

Lineare Listen (=Einfach verkettete Listen)

Einfach **verkettete Listen** oder **linked lists** sind eine **fundamentale Datenstruktur**, die ich hier anhand von Code-Beispielen und Grafiken erklären will.

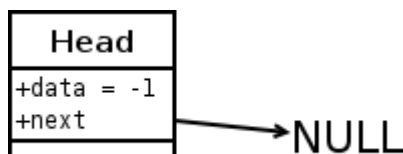
Einfach verkettete Listen zeichnen sich dadurch aus, dass man besonders einfach Elemente einfügen kann, wodurch sie sich besonders gut für Insertion Sort eignen.

Knoten

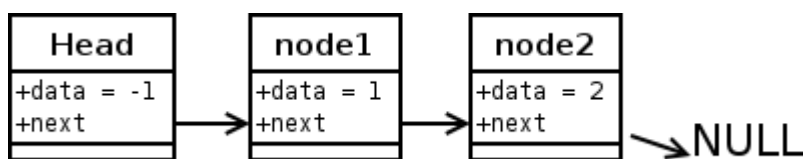
Eine einfach verkettete Liste besteht aus **Knoten**, **Englisch nodes**, die einen **Zeiger auf das nächste Element** und Daten beinhalten.

```
typedef struct listnode{  
    int data;  
    listnode *next;  
};
```

Eine **leere Liste** besteht aus einem Kopf (Head) und nichts sonst:



Wenn man mehrere Elemente einfügt, sieht das so aus:



Eine einfach verkettete Liste mit einem Kopf und zwei Knoten.

Knoten befüllen & einfügen

Wenn man einen Zeiger auf ein Element der Liste hat, ist es einfach, ein Element dahinter einzufügen.

Dazu muss man den next-Zeiger der Liste auf das neue Element setzen, und den next-Zeiger des neuen Element auf den alten Wert des next-Zeigers der Liste:

```
//1. Knoten erstellen  
listnode *node1;
```

```
node1=new listnode;
```

```
//1. Knoten befüllen  
node1->data=1; //oder (*node1).data=1;  
node1->next=NULL;
```



```
//2. Knoten erstellen  
listnode *node2;  
node2=new listnode;  
node2->data=2;  
node2->next=NULL;  
//1. Knoten zeigt auf 2. Knoten  
node1->next=node2;
```



```
//Knoten 12 zwischen Knoten 1 und 2 einfügen  
listnode *node12;  
node12=new listnode;  
node12->data=12;  
node12->next=node2;  
node1->next=node12;
```

Schritt 1



Schritt 2



Automatisiertes Hinzufügen von Knoten

Ein neuer Listenknoten wird durch Aufruf von new erzeugt. Dabei muss darauf geachtet werden, dass der Zeiger next gleich korrekt gesetzt wird. Wenn Sie nicht sofort den Nachfolger einhängen können, setzen Sie den Zeiger auf NULL.

```
#include <iostream>  
  
using namespace std;  
  
typedef struct listnode{
```

```
int data;
listnode *next;

};

int main(int argc, char** argv) {

    //Kopf der Liste
    listnode *head=NULL;

    //10 Knoten hinzufügen
    for(int i=1; i<10; i++)
    {
        listnode *node = new listnode;
        node->data=i;
        node->next=head;
        head=node;
    }

    //Knoten ausgeben
    listnode *help=NULL;
    help=head;

    while(help!=NULL)           //Solange Hilfszeiger nicht auf NULL zeigt
    {
        cout << help->data << endl;
        help=help->next;
    }

    return 0;
}
```

- [1.3.1\) Musterbeispiel](#)

From:
<http://elearn.bgamstetten.ac.at/wiki/> - Wiki

Permanent link:
http://elearn.bgamstetten.ac.at/wiki/doku.php?id=inf:inf8bi_201920:1:1_03

Last update: **2022/04/18 13:20**

