

Anlage zu ADT- Operationen***Operationen für den ADT Liste***

Java	Pascal / Delphi Operationen der Klasse TListe	Beschreibung
<code>public Liste ()</code>	constructor <code>create</code>	Eine leere Liste wird angelegt.
<code>public void insert (Object inhalt)</code>	procedure <code>insert (inhalt: TInhalt)</code>	Ein neues Listenelement mit dem angegebenen Inhalt wird angelegt und vor der aktuellen Position in die Liste eingefügt. Der interne Positionszeiger steht auf dem neu eingefügten Element.
<code>public void append (Object inhalt)</code>	procedure <code>append (inhalt: TInhalt)</code>	Ein neues Listenelement mit dem angegebenen Inhalt wird angelegt und am Ende der Liste angehängt. Der interne Positionszeiger bleibt unverändert.
<code>public Object getItem ()</code>	function <code>getItem: TInhalt</code>	Der Inhalt des aktuellen Listenelements wird gelesen.
<code>public void setItem (Object inhalt)</code>	procedure <code>setItem (inhalt: TInhalt)</code>	Der Inhalt des aktuellen Listenelements wird durch den angegebenen Inhalt ersetzt.
<code>public void delete ()</code>	procedure <code>delete</code>	Das aktuelle Listenelement wird gelöscht. Der interne Positionszeiger steht auf dem Nachfolger des gelöschten Elements. Falls kein Nachfolger existiert, zeigt er auf den Vorgänger.
<code>public void toNext ()</code>	procedure <code>toNext</code>	Der interne Positionszeiger wird auf das jeweils nachfolgende Listenelement weitergerückt. Steht der Positionszeiger auf dem letzten Listenelement, wird er nicht verändert.
<code>public void toPrevious ()</code>	procedure <code>toPrevious</code>	Der interne Positionszeiger wird auf das jeweils vorhergehende Listenelement gesetzt. Steht der Positionszeiger auf dem ersten Listenelement, wird er nicht verändert.
<code>public void toFirst ()</code>	procedure <code>toFirst</code>	Der interne Positionszeiger wird auf das erste Listenelement gesetzt.
<code>public boolean isEmpty ()</code>	function <code>isEmpty: boolean</code>	Die Anfrage liefert den Wert true , wenn die Liste keine Elemente enthält, sonst liefert sie den Wert false .
<code>public boolean endOfList ()</code>	function <code>endOfList: boolean</code>	Die Anfrage liefert den Wert true , wenn der interne Positionszeiger auf dem letzten Listenelement steht, sonst liefert sie den Wert false .
<code>public boolean topOfList ()</code>	function <code>topOfList: boolean</code>	Die Anfrage liefert den Wert true , wenn der interne Positionszeiger auf dem ersten Listenelement steht, sonst liefert sie den Wert false .

Operationen für den ADT Stapel

Java	Pascal / Delphi Operationen der Klasse TStapel	Beschreibung
<code>public Stapel ()</code>	constructor <code>create</code>	Ein leerer Stapel wird angelegt.
<code>public void push (Object inhalt)</code>	procedure <code>push (inhalt: TInhalt)</code>	Ein neues Element mit dem angegebenen Inhalt wird auf den Stapel gelegt.
<code>public void pop ()</code>	procedure <code>pop</code>	Das oberste Element wird vom Stapel entfernt.
<code>public Object top ()</code>	function <code>top: TInhalt</code>	Der Inhalt des obersten Elements wird gelesen.
<code>public boolean isEmpty ()</code>	function <code>isEmpty: boolean</code>	Die Anfrage liefert den Wert true , wenn der Stapel keine Elemente enthält, sonst liefert sie den Wert false .

Operationen für den ADT Schlange

Java	Pascal / Delphi Operationen der Klasse TSchlange	Beschreibung
<code>public Schlange ()</code>	constructor <code>create</code>	Eine leere Schlange wird angelegt.
<code>public void enqueue (Object inhalt)</code>	procedure <code>enqueue (inhalt: TInhalt)</code>	Ein neues Element mit dem angegebenen Inhalt wird angelegt und am Ende an die Schlange angehängt.
<code>public void dequeue ()</code>	procedure <code>dequeue</code>	Das Element am Kopf der Schlange wird entfernt.
<code>public Object head ()</code>	function <code>head: TInhalt</code>	Der Inhalt des Elements am Kopf der Schlange wird gelesen.
<code>public boolean isEmpty ()</code>	function <code>isEmpty: boolean</code>	Die Anfrage liefert den Wert true , wenn die Schlange keine Elemente enthält, sonst liefert sie den Wert false .

Anlage zu SQL-Klauseln

SELECT-Anweisung

```
SELECT [ALL|DISTINCT] Select-Ausdruck {,Select-Ausdruck}
FROM TabellenName {,TabelleName}
[WHERE Bedingung]
[GROUP BY SpaltenName {,SpaltenName}]
[HAVING Bedingung]
[ORDER BY Select-Ausdruck [ASC|DESC]{,Select-Ausdruck [ASC|DESC]}]
[LIMIT Zeilenzahl]
```

Ein Select-Ausdruck kann ein Spaltenname, eine Spaltenfunktion oder ein Alias-Name sein.

Operatoren

+, -, *, /,
=, != (ungleich), >, <, >=, <=,
AND, OR, NOT,
LIKE, BETWEEN, IN, IS NULL

Arithmetische Gruppenfunktionen

AVG(), COUNT(), MAX(), MIN(), SUM()