



Vorlesung „Grundlagen der Informatik und Programmierung 1“

CONTAINER

Verkettete Liste

Prof. Dr. Tom Vierjahn

Visual Computing (<https://vc.w-hs.de>)

Fachbereich Wirtschaft und Informationstechnik – Campus Bocholt



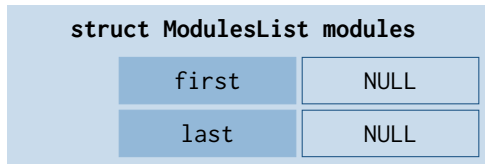
Wintersemester 2020/21



Veröffentlicht unter der Creative-Commons-Lizenz

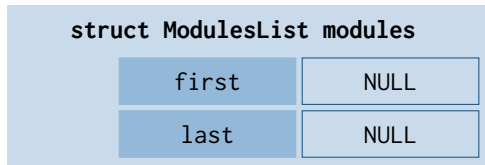
Attribution-NonCommercial-ShareAlike 4.0 International (CC BY-NC-SA 4.0)

Container:



Speicher:

Container:



Speicher:

Definition: Liste

Eine **einfach verkettete Liste** ist eine endliche Menge von Elementen, die durch eine Nachfolgeoperation miteinander **verkettet** sind. Enthält jedes Listenelement zusätzlich noch einen Verweis auf seinen Vorgänger, handelt es sich um eine **doppelt verkettete Liste**.

Definition: Listenanfang

Es gibt genau ein Listenelement ohne Vorgänger. Dies wird als **Listenanfang** bezeichnet.

Definition: Listenende

Es gibt genau ein Listenelement ohne Nachfolger. Dies wird als **Listenende** bezeichnet.

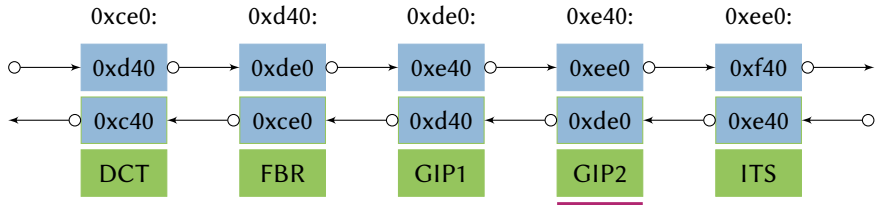
Definition: Listenanker

Damit auf die einzelnen Elemente (sequenziell) zugegriffen werden kann, muss der zugehörige Container als **Listenanker** einen Verweis auf den Listenanfang speichern.

Löschen von Elementen

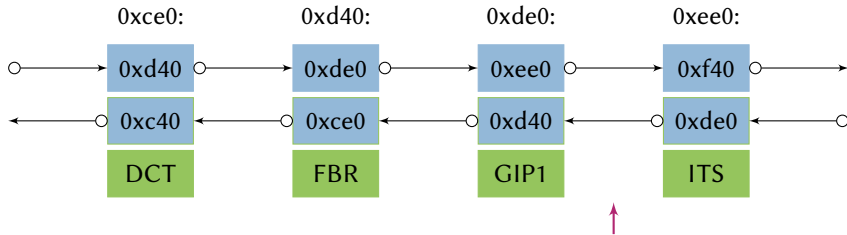
an beliebiger Stelle

vorher:



Einfügen von Elementen

an beliebiger Stelle



Container:

```
struct ModulesList {  
    struct ModulesListElement* first;  
    struct ModulesListElement* last;  
};
```

Listenelemente:

```
struct ModulesListElement {  
    struct Module module;  
    struct ModulesListElement* next;  
    struct ModulesListElement* previous;  
};
```

Iteration über alle Listenelemente

z.B. zur Ausgabe

- ▶ Listencontainer: modules
- ▶ Ausgabefunktion: void PrintModule(struct Module* module)

als Listenelemente:

```
for (struct ModulesListElement* current = modules.first;
     current != NULL;
     current = current->next) {
    PrintModule(&current->module);
}
```


Iteration über alle Listenelemente

z.B. zur Ausgabe

- ▶ Listencontainer: modules
- ▶ Ausgabefunktion: void PrintModule(struct Module* module)

Zeiger-Cast auf erstes Datenfeld:


```
for (struct ModulesListElement* current = modules.first;
     current != NULL;
     current = current->next) {
    PrintModule((Module*)current);
}
```

Dynamisches Array vs. (doppelt) verkettete Liste


Auswahlkriterien

- ▶ einfach verkettete Liste
- ▶ doppelt verkettete Liste
- ▶ Einfügen von Elementen
- ▶ Löschen von Elementen
- ▶ Deklaration von Liste und deren Elementen
- ▶ Iteration
- ▶ verkettete Liste vs. dynamisches Array


Prof. Dr. Tom Vierjahn

▶  tom.vierjahn@w-hs.de

Visual Computing

▶  <https://vc.w-hs.de>

▶  VisualComputingWH

▶  Visual Computing WH

▶  @VisComputingWH

Westfälische Hochschule

Fachbereich Wirtschaft und Informationstechnik

Campus Bocholt



Veröffentlicht unter der Creative-Commons-Lizenz

Attribution-NonCommercial-ShareAlike 4.0 International (CC BY-NC-SA 4.0)