

DER DATENTYP BINÄRBAUM – SCHNITTSTELLE

```

{ ----- }
{ }
{ Die Unit stellt einen inhaltslosen Datentypen TBinBaum zur Verfügung. Um einen Binärbaum }
{ mit Inhalt zu erhalten, muss eine hiervon abgeleitete Klasse erstellt werden, z. B.: }
{ }
{ type TIntBinBaum = class(TBinBaum) }
{   public }
{     Wurzelinhalt: integer; }
{     constructor Create(w: Integer); }
{     function WurzelinhaltToString: string; override; }
{   end; }
{ }
{ Ein Binärbaum kann entweder leer sein (NIL) oder eine Wurzel mit zwei evtl. leeren }
{ Teilbäumen rtb, ltb }
{ }
{ ----- Tool zum Binärbaum, (C) 2001 D. Garmann - }

```

unit Binbaum;

interface

type

```

TBinBaum = class
  private
    ltb, rtb: TBinBaum;

  public
    constructor Create;
    { Erzeugt ein Blatt, d.h. "leeres Element mit rtb = ltb = NIL }
    destructor Destroy;
    { Zerstört das Element inklusive seiner Teilbäume }
    function Leer: boolean;
    { Liefert "true" wenn der Baum leer ist, d. h. der Baum ist = NIL }
    function WurzelinhaltToString: string; virtual; abstract;
    { Abstrakte Methode zur Darstellung des Inhalts auf dem Bildschirm (Editfeld, etc.) }

    function LinkerTeilbaum: TBinBaum;
    { Liefert den Linken Teilbaum im Funktionsergebnis zurück. }
    function RechterTeilbaum: TBinBaum;
    { Liefert den Rechten Teilbaum im Funktionsergebnis zurück. }

    procedure FuegeTeilbaumLinksAn(tb: TBinBaum);
    { Fügt einen Teilbaum an das aktuelle Bauelement links an. }
    { VORSICHT! Ein eventuell vorhandener linker Teilbaum geht verloren! }
    procedure FuegeTeilbaumRechtsAn(tb: TBinBaum);
    { Fügt einen Teilbaum an das aktuelle Bauelement rechts an. }
    { VORSICHT! Ein eventuell vorhandener rechter Teilbaum geht verloren! }

    function LinksLeer: Boolean;
    { Liefert "true" wenn der linke Teilbaum leer ist, d. h. ltb = NIL }
    { VORSICHT! Ein leerer Baum hat keinen linken Teilbaum => Laufzeitfehler }
    function RechtsLeer: Boolean;
    { Liefert "true" wenn der rechte Teilbaum leer ist, d. h. rtb = NIL }
    { VORSICHT! Ein leerer Baum hat keinen rechten Teilbaum => Laufzeitfehler }
    function IstBlatt: Boolean;
    { Liefert "true", wenn das aktuelle Element leere Teilbäume hat. }

    function ErsetztDurchLinkenTeilbaum: TBinBaum;
    { Liefert den Linken Teilbaum im Funktionsergebnis zurück und LÖSCHT den Rest. }
    function ErsetztDurchRechtenTeilbaum: TBinBaum;
    { Liefert den Rechten Teilbaum im Funktionsergebnis zurück und LÖSCHT den Rest. }
end;

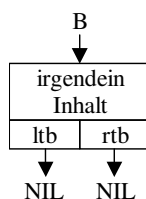
```

Grafisch lässt sich dieser Datentyp wie folgt darstellen: **var** B: TBinBaum;

leerer Baum



Blatt



beliebiger Baum

