

sum.strukturen

Class Baum<Typ>

[java.lang.Object](#)



All Implemented Interfaces:
[Serializable](#)

Direct Known Subclasses:
[Suchbaum](#)

```
public class Baum<Typ>
extends Object
implements Serializable
```

Die Klasse Baum ist ein Binärbaum nach der Schnittstelle aus dem Buch "Informatik mit Java".
Er kann als Baum mit einem Inhaltstyp deklariert werden. Wird dieser weggelassen, liefert die Anfrage inhalt ein Object und es wird Typkonversion nötig.

Version:
7.3 vom 11.10.2011
Author:
Bernard Schriek
See Also:
[Serialized Form](#)

Field Summary	
(package private) static long	serialVersionUID

Constructor Summary	
Baum (Typ pInhalt)	Ein neuer Baum mit leeren Teilbäumen wird erzeugt
Baum (Typ pInhalt, Baum<Typ> pLinks, Baum<Typ> pRechts)	Ein neuer Baum mit zwei Teilbäumen wird erzeugt

Method Summary	
Typ inhalt ()	Der Inhalt der Wurzel wird zurückgegeben
boolean istBlatt ()	Wenn der Baum keine Teilbäume hat, wird true zurückgegeben
boolean istLeer ()	Wenn die Wurzel keinen Inhalt besitzt, wird true zurückgegeben
boolean istWurzel ()	Wenn der Baum keinen Vater hat, wird true zurückgegeben
Baum<Typ> linkerTeilbaum ()	Der linke Teilbaum wird zurückgegeben
Baum<Typ> rechterTeilbaum ()	Der rechte Teilbaum wird zurückgegeben
void setzeInhalt (Typ pInhalt)	Der Inhalt der Wurzel wird geändert
void setzeLinkenTeilbaum (Baum<Typ> pBaum)	Der rechte Teilbaum wird verändert
void setzeRechtenTeilbaum (Baum<Typ> pBaum)	Der rechte Teilbaum wird verändert
void setzeVater (Baum<Typ> pBaum)	

	Der Vater des Baums wird geändert
boolean	teilbaeumeLeer() Wenn die Teilbäume leer sind, wird true zurückgegeben
String	toString() Eine Stringrepräsentation des Baum mit seinen Unterbäumen wird zurückgegeben
Baum<Typ>	vater() Der Vater des Baums wird zurückgegeben

Methods inherited from class java.lang.Object
clone , equals , finalize , getClass , hashCode , notify , notifyAll , wait , wait , wait

Field Detail

serialVersionUID

static final long **serialVersionUID**

See Also:
[Constant Field Values](#)

Constructor Detail

Baum

public **Baum**([Typ](#) pInhalt)

Ein neuer Baum mit leeren Teilbäumen wird erzeugt

Parameters:
pInhalt - das Objekt, das der Inhalt der Wurzel wird

Baum

public **Baum**([Typ](#) pInhalt,
[Baum<Typ>](#) pLinks,
[Baum<Typ>](#) pRechts)

Ein neuer Baum mit zwei Teilbäumen wird erzeugt

Parameters:
pInhalt - das Objekt, das der Inhalt der Wurzel wird
pLinks - der linke Teilbaum
pRechts - der rechte Teilbaum

Method Detail

inhalt

public [Typ](#) **inhalt()**

Der Inhalt der Wurzel wird zurückgegeben

Returns:
der Inhalt der Wurzel

istBlatt

public boolean **istBlatt()**

Wenn der Baum keine Teilbäume hat, wird true zurückgegeben

Returns:
ob keine Teilbäume vorhanden sind

istLeer

public boolean **istLeer()**

Wenn die Wurzel keinen Inhalt besitzt, wird true zurückgegeben

Returns:

ob der Inhalt der Wurzel leer ist

istWurzel

public boolean **istWurzel**()

Wenn der Baum keinen Vater hat, wird true zurückgegeben

Returns:

ob kein Vater vorhanden ist

linkerTeilbaum

public [Baum](#)<[Typ](#)> **linkerTeilbaum**()

Der linke Teilbum wird zurückgegeben

Returns:

den linken Teilbaum

rechterTeilbaum

public [Baum](#)<[Typ](#)> **rechterTeilbaum**()

Der rechte Teilbum wird zurückgegeben

Returns:

den rechten Teilbaum

setzeInhalt

public void **setzeInhalt**([Typ](#) pInhalt)

Der Inhalt der Wurzel wird geändert

Parameters:

pInhalt - der neue Inhalt der Wurzel

setzeLinkenTeilbaum

public void **setzeLinkenTeilbaum**([Baum](#)<[Typ](#)> pBaum)

Der rechte Teilbaum wird verändert

Parameters:

pBaum - der neue rechte Teilbaum

setzeRechtenTeilbaum

public void **setzeRechtenTeilbaum**([Baum](#)<[Typ](#)> pBaum)

Der rechte Teilbaum wird verändert

Parameters:

pBaum - der neue rechte Teilbaum

setzeVater

public void **setzeVater**([Baum](#)<[Typ](#)> pBaum)

Der Vater des Baums wird geändert

Parameters:

pBaum - der neue Vater

teilbaeumeLeer

public boolean **teilbaeumeLeer**()

Wenn die Teilbäume leer sind, wird true zurückgegeben

Returns:
ob die Teilbäe leer sind

toString

public [String](#) **toString**()

Eine Stringrepräsentation des Baum mit seinen Unterbäumen wird zurückgegeben

Overrides:
[toString](#) in class [Object](#)

Returns:
einen String, der den Baum in mehreren Zeilen darstellt

vater

public [Baum](#)<[Typ](#)> **vater**()

Der Vater des Baums wird zurückgegeben

Returns:
den Vater des Baums

[Overview](#) [Package](#) **Class** [Tree](#) [Index](#) [Help](#)

PREV CLASS [NEXT CLASS](#)

SUMMARY: NESTED | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#) [All Classes](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

sum.strukturen

Class Binaerbaum<Typ>

[java.lang.Object](#)
 `sum.strukturen.Binaerbaum<Typ>`

All Implemented Interfaces:
[Serializable](#)

Direct Known Subclasses:
[Suchbinaerbaum](#)

```
public class Binaerbaum<Typ>  
extends Object  
implements Serializable
```

Binaerbaum ist die Klasse für einen binären Baum nach der Schnittstelle aus dem learn:line Arbeitsbereich "Von Stiften und Mäusen".
Er kann als Binaerbaum mit einem Inhaltstyp deklariert werden. Wird dieser weggelassen, liefert die Anfrage wurzelInhalt ein Object und es wird Typkonversion nötig.

Version:
7.3 vom 11.10.2011
Author:
Bernard Schriek
See Also:
[Serialized Form](#)

Field Summary	
(package private) Typ	kenntInhalt
(package private) Binaerbaum<Typ>	kenntLinkenNachfolger
(package private) Binaerbaum<Typ>	kenntRechtenNachfolger
(package private) Binaerbaum<Typ>	kenntVater
(package private) static long	serialVersionUID

Constructor Summary	
Binaerbaum ()	Es wurde ein leerer Binaerbaum erzeugt.
Binaerbaum (Typ pInhalt)	Es wurde ein Binaerbaum erzeugt, dessen Wurzel den angegebenen Inhalt hat und dessen Teilbäume leer sind.
Binaerbaum (Typ pInhalt, Binaerbaum<Typ> pLinkerBaum, Binaerbaum<Typ> pRechterBaum)	Es wurde ein Binaerbaum erzeugt, dessen Wurzel den angegebenen Inhalt hat und der die angegebenen Teilbäume hat.

Method Summary	
void	haengeLinksAn (Binaerbaum<Typ> pBaum) Der linke Teilbaum ist nun der angegebene Binaerbaum.
void	haengeRechtsAn (Binaerbaum<Typ> pBaum) Der rechte Teilbaum ist nun der angegebene Binaerbaum.
boolean	istLeer () Die Anfrage gibt an, ob der Binaerbaum leer ist.
Binaerbaum<Typ>	linkerTeilbaum ()

	Die Anfrage liefert den linken Teilbaum des Binaerbaums.
Binaerbaum<Typ>	rechterTeilbaum() Die Anfrage liefert den rechten Teilbaum des Binaerbaums.
boolean	teilbaeumeLeer() Die Anfrage gibt an, ob der Binaerbaum ein Blatt ist.
String	toString() Die Anfrage liefert eine Stringrepräsentation des Baum mit seinen Unterbäumen.
void	ueberschreibeWurzel(Typ pInhalt) Die Wurzel hat den angegebenen Inhalt.
Binaerbaum<Typ>	vater() Die Anfrage liefert den Vater des Binaerbaums.
Typ	wurzelInhalt() Die Anfrage liefert den Inhalt der Wurzel.

Methods inherited from class java.lang.Object
clone , equals , finalize , getClass , hashCode , notify , notifyAll , wait , wait , wait

Field Detail

kenntInhalt

[Typ](#) kenntInhalt

kenntLinkenNachfolger

[Binaerbaum<Typ>](#) kenntLinkenNachfolger

kenntRechtenNachfolger

[Binaerbaum<Typ>](#) kenntRechtenNachfolger

kenntVater

[Binaerbaum<Typ>](#) kenntVater

serialVersionUID

static final long serialVersionUID

See Also:

[Constant Field Values](#)

Constructor Detail

Binaerbaum

public Binaerbaum()

Es wurde ein leerer Binaerbaum erzeugt.

Binaerbaum

public Binaerbaum([Typ](#) pInhalt)

Es wurde ein Binaerbaum erzeugt, dessen Wurzel den angegebenen Inhalt hat und dessen Teilbäume leer sind.

Binaerbaum

public Binaerbaum([Typ](#) pInhalt,
 [Binaerbaum<Typ>](#) pLinkerBaum,
 [Binaerbaum<Typ>](#) pRechterBaum)

Es wurde ein Binaerbaum erzeugt, dessen Wurzel den angegebenen Inhalt hat und der die angegebenen

Teilbäume hat.

Method Detail

haengeLinksAn

public void haengeLinksAn([Binaerbaum](#)<[Typ](#)> pBaum)

Der linke Teilbaum ist nun der angegebene Binaerbaum.

haengeRechtsAn

public void haengeRechtsAn([Binaerbaum](#)<[Typ](#)> pBaum)

Der rechte Teilbaum ist nun der angegebene Binaerbaum.

istLeer

public boolean istLeer()

Die Anfrage gibt an, ob der Binaerbaum leer ist.

linkerTeilbaum

public [Binaerbaum](#)<[Typ](#)> linkerTeilbaum()

Die Anfrage liefert den linken Teilbaum des Binaerbaums.

rechterTeilbaum

public [Binaerbaum](#)<[Typ](#)> rechterTeilbaum()

Die Anfrage liefert den rechten Teilbaum des Binaerbaums.

teilbaeumeLeer

public boolean teilbaeumeLeer()

Die Anfrage gibt an, ob der Binaerbaum ein Blatt ist.

toString

public [String](#) toString()

Die Anfrage liefert eine Stringrepräsentation des Baum mit seinen Unterbäumen.

Overrides:
[toString](#) in class [Object](#)

ueberschreibeWurzel

public void ueberschreibeWurzel([Typ](#) pInhalt)

Die Wurzel hat den angegebenen Inhalt. Die beiden Teilbäume sind leer.

vater

public [Binaerbaum](#)<[Typ](#)> vater()

Die Anfrage liefert den Vater des Binaerbaums.

wurzelInhalt

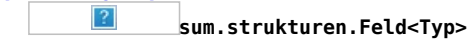
public [Typ](#) wurzelInhalt()

Die Anfrage liefert den Inhalt der Wurzel.

sum.strukturen

Class Feld<Typ>

[java.lang.Object](#)



Direct Known Subclasses:

[Tabelle](#)

```
public class Feld<Typ>
extends Object
```

Ein Feld verallgemeinert ein Array so, dass es beliebige Grenzen hat.
Es kann als Feld mit einem Inhaltstyp deklariert werden. Wird dieser weggelassen, liefert die Anfrage inhalt ein Object und es wird Typkonversion nötig.

Version:
7.3 vom 11.10.2011

Author:
Horst Hildebrecht

Field Summary	
(package private) Object []	hatArray
(package private) static long	serialVersionUID
(package private) int	zObereGrenze
(package private) int	zUntereGrenze

Constructor Summary	
Feld (int pUntereGrenze, int pObereGrenze)	nachher: Das Feld der angegebenen Größe wurde erzeugt.

Method Summary	
Typ inhalt (int pPosition)	nachher: Der Inhalt des Feldes an der angegebenen Position wurde geliefert.
int obereGrenze ()	nachher: Die obere Grenze des Indexbereichs wurde geliefert.
void setzeInhalt (int pPosition, Typ pInhalt)	nachher: Als Inhalt des Feldes an der angegebenen Position wurde das angegebene Objekt gesetzt.
int untereGrenze ()	nachher: Die untere Grenze des Indexbereichs wurde geliefert.

Methods inherited from class java.lang. Object	
clone , equals , finalize , getClass , hashCode , notify , notifyAll , toString , wait , wait , wait	

Field Detail	
hatArray	
Object []	hatArray

serialVersionUID

static final long **serialVersionUID**

See Also:
[Constant Field Values](#)

zObereGrenze

int **zObereGrenze**

zUntereGrenze

int **zUntereGrenze**

Constructor Detail

Feld

```
public Feld(int pUntereGrenze,
            int pObereGrenze)
```

nachher: Das Feld der angegebenen Größe wurde erzeugt. Es hat noch keinen Inhalt.

Method Detail

inhalt

```
public Typ inhalt(int pPosition)
```

nachher: Der Inhalt des Feldes an der angegebenen Position wurde geliefert.

obereGrenze

```
public int obereGrenze()
```

nachher: Die obere Grenze des Indexbereichs wurde geliefert.

setzeInhalt

```
public void setzeInhalt(int pPosition,
                        Typ pInhalt)
```

nachher: Als Inhalt des Feldes an der angegebenen Position wurde das angegebene Objekt gesetzt.

untereGrenze

```
public int untereGrenze()
```

nachher: Die untere Grenze des Indexbereichs wurde geliefert.

sum.strukturen

Class Hashtabelle<Typ extends [Schluesselobjekt](#)>



All Implemented Interfaces:
[Serializable](#)

```
public class Hashtabelle<Typ extends Schluesselobjekt>
extends Object
implements Serializable
```

Eine Hashtabelle enthält Schlüsselobjekte, um nach dem Schlüssel suchen zu können. Sie kann als Hashtabelle mit einem Inhaltstyp, der Ordnungsobjekt implementiert, deklariert werden. Wird dieser weggelassen, liefert die Anfrage suche ein Object und es wird Typkonversion nötig.

Version:
7.3 vom 11.10.2011

Author:
Bernard Schriek

See Also:
[Serialized Form](#)

Field Summary	
(package private) static long	serialVersionUID

Constructor Summary	
Hashtabelle ()	Eine neue leere Hashtabelle mit der Größe 11 wird angelegt
Hashtabelle (int pGroesse)	Eine neue leere Hashtabelle wird angelegt

Method Summary	
void	entferneAlleElemente () Alle Schlüsselobjekte werden aus der Hashtabelle entfernt
void	fuegeEin (Schluesselobjekt pSchluesselobjekt) Ein neues Schlüsselobjekt wird eingefügt
void	loesche (Object pSchluessel) Ein Schlüsselobjekt wird aus der Hashtabelle entfernt
Typ	suche (Object pSchluessel) Ein gesuchtes Schlüsselobjekt wird zurückgegeben bzw.null, wenn es nicht vorhanden ist
String	toString () Ein Stringrepräsentation der Hashtabelle wird zurückgegeben

Methods inherited from class java.lang. Object	
clone , equals , finalize , getClass , hashCode , notify , notifyAll , wait , wait , wait	

Field Detail

serialVersionUID

static final long **serialVersionUID**

See Also:
[Constant Field Values](#)

Constructor Detail

Hashtabelle

public **Hashtabelle**()

Eine neue leere Hashtabelle mit der Größe 11 wird angelegt

Hashtabelle

public **Hashtabelle**(int pGroesse)

Eine neue leere Hashtabelle wird angelegt

Parameters:
pGroesse - die Größe der neuen Hashtabelle

Method Detail

entferneAlleElemente

public void **entferneAlleElemente**()

Alle Schlüsselobjekte werden aus der Hashtabelle entfernt

fuegeEin

public void **fuegeEin**([Schluesselobjekt](#) pSchluesselobjekt)

Ein neues Schlüsselobjekt wird eingefügt

Parameters:
pSchluesselobjekt - das Schlüsselobjekt, das eingefügt wird

loesche

public void **loesche**([Object](#) pSchluessel)

Ein Schlüsselobjekt wird aus der Hashtabelle entfernt

Parameters:
pSchluessel - der Schlüssel, dessen Schlüsselobjekt entfernt werden soll

suche

public [Typ](#) **suche**([Object](#) pSchluessel)

Ein gesuchtes Schlüsselobjekt wird zurückgegeben bzw.null, wenn es nicht vorhanden ist

Parameters:
pSchluessel - der Schlüssel, dessen zugehöriges Schlüsselobjekt zurückgegeben werden soll

Returns:
das gesuchte Schlüsselobjekt oder null, wenn es nicht vorhanden ist

toString

public [String](#) **toString**()

Ein Stringrepräsentation der Hashtabelle wird zurückgegeben

Overrides:
[toString](#) in class [Object](#)

Returns:
einen String, der in mehreren Zeilen die Hashtabelle darstellt

sum.strukturen

Class Liste<Typ>



All Implemented Interfaces:
[Serializable](#)

Direct Known Subclasses:
[Schlange](#), [Stapel](#)

```
public class Liste<Typ>
extends Object
implements Serializable
```

Die Klasse Liste ist die Realisierung einer doppelt verketteten Liste mit zwei Sentinels (Bug und Heck) am Anfang und am Ende der Liste.
Sie kann als Liste mit einem Inhaltstyp deklariert werden. Wird dieser weggelassen, liefert die Anfrage aktuelles ein Object und es wird Typkonversion nötig.

Version:
7.3 vom 11.10.2011

Author:
Bernard Schriek

See Also:
[Serialized Form](#)

Field Summary	
(package private) static long	serialVersionUID

Constructor Summary	
Liste()	Eine neue leere Liste wird erzeugt.

Method Summary	
int	aktuellePosition() Das Position des aktuellen Knotens wird zurückgegeben.
Typ	aktuelles() Das Objekt an der aktuellen Position in der Liste wird zurückgegeben.
Typ	aktuellesElement() Das Objekt an der aktuellen Position in der Liste wird zurückgegeben.
void	entferneAktuelles() Das aktuelle Listenelement wird gelöscht.
void	entferneAlleElemente() Alle Listenelemente werden aus der Liste entfernt.
void	ersetzeAktuelles(Typ pObject) Das aktuelle Listenelement wird ersetzt.
protected sum.strukturen.Liste.Knoten	ersterKnoten() interner Dienst zur Unterstützung von haengeListeAn
void	fuegeDahinterEin(Typ pInhalt) Ein neues Objekt wird hinter der aktuellen Position in die Liste eingefügt.
void	fuegeDavorEin(Typ pInhalt) Ein neues Objekt wird vor der aktuellen Position in die Liste eingefügt.
void	geheZuPosition(int pPosition) Die aktuelle Position wird zu einer bestimmten Position bewegt.
void	haengeAn(Typ pInhalt)

	Ein neues Objekt wird an das Ende der Liste angehängt.
void	haengeListeAn (Liste pListe) Eine weitere Liste wird an die Liste angehängt.
boolean	istAmAnfang () Wenn die aktuelle Position das erste Element der Liste ist, wird true zurückgegeben.
boolean	istAmEnde () Wenn die aktuelle Position das letzte Element der Liste ist, wird true zurückgegeben.
boolean	istDahinter () Wenn die aktuelle Position hinter der Liste ist, wird true zurückgegeben.
boolean	istDavor () Wenn die aktuelle Position vor der Liste ist, wird true zurückgegeben.
boolean	istLeer () Es wird zurückgegeben, ob die Liste leer ist.
int	laenge () Die Länge der Liste wird zurückgegeben.
protected sum.strukturen.Liste.Knoten	letzterKnoten () interner Dienst zur Unterstützung von haengeListeAn
void	loescheAktuelles () Das aktuelle Listenelement wird gelöscht.
int	position (Typ pObject) Die Position eines Objekts in der Liste wird ermittelt.
void	setzeDavor (Typ pInhalt) Ein neues Objekt wird vor den Anfang der Liste eingefügt.
void	setzeListeDavor (Liste pListe) Eine weitere Liste wird vor die Liste angehängt.
String	toString () Der Inhalt der Liste wird als String zurückgegeben.
void	vor () Die aktuelle Position wird vor gesetzt.
void	zumAnfang () Die aktuelle Position wird auf den Listenanfang gesetzt.
void	zumEnde () Die aktuelle Position wird auf das Listenende gesetzt.
void	zurueck () Die aktuelle Position wird zurück gesetzt.

Methods inherited from class java.lang.Object
clone , equals , finalize , getClass , hashCode , notify , notifyAll , wait , wait , wait

Field Detail

serialVersionUID

static final long **serialVersionUID**

See Also:
[Constant Field Values](#)

Constructor Detail

Liste

public **Liste**()

Eine neue leere Liste wird erzeugt.

Method Detail

aktuellePosition

public int **aktuellePosition**()

Das Position des aktuellen Knotens wird zurückgegeben.

Returns:

die Position des aktuellen Knotens in der Liste

aktuelles

public [Typ](#) **aktuelles**()

Das Objekt an der aktuellen Position in der Liste wird zurückgegeben.

Returns:

das aktuelle Objekt der Liste

aktuellesElement

public [Typ](#) **aktuellesElement**()

Das Objekt an der aktuellen Position in der Liste wird zurückgegeben.

Returns:

das aktuelle Objekt der Liste

entferneAktuelles

public void **entferneAktuelles**()

Das aktuelle Listenelement wird gelöscht.

entferneAlleElemente

public void **entferneAlleElemente**()

Alle Listenelemente werden aus der Liste entfernt. Die Liste ist danach leer.

ersetzeAktuelles

public void **ersetzeAktuelles**([Typ](#) pObject)

Das aktuelle Listenelement wird ersetzt.

Parameters:

pObject - der neue Inhalt des aktuellen Listenelements

ersterKnoten

protected sum.strukturen.Liste.Knoten **ersterKnoten**()

interner Dienst zur Unterstützung von haengeListeAn

fuegeDahinterEin

public void **fuegeDahinterEin**([Typ](#) pInhalt)

Ein neues Objekt wird hinter der aktuellen Position in die Liste eingefügt. Die aktuelle Position bleibt unverändert.

Parameters:

pInhalt - der neue Inhalt (Objekt)

fuegeDavorEin

public void **fuegeDavorEin**([Typ](#) pInhalt)

Ein neues Objekt wird vor der aktuellen Position in die Liste eingefügt. Die aktuelle Position bleibt unverändert.

Parameters:

pInhalt - der neue Inhalt (Objekt)

geheZuPosition

```
public void geheZuPosition(int pPosition)
```

Die aktuelle Position wird zu einer bestimmten Position bewegt.

haengeAn

```
public void haengeAn(Typ pInhalt)
```

Ein neues Objekt wird an das Ende der Liste angehängt.

Parameters:
pInhalt - der neue Inhalt (Objekt)

haengeListeAn

```
public void haengeListeAn(Liste pListe)
```

Eine weitere Liste wird an die Liste angehängt.

Parameters:
pListe - die Liste, die an diese Liste angehängt wird.

istAmAnfang

```
public boolean istAmAnfang()
```

Wenn die aktuelle Position das erste Element der Liste ist, wird true zurückgegeben.

Returns:
true, wenn die aktuelle Position das erste Element der Liste ist.

istAmEnde

```
public boolean istAmEnde()
```

Wenn die aktuelle Position das letzte Element der Liste ist, wird true zurückgegeben.

Returns:
true, wenn die aktuelle Position das letzte Element der Liste ist.

istDahinter

```
public boolean istDahinter()
```

Wenn die aktuelle Position hinter der Liste ist, wird true zurückgegeben.

Returns:
true, wenn die aktuelle Position hinter der Liste ist.

istDavor

```
public boolean istDavor()
```

Wenn die aktuelle Position vor der Liste ist, wird true zurückgegeben.

Returns:
true, wenn die aktuelle Position vor der Liste ist.

istLeer

```
public boolean istLeer()
```

Es wird zurückgegeben, ob die Liste leer ist.

Returns:
true, wenn die Liste leer ist

laenge

```
public int laenge()
```

Die Länge der Liste wird zurückgegeben.

Returns:

die Anzahl der Elemente in der Liste

letzterKnoten

```
protected sum.strukturen.Liste.Knoten letzterKnoten()
```

interner Dienst zur Unterstützung von haengeListeAn

loescheAktuelles

```
public void loescheAktuelles()
```

Das aktuelle Listenelement wird gelöscht.

position

```
public int position(Typ pObject)
```

Die Position eines Objekts in der Liste wird ermittelt. Falls das Objekt nicht in der Liste enthalten ist, wird -1 zurückgegeben. Falls das Objekt mehrfach in der Liste enthalten ist, wird die Position des ersten Auftretens zurückgegeben.

Parameters:

pObject - das Objekt, dessen Position in der Liste ermittelt wird.

Returns:

die Position des Objekts in der Liste

setzeDavor

```
public void setzeDavor(Typ pInhalt)
```

Ein neues Objekt wird vor den Anfang der Liste eingefügt.

Parameters:

pInhalt - der neue Inhalt (Objekt)

setzeListeDavor

```
public void setzeListeDavor(Liste pListe)
```

Eine weitere Liste wird vor die Liste angehängt.

Parameters:

pListe - die Liste, die vor diese Liste angehängt wird.

toString

```
public String toString()
```

Der Inhalt der Liste wird als String zurückgegeben.

Overrides:

[toString](#) in class [Object](#)

Returns:

der Inhalt der Liste als String

vor

```
public void vor()
```

Die aktuelle Position wird vor gesetzt.

zumAnfang

```
public void zumAnfang()
```

Die aktuelle Position wird auf den Listenanfang gesetzt.

zumEnde

```
public void zumEnde()
```

Die aktuelle Position wird auf das Listenende gesetzt.

zurueck

```
public void zurueck()
```

Die aktuelle Position wird zurück gesetzt.

[Overview](#) [Package](#) **Class** [Tree](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: [NESTED](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#) [All Classes](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

sum.strukturen

Class Ordnungsklasse

[java.lang.Object](#)

sum.strukturen.Ordnungsklasse

All Implemented Interfaces:
[Serializable](#)

Direct Known Subclasses:
[SuchbaumInhalt](#)

public abstract class **Ordnungsklasse**
extends [Object](#)
implements [Serializable](#)

Die Ordnungsklasse ist eine Oberklasse für Objekte mit einer Ordnung.

Version:
7.3 vom 11.10.2011

Author:
Horst Hildebrecht

See Also:
[Serialized Form](#)

Constructor Summary

[Ordnungsklasse](#)()

Method Summary

abstract boolean	istGleich (Ordnungsklasse pObjekt) Ist genau dann wahr, wenn dieses Exemplar der Ordnungsklasse gleich dem Vergleichsobjekt pObjekt ist.
boolean	istGroesser (Ordnungsklasse pObjekt) Ist genau dann wahr, wenn dieses Exemplar der Ordnungsklasse größ als das Vergleichsobjekt pObjekt ist.
abstract boolean	istKleiner (Ordnungsklasse pObjekt) Ist genau dann wahr, wenn dieses Exemplar der Ordnungsklasse kleiner als das Vergleichsobjekt pObjekt ist.

Methods inherited from class java.lang.Object

[clone](#), [equals](#), [finalize](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#), [toString](#), [wait](#), [wait](#), [wait](#)

Constructor Detail

Ordnungsklasse

public **Ordnungsklasse**()

Method Detail

istGleich

public abstract boolean **istGleich**([Ordnungsklasse](#) pObjekt)

Ist genau dann wahr, wenn dieses Exemplar der Ordnungsklasse gleich dem Vergleichsobjekt pObjekt ist.

istGroesser

public boolean **istGroesser**([Ordnungsklasse](#) pObjekt)

Ist genau dann wahr, wenn dieses Exemplar der Ordnungsklasse größer als das Vergleichsobjekt pObjekt ist.

istKleiner

public abstract boolean **istKleiner**([Ordnungsklasse](#) pObjekt)

Ist genau dann wahr, wenn dieses Exemplar der Ordnungsklasse kleiner als das Vergleichsobjekt pObjekt ist.

[Overview](#) [Package](#) **Class** [Tree](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

[FRAMES](#) [NO FRAMES](#) [All Classes](#)

SUMMARY: NESTED | FIELD | [CONSTR](#) | [METHOD](#)

DETAIL: FIELD | [CONSTR](#) | [METHOD](#)

sum.strukturen

Interface Ordnungsobjekt

public interface **Ordnungsobjekt**

Ein Ordnungsobjekt ist eine Interface für Objekte, die in Suchbäume eingefügt werden sollen.

Version:
7.3 vom 11.10.2011

Author:
Bernard Schriek

Method Summary

boolean	istGleichWie (Ordnungsobjekt pObjekt) Ist genau dann wahr, wenn das Ordnungsobjekt gleich dem Vergleichsobjekt pObjekt ist.
boolean	istGroesserAls (Ordnungsobjekt pObjekt) Ist genau dann wahr, wenn das Ordnungsobjekt größer als das Vergleichsobjekt pObjekt ist.
boolean	istKleinerAls (Ordnungsobjekt pObjekt) Ist genau dann wahr, wenn das Ordnungsobjekt kleiner als das Vergleichsobjekt pObjekt ist.

Method Detail

istGleichWie

boolean **istGleichWie**([Ordnungsobjekt](#) pObjekt)

Ist genau dann wahr, wenn das Ordnungsobjekt gleich dem Vergleichsobjekt pObjekt ist.

istGroesserAls

boolean **istGroesserAls**([Ordnungsobjekt](#) pObjekt)

Ist genau dann wahr, wenn das Ordnungsobjekt größer als das Vergleichsobjekt pObjekt ist.

istKleinerAls

boolean **istKleinerAls**([Ordnungsobjekt](#) pObjekt)

Ist genau dann wahr, wenn das Ordnungsobjekt kleiner als das Vergleichsobjekt pObjekt ist.

[sum.strukturen](#)

Interfaces

[*Ordnungsobjekt*](#)

[*Schlüsselobjekt*](#)

Classes

[Baum](#)

[Binaerbaum](#)

[Feld](#)

[Hashtabelle](#)

[Liste](#)

[Ordnungsklasse](#)

[Schlange](#)

[Stapel](#)

[Suchbaum](#)

[SuchbaumInhalt](#)

[Suchbinaerbaum](#)

[Tabelle](#)

Package sum.strukturen

Interface Summary	
Ordnungsobjekt	Ein Ordnungsobjekt ist eine Interface für Objekte, die in Suchbäume eingefügt werden sollen.
Schluesselobjekt	Ein Schlüsselobjekt ist das Interface für Objekte, die in Hashtabellen eingefügt werden sollen.

Class Summary	
Baum<Typ>	Die Klasse Baum ist ein Binärbaum nach der Schnittstelle aus dem Buch "Informatik mit Java".
Binaerbaum<Typ>	Binaerbaum ist die Klasse für einen binären Baum nach der Schnittstelle aus dem learn:line Arbeitsbereich "Von Stiften und Mäusen".
Feld<Typ>	Ein Feld verallgemeinert ein Array so, dass es beliebige Grenzen hat.
Hashtabelle<Typ extends Schluesselobjekt>	Eine Hashtabelle enthält Schlüsselobjekte, um nach dem Schlüssel suchen zu können.
Liste<Typ>	Die Klasse Liste ist die Realisierung einer doppelt verketteten Liste mit zwei Sentinels (Bug und Heck) am Anfang und am Ende der Liste.
Ordnungsklasse	Die Ordnungsklasse ist eine Oberklasse für Objekte mit einer Ordnung.
Schlange<Typ>	Die Schlange ist eine FIFO-Struktur.
Stapel<Typ>	Ein Stapel ist die Realisierung eines Stacks.
Suchbaum<Typ extends Ordnungsobjekt>	Ein Suchbaum ist als Binärbaum nach der Schnittstelle aus dem Buch "Informatik mit Java" realisiert und enthält Ordnungsobjekte.
SuchbaumInhalt	Ein SuchbaumInhalt ist eine Klasse mit einer Ordnung, die außerdem zählen kann, wie oft ein Exemplar mit diesem Schlüssel eingefügt wurde.
Suchbinaerbaum<Typ extends SuchbaumInhalt>	Suchbinaerbaum ist die Klasse für einen Suchbaum nach der Schnittstelle aus dem learn:line Arbeitsbereich "Von Stiften und Mäusen".
Tabelle<Typ extends Ordnungsobjekt>	Eine Tabelle ist ein nur teilweise gefülltes Feld.

Hierarchy For Package sum.strukturen

Package Hierarchies:
[All Packages](#)

Class Hierarchy

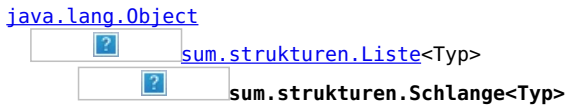
- java.lang.[Object](#)
 - sum.strukturen.[Baum](#)<Typ> (implements java.io.[Serializable](#))
 - sum.strukturen.[Suchbaum](#)<Typ> (implements java.io.[Serializable](#))
 - sum.strukturen.[Binaerbaum](#)<Typ> (implements java.io.[Serializable](#))
 - sum.strukturen.[Suchbinaerbaum](#)<Typ> (implements java.io.[Serializable](#))
 - sum.strukturen.[Feld](#)<Typ>
 - sum.strukturen.[Tabelle](#)<Typ>
 - sum.strukturen.[Hashtabelle](#)<Typ> (implements java.io.[Serializable](#))
 - sum.strukturen.[Liste](#)<Typ> (implements java.io.[Serializable](#))
 - sum.strukturen.[Schlange](#)<Typ> (implements java.io.[Serializable](#))
 - sum.strukturen.[Stapel](#)<Typ> (implements java.io.[Serializable](#))
 - sum.strukturen.[Ordnungsklasse](#) (implements java.io.[Serializable](#))
 - sum.strukturen.[SuchbaumInhalt](#)

Interface Hierarchy

- sum.strukturen.[Ordnungsobjekt](#)
- sum.strukturen.[Schluesselobjekt](#)

sum.strukturen

Class Schlange<Typ>



All Implemented Interfaces:
[Serializable](#)

```
public class Schlange<Typ>
extends Liste<Typ>
implements Serializable
```

Die Schlange ist eine FIFO-Struktur.
Sie kann als Schlange mit einem Inhaltstyp deklariert werden. Wird dieser weggelassen, liefert die Anfrage kopf bzw. erstes ein Object und es wird Typkonversion nötig.

Version:
7.3 vom 11.10.2011

Author:
Bernard Schriek

See Also:
[Serialized Form](#)

Field Summary	
(package private) static long	serialVersionUID

Constructor Summary	
Schlange()	Eine neue leere Schlange wird erzeugt.

Method Summary	
void	entferneErstes() Das erste Element der Schlange wird entfernt.
void	entferneKopf() Das erste Element der Schlange wird entfernt.
Typ	erstes() Das erste Objekt der Schlange wird zurückgegeben.
void	haengeAn(Typ pInhalt) Ein neues Objekt wird an das Ende der Schlange angehängt.
Typ	kopf() Das erste Objekt der Schlange wird zurückgegeben.

Methods inherited from class sum.strukturen. Liste	
aktuellePosition , aktuelles , aktuellesElement , entferneAktuelles , entferneAlleElemente , ersetzeAktuelles , ersterKnoten , fuegeDahinterEin , fuegeDavorEin , geheZuPosition , haengeListeAn , istAmAnfang , istAmEnde , istDahinter , istDavor , istLeer , laenge , letzterKnoten , loescheAktuelles , position , setzeDavor , setzeListeDavor , toString , vor , zumAnfang , zumEnde , zurueck	

Methods inherited from class java.lang. Object	
clone , equals , finalize , getClass , hashCode , notify , notifyAll , wait , wait , wait	

Field Detail	
serialVersionUID	

static final long **serialVersionUID**

See Also:
[Constant Field Values](#)

Constructor Detail

Schlange

public **Schlange**()

Eine neue leere Schlange wird erzeugt.

Method Detail

entferneErstes

public void **entferneErstes**()

Das erste Element der Schlange wird entfernt.

entferneKopf

public void **entferneKopf**()

Das erste Element der Schlange wird entfernt.

erstes

public [Typ](#) **erstes**()

Das erste Objekt der Schlange wird zurückgegeben.

Returns:
das erste Objekt der Schlange

haengeAn

public void **haengeAn**([Typ](#) pInhalt)

Ein neues Objekt wird an das Ende der Schlange angehängt.

Overrides:
[haengeAn](#) in class [Liste<Typ>](#)

Parameters:
pInhalt - der neue Inhalt (Objekt)

kopf

public [Typ](#) **kopf**()

Das erste Objekt der Schlange wird zurückgegeben.

Returns:
das erste Objekt der Schlange

sum.strukturen

Interface Schluesselobjekt

public interface **Schluesselobjekt**

Ein Schlüsselobjekt ist das Interface für Objekte, die in Hashtabellen eingefügt werden sollen.

Version:
7.3 vom 11.10.2011

Author:
Bernard Schriek

Method Summary

Object	schluessel() Der Schlüssel des Objekts wird geliefert.
------------------------	---

Method Detail

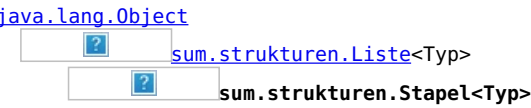
schluessel

[Object](#) **schluessel()**

Der Schlüssel des Objekts wird geliefert.

sum.strukturen

Class Stapel<Typ>



All Implemented Interfaces:
[Serializable](#)

```
public class Stapel<Typ>
extends Liste<Typ>
implements Serializable
```

Ein Stapel ist die Realisierung eines Stacks. (LIFO)
Er kann als Stapel mit einem Inhaltstyp deklariert werden. Wird dieser weggelassen, liefert die Anfrage oberstes bzw. spitze ein Object und es wird Typkonversion nötig.

Version:
7.3 vom 11.10.2011

Author:
Bernard Schriek

See Also:
[Serialized Form](#)

Field Summary	
(package private) static long	serialVersionUID

Constructor Summary	
Stapel()	Ein neuer leerer Stapel wird erzeugt

Method Summary	
void	entferneOberstes() Das oberste Objekt wird vom Stapel entfernt
void	legeAuf(Typ pObject) Ein neues Objekt wird auf den Stapel gelegt
void	nimmAb() Das oberste Objekt wird vom Stapel entfernt
Typ	oberstes() Das oberste Stapelelement wird zurückgegeben
Typ	spitze() Das oberste Stapelelement wird zurückgegeben

Methods inherited from class sum.strukturen.Liste	
aktuellePosition , aktuelles , aktuellesElement , entferneAktuelles , entferneAlleElemente , ersetzeAktuelles , ersterKnoten , fuegeDahinterEin , fuegeDavorEin , geheZuPosition , haengeAn , haengeListeAn , istAmAnfang , istAmEnde , istDahinter , istDavor , istLeer , laenge , letzterKnoten , loescheAktuelles , position , setzeDavor , setzeListeDavor , toString , vor , zumAnfang , zumEnde , zurueck	

Methods inherited from class java.lang.Object	
clone , equals , finalize , getClass , hashCode , notify , notifyAll , wait , wait , wait	

Field Detail

serialVersionUID

static final long **serialVersionUID**

See Also:
[Constant Field Values](#)

Constructor Detail

Stapel

public **Stapel**()

Ein neuer leerer Stapel wird erzeugt

Method Detail

entferneOberstes

public void **entferneOberstes**()

Das oberste Objekt wird vom Stapel entfernt

legeAuf

public void **legeAuf**([Typ](#) pObject)

Ein neues Objekt wird auf den Stapel gelegt

Parameters:
pObject - das Objekt, das auf den Stapel gelegt wird

nimmAb

public void **nimmAb**()

Das oberste Objekt wird vom Stapel entfernt

oberstes

public [Typ](#) **oberstes**()

Das oberste Stapелеlement wird zurückgegeben

Returns:
das oberste Objekt auf dem Stapel

spitze

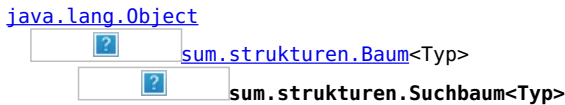
public [Typ](#) **spitze**()

Das oberste Stapелеlement wird zurückgegeben

Returns:
das oberste Objekt auf dem Stapel

sum.strukturen

Class Suchbaum<Typ extends [Ordnungsobjekt](#)>



All Implemented Interfaces:
[Serializable](#)

```
public class Suchbaum<Typ extends Ordnungsobjekt>
extends Baum<Typ>
implements Serializable
```

Ein Suchbaum ist als Binärbaum nach der Schnittstelle aus dem Buch "Informatik mit Java" realisiert und enthält Ordnungsobjekte. Er kann als Suchbaum mit einem Inhaltstyp, der Ordnungsobjekt implementiert, deklariert werden. Wird dieser weggelassen, liefert die Anfrage suche ein Object und es wird Typkonversion nötig.

Version:
7.3 vom 11.10.2011

Author:
Bernard Schriek

See Also:
[Serialized Form](#)

Field Summary	
(package private) static long	serialVersionUID

Constructor Summary	
Suchbaum (Typ pInhalt)	Ein neuer Suchbaum mit leeren Teilbäumen wird erzeugt

Method Summary	
void fuegeEin (Typ pInhalt)	Ein Objekt wird an der passenden Stelle eingefügt, wenn noch kein Objekt mit diesem Schlüssel vorhanden war.
void loesche (Typ pInhalt)	Wenn ein Objekt mit dem Schlüssel von pInhalt vorhanden ist, wird es aus dem Baum gelöscht.
Typ suche (Typ pInhalt)	Wenn ein Objekt mit dem Schlüssel von pInhalt vorhanden ist, wird es geliefert, sonst ist das Ergebnis null.

Methods inherited from class sum.strukturen. Baum	
inhalt , istBlatt , istLeer , istWurzel , linkerTeilbaum , rechterTeilbaum , setzeInhalt , setzeLinkenTeilbaum , setzeRechtenTeilbaum , setzeVater , teilbaeumeLeer , toString , vater	

Methods inherited from class java.lang. Object	
clone , equals , finalize , getClass , hashCode , notify , notifyAll , wait , wait , wait	

Field Detail	
serialVersionUID	
static final long serialVersionUID	
See Also:	

Constructor Detail

Suchbaum

public **Suchbaum**([Typ](#) pInhalt)

Ein neuer Suchbaum mit leeren Teilbäumen wird erzeugt

Parameters:

pInhalt - das Objekt, das der Inhalt der Wurzel wird

Method Detail

fuegeEin

public void **fuegeEin**([Typ](#) pInhalt)

Ein Objekt wird an der passenden Stelle eingefügt, wenn noch kein Objekt mit diesem Schlüssel vorhanden war.

Parameters:

pInhalt - das Objekt, das eingefügt werden soll

loesche

public void **loesche**([Typ](#) pInhalt)

Wenn ein Objekt mit dem Schlüssel von pInhalt vorhanden ist, wird es aus dem Baum gelöscht.

Parameters:

pInhalt - das Objekt, das gelöscht werden soll

suche

public [Typ](#) **suche**([Typ](#) pInhalt)

Wenn ein Objekt mit dem Schlüssel von pInhalt vorhanden ist, wird es geliefert, sonst ist das Ergebnis null.

Parameters:

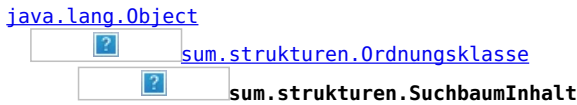
pInhalt - das Objekt, das gelöscht werden soll

Returns:

das Objekt aus dem Suchbaum mit demselben Schlüssel wie pInhalt

sum.strukturen

Class SuchbaumInhalt



All Implemented Interfaces:
[Serializable](#)

```
public abstract class SuchbaumInhalt
extends Ordnungsklasse
```

Ein SuchbaumInhalt ist eine Klasse mit einer Ordnung, die au erdem z hlen kann, wie oft ein Exemplar mit diesem Schl ssel eingef gt wurde.

Version:
7.3 vom 11.10.2011

Author:
Horst Hildebrecht

See Also:
[Serialized Form](#)

Field Summary	
(package private) int	zAnzahl

Constructor Summary	
SuchbaumInhalt ()	Der SuchbaumInhalt hat die Anzahl 1.
SuchbaumInhalt (int pAnzahl)	Der SuchbaumInhalt hat die angegebene Anzahl.

Method Summary	
int anzahl ()	Die Anzahl des SuchbaumInhalts wird geliefert.
void erhoeheAnzahl ()	Die Anzahl des SuchbaumInhalts wurde inkrementiert.
void setzeAnzahl (int pAnzahl)	Der SuchbaumInhalt hat die angegebene Anzahl.

Methods inherited from class sum.strukturen. Ordnungsklasse	
istGleich , istGroesser , istKleiner	

Methods inherited from class java.lang. Object	
clone , equals , finalize , getClass , hashCode , notify , notifyAll , toString , wait , wait , wait	

Field Detail

zAnzahl

int zAnzahl

Constructor Detail

SuchbaumInhalt

public **SuchbaumInhalt**()

Der SuchbaumInhalt hat die Anzahl 1.

SuchbaumInhalt

public **SuchbaumInhalt**(int pAnzahl)

Der SuchbaumInhalt hat die angegebene Anzahl.

Method Detail

anzahl

public int **anzahl**()

Die Anzahl des SuchbaumInhalts wird geliefert.

erhoeheAnzahl

public void **erhoeheAnzahl**()

Die Anzahl des SuchbaumInhalts wurde inkrementiert.

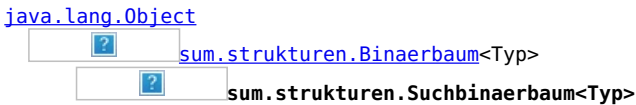
setzeAnzahl

public void **setzeAnzahl**(int pAnzahl)

Der SuchbaumInhalt hat die angegebene Anzahl.

sum.strukturen

Class Suchbinaerbaum<Typ extends [SuchbaumInhalt](#)>



All Implemented Interfaces:
[Serializable](#)

```
public class Suchbinaerbaum<Typ extends SuchbaumInhalt>
extends Binaerbaum<Typ>
implements Serializable
```

Suchbinaerbaum ist die Klasse für einen Suchbaum nach der Schnittstelle aus dem learn:line Arbeitsbereich "Von Stiften und Mäusen".
Ein verwaltet Exemplare von Unterklassen der Klasse SuchbaumInhalt.
Er kann als Suchbinaerbaum mit einem Inhaltstyp deklariert werden. Wird dieser weggelassen, liefert die Anfrage suche ein Object und es wird Typkonversion nötig.

Version:
7.3 vom 11.10.2011
Author:
Horst Hildebrecht
See Also:
[Serialized Form](#)

Field Summary	
(package private) static long	serialVersionUID

Fields inherited from class sum.strukturen. Binaerbaum	
kenntInhalt , kenntLinkenNachfolger , kenntRechtenNachfolger , kenntVater	

Constructor Summary	
Suchbinaerbaum ()	Ein leerer Suchbaum existiert.

Method Summary	
void	entferne (Typ pObjekt) Es gibt keinen Knoten im Suchbaum mit demselben Schlüssel wie pObjekt.
void	entferneAlle () Der Suchbinaerbaum ist nun leer.
void	fuegeEin (Typ pInhalt) Wenn ein Objekt mit dem Schlüssel von pInhalt noch nicht im Suchbaum war, wurde pInhalt passend eingefügt.
Suchbinaerbaum < Typ >	linkerSuchbaum () Diese Anfrage liefert den linken Teilbaum des Suchbaums.
Suchbinaerbaum < Typ >	rechterSuchbaum () Diese Anfrage liefert den rechten Teilbaum des Suchbaums.
Liste	sortierteListe () Diese Anfrage liefert eine nach dem Schlüssel sortierte Liste aller Inhalte des Suchbaums.
Typ	suche (Typ pObjekt) Wenn ein Objekt mit dem Schlüssel von pObjekt im Baum gefunden wurde, liefert die Anfrage dieses Objekt, andernfalls ist das Ergebnis null.

Methods inherited from class sum.strukturen. Binaerbaum	
haengeLinksAn , haengeRechtsAn , istLeer , linkerTeilbaum , rechterTeilbaum , teilbaeumeLeer , toString , ueberschreibWurzel ,	

[vater](#), [wurzelInhalt](#)

Methods inherited from class `java.lang.Object`

[clone](#), [equals](#), [finalize](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#), [wait](#), [wait](#), [wait](#)

Field Detail

`serialVersionUID`

static final long `serialVersionUID`

See Also:

[Constant Field Values](#)

Constructor Detail

`Suchbinaerbaum`

public `Suchbinaerbaum()`

Ein leerer Suchbaum existiert.

Method Detail

`entferne`

public void `entferne`([Typ](#) pObjekt)

Es gibt keinen Knoten im Suchbaum mit demselben Schlüssel wie pObjekt.

`entferneAlle`

public void `entferneAlle()`

Der Suchbinaerbaum ist nun leer.

`fuegeEin`

public void `fuegeEin`([Typ](#) pInhalt)

Wenn ein Objekt mit dem Schlüssel von pInhalt noch nicht im Suchbaum war, wurde pInhalt passend eingefügt. Andernfalls wurde der Zähler in dem entsprechenden Knoteninhalt um eins erhöht.

`linkerSuchbaum`

public [Suchbinaerbaum](#)<[Typ](#)> `linkerSuchbaum()`

Diese Anfrage liefert den linken Teilbaum des Suchbaums.

`rechterSuchbaum`

public [Suchbinaerbaum](#)<[Typ](#)> `rechterSuchbaum()`

Diese Anfrage liefert den rechten Teilbaum des Suchbaums.

`sortierteListe`

public [Liste](#) `sortierteListe()`

Diese Anfrage liefert eine nach dem Schlüssel sortierte Liste aller Inhalte des Suchbaums.

`suche`

public [Typ](#) `suche`([Typ](#) pObjekt)

Wenn ein Objekt mit dem Schlüssel von pObjekt im Baum gefunden wurde, liefert die Anfrage dieses Objekt, andernfalls ist das Ergebnis null.

[Overview](#) [Package](#) **Class** [Tree](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

[FRAMES](#) [NO FRAMES](#) [All Classes](#)

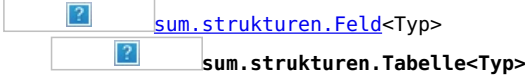
SUMMARY: [NESTED](#) | [FIELD](#) | [CONSTR](#) | [METHOD](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)

sum.strukturen

Class Tabelle<Typ extends [Ordnungsobjekt](#)>

[java.lang.Object](#)



```
public class Tabelle<Typ extends Ordnungsobjekt>
extends Feld<Typ>
```

Eine Tabelle ist ein nur teilweise gefülltes Feld.
 Sie kann als Tabelle mit einem Inhaltstyp, der Ordnungsobjekt implementiert, deklariert werden. Wird dieser weggelassen, liefert die Anfrage inhalt ein Object und es wird Typkonversion nötig.

Version:
 7.3 vom 11.10.2011

Author:
 Horst Hildebrecht

Field Summary	
(package private) static long	serialVersionUID
(package private) int	zAnzahl
(package private) int	zErsterFreierPlatz
(package private) int	zMaximaleAnzahl

Fields inherited from class sum.strukturen. Feld
hatArray , zObereGrenze , zUntereGrenze

Constructor Summary
Tabelle (int pUntereGrenze, int pObereGrenze) nachher: Die Tabelle ist initialisiert.

Method Summary	
int	anzahl () nachher: Die Anfrage liefert die Anzahl der Objekte in der Tabelle.
void	fuegeEin (Typ pInhalt) vorher: Die Tabelle ist noch nicht voll.
int	index (Typ pInhalt) nachher: Wenn sich ein Objekt mit dem identischen Suchschlüssel in der Tabelle befindet, dann liefert die Anfrage dessen Index im Indexbereich.
void	loesche (int pIndex) vorher: pIndex ist eine Zahl im Indexbereich der Tabelle.
boolean	voll () nachher: Die Anfrage liefert wahr, wenn kein weiteres Objekt eingefügt werden kann.

Methods inherited from class sum.strukturen. Feld
inhalt , obereGrenze , setzeInhalt , untereGrenze

Methods inherited from class java.lang. Object
clone , equals , finalize , getClass , hashCode , notify , notifyAll , toString , wait , wait , wait

Field Detail

serialVersionUID

static final long **serialVersionUID**

See Also:
[Constant Field Values](#)

zAnzahl

int **zAnzahl**

zErsterFreierPlatz

int **zErsterFreierPlatz**

zMaximaleAnzahl

int **zMaximaleAnzahl**

Constructor Detail

Tabelle

```
public Tabelle(int pUntereGrenze,
               int pObereGrenze)
```

nachher: Die Tabelle ist initialisiert. Die aufzunehmenden Objekte haben die Nummern pUntereGrenze bis pObereGrenze (Indexbereich). Die Tabelle enthält noch keine Objekte. Die Anzahl der gespeicherten Objekte ist also 0.

Method Detail

anzahl

```
public int anzahl()
```

nachher: Die Anfrage liefert die Anzahl der Objekte in der Tabelle.

fuegeEin

```
public void fuegeEin(Typ pInhalt)
```

vorher: Die Tabelle ist noch nicht voll.
nachher: Das eingegebene Objekt pInhalt befindet sich in der Tabelle. Die Anzahl der gespeicherten Objekte wird um 1 erhöht.

index

```
public int index(Typ pInhalt)
```

nachher: Wenn sich ein Objekt mit dem identischen Suchschlüssel in der Tabelle befindet, dann liefert die Anfrage dessen Index im Indexbereich. Wenn es sich nicht in der Tabelle befindet, liefert die Anfrage eine Zahl außerhalb des Indexbereichs.

loesche

```
public void loesche(int pIndex)
```

vorher: pIndex ist eine Zahl im Indexbereich der Tabelle.
nachher: Das Objekt an der Stelle pIndex wurde aus der Tabelle gelöscht. Die Anzahl der gespeicherten Objekte wurde um 1 verkeintert.

voll

public boolean **voll()**

nachher: Die Anfrage liefert wahr, wenn kein weiteres Objekt eingefügt werden kann.

[Overview](#) [Package](#) **Class** [Tree](#) [Index](#) [Help](#)

[PREV CLASS](#) [NEXT CLASS](#)

SUMMARY: NESTED | [FIELD](#) | [CONSTR](#) | [METHOD](#)

[FRAMES](#) [NO FRAMES](#) [All Classes](#)

DETAIL: [FIELD](#) | [CONSTR](#) | [METHOD](#)