

DER DATENTYP AVL-BAUM – SCHNITTSTELLE

```
----- }
{ }
{ Die Unit stellt einen inhaltslosen Datentypen TAVLbaum zur Verfügung. Um einen AVL-Baum mit Inhalt zu erhalten, }
{ muss eine hiervon abgeleitete Klasse erstellt werden, z. B.: }
{ }
{ type TintAVLbaum = class(TAVLbaum) }
{   public }
{     Wurzelinhalt: integer; }
{     constructor Create(w: Integer); }
{     function equal(e: TSuchbaum): boolean; override; // Diese stammt noch von TSuchbaum }
{     function higher(e: TSuchbaum): boolean; override; // Diese ebenfalls }
{     function WurzelinhaltToString: string; override; }
{   end; }
{ }
{ Die Routinen equal (gleich) und higher (größer) dienen zur Einsortierung in den Suchbaum. }
{ }
{ ----- Tool zum AVL-Baum, (C) 2002 D. Garmann - }

unit Avlbaum;

interface

uses ExtCtrls, suchbaum;

type TAVLbaum=class(TSuchbaum)
  private
    Hoehe: integer;
    { Damit die Höhe nicht immer rekursiv berechnet werden muss, wird sie lokal in jedem Knoten gespeichert }

    function IsoliereGroesstesElement(var grElem: TAVLbaum): TAVLbaum;
    { Liefert den AVLbaum ohne das groesste Element. Dieses steht abgekapselt in grElement }
    function EchtLoeschen: TAVLbaum;
    { Liefert den neuen AVLbaum ohne das zuvor existierende (!!!) Suchelement e }
    function AVLKorrektur: TAVLbaum;
    { Liefert den - falls nötig - höhenbalancierten AVLbaum }
    procedure Aktualisiere_Hoehe;
    { Aktualisiert die Variable Hoehe }

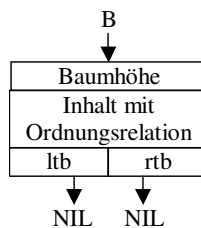
  public
    constructor create;
    function Balance: integer;
    { Liefert die Balance (rechterTeilbaum.Hoehe - linkerTeilbaum.Hoehe) }
    function BaumHoehe: integer;
    { Liefert den Wert der privaten Variablen Hoehe }
    function AVLEinfuegen(e: TAVLbaum; var ExistiertBereits: boolean): TAVLbaum;
    { Liefert den neuen AVLbaum mit dem eingefügten Suchelement }
    function AVLloeschen(e: TAVLbaum; var Existiert: boolean): TAVLbaum;
    { Liefert den neuen AVLbaum ohne das eventuell vorhandene Suchelement e }
    { war es nicht vorhanden => Existiert = false }
end;
```

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

leerer AVLbaum



Blatt



beliebiger AVLbaum

