

Listen - Einführung zu den Aufgaben 1, 2 und 3

Verkettete Listen stellen wohl die elementarsten Datenstrukturen dar. Verkettete Listen bestehen aus einer Reihe von Elementen, die in einer bestimmten Reihenfolge angeordnet oder verkettet sind. Die Form der Verwaltung erinnert stark an Arrays, allerdings bieten verkettete Listen gegenüber Arrays in vielen Fällen wichtige Vorteile. Insbesondere sind verkettete Listen wesentlich effizienter bei Einfüge- und Lösch-Operationen. Verkettete Listen nutzen darüber hinaus dynamisch allozierten Speicher, d.h. zur Laufzeit allozierten Speicherplatz. Weil bei vielen Anwendungen die Größe der Daten während der Kompilierung nicht bekannt ist, kann dies eine sehr nützliche Eigenschaft sein. Man unterscheidet zwischen:

Einfach verketteten Listen:

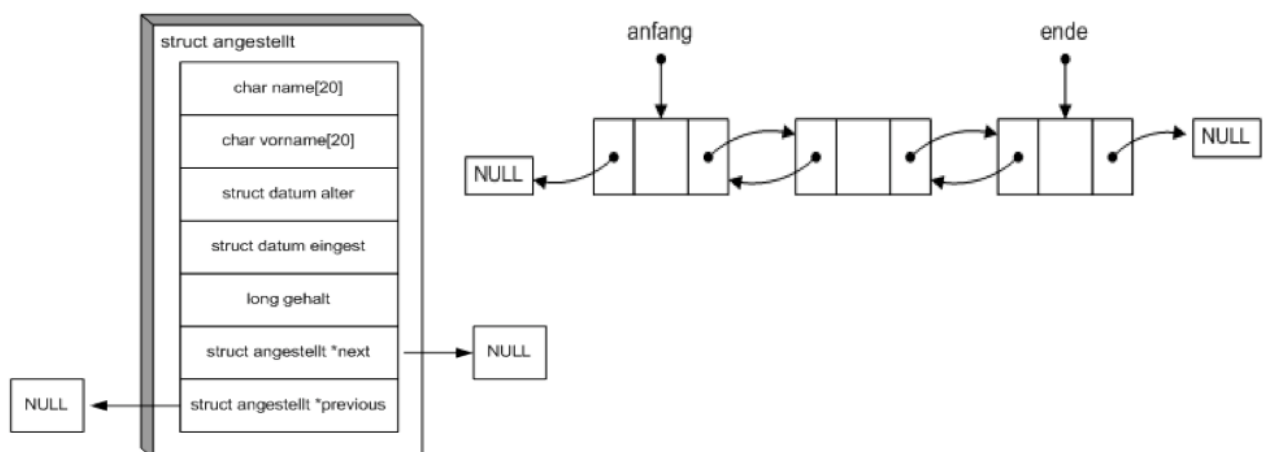
Die einfachste Form verketteter Listen, bei denen die Elemente über einen einzelnen Zeiger verknüpft sind. Diese Struktur erlaubt das Durchgehen der Liste vom ersten bis zum letzten Element.

Doppelt verketteten Listen:

Verkettete Listen, bei denen die Elemente nicht nur über einen, sondern über zwei Zeiger verkettet werden. Diese Struktur erlaubt das Durchgehen der Liste in beiden Richtungen.

Ringförmigen Listen

Verkettete Listen, bei denen das letzte Element nicht auf Null gesetzt wird, sondern wieder auf das erste Element verweist. Diese Struktur erlaubt ein ringförmiges Verarbeiten der Liste.



Das Bild zeigt ein Beispiel für eine doppelt verkettete Liste, die eine Liste mit Angestellten darstellt. Lässt man das Element am Ende wieder auf den Anfang zeigen und umgekehrt, erhält man eine zyklisch doppelt verkettete Liste.

Aufgabe 1

Es ist eine zyklisch verkettete Liste zu erstellen. Jedes Element soll als Datensatz die folgenden Werte enthalten: Name, Vorname, Geburtsdatum, Einstellungsdatum, Position und Gehalt. Benutzen Sie für Daten, wie Name, Vorname u.Ä. den Datentyp string. Dieser ist in C++ zu empfehlen, da er sehr einfach zu benutzen ist.

Aufgabe 2

Folgende Funktionen sind für die Liste zu implementieren:

init(...)

Initialisiert die Liste. Diese Operation muss angewendet werden, bevor die Liste mit anderen Operationen verwendet werden kann.

empty(...)

Gibt die Liste frei. Nach dem Aufruf sind keine Operationen mehr erlaubt, bis init erneut aufgerufen wurde. Die Operation entfernt alle Elemente aus dem Speicher und löscht diese.

remove(...)

Entfernt ein übergebenes Element aus der Liste.

insert_next(...)

Fügt ein übergebenes Element hinter einem bereits in der Liste enthaltenen Element ein.

insert_prev(...)

Fügt ein übergebenes Element vor einem bereits in der Liste enthaltenen Element ein.

head(...)

Liefert das erste Element der Liste zurück.

tail(...)

Liefert das letzte Element der Liste zurück.

Rückgabewerte sowie Parameter der Funktionen sind absichtlich nicht vorhanden. Diese sind in sinnvoller Weise zu ergänzen.

Aufgabe 3

Schreiben Sie ein kleines Rahmenprogramm, das die korrekte Funktionsweise belegt.