

## Modulare Programmierung in C

### Lernziele

Anwenden der modularen Programmierung in C sowie Vertiefen der Kenntnisse über die Gültigkeit und Sichtbarkeit von Variablen. Modulare Programmierung beschreibt die Aufteilung eines Programms in Module, die einzeln geplant, programmiert und getestet werden können.

### Aufgabe 2:

Implementieren Sie eine Datenstruktur `heap_t` mittels modularer Programmierung. Der Heap soll bei Bedarf automatisch vergrößert werden, sodass er beliebig viele Werte vom Typ `int` speichern kann. Die Datenstruktur soll folgende Operationen unterstützen:

- `heap_t *createHeap();`  
erzeugt einen leeren Heap
- `char insert(heap_t *h, int val);`  
fügt den Wert `val` in den Heap `h` ein
- `int minimum(heap_t *h);`  
liefert das minimale Element des Heaps `h`
- `char extractMin(heap_t *h);`  
entfernt das minimale Element aus dem Heap `h`
- `char getError(heap_t *h);`  
liefert den Inhalt der Fehlervariablen des Heaps `h`
- `char* toString(heap_t *h);`  
liefert den Inhalt der Heaps `h` als Zeichenkette
- `void destroyHeap(heap_t *h);`  
zerstört den Heap `h` und gibt belegten Speicherplatz frei

Wenn `minimum` oder `extractMin` auf einem leeren Heap ausgeführt wird, soll eine Fehlervariable gesetzt werden.

Erstellen Sie einen Testtreiber und Testfälle, um die Korrektheit Ihrer Implementierung nachzuweisen. Test sind grundsätzlich so zu gestalten, dass sie automatisch ablaufen und die Ergebnisse selbst überprüfen.

Informationen darüber, was ein Heap ist, wie man einen Heap mit Hilfe eines Arrays implementiert und wie die einzelnen Operationen implementiert werden können, finden Sie unter [https://de.wikipedia.org/wiki/Bin%C3%A4rer\\_Heap](https://de.wikipedia.org/wiki/Bin%C3%A4rer_Heap).

### Zusatz-Aufgabe Z2:

Um den Umgang mit Zeigern zu üben, wollen wir die aus der Vorlesung bekannte Liste als verkettete Liste (englisch: linked list) von einzelnen Elementen implementieren, siehe Abbildung 1.

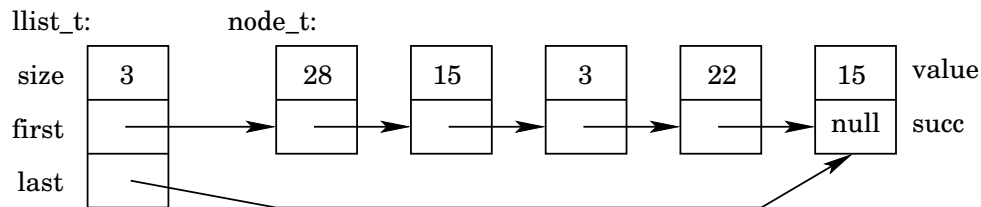


Abbildung 1: Einfach verkettete Liste.

In der Header-Datei deklarieren wir einen unvollständigen Datentypen `l1ist_t` und die Funktionen, mit denen wir auf die Liste zugreifen wollen. Beachten Sie dabei, dass wir den Datentyp `node_t` für die einzelnen Listenelemente dort nicht angeben. Die verkettete Liste soll die gleichen Operationen bereit stellen, wie die aus der Vorlesung bekannte Liste.