Informatik 09

Basisinformationen zu Informatik 09. In der Jahrgangsstufe 9 ist der Einstieg das Arbeiten mit Funktionen und Datenflüssen (anhand von Calc/ Excel) und das zentrale Themengebiet die Objektorientierte Programmierung (OOP) mit Python und die Datenbank-Verknüpfung mit SQL.

Aufgaben in Informatik 9

- **Grundlagen der Informatik** (Wdh)
  Klasse, Objekt, Attribut, Methode, Klassen- und Objektkarte, Klassendiagramm, HTML (Wdh)
- **HTML** (Wdh) webSeite mit Wordpress (wordpress.com) o.a. editieren. NEU: **CSS, js (Jquery), XML**
  codepen.io | **CodePen** online editor
- LibreOffice Calc bzw. Microsoft Excel (Fkt)
- Aufgaben aus dem Informatik-Buch der Reihe nach selbständig bearbeiten. Das Buch ist dafür zur Informatik-Stunde mitzubringen. Schreibe Deinen Lösungsvorschlag in Dein Heft.

- **Funktionale Programmierung**: Lege EINE Datei an mit mehreren Blättern (tabs unten).
- opt: Aufgabe Schreibtraining: Übe konzentriert, z.B. jede Unterrichtsstunde, zu Stundenbeginn ca 5 min.
- Referat Informatik (Präsentation) - 5 min Vortrag mit Hefteintrag (> alle)

- **OOP** (Objektorientierte Programmierung) mit **Processing**: processing.org (Vereinfachte Java-Version) bzw. openprocessing.org
  Processing ist eine objektorientierte, stark typisierte Programmiersprache mit zugehöriger integrierter Entwicklungsumgebung. Die Programmiersprache ist auf die Einsatzbereiche Grafik, Simulation und Animation spezialisiert. Processing wird in einem quelloffenen Projekt entwickelt, das am Massachusetts Institute of Technology in Boston von Ben Fry (Broad Institute) und Casey Reas (UCLA Design|Media Arts) initiiert wurde. Processing hat den Charakter einer stark vereinfachten Version der Programmiersprache Java, ermöglicht Interaktionen und visuelle Elemente zu programmieren und richtet sich vorwiegend an Gestalter, Künstler und Programmieranfänger.
- **OOP** mit **Python**: Online-Kurs z.B. auf cscircles.cemc.uwaterloo.ca/de
- **Datenbanken/** Datenmodellierung: Lege ZWEI Datenbanken (DB) an mit den benötigten Tabellen (LibreOffice Calc, Microsoft Access) zu den Themen "Musik-DB" und "Photo/ Filmarchiv".
  Digitales Audio, vgl. learningmusic.ableton.com und learningsynths.ableton.com
  Bild Archiv, vgl. architonic.com, metmuseum.org, instagram.com, typicon.de
- https://metmuseum.org/about-the-met/policies-and-documents/open-access
- **Datenmodellierung**: ZWEI Datenbanken (DB) mit den benötigten Tabellen (LibreOffice Calc, Microsoft Access) zu den Themen "Musik-DB" und "Photo/ Filmarchiv" anlegen
  (wenige (auch fiktive) Datensätze reichen für eine SQL-Abfrage
- SQL Lite (py), LibreOffice Base, Microsoft Access (DB)
  - Aufgaben aus dem Informatik-Buch der Reihe nach selbständig bearbeiten. Das Buch ist dafür zur Informatik-Stunde mitzubringen. Schreibe Deinen Lösungsvorschlag in Dein Heft.
  - Werkzeugkasten-Kapitel im Buch. Lies sie dir selbständig durch (ab S. 142 in Informatik I, Oldenbourg-Verlag)
- Digitales Audio, vgl. learningmusic.ableton.com (zur Klärung eines persönlich musikalisch favorisierten Klassendiagramms mit relevanten Attributen)
- https://metmuseum.org/about-the-met/policies-and-documents/open-access (Kunstgeschichts-Bilder mit public-domain-Lizenz)

Mögliche Referate Informatik 9 (5 min Vortrag, ca. 20+ Präsentationsfolien)

wichtig ist hierbei jeweils:

Beschreibung | Definition | Beispiel | Abbildung mit Quellenangaben für Abbildungen und Text.

Jedes Bild hat eine **Bildunterschrift**, z.B.: Abb. 1: Bildname/Beschreibung/Beispiel Quellenangabe

Bei jedem Referat soll am Ende ein kurzer Hefteintrag (1-3 Folien zum Abschreiben, Graphik oder Abbildung zum Abzeichnen) die letzte Seite sein.

Grundlagen der Informatik | basic

Geschichte/ Meilensteine der Informatik
Klasse Objekt Attribut Methode. Punktnotation
Klassenkarte Objektkarte Klassendiagramm | py
Beziehungen und Kardinalitäten
Methoden | py
HTML | atom.io oder notepad++
Algorithmus/ algorithmisches Denken
Struktogramme
Baum und Graph | Hierarchische Strukturen
Binäres Zahlensystem (vgl. S. 63)
Rechnerarchitektur

Open Source
CC, Copyright und Lizenzen
KI - Künstliche Intelligenz
Big Data - Überwachung, Online-Tracking, Datenschutz
Serious Games, technologiegestütztes Lernen

Funktionale Programmierung | py

Funktionen
Vordefinierte Funktionen (vgl. S. 39)
Logische Funktionen *UND*, *ODER*, *NICHT* - AND, OR, NOT
Logische Funktionen/ Logikgatter - NAND, NOR, XOR, XNOR
Datenflussdiagramm
Bedingte Funktion: *WENN*-Funktion
while-Schleife
if-/else - Bedingte Anweisung
Zusammenfassen von Funktionen
Verknüpfung von Tabellen
Verzweigung im Datenfluss
Geschichte der Tabellenkalkulation (vgl. S. 62)
Tabellenkalkulation - Buchhaltung
Diagramme aus Tabellendaten
Kapital-Berechnung (Kredit-Aufnahme, Geld-Anlegen, Zinsberechnung mit Zinstagen - Beginn- u. Ende Datum; Kostenkalkulationen)
Zusammengesetzte Daten (z.B. Brüche)
Verkaufspreis mit Mengenrabatt
Verkaufspreis mit Skonto
Kostenvoranschlag (Rechnung) mit Tabellenkalkulation
Adress-Listen mit Tabellen
Diagramme in Tabellenkalkulation: Entwicklung der Weltbevölkerung (vgl. S. 61)
Kopieren - relative und absolute Adressen
Links - lokal und online z.B. in Tabellen

Informatik und python

if/else - Bedingung
while - Schleife
Objekte | py
Klassen | py z.B. Liste (unsortiert, sortiert), Liste einfach verkettet, Schlange, Stapel, Baum, Graph (ungewichtet, gewichtet)
Klassenkarten | py
Objektkarten | py
Klassendiagramm
Objektdiagramm
Zustandsdiagramm
Variablen | py
Methoden | py
Anweisungen | py
Strings (Zeichenketten) | py
Vererbung | py
Verzweigung/ Rekursion
Bsp. Koch'sche Schneeflocke
Bsp. Sierpinski-Dreieck

Bsp. Fibonacci-Funktion
Lastenheft/ Pflichtenheft in der Informatik
Agile Softwareentwicklung (vs Wasserfall-Modell)
Objektorientierte Programmierung - Diff. python, Java
Biometrische Daten als Schlüssel
Künstliche Intelligenz (Artificial Intelligence AI)
Intuitive Bedienbarkeit/ User-Dialog/ GUI - Graphische Oberfläche und optisches Design
Eclipse-Programmierumgebung
Android-Programmierung
python-Turtle-Programmierung

_____(eigener Themenvorschlag)

Datenbanken | py

DB statt Fkt - warum?
Datentypen | py
Relationales Datenbanksystem
SQL-Abfragen
Datenbankschema
Datenpflege und Änderungsanomalien (UPDATE, INSERT, DELETE)
Datensicherheit und Datenschutz
Klassendiagramm und Kardinalitäten
n:m - Beziehung
Das kartesische Produkt
Aggregat-Funktionen von SQL (COUNT, AVG, MAX, MIN, SUM vgl. S. 103)
Schlüssel, Fremdschlüssel und referentielle Integrität
Datenpflege (Ändern, Einfügen, Löschen von Datensätzen | UPDATE, INSERT, DELETE)
Datenkonsistenz
Datensicherheit und Datenschutz
Datenbankschema
Datenbank-Design
Prüfziffer am Bsp. der ISBN-Nummer

Informatik 09

Grundwissen Informatik 06: Klasse, Objekt, Attribut, Methode

Objekte und Klassen

Objekt

Man kann die analoge und auch digitale Welt in Objekten verschiedener Art beschreiben. In Vektorgrafiken finden sich Linien, Kreise, Rechtecke oder Textblöcke; in (reinen) Rastergrafiken dagegen nur Objekte der Art Bildpunkt. Jedes Objekt benötigt zur eindeutigen Identifizierung einen innerhalb des jeweiligen Dokumentes eindeutigen Bezeichner, z.B. *kreis1*, *rechteck7* oder *linie13*.

Attribut

Die Eigenschaften der Objekte werden durch die Werte ihrer Attribute beschrieben. Bsp. (in Punktschreibweise):

Objektname.Attributname=Wert

kreis1.füllfarbe=rot #rot ist der Attributwert
Klasse:KREIS

Für eine Attributwert-Zuweisung schreiben wir allgemein:
Objektname.Attributname=Wert

Klasse

Je nach Art der Attribute teilt man Objekte in Klassen ein. Objekte der gleichen **Klasse** haben dieselben Attribute und Methoden. Eine Klasse stellt einen **Konstruktionsplan** für bestimmte Objekte dar, der mit all seinen Informationen auch ohne diese Objekte existiert.

Objekt:KLASSE

Methode

Objekte können auf Befehl bestimmte Operationen (Methoden) ausführen. Diese Operationen werden durch den Aufruf einer Methode ausgelöst. Bsp.:

Objektname.Methodenname(Wert)

buchstabe1.horizontalSpiegeln()
buchstabe5.horizontalVerschieben(2cm) #2cm ist der Methodenwert
Klasse: BUCHSTABE

Für einen Methodenaufruf schreiben wir allgemein:

Objektname.Methodenname(Wert)

Parameter

Bei vielen Methoden muss man durch ein oder mehrere Argumente (Parameter) festlegen, mit welchen Eingaben sie ausgeführt werden sollen, z.B. um welche Strecke ein Objekt verschoben werden soll. Die Parameter werden in einer festgelegten Reihenfolge in Klammern notiert. Bsp.:
buchstabe1.verschieben(2cm, -3cm)

Variable

Klassenkarte, Klassendiagramm

Objektkarte > Zustände

Algorithmus

Forderungen an einen Algorithmus:

Eindeutigkeit: ein Algorithmus darf keine widersprüchliche Beschreibung haben. Diese muss eindeutig sein.

Ausführbarkeit: jeder Einzelschritt muss ausführbar sein.

Finitheit (= Endlichkeit): die Beschreibung des Algorithmus muss endlich sein.

Terminierung: nach endlich vielen Schritten muss der Algorithmus enden und ein Ergebnis liefern.

Determiniertheit: der Algorithmus muss bei gleichen Voraussetzungen stets das gleiche Ergebnis liefern.

Determinismus: zu jedem Zeitpunkt der Ausführung besteht höchstens eine Möglichkeit der Fortsetzung. Der Folgeschritt ist also eindeutig bestimmt.

Modellierung in der Informatik

reales System: **Systemanalyse**. abgrenzen, abstrahieren, idealisieren, zusammenfassen
mentales Modell: **Modellbildung**
reales Modell: Normierte Darstellung, **Programm**

> neue Erkenntnisse (über das reale System)

Modellbilden bedeutet:
- einen Ausschnitt aus dem realen System wählen und alle Einflüsse von außen weglassen,
- nur die wichtigen Dinge in diesem Ausschnitt betrachten.
- diese wichtigen Dinge so einfach wie möglich und so umfassend wie nötig wiedergeben,
- das Modell mit einer normierten Darstellungsform beschreiben.

Textverarbeitung (II)

- Serienbriefe (u. a. Seriendruckfelder, Bedingungen, Datenquellen, Hauptdokument)
- Privatbrief (u. a. Aufbau, Vorlagenerstellung, Privatanschriftfeld)
- Geschäftsbriefe (u. a. Aufbau, Vorlagenerstellung, Anschriftfeld, Briefschluss)
- Unterscheidung zwischen Privat- und Geschäftsbrief
- Autotexte, Textbausteine
- (Online-)Formulare
- Bewerbungsunterlagen (u. a. Online-Bewerbung)
- Layout anspruchsvoller Dokumente (u. a. Absatz-, Zeichen-, Seitenformatierungen, Tabellen, Verknüpfungen, Formatvorlagen, Inhaltsverzeichnis)
- Objekte der Textverarbeitung, z. B. Textfelder, Tabulatoren, Spalten, Umbrüche
- Funktionen eines Textverarbeitungsprogramms, z. B. Nummerierung, Aufzählung, Feldfunktionen wie Datum und Seitenzahl
- Automatisches Inhaltsverzeichnis und Fußnoten

(Fkt) Funktionale Modellierung | Literatur-Recherche

Praxis-Projekt Literatur-Recherche | Inhalt: individuell

Autor-Nachname, Vorname, Buchtitel, Jahr, Verlag, Inhalt-Überblick, Notizen, Anmerkung
bzw.
Heft-Name, Jahr, Heft-Nr., Inhalt-Überblick, Notizen, Anmerkung

Funktionale Modellierung | Funktionen als Prozessbeschreibungen

- grundlegende Funktionsweise eines Tabellenkalkulationsprogramms
- Modelle zur Analyse und Lösung von Aufgaben, z. B. Struktogramm, Datenflussdiagramm, Aktivitätsdiagramm
- Datentypen, z.B. Text, Zahl, Datum
- Formeln und ihre Bestandteile
- relative und absolute Zelladressierung
- einfache Funktionen und ihr Aufbau, z. B. zur Berechnung von Minimum, Maximum, Summe, Mittelwert
- verschiedene Diagrammtypen, z. B. Kreis-, Säulen-, Liniendiagramm
- Modelle (z. B. Struktogramm, Datenflussdiagramm, Aktivitätsdiagramm) zur Analyse und Lösung von Aufgabenstellungen mit diversen Funktionen
- zweiseitige Auswahlstruktur
- Sortierfunktion

- erweiterte Formatierungsmöglichkeiten: bedingte Formatierung, benutzerdefinierte Zahlenformate
- Verknüpfung von Zellinhalten über mehrere Tabellenblätter
- mehrstufige und mehrseitige Auswahlstrukturen
- Wahrheitswert, logische Funktionen und ihre Verknüpfungen
- Gültigkeitsprüfung für Eingabewerte (Validierung)
- Filterung von Daten
- Schutz von Zellen und Tabellenblättern
- weitere Funktionen, z. B. aus den Bereichen Mathematik, Statistik, Finanzen, Datum und Uhrzeit

Viele **Prozesse** in Arbeitswelt und Technik lassen sich wegen ihrer Komplexität nur dadurch überschauen, dass man sie in Teilprozesse gliedert. In einem geeigneten Diagramm stellt man diese **Teilprozesse** dar und beschreibt den Materialfluss zwischen ihnen.

In der Informatik werden **Daten** verarbeitet. Ein Datum (Datenelement) umfasst dabei sowohl die **Darstellungsform** als auch die zugehörige **Bedeutung**.

Eine **Funktion** beschreibt einen klar umrissenen Vorgang (Prozess) innerhalb eines größeren Zusammenhangs. Sie ermittelt aus **Eingabewerten** nach einer festgelegten und **eindeutigen Zuordnungsvorschrift** einen **Ausgabewert**.
Jede Funktion hat einen Bezeichner und eine Zuordnungsvorschrift. Die Platzhalter für die nötigen **Eingabewerte** werden mit eindeutigen Bezeichnern festgelegt und heißen **Eingangsparameter**.
Schreibweise:

> *Funktionsname*(Eingangsparameter1; Eingangsparameter2; Eingangsparameter3; ...)

Beim Aufruf der Funktion werden die Eingangsparameter entsprechend ihrer Reihenfolge durch aktuelle Eingabewerte ersetzt. Mit diesen Werten wird der **Ausgabewert** bestimmt.

Logische Funktionen UND, ODER, NICHT

Mit Wahrheitswerten kann man logisch rechnen.
Melde Dich, WENN du die Antwort weißt ODER eine Frage hast.
Melde Dich, WENN jemand eine Frage gestellt hat UND du die Antwort kennst.
Frag nach, WENN du etwas NICHT verstanden hast.

Präs2 | Tabellenkalkulationssystem. Die Klasse ZELLE. **Datentypen**
Präs3 | **Zusammenfassen von Funktionen**. Treppenvolumen berechnen.
Präs4 | **Verzweigung** im Datenfluss. Malerkosten berechnen.
Präs5 | Vordefinierte Funktionen. ANZAHL. SUMME. **MITTELWERT**.
Präs6 | Zusammengesetzte Daten. **Bruchaddition**.
Präs7 | Bedingte Funktionen. **Mengenrabattsatz**.

Datenflussdiagramm
Zur Berechnung komplexer Zusammenhänge lassen sich Funktionen kombinieren. Dabei wird der Ausgabewert einer Funktion zum Eingabewert einer anderen Funktion. In der grafischen Darstellung entsteht ein **Datenflussdiagramm**.
Elemente eines Datenflussdiagramms sind:

- Funktionen, die Daten verarbeiten (Rechtecke, abgerundet)
- Verbindungspfeile (zeigen von Datenflüssen). Bei vielen Tabellenkalkulationsprogrammen kann man sich die Datenflusspfiele aufgrund der Zuordnungsvorschriften nachträglich automatisch eintragen lassen.
- Eingabewerte (Quellen)
- Ausgabewert (Senke)

Bei der **funktionalen Modellierung** werden komplexe Prozesse in einzelne Teilprozesse zerlegt und der Datenfluss zwischen den Prozessen analysiert. Das Datenflussidagramm ist eine grafische Darstellungsform für ein funktionales Modell.
Das funktionale Modellieren besteht in erster Linie aus den Schritten:

- Identifizieren der Eingabewerte und des Ausgabewertes des Gesamtprozesses
- Bestimmen der Teilprozesse und Datenflüsse
- Erstellen des Datenflussdiagramms

- Beschreiben der Teilprozesse durch Zuordnungsvorschriften.

Hinweis: Beim Erstellen eines Datenflussdiagramms betrachtet man jeden Teilprozess im ersten Moment als sogenannte "Blackbox". Das heißt, man weiß zwar, was der Teilprozess machen soll, aber noch nicht, wie dies im Detail durchzuführen ist. Erst im letzten Schritt beschäftigt man sich bei der funktionalen Modellierung mit dem "wie", also dem Entwickeln einer geeigneten und detaillierten Zuordnungsvorschrift.

Tabellenkalkulationssysteme

Die Dokumente einer Tabellenkalkulation enthalten **Rechenblätter**. Ein Rechenblatt enthält Objekte der Klassen ZELLE, ZEILE und SPALTE.
Der **Datentyp** einer Zelle legt fest, welche Werte das Attribut **Zellwert** annehmen kann und welche Operationen (Rechenverfahren) auf diesen Werten möglich sind. Wichtige Datentypen in Tabellenkalkulationsprogrammen sind (Komma-)Zahl, Text, Zeitangabe, Wahrheitswert.
Der Wert des Attributes Zellwert einer Zelle kann eine vom Benutzer eingegebene **Konstante** oder das Ergebnis der Ausführung einer Zuordnungsvorschrift sein. Diese Zuordnungsvorschrift wird im Attribut **Formel** abgelegt, sie beginnnt mit einem "="-Zeichen und kann als formale Parameter die **Adressen** anderer Zellen enthalten.

Datentypen

Eine Folge von Ziffern kann manchmal auch ein Text sein. Je nach festgelegtem Datentyp wird die Zeichenfolge "123" als die Zahl einhundertdreiundzwanzig oder als der Text eins-zwei-drei behandelt.
Auch die Zeitangaben werden intern als Zahl gespeichert, damit sie für Berechnungen verwendet werden können. Wahrheitswerte (logische Werte, boolsche Werte) werden ebenfalls intern als Zahlen dargestellt.

Datentypen (in python)

- für ganze Zahlen: `int` (integer) und `long`
- für Gleitkommazahlen: `float (Kommazahl) und double`
- für komplexe Zahlen: `complex`
- für boolsche Werte: `bool (Wahrheitswert true/false)`
- einzelnes Zeichen: char (character)
- Wort, Zeichenkette: string

Primitive Datentypen - Übersicht

Python, Java

		Typname	Beschreibung	Länge in Byte
		boolean	Boole'scher Wert (wahr oder falsch, 1 Bit)	1
		char	einzelnes Zeichen (16 Bit)	2
		byte	ganze Zahl (8 Bit)	1
		short	ganze Zahl (16 Bit)	2
		int	ganze Zahl (32 Bit)	4
		long	ganze Zahl (64 Bit)	8
		float	Fließkommazahl (32 Bit)	4
		double	Fließkommazahl (64 Bit)	8

Referenztypen (Objekttypen)

Python, Java
Zu den Referenztypen gehören Objekte, Strings und Arrays.

Typname	Beschreibung	Bsp
String	Zeichenkette (Text)	"Hallo!"
Array (z.B. int[])	Feld (hier ganzzahlig)	{1, 2, 3, 4, 5}

Stings und Arrays sind streng genommen auch Objekte, können aber (in Java) ohne Aufruf des new-Operators erzeugt werden.

Exkurs: Python als Taschenrechner

Für numerische Datentypen sind folgende arithmetische Operatoren definiert:

- Addition: `x+y`
- Subtraktion: `x-y`
- Multiplikation: `x*y`
- Division: `x/y`
- Rest beim ganzzahligen Teilen: `x%y`
- Ganzzahliger Anteil der Integer Division: `x//y`
- Potenzieren: `x**y`
- Negatives Vorzeichen: `-x`

In Python gibt es **keine** Operatoren für das Inkrementieren (`x++`) und Dekrementieren (`x--`). Es sind jedoch sogenannte *erweiterte Zuweisungen* der Form `x+=y` als kürzere Form für `x=x+y` möglich.

hdm-stuttgart.de/~maucher/Python/html/Datentypen.html

Zusammenfassen von Funktionen

Mehrere Funktionen in einem Datenflussdiagramm kann man zu einer Funktion zusammenfassen und diese mit Eingangsparameter (ggf. mehreren) und einem Ausgang versehen.

Die Termschreibweise ist eine mathematische Darstellungsform eines funktionalen Modells. Bei dieser Schreibweise werden die einzelnen Funktionen entsprechend des Datenflusses zu einem Term zusammengesetzt.

Verzweigung im Datenfluss

Mit dem Verteiler-Symbol wird ein Datenfluss im Datenflussdiagramm verzweigt, wobei in beiden Zweigen dieselben Daten fließen. In der Termnotation erfolgt diese Verzweigung durch Mehrfachverwendung desselben Parameterbezeichners.

Vordefinierte Funktionen

Tabellenkalkulationsprogramme stellen **vordefinierte Funktionen** bereit. Einige vordefinierte Funktionen haben eine beliebige Anzahl Eingangsparameter. In diesem Fall können **Zellenbereiche** angeben werden.
Vordefinierte Funktionen gibt es in allen möglichen Programmen jeweils unterschiedliche.

Übersicht über wichtige vordefinierte Funktionen	
mathematische Funktion	*Beschreibung des Ausgabewerts*
RUNDEN(Zahl; Nachkommastellen)	Zahl, gerundet
ABRUNDEN(Zahl; Nachkommastellen)	Zahl, abgerundet
AUFRUNDEN(Zahl; Nachkommastellen)	Zahl, aufgerundet
GANZZAHL(Zahl)	nächst kleinere ganze Zahl
GGT(Zahl1; Zahl2)	größter gemeinsamer Teiler
KGV(Zahl1; Zahl2)	kleinstes gemeinsames Vielfaches
QUOTIENT(Dividend; Divisor)	ganzzahliger Anteil der Division
REST(Dividend; Divisor)	Rest bei der ganzzahligen Division
ABS(Zahl)	Absolutbetrag der Zahl
VORZEICHEN(Zahl)	-1/0/+1 je nach Vorzeichen der Zahl
SUMME(Zahl1; Zahl2; ...)	Summe aller Zahlen (*)
PRODUKT(Zahl1; Zahl2; ...)	Produkt aller Zahlen (*)
ANZAHL(Zahl1; Zahl2; ...)	Anzahl aller Zahlen (*)
MAX(Zahl1; Zahl2; ...)	größte aller Zahlen (*)
MIN(Zahl1; Zahl2; ...)	kleinste aller Zahlen (*)
MITTELWERT(Zahl1; Zahl2; ...)	Mittelwert der Zahlen (*)
	(*)auch Angabe eines Zellenbereichs möglich

ZUFALLSZAHL()	zufällige Zahl zwischen 0 und 1
ZUFALLSBEREICH(UntereZahl; ObereZahl)	zufällige Ganzzahl im Bereich mindestens UntereZahl und höchstens ObereZahl
PI()	der Wert der Kreiszahl π
Zeitfunktionen	*Beschreibung des Ausgabewerts*
HEUTE()	aktuelles Tagesdatum
JETZT()	aktuelle Zeitangabe; Tag und Uhrzeit
JAHR(Zeitangabe)	Jahreszahl
MONAT(Zeitangabe)	Zahl des Monats im Jahr
TAG(Zeitangabe)	Zahl des Tages im Monat
WOCHENTAG(Zeitangabe)	Zahl für den Wochentag innerhalb der Woche

Zusammengesetzte Daten

Datentypen, die aus zwei oder mehr Bestandteilen zusammengesetzt sind, bezeichnet man als **Verbund**. Im Datenflussdiagramm gibt es Symbole zum Aufspalten und Zusammenfassen von Verbunden.

vgl. Datentypen (Buch S. 75)

Bedingte Funktionen

Die *WENN*-Funktion hat drei Eingangsparameter: einen Wahrheitswert und zwei alternative Werte für die Ausgabe. Je nachdem, ob der Wahrheitswert WAHR oder FALSCH ist, nimmt der Ausgabewert den Wert der ersten oder der zweiten Alternative an. Schreibweise:

WENN(Wahrheitswert; AlternativeBeiWahr; AlternativeBeiFalsch)

Eine Funktion mit beliebigen Eingangsparametern, die als Ausgabewert einen Wahrheitswert liefert, bezeichnet man als **Aussagefunktion**. Eine Funktion mit Eingangsparametern vom Typ Wahrheitswert, die als ausgabewert einen Wahrheitswert liefert, bezeichnet man als **logische Funktion**. Beispiele sind *UND*, *ODER* und *NICHT*.

Notationen

Infixnotation: a*b+c

Präfixnotation: +*abc oder plus(mal(a;b);c)

Postfixnotation: ab*c+

Zahlendarstellung - Dezimal- und Dualsystem

Zahlen werden im Computer intern im Zweiersystem (auch Dualsystem oder Binärsystem genannt) dargestellt. Die Stufenzahlen sind also nicht 1; 10; 100; 1000; 10000 ..., sondern 1; 2; 4; 8; 16; 32; 64 ...
Das setzt sich bei den Nachkommastellen fort: Statt 1/10; 1/100; 1/1000; 1/10000 ... lauten die Stellenwerte 1/2; 1/4; 1/8; 1/16; 1/32; 1/64 usw.

Ein Rechenblatt besteht aus Zellen, die in Zeilen und Spalten angeordnet sind. Eine Aufteilung dieser Art wird auch als zweidimensionale Matrix bezeichnet. Die Zeilen werden mit Zahlen benannt und die Spalten mit Buchstaben. Die Position einer Zelle wird mit der Kombination aus zugehörigem Spaltenbuchstaben und Zeilennummer bestimmt. Die Anzahl an Zeilen und Spalten ist in der Regel begrenzt.

Mittelwert

Man berechnet den **Mittelwert** wie folgt: Summe der betrachteten Zahlen geteilt durch ihre Anzahl.
Das **Arithmetische Mittel**, auch **arithmetischer Mittelwert** genannt (umgangssprachlich auch als Durchschnitt bezeichnet) ist ein Begriff in der Statistik. Es ist ein Lageparameter.

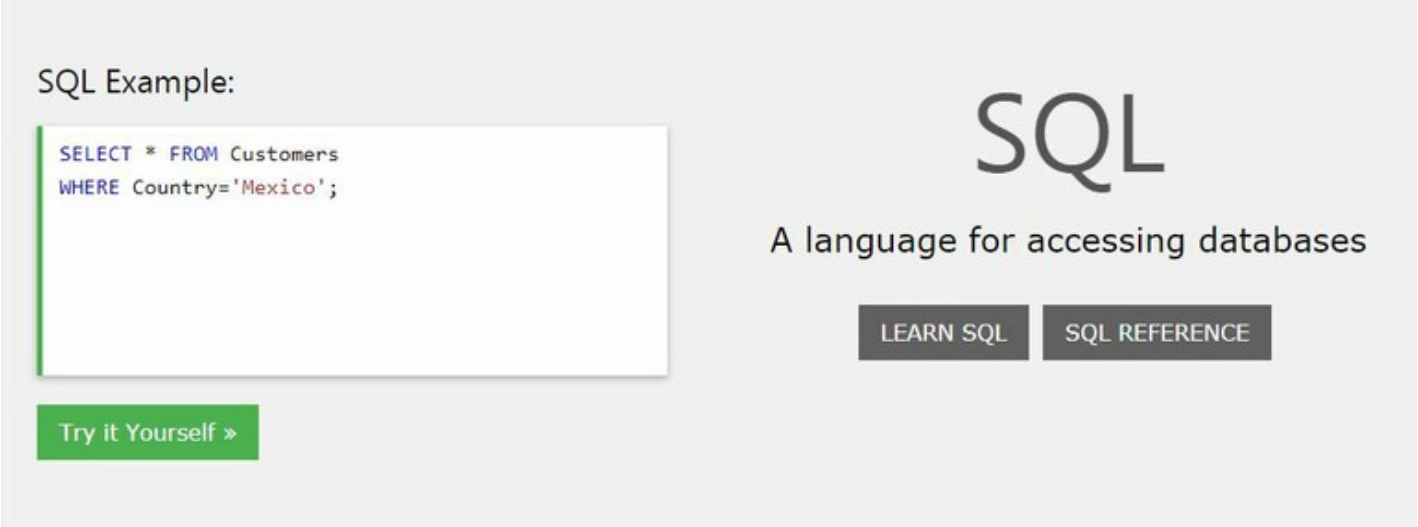
Mengenrabattsatz

Bruchaddition

Selbsttest s Funktionale Programmierung Fkt final

- mit Tabellenblättern/Arbeitsblättern arbeiten und diese in unterschiedlichen Dateiformaten abspeichern können,
- integrierte Funktionen wie die Hilfe verwenden können, um die Produktivität zu steigern,
- Daten in Tabellen eingeben können und gute Praxis beim Erstellen von Listen beachten,
- Daten auswählen, sortieren, kopieren, verschieben und löschen können,
- Zeilen und Spalten in einem Tabellenblatt/Arbeitsblatt bearbeiten können.
- Arbeitsblätter/Tabellenblätter kopieren, verschieben, löschen und passend umbenennen können,
- logische und mathematische Formeln unter Verwendung der Standardfunktionen der Tabellenkalkulation erstellen,
- gute Praxis beim Erstellen von Formeln beachten und Fehlerwerte kennen und interpretieren können,
- Zahlen und Text in einem Arbeitsblatt/Tabellenblatt formatieren können,
- Diagramme auswählen, erstellen und formatieren können, um Information verständlich darzustellen,
- Seiteneigenschaften eines Arbeitsblattes/Tabellenblattes anpassen und die Rechtschreibung überprüfen können, bevor das Tabellenblatt gedruckt wird.

SQL Lernen | Literatur in Englisch | w3schools.com



w3schools.com/sql/trysql.asp?filename=trysql_select_where | SQL try it
und
w3schools.com/sql | SQL tutorial
w3schools.com/sql/sql_ref_keywords.asp | SQL keywords Reference

(DB) Datenbanken/ Datenmodellierung

Schlüssel: Primärschlüssel, Fremdschlüssel, künstlicher Schlüssel

Buch Inf 9. Merkekästchen S. 71, 79, 88, 92. Werkzeugkasten "Microsoft Access". S. 154/155 8.1 + 8.3 notieren

Tabellen anlegen > DB

Thema: Erstellung eigener Datenbanken auf Grundlage entwickelter Klassendiagramme und beispielhafter Datensätze.

DB Einkauf
DB Bibliothek
DB Musik
DB Sportverein
DB Photo/ Filmarchiv

Die Tabellen anlegen und schrittweise erweitern, ergänzen, optimieren.

Aufgaben:
a) Erstellung eines **Klassendiagramms** mit den jeweiligen Attributen und den Beziehungen zwischen den Klassen zu einem der vorgeschlagenen Anwendungsfälle.
b) Entwickle danach zu ALLEN folgenden Themen Klassendiagramme (auf Papier, ins Heft).
c) Lege ZWEI Datenbanken an mit den benötigten Tabellen (in Microsoft Access) zu den Themen "Musik" und "Photo/ Filmarchiv".
d) Erstellung der jeweiligen Tabellen in der Datenbank, Realisierung der Beziehungen. Dabei ist zu prüfen, ob referentielle Integrität gegeben ist.
Es müssen genügend viele Daten in die Tabelle eingegeben werden, um brauchbare Ergebnistabellen bei Abfragen zu erhalten.
e) Überlegung, welche Benutzergruppen welche Rechte auf der Datenbank haben sollten.
f) Erstellung von Abfragen mit SQL für eine Auswahl realitätsbezogener Fragestellungen an die Datenbank.

Speichern großer Datenmengen in Datenbanken

Für die Verwaltung großer Mengen strukturierter Daten gibt es spezielle Werkzeuge, die **Datenbankverwaltungssysteme** (**Datenbanksysteme**).
Diese bieten Möglichkeiten, die Struktur der Daten auf eine **Datenbank** zu übertragen, die Daten in dieser Datenbank zu speichern und zu pflegen und die gespeicherten Daten vielfältig abzufragen und auszuwerten.

Objektorientiertes Datenmodell

Zur Analyse der benötigten **Datenstruktur** bewährt sich die objektorientierte Sichtweise. Das Erstellen des **objektorientierten Datenmodells** (**Objektmodell**) besteht in erster Linie aus den folgenden Schritten:
Welche **Objekte** sind für die Aufgabenstellung wichtig?
Zu welchen **Klassen** gehören diese Objekte?
Welche aufgabenbezogenen **Attribute** haben diese Klassen?
Welche **Beziehungen** bestehen zwischen den Klassen?
Die Struktur der Daten lässt sich im **Klassendiagramm** übersichtlich darstellen.
Im **Pflichtenheft** werden neben der Datenstruktur die Nutzungsmöglichkeiten der geplanten Datenbank festgehalten.

Relationales Datenbanksystem

In r**elationalen Datenbanken** werden die Daten in Form von **Tabellen** abgelegt. Das **objektorientierte Datenmodell** kann auf ein **relationales Datenbankschema** umgesetzt werden. Dabei werden den Klassen Tabellen zugeordnet, den Objekten Tabellenzeilen. Die Tabellenspalten entsprechen den Attributen der Klassen. In den Feldern eines Datensatzes (Tabellenzeile) stehen die Attributwerte eines Objekts. Der Wert **NULL** bedeutet "kein Wert eingegeben".
Ein **Schlüssel** ist eine Spalte (eine Gruppe von Spalten), deren Wert jeweils einen Datensatz in der Tabelle eindeutig bestimmt (identifiziert). Muss eine neue Spalte zur eindeutigen Identifikation der Datensätze eingeführt werden, nennt man sie **künstlichen Schlüssel**.

Datenbankabfragen

SQL

Einfache Beziehungen zwischen Klassen

Das Resultat einer SQL-Abfrage ist eine **Ergebnistabelle**, die durch **Selektion** und **Projektion** festgelegt wird.
Zur Formulierung der Abfrage wird die Sprache **SQL** verwendet. Dort hat eine Abfrage den Aufbau:

SELECT <Spaltenliste> **FROM** <Tabellenliste> **WHERE** <Bedingung>

Eine **Sortierung** des Ergebnisses einer SQL-Abfrage wird durch den Zusatz **ORDER BY** festgelegt, die Sortierung ist normalerweise **aufsteigend**. Bei jeder Sortierspalte kann durch den Zusatz **DESC** eine **absteigende** Sortierung erreicht werden.

Abfragen über mehrere Tabellen

Im **Klassendiagramm** werden Beziehungen beschrieben.
Neben einem beschreibenden Kurztext der Beziehung wird ihre **Kardinalität** angegeben.
Im relationalen Datenbankschema:
Bei einer **1:n-Beziehung** bekommen die Datensätze auf der "n"-Seite eine zusätzliche Spalte (**Fremdschlüssel**), in dieser wird der Schlüssel des Datensatzes der "1"-Seite gespeichert, mit dem sie in Beziehung stehen.

Anfragen an mehrere Tabellen: **kartesisches Produkt**

Mehr Beziehungen zwischen Klassen

Eine **Beziehungstabelle** muss für jede **n:m-Beziehung** erfasst werden und nimmt die Schlüssel der zueinander in Beziehung stehenden Datensätze als Fremdschlüssel auf. Der Schlüssel für diese Tabelle ist das Paar der beiden Fremdschlüssel.
Abfragen bei n:m-Beziehungen müssen drei Tabellen verknüpfen!

SQL-Abfragen - GROUP BY, ORDER BY

Aggregatfunktionen COUNT, AVG, SUM, MAX oder MIN
... **AS <Name>** | Namen für die Spalten der Ergebnistabelle

GROUP BY | Gruppieren der Ergebnisse einer Abfrage nach bestimmten Spalten, insbesondere für die Anwendung der Aggregatfunktionen. Bedingungen an diese Gruppen werden mit **HAVING** festgelegt.

Die erweiterte Abfrageanweisung lautet:

SELECT <Spaltenliste>
FROM <Tabellenliste>
WHERE <Bedingung>
GROUP BY <Spalte>
HAVING <Bedingung>
ORDER BY <Spaltenliste>

Ergebnisse von **Unterabfragen** können im WHERE-Teil für Bedingungen verwendet werden.
Ergebnisse von Abfragen können als Mengen interpretiert werden; der "Enthalten"-Operator wird als **IN** geschrieben.

Datenbanken - komplett und gut entworfen (Qualitätsbetrachtung)

Betrachtung exemplarischer Datenbanken

- DB Einkauf
- DB Bibliothek
- DB Musik
- DB Sportverein
- DB Photo/ Filmarchiv

Verändern von Datensätzen | INSERT, UPDATE, DELETE

Datenpflege
SQL-Anweisungen **INSERT**, **UPDATE** und **DELETE** (**Einfügen**, **Ändern** und **Löschen** von Datensätzen)

Datenkonsistenz

Referenzielle Integrität: für jeden Fremdschlüsselwert existiert auch ein Datensatz mit diesem Schlüsselwert. > Angabe von **Fremdschlüsselbedingungen** können dies sicherstellen.
Fremdschlüssel löschen oder ändern:
- Beim Versuch, einen referenzierten Schlüssel zu löschen, kann das Datenbanksystem die Löschung verbieten, den referenzierenden Datensatz ebenfalls löschen oder den referenzierenden Fremdschlüssel auf den Wert NULL setzen.
- Beim Ändern eines referenzierten Schlüssels kann das Datenbanksystem die Änderung verbieten oder alle Referenzen mit ändern.

Datensicherheit und Datenschutz

Datenschutz

Datensicherheit

Normalformen

1. Normalform: Eine Tabelle enthält ausschließlich atomare (einelementige) Feldwerte.
Definition: Ein Attribut Y ist von einem Attribut X **abhängig**, wenn der Wert von X den Wert von Y festlegt.

2. Normalform: Zusätzlich zur 1. Normalform muss gelten:
Jedes Nicht-Schlüsselattribut muss vom **gesamten** Schlüssel abhängen.
Anmerkung: Besteht der Schlüssel nur aus einer Spalte, ist die Tabelle immer in der zweiten Normalform.

3. Normalform: Zusätzlich zur 2. Normalform muss gelten:
Attribute sind nur vom Schlüssel, nicht voneinander abhängig.

Anwendungsbeispiele DB - Analyse

Analyse konkreter Datenbanken.

Quelle u.a: vgl. Brichzin, u.a. (Hg), "Informatik I. Funktionale Modellierung. Datenmodellierung", Oldenburg-Verlag

Selbsttest s Multimedia

Anmerkung: Die Programme GIMP, Inkscape, Blender, Audacity, 'Video-Pad Basis', 'Vectorian Giotto' oder 'Libre Office Impress' sind als Freeware oder OpenSource frei erhältlich.

Computergrafik

- Grundlegende theoretische Kenntnisse zum Thema Computergrafik
- Eigene Gestaltung in Variationen mit GIMP
- Pixelbilder erzeugen, bearbeiten, speichern und exportieren
- Eigene Gestaltung in Variationen mit Inkscape
- Vektorgrafiken erzeugen, bearbeiten, speichern und exportieren

Computeranimation

- Grundlegende theoretische Kenntnisse zum Thema Computeranimation
- Bilder für Animationen planen und erstellen
- Bildanimation mit dem Programm GIMP
- Schlüsselbildanimation mit Blender (3D) (oder mit Vectorian Giotto)
- Dateiformate (Gif-Animation vs. Filmdateien)

Audio und Video

- Grundlegende theoretische Kenntnisse zum Thema Audio und Video
- Umgang mit einem Audiobearbeitungsprogramm: Audacity
- Aufnehmen, Bearbeiten, Exportieren
- Speichern und exportieren (Dateiformate)
- Umgang mit einem Videobearbeitungsprogramm: VideoPad-Basis
- Medien importieren, Clips schneiden, Titel einblenden
- Filter
- Speichern und exportieren (Dateiformate)

Multimedia

- Grundlegende theoretische Kenntnisse zum Thema Multimedia
- Erweiterter Umgang mit einem Präsentationswerkzeug (z. B. PowerPoint, Impress)
- Multimediainhalte in eine Präsentation integrieren
- Ein Multimediaprodukt speichern, exportieren und versenden

Scratch | scratch.mit.edu (Wdh)

Programmieren mit Puzzle-Bausteinen

scratch.mit.edu > Entwickeln

de.scratch-wiki.info/wiki/Einsteiger-Tutorials

OOP Objektorientierte Programmierung mit Python/ Java in Informatik 9

- **Java/ Processing** | OOP (Objektorientierte Programmierung) mit Java/ Processing: processing.org (Vereinfachte Java-Version), openprocessing.org
  Processing ist eine objektorientierte, stark typisierte Programmiersprache mit zugehöriger integrierter Entwicklungsumgebung. Die Programmiersprache ist auf die Einsatzbereiche Grafik, Simulation und Animation spezialisiert. Processing wird in einem quelloffenen Projekt entwickelt, das am Massachusetts Institute of Technology in Boston von Ben Fry (Broad Institute) und Casey Reas (UCLA Design|Media Arts) initiiert wurde. Processing hat den Charakter einer stark vereinfachten Version der Programmiersprache Java, ermöglicht Interaktionen und visuelle

Elemente zu programmieren und richtet sich vorwiegend an Gestalter, Künstler und Programmieranfänger.

processing.org | OOP (Objektorientierte Programmierung) mit Java/ Processing. Vereinfachte Java-Version
processing.org/examples | Beispiele
hello.processing.org/editor | Tutorial

py.processing.org/reference | Python/ Processing

- **Python** | OOP
  vgl. cscircles.cemc.uwaterloo.ca
  cscircles.cemc.uwaterloo.ca > cheatsheet.pdf

OOP Java/ Processing | openprocessing

processing.org | OOP (Objektorientierte Programmierung) mit Java/ Processing. Vereinfachte Java-Version
processing.org/examples | Beispiele

openprocessing.org
openprocessing.org/browse/#

Python auch im Jahr 2020 im tiobe-index weiterhin aufsteigend

Python als Programmiersprache und Gewinner des Jahres 2018 ist weiterhin in der Relevanz der Programmiersprache aufsteigend.
vgl. tiobe.com/tiobe-index

OOP Python

- OOP Referate (**Python** | Java) (Lit. z.B. python4kids.net > Wie ein Informatiker denken lernen)

  Kap02 **Variablen, Ausdrücke und Anweisungen**
  Kap03 **Funktionen**
  Kap04 **Verzweigung und Rekursion**
  Kap05 **Funktionen mit Wert**
  Kap06 **Iteration**
  Kap07 **Strings**
  Kap08 **Listen**
  Kap09 **Tupel**
  Kap10 **Datentyp Dictionary**
  Kap11 **Dateien und Ausnahmen**
  Kap12 **Klassen und Objekte**
  Kap13 **Klassen und Funktionen**
  Kap14 **Klassen und Methoden**
  Kap15 **Mengen von Objekten**
  Kap16 **Vererbung**
  Kap17 **Verkettete Listen**
  Kap18 **Stacks**
  Kap19 **Queues**
  Kap20 **Bäume**

py | Literatur zu Python und Games (englisch)

Teil 1
inventwithpython.com/invent4thed

Teil 2
inventwithpython.com/pygame

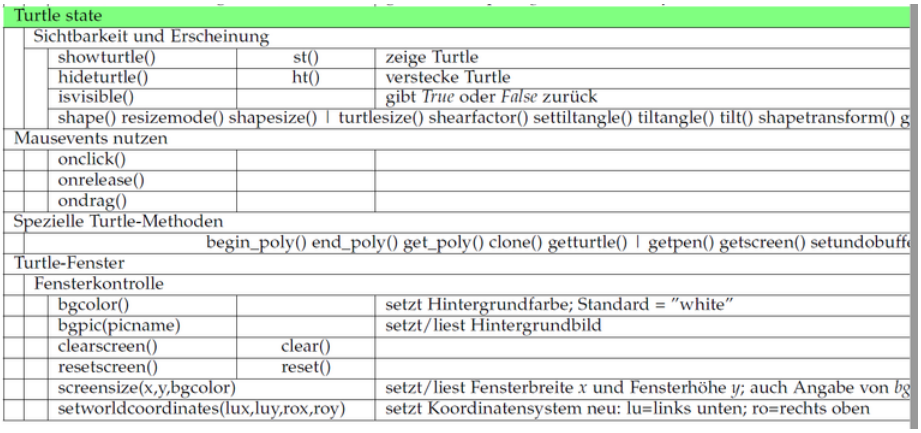
- **OOP Programmierprojekt** mit objektorientierter Programmierung z.B. turtle Grafik (Python)
  #Codierung aufschlussreich kommentieren.
  #Eigene Projektdokumentation erstellen
- Alternativ: **5-Seiten-Artikel schreiben**. Programmier- bzw. Rechercheprojekt
  Ihr könnte einen Artikel, d.h. gut recherchierten Aufsatz schreiben, der etwas umfangreicher ist als ein Referat und das in der Erstellungsphase im Kurs wie in einem Kolloquium vorgestellt wird - von Internet-Kunst, webDesign, Anwendung von QR-codes, pygame py-Bibliotheken, jupyter-notebooks for python oder dergleichen -
  einfach ein informatisches Spezialthema eures Interesses. Seitenumfang: ca. 5 Seiten Text (ohne Abb. oder Programm-Code).
  Die Vorstellung im Kurs/ in der Klasse kann sein:
  - Vorstellung der Kurz-Gliederungspunkte mit einem Beispiel-Thema
  - Vorstellung einer Argumentation oder eines Kernaspekts
  - Vorstellung einer geschriebenen Textpassage
  - Vorstellung von Code-Beispielen

Wir wiederholen Grundlagen aus der Informatik der Vorjahre und programmieren mit objektorientierter Programmiersprache (Python | Java), OOP.
Konkret fangen wir mit kleineren Übungen an, z.B. mit dem Blocksystem von Scratch (http://scratch.mit.edu) und der graphischen Python turtle Programmierung.
Danach gehts systematisch in Programmierkenntnisse von Python und später auch Java - wir behandeln hier die Unterschiede von Python und Java.
Jeder hat als Aufgabe, ein größeres oder mehrere kleinere Informatik-Projekte mit Python oder Java umzusetzen.

python turtle Grafik

Mit der python IDLE grafische Objekte zeichnen
vgl. Skript Kurzbefehle

Befehl, Langform	Kurzformen	Bedeutung
Turtle-Aktionen		
Bewegen und Zeichnen		
forward(px)	fd()	bewegt T. um *px* Pixel nach vorne
backward(px)	bk(), back()	bewegt T. um *px* Pixel nach hinten
right(winkel)	rt()	dreht T. um *winkel* im Uhrzeigersinn
left(winkel)	lt()	dreht T. um *winkel* gegen den Uhrzeigersinn
setposition(x,y)	setpos(), goto()	positioniert T. auf *x- und y*-Koordinaten
setx(x), sety(y)		positioniert T. auf *x- oder y*-Koordinate
setheading()	seth()	setzt absolute Blickrichtung der T.
home()		positioniert T. auf Startkoordinaten
circle(radius, winkel, ecken)		zeichnet Kreis gemäß radius; falls *winkel* entspr. Kreisbogen; falls *ecken* entspr. Polygon
dot(größe, farbe)		zeichnet Punkt ggf. mit entspr. *größe* und *farbe*
stamp(), clearstamp()		
speed(zahl)		steuert T.geschwindigkeit gemäß *zahl*: 1-10
undo()		macht letzte Anweisung rückgängig
Status abfragen		
position()	pos()	gibt Position als Koordinatentupel zurück
towards()		
xcor(), ycor()		gibt x- oder y-Koordinate zurück
heading()		gibt T.orientierung zurück; 0 = ost, 90 = nord
distance(x,y)		berechnet Abstand der T. zum Punkt P(x|y)
Kontrolle des Stifts		
Stift setzen		
pendown()	down(), pd()	bringt S. aufs Papier
penup()	up(),up()	nimmt S. vom Papier
pensize(breite)	width()	setzt Stiftbreite auf Pixel*breite*
pen()		gibt Stiftstatus als Dictionary zurück
isdown()		gibt *True* zurück, falls Stift schreibbereit
Farben		
color()		gibt pencolor() und fillcolor() zurück
pencolor(), fillcolor()		gibt Strich-/Füllfarbe als Namen oder RGB-Tupel zurück
pencolor(farbname)		setzt Strichfarbe gemäß Farbnamenszeichenkette
pencolor((r,g,b))		setzt Strichfarbe gemäß RGB-Komponenten (1-255)
filling()		gibt True zurück, falls Füllmodus aktiviert
begin_fill(), end_fill()		Füllmodus (de-)aktivieren
Weitere Zeichenoptionen		
reset()		Gezeichnetes löschen und T. in Ausgangsposition zurück
clear()		nur Gezeichnetes löschen
write(text, move, align, font)		gibt *text* entspr. *align* ='left' mittels *font* =('Arial',8,'normal') aus; *m*



Turtle state			
Sichtbarkeit und Erscheinung			
	showturtle()	st()	zeige Turtle
	hideturtle()	ht()	verstecke Turtle
	isvisible()		gibt *True* oder *False* zurück
	shape() resizemode() shapesize() | turtlesize() shearfactor() settiltangle() tiltangle() tilt() shapetransform() g		
Mausevents nutzen			
	onclick()		
	onrelease()		
	ondrag()		
Spezielle Turtle-Methoden			
	begin_poly() end_poly() get_poly() clone() getturtle() | getpen() getscreen() setundobuff		
Turtle-Fenster			
Fensterkontrolle			
	bgcolor()		setzt Hintergrundfarbe; Standard = "white"
	bgpic(picname)		setzt/liest Hintergrundbild
	clearscreen()	clear()	
	resetscreen()	reset()	
	screensize(x,y,bgcolor)		setzt/liest Fensterbreite *x* und Fensterhöhe *y*; auch Angabe von *bg*
	setworldcoordinates(lux,luy,rox,roy)		setzt Koordinatensystem neu: lu=links unten; ro=rechts oben

Python-Referat I | mit py computer science circles (Uni Waterloo)

Es gibt hier 18 gut verständliche Kapitel zur Objektorientierten Programmierung mit Python anhand von Programmier-Beispielen.

Referat eines der Kapitel:
Vortrag auf deutsch mit Verwendung cscircles.cemc.uwaterloo.ca/de
Präsentation in englisch cscircles.cemc.uwaterloo.ca
Übungen mit Codierung direkt ausführen und kommentieren.

Nachschlage-Werk: Pythonbuch (auf deutsch)

pythonbuch.com

IDE Wing

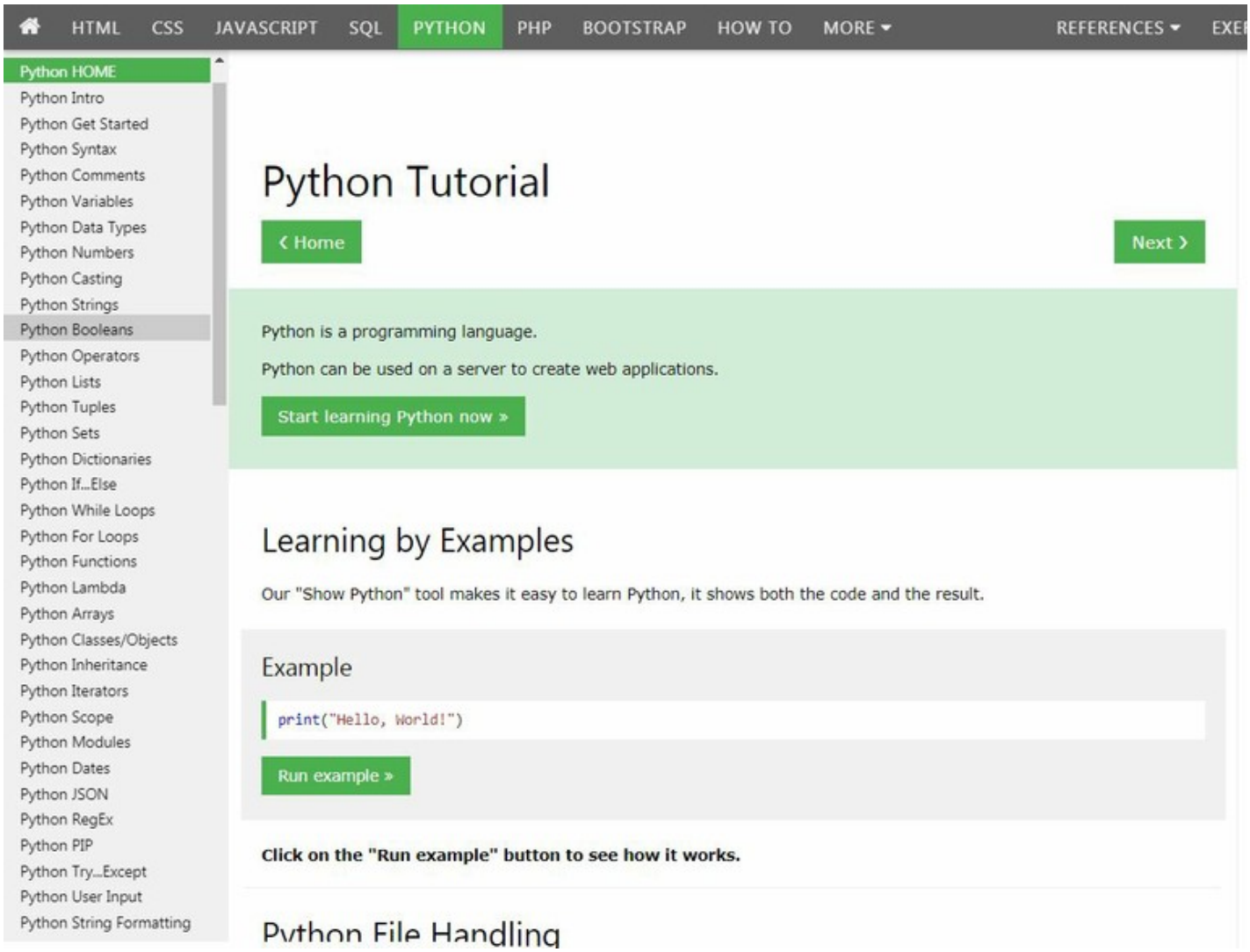
Zum Editieren von Python, statt der python-IDLE:
wingware.com/downloads/wing

Überblicks-Python-Referate II (in Teamarbeit zu zweit)

Recherche z.B. in Online-Quellen, Lit:
Python für kids: http://python4kids.net/how2think/index.html

Kapitel 1: Über das Programmieren
Kapitel 2: Variablen, Ausdrücke und Anweisungen
Kapitel 3: Funktionen
Kapitel 4: Verzweigung und Rekursion
Kapitel 5: Funktionen mit Wert
Kapitel 6: Iteration
Kapitel 7: Strings
Kapitel 8: Listen
Kapitel 9: Tupel
Kapitel 10: Der Datentyp Dictionary
Kapitel 11: Dateien und Ausnahmen
Kapitel 12: Klassen und Objekte
Kapitel 13: Klassen und Funktionen
Kapitel 14: Klassen und Methoden
Kapitel 15: Mengen von Objekten
Kapitel 16: Vererbung
Kapitel 17: Verkettete Listen
Kapitel 18: Stacks
Kapitel 19: Queues
Kapitel 20: Bäume

Python Lernen | Literatur in Englisch | w3schools.com



w3schools.com/python | py
und
w3schools.com/python/python_examples.asp | py examples
w3schools.com/python/exercise.asp | py exercise
w3schools.com/quiztest/quiztest.asp?qtest=PYTHON | py Quiz
w3schools.com/python/showpython.asp?filename=demo_default | py run example

Python-Projekt, eigenes (kleines) | Memory/ Puzzle-App

vgl.
pythonprogramming.altervista.org/memory-game-with-python-in-no-time/
oder
inventwithpython.com/pygame/chapter3.html

Mögliche weitere Python-Referate (5 min Vortrag)

wichtig ist hierbei jeweils: Beschreibung | Definition | Beispiel | Abbildung

Bei jedem Referat soll am Ende ein kurzer Hefteintrag (Textpassagen zum Abschreiben bzw./und Graphik oder Abbildung zum Abzeichnen) die letzte Seite sein.

Informatik | Objektorientierte Programmierung mit Python oder Java

if/else - Bedingung
while - Schleife
Objekte | py
Klassen | py z.B. Liste (unsortiert, sortiert), Liste einfach verkettet, Schlange, Stapel, Baum, Graph (ungewichtet, gewichtet)

Klassenkarten | py
Objektkarten | py
Klassendiagramm
Objektdiagramm
Zustandsdiagramm
Variablen | py
Methoden | py
Anweisungen | py
Strings (Zeichenketten) | py
Vererbung | py
Verzweigung/ Rekursion
Bsp. Koch'sche Schneeflocke
Bsp. Sierpinski-Dreieck
Bsp. Fibonacci-Funktion
Lastenheft/ Pflichtenheft in der Informatik
Agile Softwareentwicklung (vs Wasserfall-Modell)
Objektorientierte Programmierung - Diff. python, Java
Biometrische Daten als Schlüssel
Künstliche Intelligenz (Artificial Intelligence AI)
Intuitive Bedienbarkeit/ User-Dialog/ GUI - Graphische Oberfläche und optisches Design
Eclipse-Programmierumgebung
Android-Programmierung
python-Turtle-Programmierung
App-Programmierung

Grundlagen der Informatik | basic

Open Source
CC, Copyright und Lizenzen
KI - Künstliche Intelligenz
Big Data - Überwachung, Online-Tracking, Datenschutz
Serious Games, technologiegestütztes Lernen

Geschichte/ Meilensteine der Informatik
Klasse Objekt Attribut Methode. Punktnotation
Klassenkarte Objektkarte Klassendiagramm | py
Beziehungen und Kardinalitäten
Methoden | py
HTML | atom.io oder notepad++
Algorithmus/ algorithmisches Denken
Struktogramme
Baum und Graph | Hierarchische Strukturen
Binäres Zahlensystem
Rechnerarchitektur

Mögliche Themen als Wiederholung aus Informatik 09, Funktionale Programmierung (Excel/ Calc) und Datenbanken (Access/ Base)
mit Blick auf Umsetzungen in Python

Funktionale Programmierung | py

Funktionen
Vordefinierte Funktionen
Funktionen *UND*, *ODER*, *NICHT*
Logische Funktionen
Datenflussdiagramm
Bedingte Funktion: *WENN*-Funktion
while-Schleife
if-/else - Bedingte Anweisung
Zusammenfassen von Funktionen
Verknüpfung von Tabellen
Verzweigung im Datenfluss
Geschichte der Tabellenkalkulation
Tabellenkalkulation - Buchhaltung
Diagramme aus Tabellendaten

Kapital-Berechung (Kredit-Aufnahme, Geld-Anlegen, Zinsberechnung mit Zinstagen)
Zusammengesetzte Daten (z.B. Brüche)
Verkaufspreis mit Mengenrabatt
Verkaufspreis mit Skonto
Kostenvoranschlag (Rechnung) mit Tabellenkalkulation
Adress-Listen mit Tabellen
Diagramme in Tabellenkalkulation: Entwicklung der Weltbevölkerung
Kopieren - relative und absolute Adressen
Links - lokal und online z.B. in Tabellen

Datenbanken | py

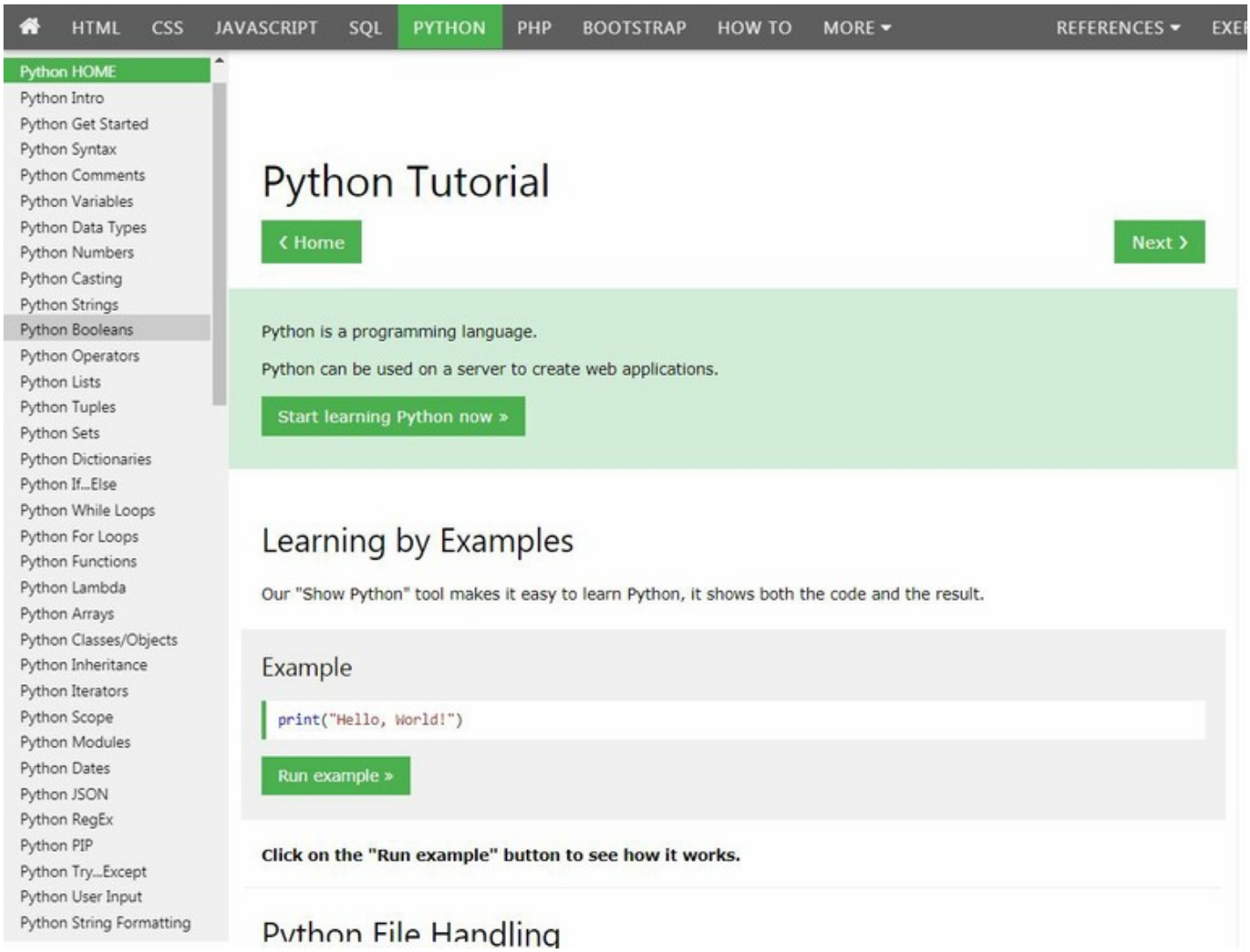
DB statt Fkt - warum?
Datentypen | py
Relationales Datenbanksystem
SQL-Abfragen
Datenbankschema
Datenpflege (UPDATE, INSERT, DELETE)
Datensicherheit und Datenschutz
Klassendiagramm und Kardinalitäten
n:m - Beziehung
Das kartesische Produkt
Aggregat-Funktionen von SQL (COUNT, AVG, MAX, MIN, SUM)
Schlüssel, Fremdschlüssel und referentielle Integrität
Datenpflege (Ändern, Einfügen, Löschen von Datensätzen | UPDATE, INSERT, DELETE)
Änderungsanomalien bei der Datenpflege
Datenkonsistenz
Datensicherheit und Datenschutz
Datenbankschema
Datenbank-Design
Prüfziffer am Bsp. der ISBN-Nummer

Python mit py computer science circles (Uni Waterloo) | Objektorientierte Programmierung (OOP)

Es gibt hier 18 gut verständliche Kapitel (in deutsch) zur Objektorientierten Programmierung mit Python.

Möglich wäre auch ein Referat über eines der Kapitel:
Vortrag auf deutsch mit Verwendung cscircles.cemc.uwaterloo.ca/de
Präsentation in englisch cscircles.cemc.uwaterloo.ca
Übungen mit Codierung direkt ausführen und kommentieren.

Python Lernen | Literatur in Englisch | w3schools.com



w3schools.com/python | py
und
w3schools.com/python/python_examples.asp | py examples
w3schools.com/python/exercise.asp | py exercise
w3schools.com/quiztest/quiztest.asp?qtest=PYTHON | py Quiz
w3schools.com/python/showpython.asp?filename=demo_default | py run example

Informatik 09 Grundwissen Funktionale Modellierung

Zellbezug
z. B. A4: A ist der Spaltenbezeichner, 4 ist die Zeilennummer

relativer Zellbezug
Beim Kopieren von Zellen, die Formeln enthalten, werden die Zellbezüge in der Zielzelle entsprechend angepasst. z. B. A4

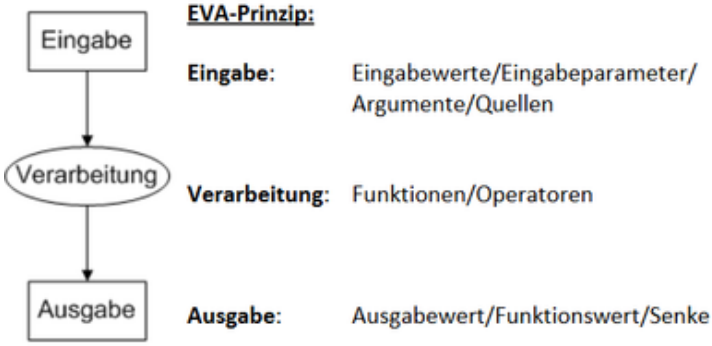
absoluter Zellbezug
Soll beim Kopieren einer Zelle mit Formel ein Zellbezug gleich bleiben, verwendet man absolute Zellbezüge. Dazu muss man der Zeilennummer und/oder dem Spaltenbezeichner ein $-Zeichen voran stellen. z. B. A4, $A4, A$4

iterative Berechnungen
Verwendet man das Ergebnis einer Formel in der jeweils nachfolgenden Formel, spricht man von iterativen Berechnungen.

Datentypen
Tabellenkalkulationen arbeiten intern nur mit Zahlen und Texten, können aber über die Zellformatierung viele weitere Datentypen darstellen: Zahl, Prozent, Währung, Datum, Zeit, Bruch, Wahrheitswert, Text.

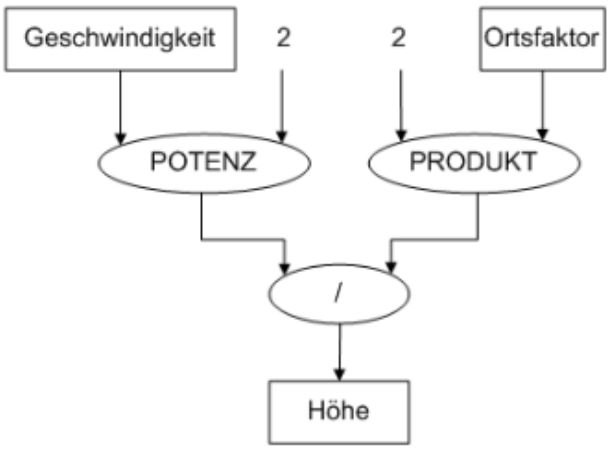
Funktionen
Eindeutige Zuordnungen heißen Funktionen. Sie ordnen jedem Argument höchstens einen Funktionswert zu.

Datenflussdiagramme



Schreibweisen von zweistelligen Funktionen
Präfixschreibweise: Der Funktionsbezeichner wird vor den beiden Parametern geschrieben. z. B. SUMME(3;4)
Infixschreibweise: Der Funktionsbezeichner wird zwischen den beiden Parametern geschrieben. z. B. 3+4

Verkettung von Funktionen
Funktionen werden verkettet, indem man den Wert einer Funktion einer weiteren Funktion als Argument übergibt. z. B.:



Umwandlung des Datenflussdiagramms in einen Term: Höhe = POTENZ(Geschwindigkeit, 2) / PRODUKT(2, Ortsfaktor)

Wenn-Funktion | Bedingte Terme
Eine Funktion kann, abhängig von Bedingungen, unterschiedliche Ergebnisse liefern:

WENN(Bedingung, Term1, Term2)

Term1 wird ausgeführt, wenn die Bedingung den Wahrheitswert WAHR liefert, sonst wird Term2 ausgeführt.

Logische Funktionen | UND ODER NICHT
Man verwendet die **UND**-Funktion zur Verknüpfung mehrerer Aussagen. Sie liefert den Wert WAHR, wenn alle Argumente WAHR sind.
Man verwendet die **ODER**-Funktion zur Verknüpfung mehrerer Aussagen. Sie liefert den Wert WAHR, wenn mindestens eines der Argumente den Wert WAHR hat.
Die Funktion **NICHT** kehrt den Wert einer logischen Aussage um.

Informatik 09 Grundwissen Datenmodellierung

Datenbanksystem
Ein Datenbanksystem (DBS) ist eine systematische und strukturierte Zusammenfassung von Daten eines Problembereichs (Datenbasis) einschließlich der zur Eingabe, Verwaltung, Auswertung und Ausgabe erforderlichen Software (Datenbankmanagementsystem, DBMS):
DBS = Datenbasis + DBMS

Schema
Jede Tabelle wird durch ein Schema charakterisiert:

- Name der Tabelle
- Liste der Attribute (Spaltenliste)
- Datentypen der jeweiligen Attribute

Darstellung eines Schemas: **TABELLE(Attribut_1: Datentyp_1, Attribut_2: Datentyp_2, ...)**

Datentypen
Datentypen in einer MySQL-Datenbank: z.B. für Buchstaben char (=Character = Buchstabe) sowie für Texte das Format varchar (= Character-Attribut mit vaiabler Länge). Für Zahlen finden wir integer (=Ganzzahl), float (=Gleitkommazahlen), für Zeitangaben z.B. date und time...

SQL-Abfrage
Das Ergebnis jeder Abfrage ist immer eine Tabelle, auch wenn das Ergebnis nur aus einem Datensatz mit einem Attribut besteht. Mit folgender Abfrage können wir uns den kompletten Inhalt einer Tabelle ausgeben lassen:
SELECT *FROM Tabellenname;

Selektion
Bei der Selektion werden die Datensätze einer Tabelle, die die angegebene Bedingung erfüllen, in einer neuen Tabelle ausgegeben:
SELECT*FROM Tabellenname WHERE Bedingung;

Projektion
Bei der Projektion werden von allen Datensätzen die angegebenen Spalten ausgegeben.
SELECT Spalte_1 [, Spalte_2, …, Spalte_n] FROM Tabelle;

Aggregatsfunktionen

- COUNT(*)
- MAX(...)
- MIN(...)
- SUM(...)
- AVG(...)

Möchte man neben demerhalten Wert noch weitere Attribute ausgeben, so muss man mit **GROUP BY** gruppieren!

Komplexere SQL-Abfrage

SELECT Spalte_1 [, Spalte_2, ..., Spalte_n]	*Projektion*
FROM Tabelle(n)	*Tabellenname(n)*
WHERE Bedingung(en)	*Selektion*
GROUP BY Spalte	*Gruppierung*
HAVING Bedingung	*Selektion nach Gruppierung*
ORDER BY Spalte;	*Sortierung*

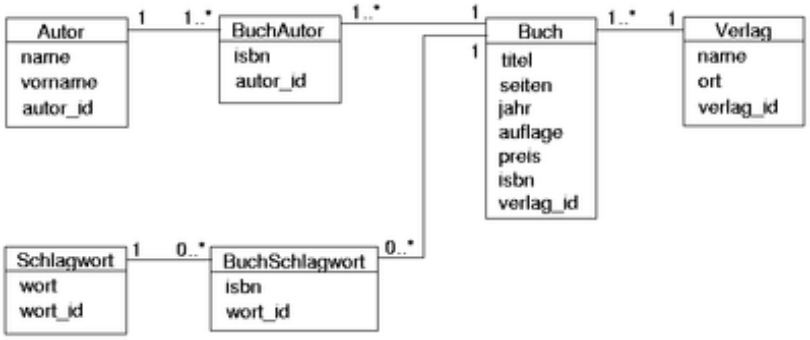
Schlüssel
Der (Primär-) Schlüssel setzt sich aus einer (minimalen) Menge von Attributen zusammen, deren Wert jeweils einen Datensatz in der Tabelle eindeutig identifiziert. Muss eine neue Spalte zur eindeutigen Identifikation der Datensätze eingeführt werden, nennt man sie künstlichen Schlüssel (z. B. fortlaufende Nummerierung).
Im Tabellenschema wird der Schlüssel unterstrichen.

Redundanz und Konsistenz
Unter Redundanz versteht man die (überflüssige) Mehrfachspeicherung von Daten. Beim Einfügen oder Ändern von Datensätzen können hierbei Probleme und Fehler, sogenannte Anomalien, auftreten: UPDATE-/DELETE-/INSERT-Anomalie.
Ist die Datenbank ohne Widersprüche, so ist sie konsistent, enthält sie Unstimmigkeiten, so ist sie inkonsistent.

Datenmodellierung
Objekte mit gleichen Attributen legen eine Klasse fest. Die Attribute werden in einer Klassenkarte aufgelistet. Klassendiagramme zeigen, welche Beziehungen die Objekte der betrachteten Klassen eingehen können. Die Beziehungen zwischen Klassen werden durch Verbindungslinien dargestellt. Man unterscheidet dabei drei Beziehungsarten: 1:1/1:n/n:m-Beziehung
Klassendiagramme umfassen somit:

- alle Klassen mit ihren Attributlisten –jeweils als Klassenkarte dargestellt
- alle Beziehungslinien mit den zugehörigen Kardinalitäten und Beziehungsnamen.

Bsp.:



Quelle Bild: http://www.peter-junglas.de/fh/vorlesungen/praktinfWI1/html/kap5-5.html

n:m-Beziehungen und 1:n-Beziehungen:
Für jede n:m-Beziehung wird eine Beziehungstabelle festgelegt. Diese enthält die Primärschlüssel der beiden Tabellen. Diese werden als Fremdschlüssel bezeichnet. Der Primärschlüssel dieser Beziehungstabelle sind alle Fremdschlüssel.
1:n-Beziehungen: Bei einer 1:n-Beziehung ist eine Beziehungstabelle nicht notwendig. Es wird die Tabelle der Klasse mit der Kardinalität n um den Primärschlüssel der Klasse mit der Kardinalität 1 erweitert. Dieser wird als Fremdschlüssel eingefügt.
1:1-Beziehungen: Bei einer 1:1-Beziehung ist auch keine Beziehungstabelle notwendig. Es wird die Tabelle einer beliebigen Klasse um den Primärschlüssel der anderen Klasse erweitert. Dieser wird als Fremdschlüsseleingefügt. Fremdschlüssel werden gestrichelt unterstrichen.

Kreuzprodukt
Das Kreuzprodukt (kartesisches Produkt) TABELLE1 x TABELLE2 bildet aus zwei Tabellen eine neue Tabelle, die jeden Datensatz der ersten Tabelle mit jedem Datensatz der zweiten Tabelle verknüpft:
SELECT * FROM Tabelle_1, Tabelle_2;

Join
Eine Hintereinanderausführung von Kreuzprodukt und Selektion nennt sich Join. Nur die Zeilen des Kreuzproduktes, bei denen der Wert der Fremdschlüsselattribute mit den zugehörigen Werten der Primärschlüsselattribute übereinstimmen, enthalten die Daten der korrekten Beziehungspartner. Diese werden mit der anschließenden Selektion gefiltert.
SELECT * FROM Tabelle_1 t1, Tabelle_2 t2 WHERE t1.Attribut = t2.Attribut;

Quelle: angelehnt an: http://www.mgf-kulmbach.de/neu/images/medien/unterrichtsfaecher/Informatik/grundwissen/inf9-grundwissen.pdf

s Tipps | Informatik

Programmiere ein eigenes Informatik-Projekt! Bei Fragen, die du trotz eigenem Ausprobieren bzw. Nachlesen alleine nicht klären kannst, wendest du dich an mich.

Sonnja Genia Riedl

—