

M162 LB1

Lernziele

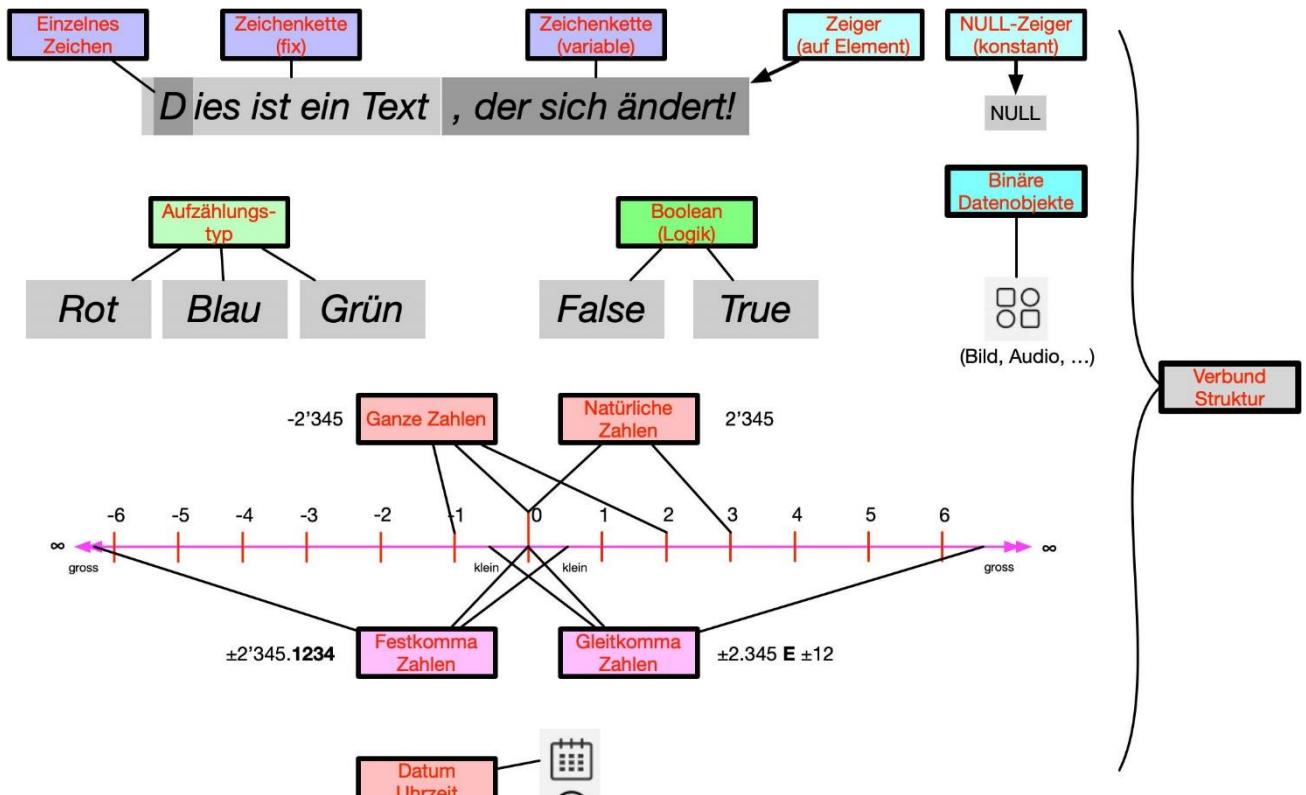
- (T) - Wissenstreppe
- (T) - Datentypen
- (T) - Datenstrukturen
- (T) - Fragetypen

-
- (P) - Excel: Statistik, Diagramme
 - (P) - Datengenerator -> txt, csv, JSON

Wissenstreppe

Auf Beiblatt ->

Datentypen



Ganze Zahlen

Bezeichnung: BIGINT, BIN, BIN FIXED, BINARY, BYTE, COMP, INT, INTEGER, LONG, LONG INT, LONGINT, MEDIUMINT, SHORT, SHORTINT, SMALLINT

Wertebereich: Meist 32 Bit ($-2^{31} \dots 2^{31}-1$), 16 Bit, 64 Bit

Operationen: +, -, *, <, >, =, Division mit Rest und Modulo

Natürliche Zahlen

Bezeichnung: BYTE, CARDINAL, NATURAL, UNSIGNED, UNSIGNED CHAR, UNSIGNED INT, UNSIGNED LONG, UNSIGNED SHORT, WORD

Wertebereich: Meist 32 Bit, (0...2³²-1), 8 Bit, 16 Bit, 64 Bit

Operationen: +, -, *, <, >, =, Division mit Rest und Modulo

Festkommazahlen (Dezimalzahlen)

Bezeichnung: COMP-3, CURRENCY, PACKED DECIMAL, DEC, DECIMAL, NUMERIC

Wertebereich: Wertebereich direkt abhängig von der maximalen Stellenanzahl, die meist vorzugeben ist; CURRENCY (64 Bit): -922'337'203'685'477.5808...922'337'203'685'477.5807

Operationen: +, -, *, <, >, =, Division mit Rest und Modulo

Aufzählungstypen

Bezeichnung: ENUM, SET oder implizit

Wertebereich: Frei wählbar, beispielsweise (SCHWARZ, ROT, BLAU, GELB)

Operationen: <, >, =

Boolean (logische Werte)

Bezeichnung: BOOL, BOOLEAN, LOGICAL, oder (implizit ohne Bezeichner)

Wertebereich: (TRUE, FALSE) oder (= 0, ≠ 0) oder (= -1, = 0)

Operationen: NOT, AND, XOR, NOR, NAND, OR, =, ≠

Zeichen (einzelnes Zeichen)

Bezeichnung: CHAR, CHARACTER

Wertebereich: Alle Elemente des Zeichensatzes (zum Beispiel Buchstaben)

Operationen: <, >, =, Konvertierung in INTEGER, ...

Gleitkommazahlen

Bezeichnung: DOUBLE, DOUBLE PRECISION, EXTENDED, FLOAT, HALF, LONGREAL, REAL, SINGLE, SHORTREAL

Wertebereich: Verschiedene Definitionen

Operationen: +, -, *, /, <, >, =

Zeiger

Bezeichnung: ACCESS, POINTER, IntPtr oder auch nur kurz Stern (*)

Wertebereich: Adresse des Basistyps (oft anonym)

Operationen: Referenz, Dereferenz, in einigen Sprachen: +, -, *, /

Konstanter Nullzeiger

Bezeichnung: NULL, VOID, None, NIL, Nothing

Wertebereich: keiner

Operationen: =

Bedeutung: Dieser Zeiger ist verschieden von allen Zeigern auf Objekte.

Zusammengesetzte Datentypen

Zeichenkette fester Länge

(Im Grunde sind Zeichenketten selbst nur eine Reihung des Typs Character (Zeichen). Da sie jedoch in vielen Programmiersprachen vordefiniert sind, werden sie hier gesondert aufgelistet.)

Bezeichnung: Array of CHAR, CHAR(n), CHAR[n]

Wertebereich: Alle möglichen Zeichenketten

Operationen: Stringfunktionen (Teilstring, Konkatenation [Zusammensetzung]), <, >, =

Zeichenkette variabler Länge

Die Länge ist feststellbar, implizit durch ein Metazeichen als String-Endezeichen (ASCII \0), explizit durch eine Variable, oder über eine Standardfunktion. Häufig als Abstrakter Datentyp in einer Standardbibliothek.

Bezeichnung: String, Array of CHAR, VARCHAR, CLOB, Text

Wertebereich: Zeichenketten variabler Länge

Operationen: Stringfunktionen (Teilstring, Länge, Konkatenation [Zusammensetzung]), <, >, =

Datum & Zeit

Bezeichnung: DATE, TIME, DATETIME, TIMESTAMP

Wertebereich: Verschiedene Definitionen. Oft dient die Anzahl Milisekunden, seit 1900 als Grundlage.

Operationen: Manchmal sind Operatoren wie +, - möglich, normalerweise werden Methoden verwendet. Vergleich möglich mit = oder Methode.

Format: Das Datumsformat wird normalerweise mit einer Zeichenkette definiert, z.B. "dd.MM.yyyy - HH:mm:ss" oder "d.M.yy / H:m"

Binäre Datenobjekte variabler Länge.

Die Länge kann durch eine Variable oder über eine Standardfunktion ermittelt werden. Bezeichnung: BLOB, Bild, Audio, Video, ...

Wertebereich: binäre Datenobjekte variabler Länge

Operationen: Länge, Konkatenation [Zusammensetzung], =

Verbund, Satz, Struktur, Bereich

Bezeichnung: RECORD, STRUCT, CLASS (erweiterte Bedeutung)

Wertebereich: Ein Verbund enthält eine Folge verschiedener Komponenten, welche verschiedene Datentypen haben können. Als Komponententyp ist jeder Typ zulässig. In einigen objektorientierten Programmiersprachen (zum Beispiel Oberon) können Verbunde zur Beschreibung des Verhaltens der Komponenten des Verbunds mittels Methoden auch typengebundene Prozeduren haben.

Operationen: Vergleich (nur Gleichheit oder Verschiedenheit), Zuweisung mit oder ohne Zuweisungskompatibilität (stark programmiersprachenabhängig)

Beispiel: type Prüfung is RECORD (Fach: STRING, Schueler: STRING, Punkte: INTEGER, Lehrer: STRING, Termin: DATUM)

Aufgabe 1

Welche Datentypen könnten den folgenden Werten zugrunde liegen?

Wert	Datentyp
A	CHAR, STRING, Aufzählungstyp
23	Natürliche Zahl, Ganze Zahl
12021.25	Dezimalzahl, Gleitkommazahl
0	Natürliche Zahlen, Boolean, Ganze Zahl
Ciara	String
12.2341235211	Dezimalzahl, Gleitkomma Zahl
true	Boolean
oben,unten	Aufzählungstyp
[12,53,23,22,55]	Array
.f},lúpÃ,ïïbõ©/=¹ryìøfÇâ´hðÂ.šžã´q	Blob
{name: "Meier", vorname: "Maxwell", alter: 21}	Verbund

Datenstrukturen

Text

Text ist eine „**Anhäufung**“ von **zusammengehörenden Zeichen** (ASCII, UTF-8, etc) die einen **Sachverhalt beschreiben**. Diese Form ist besonders gut geeignet für Volltextsuche. Text-Strukturen sind im Sinne der Strukturierung sehr schwach strukturiert. Da eine Darstellung in Textdateien bei einer Suche nicht hilft. Allerdings sind diese „Strukturen“ ganz leicht zu bearbeiten.

Typische Dateierweiterungen: TXT, JSON; XML, CSV, HTML

Grafik

Eine grafische Darstellung ist „nur“ eine **Momentaufnahme**. Sie besitzt gar keine Struktur. Von der elektronischen Speicherung gibt es zwei grundlegend verschiedene Möglichkeiten:

- die punktförmige Speicherung (**Pixelgrafik**) oder
- die berechnete Darstellung (**Vektorgrafik**). Hier wird jeder Punkt berechnet.

Die Grafik wird als gar nicht strukturierte Daten betrachtet. Unterschiede bei den Bildformaten:

Typische Dateierweiterungen: PNG, JPG, GIF, SVG

Video / Sound

Sowohl Video und Sound sind **analoge Signale, die digitalisiert werden**. Auch hier sprechen wir von KEINER Struktur.

Typische Dateierweiterungen: WAV, MP3, MP4, AVI, MPEG, MKV

Tabellen

Eine Tabelle ist eine **geordnete Zusammenstellung von Texten oder Daten**. Die darzustellenden Inhalte werden dabei in **Zeilen** (waagerecht) und **Spalten** (senkrecht) gegliedert, die aneinander ausgerichtet werden. Die erste Zeile nennt man oft Kopfzeile oder Spaltenbeschriftung. Weiterhin bestehen Tabellen aus Tabellenfeldern (**Zellen**). Wir verstehen unter dem Begriff Tabelle mehrere Bedeutungen:

1. Eine **Tabellenkalkulation** ist ein Computerprogramm, das vor allem zur Darstellung und Berechnung auch hypothetischer Zahlendaten in großem Umfang geeignet ist.
2. Bei **relationalen Datenbanken** steht der Ausdruck Tabelle für eine Sammlung ähnlich strukturierter Daten, die technisch meist zusammengehörig gespeichert sind. Sie hat je nach System unterschiedliche Bezeichnungen: Tabelle, Entität, Relation.

Tabellen sind mässig oder stark strukturiert. Ihre Struktur ist für einen Fokus generiert und kann anschliessend nur mit viel Aufwand verändert werden. Innerhalb dieses Fokus ist die Tabelle sehr gut geeignet um zu sortieren und Auszuwerten.

Datensatz / Record / Tupel

Datensätze (auch 'Tupel' oder engl. Record genannt) gehören zu den einfachsten Datenstrukturen. **Sie verkörpern Werte, die andere Werte enthalten**, üblicherweise in einer **fest definierten Anzahl und Folge**. Datensätze identifizieren sich meist durch eines oder mehrere der in ihnen enthaltenen Elemente, oft Datenfeld genannt.

Beispiele:

- (ID=12, Vorname=Hans, Nachname=Wenger, Telefonnummer=07911111111, Adresse=Zuercherstrasse 12)
- [Länge=4.5, Song=Patience, Band=Guns'n'Roses]

Ob die Klammern rund oder eckig sind, spielt hier keine Rolle. Wichtig ist, dass die Daten thematisch zusammen gehören und normalerweise mehrere Tuples des gleichen Typs existieren. Ein Tuple/Record kann als eine Zeile einer Tabelle angeschaut werden.

Array

Das Array ist die einfachste verwendete Datenstruktur. Es werden hierbei **mehrere Variablen vom selben Datentyp gespeichert**. Ein **Zugriff auf die einzelnen Elemente wird über einen Index möglich**. Jedes Feld kann wieder ein neues Array enthalten und bekommt dadurch eine zweite Dimension (zweidimensionales Array). Arrays sind aber keinesfalls nur auf zwei Dimensionen beschränkt, sondern werden beliebig mehrdimensional verwendet. Wegen ihrer Einfachheit und grundlegenden Bedeutung bieten die allermeisten Programmiersprachen eine konkrete Umsetzung dieser Datenstruktur als zusammengesetzten Datentyp Array im Grundsprachumfang an.

Einen Sonderfall bildet das assoziative Array (oft auch Dictionary genannt), bei dem nicht (nur) über einen numerischen Index, sondern über einen Schlüssel auf die gespeicherten Daten zugegriffen wird. Eine mögliche Art, ein assoziatives Array zu implementieren, ist die Hashtabelle.

Beispiele:

- [12, 45, 23, 38, 28]
- ["Hans", "Werner", "Sabine", "Rafael", "Susanne"]
- ["pos01" = "Position 01", "pos02" = "Position 02", ...] => Hier handelt es sich um ein assoziatives Array und darum muss der Schlüssel mit angegeben werden. Also an der Position "pos01" befindet sich der Wert "Position 01"

Verkettete Liste

Die verkettete Liste ist eine Datenstruktur zur **dynamischen Speicherung von beliebig vielen Objekten**. Dabei beinhaltet jedes Listenelement einer verketteten Liste als Besonderheit einen **Verweis auf das nächste Element**, wodurch die Gesamtheit der Objekte zu einer Verkettung von Objekten wird. Die zu einer Liste gehörenden Operationen sind relativ unspezifiziert. Sie werden oft in komplizierteren Datenstrukturen verwendet und meist direkt auf ihre Elemente zugegriffen.

Die Blockchain-Technologie verwendet eine solche Verknüpfung von Elementen (Blöcke).

Stapelspeicher

In einem Stapelspeicher (engl. stack oder ‚Kellerspeicher‘) kann eine beliebige Anzahl von Objekten gespeichert werden, jedoch können die gespeicherten Objekte **nur in umgekehrter Reihenfolge wieder gelesen werden**. Dies entspricht dem **LIFO-Prinzip**. Zu einem Stapelspeicher gehören zumindest die Operationen

- push, um ein Objekt im Stapelspeicher abzulegen und
- pop, um das zuletzt gespeicherte Objekt wieder zu lesen und vom Stapel zu entfernen.

Weitere Optionen können top oder peek sein, um das oberste Element zu lesen, ohne es zu löschen.

Warteschlange

In einer Warteschlange (engl. queue) kann eine beliebige Anzahl von Objekten gespeichert werden, jedoch können die gespeicherten Objekte **nur in der gleichen Reihenfolge wieder gelesen werden**, wie sie gespeichert wurden. Dies entspricht dem **FIFO-Prinzip**. Zu einer Queue gehören zumindest die Operationen

- enqueue, um ein Objekt in der Warteschlange zu speichern und
- dequeue, um das zuerst gespeicherte Objekt wieder zu lesen und aus der Warteschlange zu entfernen.

Vorrangwarteschlange

Eine Spezialisierung der Warteschlange ist die Vorrangwarteschlange, die auch Prioritätswarteschlange bzw. engl. **Priority Queue** genannt wird. Dabei wird vom FIFO-Prinzip abgewichen. Die Durchführung der enqueue-Operation, die in diesem Fall auch insert-Operation genannt wird, **sortiert das Objekt gemäß einer gegebenen Priorität**, die jedes Objekt mit sich führt, in die Vorrangwarteschlange ein. **Die dequeue-Operation liefert immer das Objekt mit der höchsten Priorität**. Vorrangwarteschlangen werden meist mit Heaps implementiert.

Graph

Ein Graph ermöglicht es als Datenstruktur die Unidirektionalität der Verknüpfung zu überwinden und können **Referenzen auf mehrere Objekte** enthalten. Die Operationen sind auch hier das Einfügen, Löschen und Finden eines Objekts.

Baum

Bäume sind **spezielle Formen von Graphen** in der Graphentheorie. **Bäume besitzen nur eine eingehende Verknüpfung, aber mehrere ausgehende Verknüpfungen**.

Eine **Spezialform ist der Binärbäumen**, bei dem die Anzahl der Kinder höchstens zwei ist und in höhen-balancierten Bäumen angelegt wird, bei dem sich die Höhen des linken und rechten Teilbaums an jedem Knoten nicht zu sehr unterscheiden.

Bei **geordneten Bäumen, insbesondere Suchbäumen**, sind die Elemente in der Baumstruktur **geordnet abgelegt, sodass man schnell Elemente im Baum finden kann**.

Heap

Der **Heap** (auch Halde oder Haufen) **vereint die Datenstruktur eines Baums mit den Operationen einer Vorrangwarteschlange**. Häufig hat der Heap neben den minimal nötigen Operationen wie insert, remove und extractMin auch noch weitere Operationen wie merge oder changeKey.

Hashtabelle

Die Hashtabelle bzw. Streuwerttabelle ist eine **spezielle Indexstruktur, bei der die Speicherposition direkt berechnet werden kann**. Hashtabellen stehen dabei in Konkurrenz zu Baumstrukturen, die im Gegensatz zu Hashtabellen alle Indexwerte in einer Ordnung wiedergeben können, aber einen größeren Verwaltungsaufwand benötigen, um den Index bereitzustellen. Beim Einsatz einer Hashtabelle zur Suche in Datenmengen spricht man auch vom Hashverfahren. Bei sehr großen Datenmengen kann eine verteilte Hashtabelle zum Einsatz kommen.

Repetitionsfragen:

Sie müssen eine Adressliste verwalten mit mehreren 100 Datensätzen. Welche Strukturen kommen in Frage und warum?

In diesem Fall würde ich mich für eine Hashtabelle entscheiden. Bei so vielen Daten kann man die Hashfunktion anwenden also man kann Einträge suchen.

Sie müssen viele Werte speichern und mit wenig Zeitverlust darauf zugreifen können. Welche Strukturen kommen in Frage und warum?

In diese Fall würde ich mich für eine Baum oder Hashtabelle entscheiden, da man auf diese Daten sehr schnell zugreifen kann. In Hashtabelle sogar suchen

Sie müssen einen Sortieralgorithmus für Zahlen programmieren. Welche Strukturen kommen in Frage und warum?

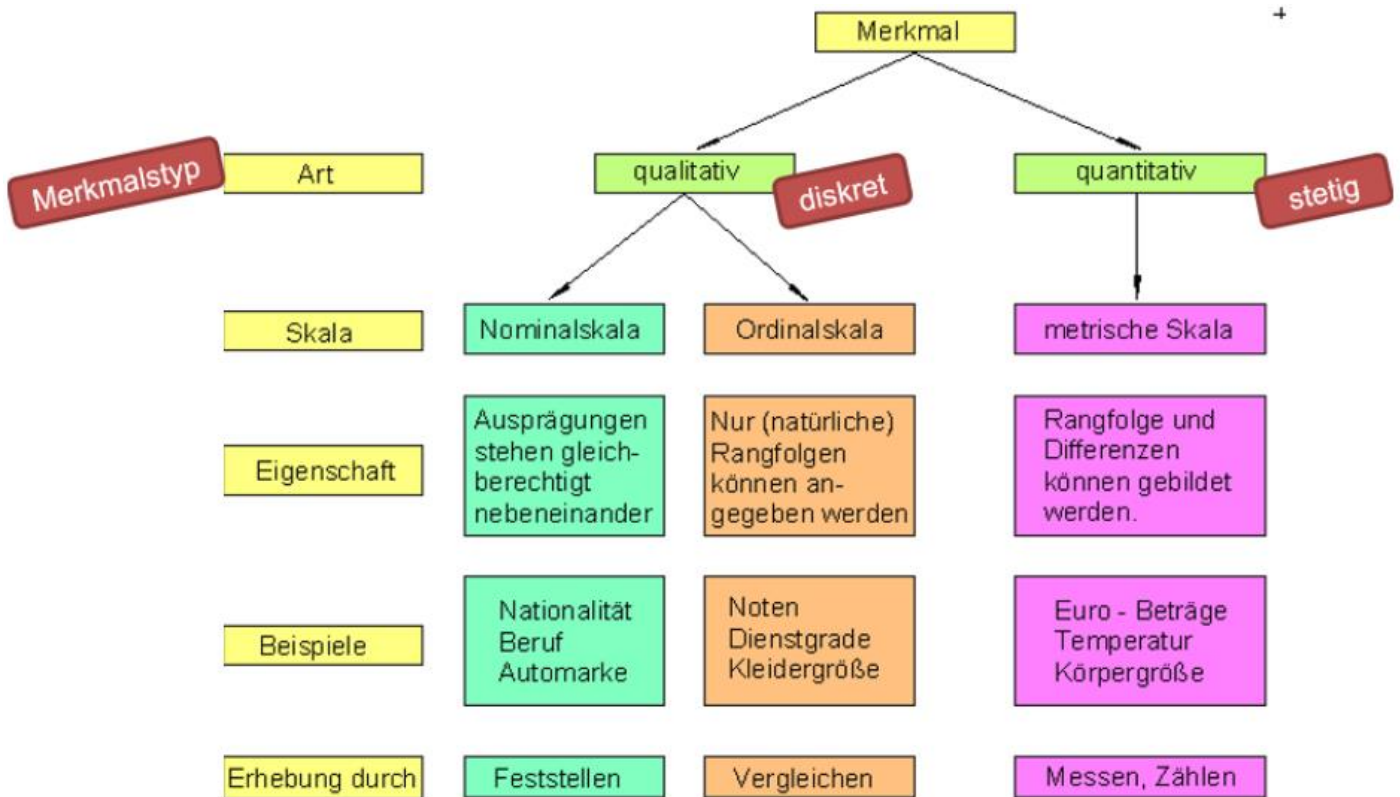
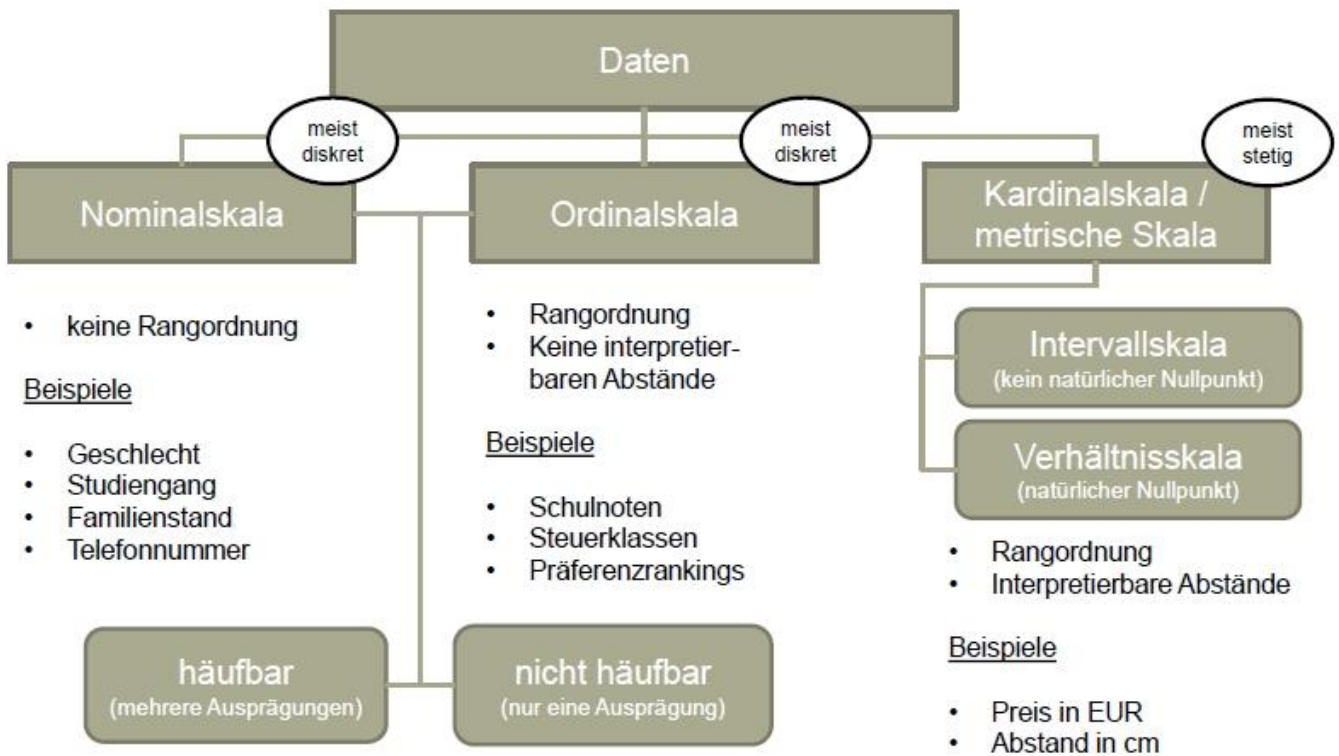
Die verkettete Liste ist eine Datenstruktur zur dynamischen Speicherung von beliebig vielen Objekten. So kann man immer wieder Sachen hinzufügen und löschen und alles bleibt ersichtlich.

Die Universität Zürich hatte in jedem Semester das Problem, dass ihre Server überlastet waren während der Modulregistrierung in neuen Semestern. Jeder Student wollte so schnell als möglich einen Platz für spezifische Vorlesungen reservieren, weil die Plätze jeweils limitiert sind. Mit welcher Struktur kann man dieses Problem lösen (und wurde so auch gelöst)

Mit einer Warteschlange (Queue) oder einer "Priority Queue", da der erste der schnellste ist. Wenn eine Reservation gelöscht wird, kann jemand erneut reservieren

Zweite Option wär ein Stapelspeicher. Die ersten sind auch die ersten von unten im Stapel.

Fragetypen



	schwach strukturiert	mässig strukturiert	stark strukturiert
Daten:	beliebiger Text	Tabellen, Listen	Adressen, Lagerverwaltung usw.
Suchen:	schlecht	gut nach sortiertem Kriterium	nach vielen Kriterien möglich
Werkzeug:	gedruckte Unterlagen	Karteikasten	Archiv
Einfügen:	beliebige Position, einfach	nach Position sortiert	Position und Beziehungen, aufwendig
elektronische Verarbeitung:	Suchmaschine	Tabellenkalkulation	Datenbank
Software-Tools	Word	Excel	Access / Workbench

Merkmal	Merkmalsausprägung	Skala	diskret/ stetig
Geschlecht	m, w	Nominal	diskret
Sieger Wettbewerb	1, 2, 3 Rang	Ordinal	diskret
Arbeitsweg	1...100 km	Metrisch	stetig
Schulnote	1, 2, 3...6	Ordinal	diskret
Temperatur	1°90°C	Metrisch	stetig

Nominal- und ordinalskalierte Merkmale = diskret
 Metrisch skalierte Merkmale = stetig oder diskret

Praxis LB1

Anschauen: https://gitlab.com/ch-tbz-it/Stud/m162/-/blob/v2_by_topics/Auswertungen_Excel/Excel.md

Diagramm erstellen in Excel

Histogramm

Studyflix

Mittelwert

Modus

Median -----> [Einführung Statistik V2.5.pdf](#)

Standartabweichung