse TOrderedTree

Konstruktor `create`

nachher Der geordnete Binärbaum existiert und ist leer.

Anfrage `isEmpty: Boolean`
`// ist eigentlich unnötig, da in TBinTree definiert`

nachher Diese Anfrage liefert den Wahrheitswert ***true***, wenn der geordnete Baum leer ist, sonst liefert sie den Wert ***false***.

Auftrag `insertItem (pItem: TItem)` `// besser Anfrage ... : TOrderedTree`

nachher *pItem* ist entsprechend der Ordnungsrelation in den Baum eingeordnet.

Anfrage `searchItem (pItem: TItem):TItem`

nachher Falls ein bezüglich der verwendeten Ordnungsrelation mit *pItem* übereinstimmendes Objekt im geordneten Baum enthalten ist, liefert die Anfrage dieses, ansonsten wird *nil* zurückgegeben.

Auftrag `deleteItem (pItem: TObject)` `// besser Anfrage ... : TOrderedTree`

nachher Falls ein bezüglich der verwendeten Ordnungsrelation mit *pItem* übereinstimmendes Objekt im Baum enthalten war, wurde dieses entfernt. Das Inhaltsobjekt des gelöschten Knotens wurde nicht freigegeben.

Anfrage `getSortedList: TList`

nachher Die Knoteninhalte des geordneten Binärbaums werden als sortierte Liste zurückgegeben. Ist der geordnete Baum leer, wird *nil* zurückgegeben.

Destruktor `destroy`
`// ist eigentlich unnötig, da in TBinTree definiert`

nachher Der sortierte Binärbaum existiert nicht mehr.