ALGORITHMEN ZUM SUCHBAUM

```
TSuchBaum = class(TBinBaum)
  private
    function IsoliereGroesstesElement(var grElem: TSuchbaum): TSuchbaum;
    function EchtLoeschen: TSuchbaum;
  public
    function equal(e: TSuchbaum): boolean; virtual; abstract;
    function higher(e: TSuchbaum): boolean; virtual; abstract;
    {... und die übrigen Ordnungsrelationen ...}
    function SuchEinfuegen(e: TSuchbaum; var ExistiertBereits: boolean): TSuchbaum;
    function SuchLoeschen(e: TSuchbaum; var Existiert: boolean): TSuchbaum;
end;
```

ALGORITHMUS	***SuchEinfuegen***		
Outputobjekte:	SuchEinfuegen:	TSuchBaum	*{ als Funktionsergebnis }*
	ExistiertBereits:	wahr oder falsch	
Inputobjekte:	Suchelement:	TSuchBaum	*{ natürlich ein Blatt }*
Hilfsobjekte:	Teilbaum:	TSuchBaum	

- Falls *Leer* ist,
  - dann
    - *SuchEinfuegen* ← Suchelement
    - ExistiertBereits ← falsch
  - sonst
    - Falls *Equal(* Suchelement *)*,
      - dann
        - ExistiertBereits ← wahr
        - *SuchEinfuegen* ← *Self*
      - sonst
        - Falls *Higher(* Suchelement *)*,
          - dann
            - Teilbaum ← *LinkerTeilbaum*
            - Teilbaum.***SuchEinfuegen(Suchelement, ExistiertBereits)***
            - *FuegeTeilbaumLinksAn (* Teilbaum *)*
            - *SuchEinfuegen* ← *Self*
          - sonst
            - Teilbaum ← *RechterTeilbaum*
            - Teilbaum.***SuchEinfuegen(Suchelement, ExistiertBereits)***
            - *FuegeTeilbaumRechtsAn (* Teilbaum *)*
            - *SuchEinfuegen* ← *Self*

ALGORITHMUS ***SuchLoeschen***

Outputobjekte:	SuchLoeschen:	TSuchBaum	*{ veränderter Suchbaum }*
	Existiert:	wahr oder falsch	
Inputobjekte:	Suchelement:	TSuchBaum	*{ natürlich ein Blatt }*
Hilfsobjekte:	Teilbaum:	TSuchBaum	

- Falls *Leer* ist,
  - dann
    - *SuchLoeschen* ← *Self*
    - Existiert ← falsch
  - sonst
    - Falls *Equal(* Suchelement *)*,
      - dann
        - *SuchLoeschen* ← ***Echtloeschen***
        - Existiert ← wahr
      - sonst
        - Falls *Higher(* Suchelement *)*,
          - dann
            - Teilbaum ← *LinkerTeilbaum*
            - Teilbaum.***SuchLoeschen(Suchelement, Existiert)***
            - *FuegeTeilbaumLinksAn (* Teilbaum *)*
            - *SuchLoeschen* ← *Self*
          - sonst
            - Teilbaum ← *RechterTeilbaum*
            - Teilbaum.***SuchLoeschen(Suchelement, Existiert)***
            - *FuegeTeilbaumRechtsAn (* Teilbaum *)*
            - *SuchLoeschen* ← *Self*

ALGORITHMUS ***EchtLoeschen***

Outputobjekte:	EchtLoeschen:	TSuchBaum	*{ veränderter Suchbaum }*
Hilfsobjekte:	Teilbaum:	TSuchBaum	
	GroesstesElement	TSuchbaum	*{ größtes El. im linken Teilb. }*

- Falls *RechtsLeer* ist,
  - dann
    - *EchtLoeschen* ← *ErsetztDurchLinkenTeilbaum*
  - sonst
    - Falls *LinksLeer*,
      - dann
        - *EchtLoeschen* ← *ErsetztDurchRechtenTeilbaum*
      - sonst
        - Teilbaum ← *LinkerTeilbaum.*
        - Teilbaum ← Teilbaum.***IsoliereGroesstesElement(GroesstesElement)***
        - *FuegeTeilbaumLinksAn (* Teilbaum *)*
        - GroesstesElement.*FuegeTeilbaumLinksAn(LinkerTeilbaum)*
        - GroesstesElement.*FuegeTeilbaumRechtsAn(RechterTeilbaum)*
        - *FuegeTeilbaumLinksAn(* NIL *)*
        - *FuegeTeilbaumRechtsAn(* NIL *)*
        - *Free*
        - *EchtLoeschen* ← GroesstesElement

ALGORITHMUS ***IsoliereGroesstesElement***

Outputobjekte:	IsoliereGroesstesElement:	TSuchBaum	*{ veränderter Suchbaum }*
	GroesstesElement:	TSuchBaum	*{ natürlich ein Blatt }*
Hilfsobjekte:	Teilbaum:	TSuchBaum	

- Falls *RechtsLeer* ist,
  - dann
    - *IsoliereGroesstesElement* ← *LinkerTeilbaum*
    - GroesstesElement ← *Self*
  - sonst
    - Teilbaum ← *RechterTeilbaum*
    - Teilbaum ← Teilbaum.***IsoliereGroesstesElement(GroesstesElement)***
    - *FuegeTeilbaumRechtsAn (* Teilbaum *)*
    - *IsoliereGroesstesElement* ← *Self*