# L3\_1 Dynamische Datenstrukturen: Verkettete Liste

Eine verkettete Liste ist eine dynamische Datenstruktur, die eine geordnete Speicherung von Datenelementen ermöglicht. Die verkettete Liste ist auf keine Größe festgelegt, sondern kann beliebig erweitert oder verringert werden, d. h. eine verkettete Liste kann sich dynamisch dem Speicherbedarf anpassen.

**Elemente und Darstellung von verketteten Listen**

* Listen bestehen aus einem Anker und einem oder mehreren Knoten.
* Der Anker enthält einen Verweis auf das erste Listenelement / den ersten Knoten.
* Der erste Knoten wird auch Listenkopf (Kopf der Liste) genannt.
* Ein Knoten besteht aus Daten und einem Zeiger, der auf den nächsten Knoten zeigt.
* Der Zeiger des letzten Knotens zeigt auf die Endmarke NULL.
* Listen können unterschiedliche Datentypen speichern.
* Listen können beliebig erweitert oder verringert werden.

**Anker**

**Daten**

**Zeiger**

**Daten**

**Zeiger**

**Daten**

**Zeiger  
(NULL)**

Listenkopf K1 Knoten K2 Knoten K3

**Operationen**

Zwei Operationen auf der Datenstruktur verketteten Liste sind von großer Bedeutung:

Das Einfügen und Löschen von Knoten.

**Einfügen**

Die Datenstruktur verkettete Liste kann beliebig erweitert werden. D. h. es können beliebig viele Knoten an einer beliebigen Stelle eingefügt werden. Beim Einfügen kann zwischen Einfügen am Anfang (Kopf) der Liste, einfügen in der Liste oder einfügen am Ende der Liste unterschieden werden.

Einfügen am **Anfang** der Liste

1. Einen neuen Knoten Kneu erstellen.
2. Zeiger Knoten Kneu auf die Adresse des ersten Knoten K1 richten.
3. Zeiger vom Anker erhält die Adresse des Knoten Kneu.

Einfügen am **Ende** der Liste

1. Einen neuen Knoten Kneu erstellen.
2. Zeiger des Knoten Kneu erhält den Wert NULL.
3. Zeiger des bisherigen letzten Knotens Kn erhält die Adresse des Knotens Kneu.

Einfügen **in der Liste** (z.B. zwischen die Knoten K2 und K3)

1. Einen neuen Knoten Kneu erstellen.
2. Zeiger des Knotens Kneu erhält die Adresse des gewünschten Nachfolgers K3.
3. Zeiger des Vorgängerknotens K2 erhält die Adresse des Knotens Kneu.

**Graphische Darstellung:**

**Anker**

**Daten**

**Zeiger**

**Daten**

**Zeiger**

**Daten**

**Zeiger  
(NULL)**

**Daten**

**Zeiger**

Listenkopf K1 Knoten K2 Knoten K3

➂

➁

➀

Knoten Kneu

**Wichtig:** Immer erst den Zeiger des neuen Knotens Kneu auf das nächste Element richten, bevor der alte Zeiger verschoben wird.

**Löschen**

Ein Knoten in einer verketteten Liste kann ebenfalls gelöscht werden. Wie beim Einfügen können auch beim Löschen verschiedene Vorgänge unterschieden werden.

Den ersten bzw. letzten Knoten löschen oder das Löschen eines Knotens in der verketteten Liste.

Löschen des ersten Knotens K1

1. Zeiger des ersten Knotens K1 (🡪 Knoten K2) wird an den Anker Übergeben.

🡪 Der Knoten K1 bleibt zwar im Speicher, ist aber nicht mehr erreichbar.

Löschen des letzten Knotens Kn

1. Zeiger des Vorgängers von Kn (= Kn-1) erhält den Wert NULL.

🡪 Der Knoten Kn bleibt zwar im Speicher, ist aber nicht mehr erreichbar.

Löschen eines Knoten Kx in der verketteten Liste.

1. Zeiger des Vorgängers von Kx (= Kx-1) erhält die Adresse des Zeigers von Kx (🡪 Kx+1).

🡪 Der Knoten Kx bleibt zwar im Speicher, ist aber nicht mehr erreichbar.

🡺 Das Entfernen der „verwaisten“ Knoten aus dem Speicher wird vom Betriebssystem übernommen.

**Graphische Darstellung**

**Anker**

**Daten**

**Zeiger**

**Daten**

**Zeiger**

**Daten**

**Zeiger  
(NULL)**



Listenkopf Kx-1 Knoten Kx Knoten Kx+1

**Anwendung von verketteten Listen**

* Verkettete Listen können in ähnlichen Fällen wie Arrays verwendet werden. Möchte man in der Mitte der Liste einen neuen Eintrag hinzufügen, müsste man im Array n/2 Elemente verschieben. In der Liste muss lediglich ein neuer Knoten erzeugt und zwei Zeiger geändert werden.
* Mit verketteten Listen werden in der Informatik Bäume implementiert.