

ALGORITHMEN FÜR DAS BEISPIEL ZUM BINÄRBAUM: TIERERATEN

Objektorientierte Datenstruktur zum TWissensbaum:

```

TInhalt = string[50];
TWissensbaum = class (TBinBaum)
  private
    Inhalt: TInhalt;
    procedure lerne_neuen_Begriff;
  public
    constructor create(i: TInhalt);
    function WurzelinhaltToString: string; override;
    procedure durchlaufe_die_Wissensstruktur_und_lerne;
end;
    
```

ALGORITHMUS <i>Durchlaufe_Wissensstruktur_und_Lerne</i>	
Hilfsobjekte: Antwort: „ja“ oder „nein“	
<ul style="list-style-type: none"> • Falls <i>not IstBlatt</i> { also kein Tiername } <li style="padding-left: 20px;">dann <ul style="list-style-type: none"> • Stelle Frage <i>Inhalt</i> + „?“ { Inhalt = Frage } • Antwort ← <i>Benutzereingabe</i> { „ja“ oder „nein“ } • Falls Antwort = „ja“ <ul style="list-style-type: none"> • <i>LinkerTeilbaum.Durchlaufe_die_Wissensstruktur_und_Lerne</i> • <i>RechterTeilbaum.Durchlaufe_die_Wissensstruktur_und_Lerne</i> sonst <ul style="list-style-type: none"> • Stelle Frage „Ist es ein“ + <i>Inhalt</i> + „?“ { Inhalt = Tiername } • Antwort ← <i>Benutzereingabe</i> { „ja“ oder „nein“ } • Falls Antwort = „ja“ <ul style="list-style-type: none"> • FERTIG { , da gefunden } • <i>Lerne_neuen_Begriff</i> 	

ALGORITHMUS <i>Lerne_neuen_Begriff</i>	
Hilfsobjekte: JA_Wissen, NEIN_Wissen: TWissensBaum	
neuer_Tiername, alter_Tiername,	
Unterscheidungsfrage: TInhalt	
<ul style="list-style-type: none"> • alter_Tiername ← <i>Inhalt</i> { String } • neuer_Tiername ← <i>Benutzereingabe</i> { String } • Unterscheidungsfrage ← <i>Benutzereingabe</i> { String } • <i>Inhalt</i> ← Unterscheidungsfrage • JA_Wissen ← TWissensbaum.Create(neuer_Tiername) • NEIN_Wissen ← TWissensbaum.Create(alter_Tiername) • <i>FuegeTeilbaumLinksAn</i> (JA_Wissen) • <i>FuegeTeilbaumRechtsAn</i> (NEIN_Wissen) 	