

## L5\_1 Datenstrukturen

### 1 Begriff

#### Pausengespräch zweier Schüler:

- Sophia: Hallo Jonas! Warum bist du denn heute Morgen schon wieder zu spät zum Unterricht gekommen?
- Jonas: Hör auf. Ich war total in Hektik. Zuerst habe ich mein gelbes T-Shirt gesucht. Bis ich das gefunden hatte ist schon mal eine Ewigkeit vergangen. Und dann ging die Suche nach dem Mathe-Buch los. Wir haben aber auch so viele Schulbücher, da geht der Überblick im Bücherschrank schon mal verloren.  
Also kein Wunder, dass ich die Bahn verpasst habe.
- Sophia: Du bist halt ein Chaot. Würdest du deine T-Shirts nach der Farbe sortieren, hättest du dein Outfit mit einem Handgriff zusammen gehabt.  
Die Schulbücher kann man auch nach Fächern ordnen, dann findest du rucki zucki, was du suchst.  
Ich sag' doch schon immer: Ordnung ist das halbe Leben.

In der Informatik geht ohne Ordnung überhaupt nichts. Diese Ordnung kann mit Hilfe von Datenstrukturen geschaffen werden, die dazu dienen, Daten zu speichern und zu organisieren.

Eine Datenstruktur ist ein Objekt zur Speicherung und Organisation von Daten, die in einer bestimmten Art und Weise angeordnet und verknüpft werden, um den Zugriff auf sie und ihre Verwaltung effizient zu ermöglichen.

Vor allem Such- und Sortiervorgänge benötigen spezielle Datenstrukturen. Sie gewährleisten, dass die Programme mit einfachem, kurzem Programmcode und wenig Anweisungen entwickelt werden können, die nur geringe Rechnerleistung in Anspruch nehmen und einen geringen Speicherbedarf haben.

Zu den wichtigsten Datenstrukturen zählen:

- Array
- Verkettete Liste
- Stapelspeicher
- Warteschlange
- Baum

Die Entscheidung für eine bestimmte Datenstruktur ist immer abhängig von der Zielsetzung der Datenspeicherung und von den Operationen auf die zu speichernden Daten.

## 2 Merkmale ausgewählter Datenstrukturen

### 2.1 Das Array

Ein Array ist eine Kombination mehrerer Datenelemente des gleichen Datentyps, die in einem Speicher direkt hintereinander gespeichert werden.

Die Datenstruktur Array kann unterschieden werden in statische und dynamische Arrays. Bei der Verwendung statischer Arrays muss festgelegt werden, wie viele Elemente im Array erfasst werden sollen. Seine Größe ist somit für immer fest bestimmt und kann nicht geändert werden. Um diesen Nachteil zu vermeiden, können dynamische Arrays verwendet werden. Hier kann die Zahl der Elemente verändert und an die jeweils aktuellen Bedürfnisse angepasst werden.

Jedes zusätzliche Element wird am Ende des Arrays "angehängt". Der Zugriff auf die Array-Elemente erfolgt über fortlaufende Index-Werte (0, 1, 2, 3, usw.), die beim Hinzufügen der Elemente automatisch vergeben werden. Daraus ergibt sich, dass auf jedes Element direkt zugegriffen werden kann.

Das Entfernen von Elementen aus Arrays ist dagegen technisch aufwändiger.

Beispiel: In einem Krankenhaus wird die Körpertemperatur der Patienten täglich mehrmals gemessen und ausgewertet. Daraus ergibt sich, dass ein Array vom Datentyp Dezimalzahl – z.B. temp[ ] – benötigt wird.

38.3	37.9	38.7
------	------	------

### 2.2 Die verkettete Liste

Eine verkettete Liste ist eine Datenstruktur, in dem mehrere Elemente beliebiger Datentypen gespeichert werden können. Die zu speichernden Daten werden in den Listenelementen, den sogenannten Knoten erfasst. Jeder Knoten enthält einen Verweis zum nächsten Knoten, dem sogenannten Zeiger.

Bei Bedarf können weitere Knoten in die verkettete Liste eingefügt oder aus ihr entfernt werden. Dazu müssen lediglich die Zeiger entsprechend neu festgelegt werden.

Da jedes Listenelement nur den Weg zu seinen unmittelbaren Nachfolgeelement kennt, kann der Zugriff auf die Elemente einer verketteten Liste nur schrittweise entlang der Zeiger von einem Knoten zum nächsten erfolgen.

Bei der verketteten Liste handelt es sich um eine dynamische Datenstruktur, da die Länge der Liste beliebig erweitert werden kann.

Beispiel: Bei einem Skisprung-Wettkampf sollen die Haltungsnoten der Sprünge erfasst und ausgewertet werden. Dabei müssen folgende Regeln beachtet werden:

- Jedem Teilnehmer wird ein Versuch gewährt.
- Fünf Sprungrichter vergeben bei jedem Sprung Noten, von denen die beste und die schlechteste gestrichen werden.
- Den acht Springern mit den besten Leistungen steht ein weiterer Sprung zu.

Springer 1: 17,5 / 18,0 / ~~17,0~~ / ~~18,5~~ / 18,0

Springer 2: 19,0 / 19,0 / ~~18,5~~ / 19,5 / ~~20,0~~ / 19,5 / ~~19,0~~ / 19,5 / ~~20,0~~ / 19,5

Hier bietet sich der Datentyp der verketteten Liste an, da einerseits die Anzahl der Einträge im Vorhinein nicht feststeht und andererseits Einträge wieder aus der Liste entfernt werden sollen.

## 2.3 Der Stapelspeicher (Stack)

Ein Stapelspeicher oder Stack ist eine dynamische Datenstruktur, mit der mehrere Datenobjekte erfasst werden können. Sie können nur oben auf den Stapel gelegt und auch nur von dort wieder abgerufen werden.

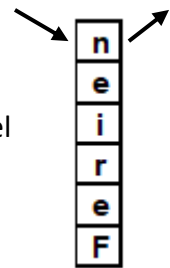
Beispiel: Ein Textverarbeitungsprogramm soll es ermöglichen, das jeweils zuletzt geschriebene Zeichen mit der Backspace-Taste (←) zu löschen.  
Da das zu löschende Zeichen immer das zuletzt geschriebene ist, bietet sich die Datenstruktur *Stapelspeicher* an

Geschriebener Text: Ferien|

←  
Löschen des letzten Zeichens.

Eintrag im Stapelspeicher:

letztes eingefügte Element: n  
Wird als erstes aus dem Stapel entfernt.



Dieses Prinzip wird auch als Last-in-First-Out-Prinzip (**LIFO**-Prinzip) bezeichnet.

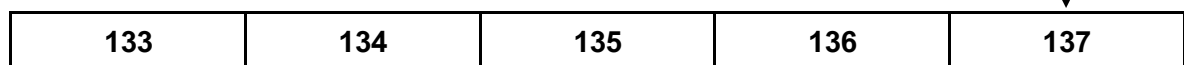
## 2.4 Die Warteschlange (Queue)

Eine Warteschlange ist eine dynamische Datenstruktur, mit der mehrere Datenobjekte erfasst werden können. Die Datenobjekte werden immer an das Ende der Warteschlange angefügt. Die Rückgabe der Datenobjekte erfolgt in der Reihenfolge ihres Einfügens. Das bedeutet, dass zuerst auf das Datenelement zugegriffen wird, das als erstes eingefügt wurde. Danach auf das, das als zweites eingefügt wurde, usw.

Beispiel: Beim Besuch einer Arztpraxis muss jeder Patient eine fortlaufende Nummer ziehen. Die Patienten werden dann in entsprechender Reihenfolge zur Behandlung aufgerufen. Hier bietet sich die Datenstruktur *Warteschlange* an, da der am längsten wartende Patient als erstes an die Reihe kommt. Seine Nummer steht am Anfang der Warteschlange.

1. Patient

letzter Patient



Als erstes wird der Patient mit der Nummer 133 aufgerufen.

Dieses Prinzip wird auch als First-in-First-Out-Prinzip (**FIFO**-Prinzip) bezeichnet.

## 2.5 Der Baum

Ein Baum ist eine dynamische Datenstruktur, die zur Abbildung hierarchischer Strukturen verwendet wird. Die Datenelemente werden dabei in Objekten, sogenannten Knoten, erfasst, die durch die Hierarchie vorgegeben sind. Jeder Knoten verweist auf die ihm untergeordneten Knoten. Bei den verweisenden Knoten wird von einem *Elternteil*, bei den untergeordneten Knoten von *Kindern* gesprochen. Alle Knoten bis auf den ersten haben einen Vorfahren. Dieser wird als *Wurzel* bezeichnet.

Beispiel: Die Holzwurm GmbH, möchte ihre Unternehmensstruktur digital erfassen. Der Geschäftsleitung sind die Abteilungen Beschaffung, Vertrieb und Verwaltung direkt unterstellt. Die Beschaffung gliedert sich in die Bereiche Einkauf und Lagerhaltung. Der Vertrieb in Marketing und Verkauf, die Verwaltung in Rechnungswesen, Personal und IT-Koordination.

Hier bietet sich die Datenstruktur *Baum* an, da die zu erfassenden Daten eine hierarchische, baumartige Beziehung untereinander haben.

