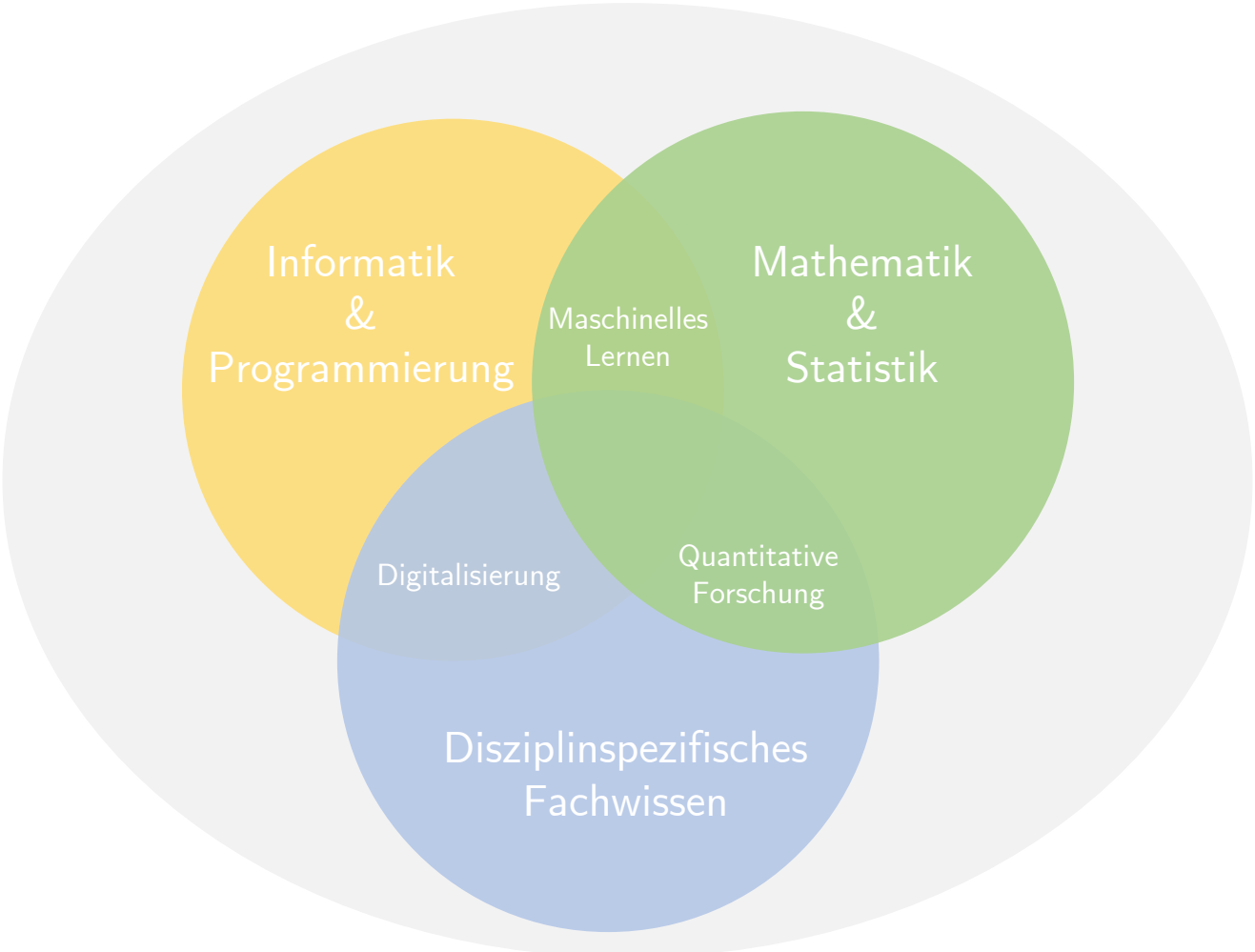
Computergestützte Datenanalyse

BSc Psychologie SoSe 2021

Prof. Dr. Dirk Ostwald

(1) Datenanalyse und Programmierung

Datenwissenschaft



Datenanalyse und Programmierung

- Computergestützte Datenanalyse
- Informatik und Rechnerarchitektur
- Algorithmen und Programme
- Selbstkontrollfragen

Datenanalyse und Programmierung

- **Computergestützte Datenanalyse**
- Informatik und Rechnerarchitektur
- Algorithmen und Programme
- Selbstkontrollfragen

Computergestützte Datenanalyse

- Wissenschaftliche Daten liegen typischerweise in digitaler Form vor.
- Daten werden typischerweise mit Hilfe eines Computers analysiert.
- Zur Analyse von digitalen schreibt man Computerprogramme.
- Diese Computerprogramme heißen *Datenanalyseskripte*.

Struktur computergestützter Datenanalyse

1. Einlesen und Bereinigen eines Datensatzes
2. Berechnung und Visualisierung deskriptiver Statistiken
3. Probabilistische Datenmodellierung, Modellinferenz, Visualisierung
4. Dokumentation und Präsentation der Ergebnisse

Typische Werkzeuge zur Analyse psychologischer Daten

Heute

- R (frei verfügbar, Datenwissenschaft, Psychologie)
- Python (frei verfügbar, Datenwissenschaft)
- Matlab (kommerziell, Ingenieurswissenschaften, Neuroimaging)

Früher

- SPSS (kommerziell, Sozialwissenschaften, Psychologie)
- JMP (kommerziell, Biologie, Psychologie)

PYPL Index März 2021

Worldwide, Mar 2021 compared to a year ago:

Rank	Change	Language	Share	Trend
1		Python	30.17 %	-0.2 %
2		Java	17.18 %	-1.2 %
3		JavaScript	8.21 %	+0.2 %
4		C#	6.76 %	-0.6 %
5	↑	C/C++	6.71 %	+0.8 %
6	↓	PHP	6.13 %	+0.0 %
7		R	3.81 %	+0.0 %
8		Objective-C	3.56 %	+1.1 %
9		Swift	1.82 %	-0.4 %
10	↑	Matlab	1.8 %	-0.0 %

- PopularitY of Programming Language
- Googlesuchanfragen zu Programmiersprachentutorials

Datenanalyseskripte

- Dokumentation aller Schritte vom Rohdatum zur Datenvisualisierung.
- Reproduktion wissenschaftlicher Ergebnisse durch Dritte.
- Essentieller Teil wissenschaftlicher Publikationen.
- Wesentlicher Teil wissenschaftlicher Arbeit im 21. Jahrhundert.

Programmierung ist das zentrale Handwerkzeug wissenschaftlicher Arbeit

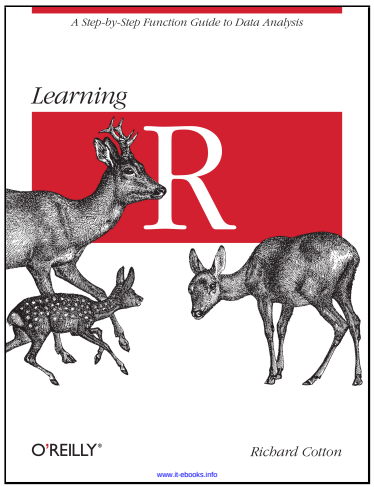
Kursziel ist das Sammeln erster datenanalytischer Programmiererfahrung

Modul C2 Computergestützte Datenanalyse

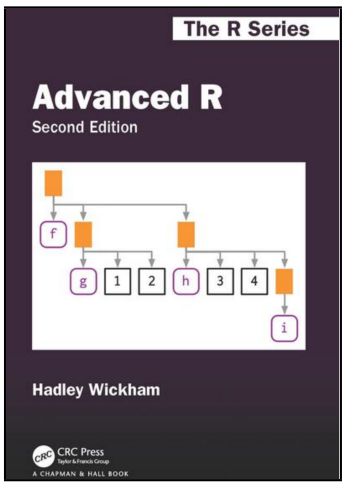
Einheit	Datum	Thema
(1) Datenanalyse und Programmierung	08.04.2021	Grundkenntnisse
(2) R und RStudio Grundlagen	15.04.2021	Programmierung
(3) Vektoren, Matrizen, Arrays	22.04.2021	Programmierung
(4) Listen und Dataframes	29.04.2021	Programmierung
(5) Environments und Funktionen	06.05.2021	Programmierung
(6) Kontrollstruktur, Schleifen, Apply	13.05.2021	Programmierung
(7) Datenimport und Datenbereinigung	20.05.2021	Datenanalyse
(8) Deskriptive Statistiken	27.05.2021	Datenanalyse
(9) Visualisierung I	03.06.2021	Visualisierung
(10) Visualisierung II	10.06.2021	Visualisierung
(11) T-Tests	17.06.2021	Datenanalyse
(12) Optimale Stichprobenumfänge	24.06.2021	Datenanalyse
(13) Einfaktorielle Varianzanalyse	01.06.2021	Datenanalyse
(14) Zweifaktorielle Varianzanalyse	08.07.2021	Datenanalyse
Klausurtermin	Juli	
Klausurwiederholungstermin	September	

Modul C2 Computergestützte Datenanalyse

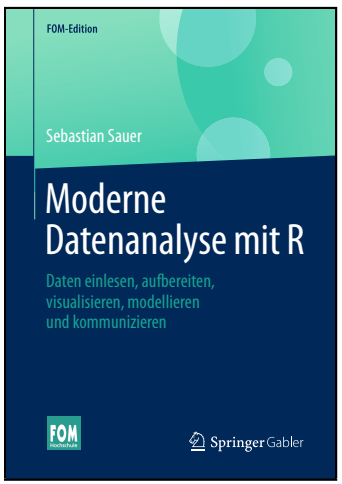
Weiterführende Literatur



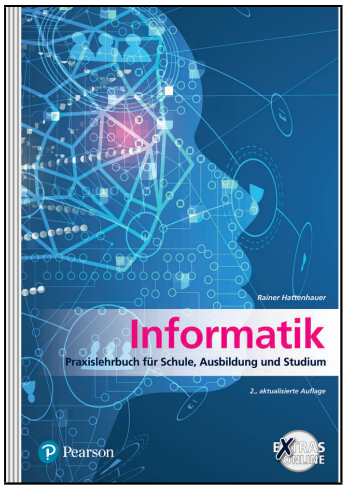
Cotton (2013)



Wickham (2019)



Sauer (2019)



Hattenhauer (2020)

Modul C2 Computergestützte Datenanalyse

Ziele des Moduls

- Grundlegende Datenanalysekompetenz
- Grundlagen für den Erwerb fortgeschrittener Datenanalysekompetenz
- Kompetenz im Umgang mit fachspezifischer statistischer Anwendungssoftware
- Anwendung von in Modul B2 Inferenzstatistik vorgestellten Methoden

Leistungsnachweis

- Klausur am Ende des Semesters
- Prüfungsvorleistung: Erfolgreiche Teilnahme Teilmodul 1
- Die Modulnote entspricht der Klausurnote des Teilmoduls 2
- Klausurfragen angelehnt an Selbstkontrollfragen

Datenanalyse und Programmierung

- Computergestützte Datenanalyse
- **Informatik und Rechnerarchitektur**
- Algorithmen und Programme
- Programmiersprachen
- Selbstkontrollfragen

Informatik (engl. Computer Science)

Bei der Informatik handelt es sich um die Wissenschaft von der systematischen Darstellung, Speicherung, Verarbeitung und Übertragung von Informationen, wobei besonders die automatische Verarbeitung mit Computern betrachtet wird. Sie ist zugleich Grundlagen- und Formalwissenschaft als auch Ingenieurdisziplin.

Wikipedia

Zentrale Komponenten der Informatik

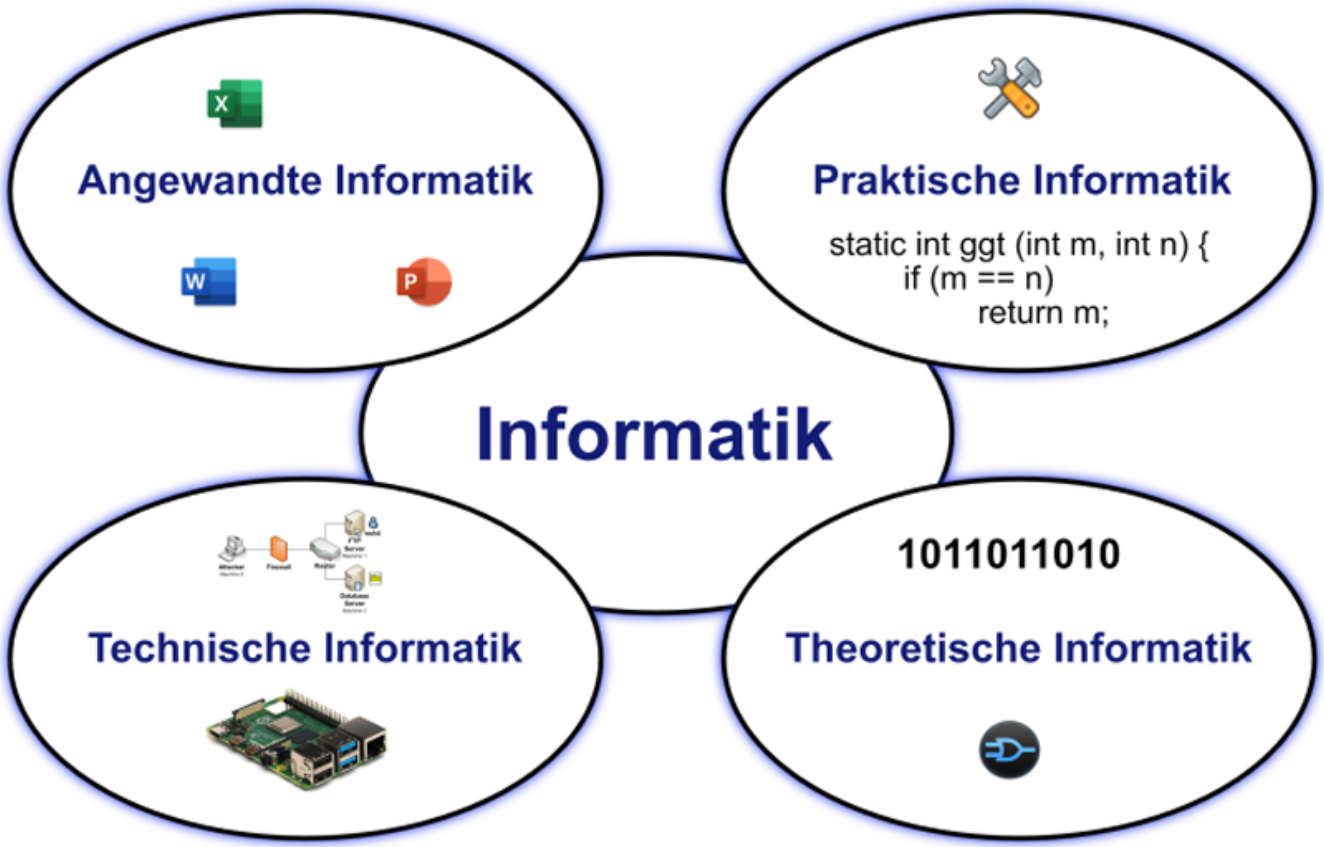
Computer

- Maschinen zum Datenspeichern und Ausführen einfacher Datenoperationen.
- Einfache Operationen mit extrem hoher Geschwindigkeit.
- Universalität durch Speicherung von Daten und Programmen.

Algorithmen und Programme

- *Programme* sind in einer *Programmiersprache* verfasste *Algorithmen*.
- Algorithmen sind Folgen von Anweisungen durchzuführender Operationen.
- Bei Algorithmen unterscheidet man
  - Beschreibung (Kochrezept, IKEA Bauanleitung, R Skript)
  - Anweisungen ("Mehl und Wasser vermengen", o - - -, x = c(1,2,3))
  - Durchführung (Kochvorgang, Zusammenbau, R Skript laufen lassen)

Teilgebiete der Informatik



Hattenhauer (2020) Informatik

Teilgebiete der Informatik (mit Relevanz für Psychologie)

Angewandte Informatik

Anwendungssoftware, Human-Computer-Interaction, Informatik und Gesellschaft

Technische Informatik

Mikroprozessortechnik, Rechnerarchitektur, Netzwerktechnik

Praktische Informatik

Programmierung, Algorithmen, Datenbanken

Theoretische Informatik

Automatentheorie, Berechenbarkeitstheorie, Komplexitätstheorie

Spezialgebiete der Informatik (mit Relevanz für Psychologie)

Maschinelles Lernen und Künstliche Intelligenz

Datenanalyse aus Sicht der Informatik

Computervisualistik

Bilderkennung und Bildsynthese, Virtuelle Realität, Augmented Reality

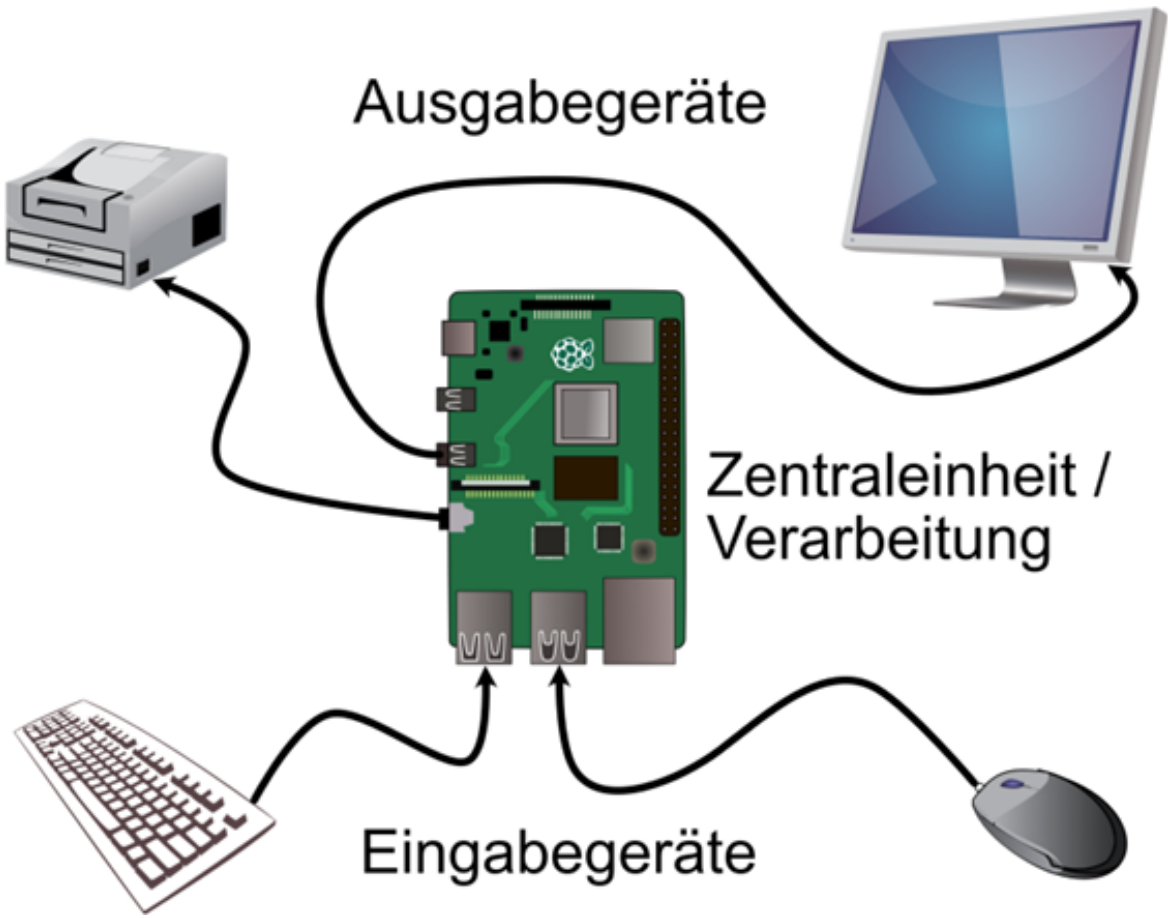
Computerlinguistik

Spracherkennung und Sprachsynthese

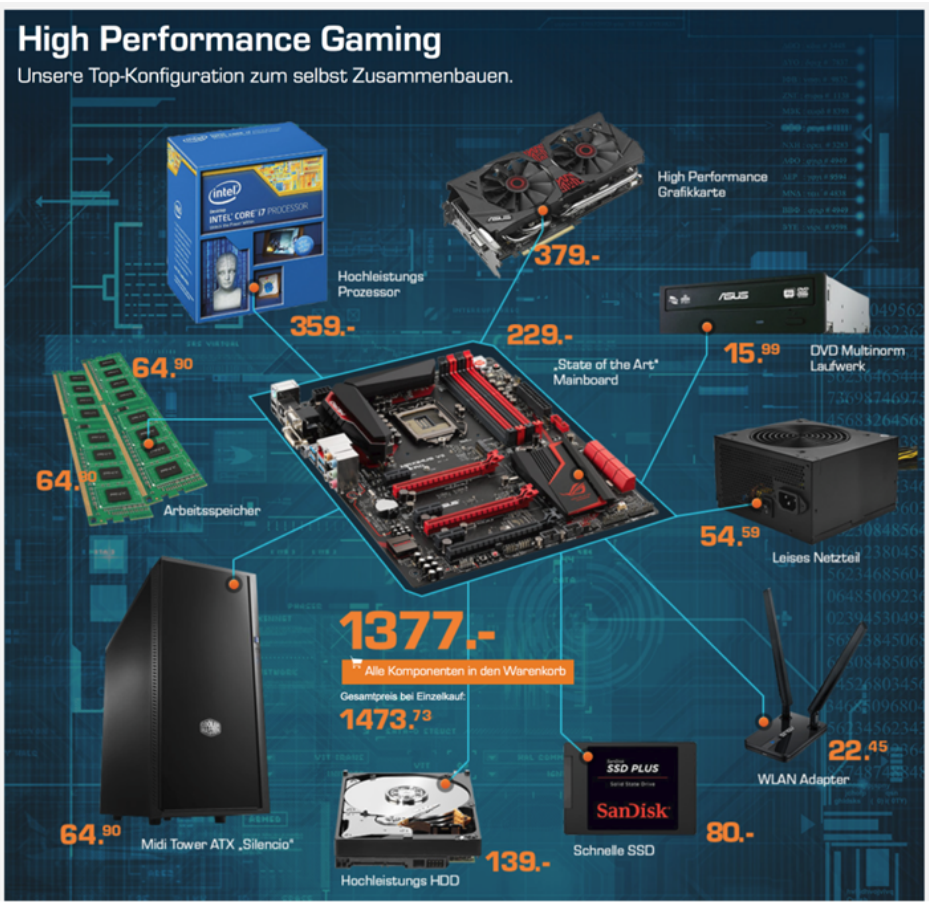
Bioinformatik

Lebenswissenschaften, Genomik, Bildgebende Verfahren der Medizin

Hardwarekomponenten eines Computers



Zentraleinheit eines Computers



Zentraleinheit (Hauptplatine, Motherboard, Mainboard)

CPU (Central Processing Unit/Mikroprozessor)

- Rechenwerk, Steuerwerk, und Leitwerk des Systems
- Cache (flüchtiger schneller Speicher)
- Intel(R) Core(TM) i5-7300HQ CPU @ 2.50GHz

RAM (Random Access Memory)

- Temporärer, flüchtiger Arbeitsspeicher des Systems
- Begrenzt, z.B. 16 GB

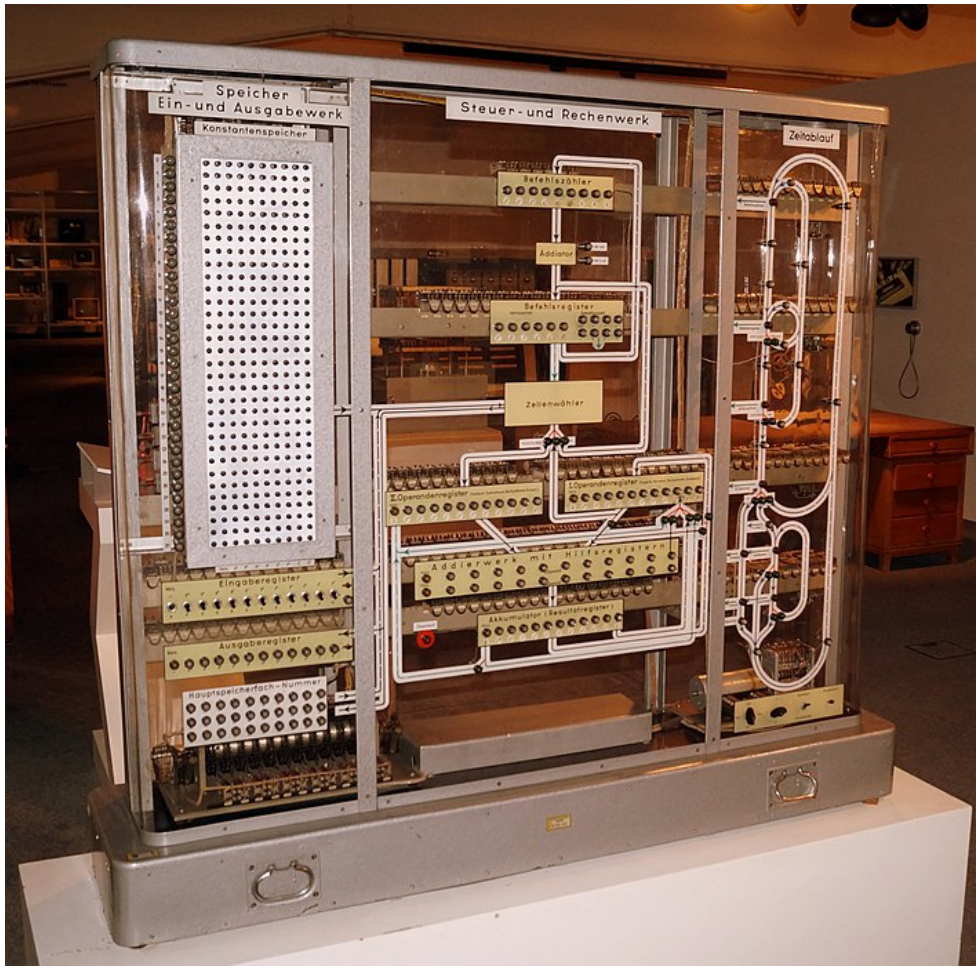
Massenspeicher

- Stationärer Speicher des Systems
- SSD (Solid State Drive), Cloudspeicher

GPU (Graphical Processing Unit)

- Leistungsstarke, speziell für Visualisierung optimierte Prozessoren
- Unterstützung der CPU in manchen Anwendungen, z.b. Neuronale Netze

Von-Neumann-Architektur

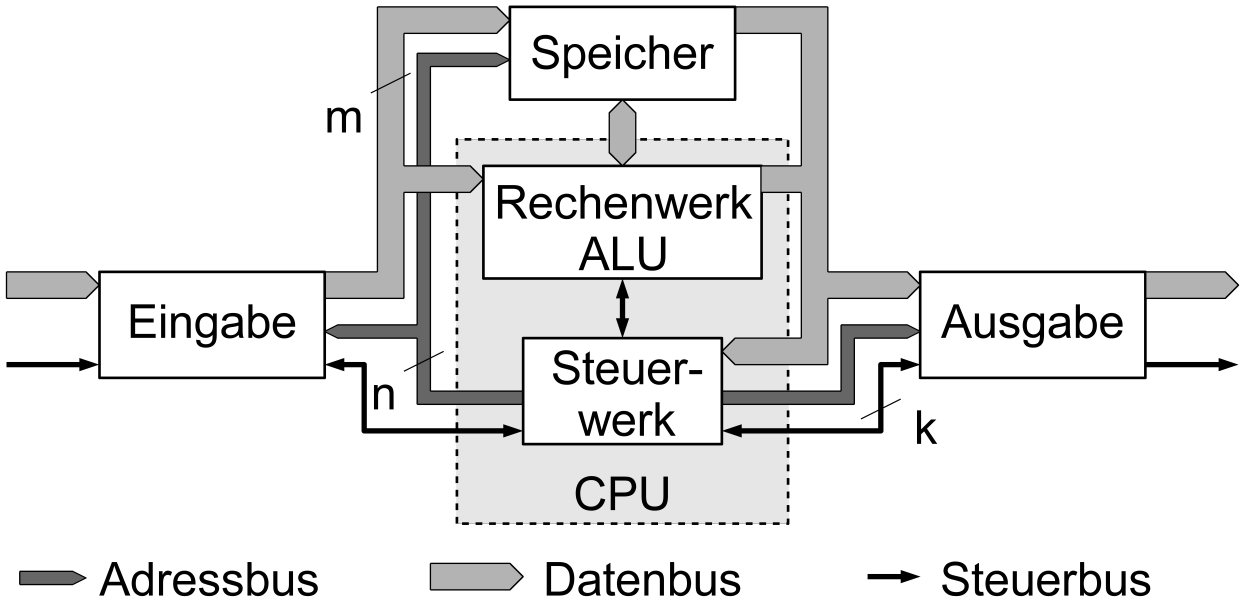


Quelle: Wikipedia

Von-Neumann-Architektur

- Rechner := Steuerwerk, Rechenwerk, Speicher, Eingabewerk, Ausgabewerk.
- Eingabe von Programmen und Daten in den Speicher.
- Daten, Programme, Zwischen- und Endergebnisse liegen im gleichen Speicher.
- Speicher ist in gleichgroße nummerierte (addressierte) Zellen unterteilt.
- Über die Adresse einer Speicherzelle kann deren Inhalt abgerufen/verändert werden.
- Aufeinanderfolgende Befehle eines Programms liegen in benachbarten Speicherzellen.
- Steuerwerk ruft den nächsten Befehl durch Erhöhen der Befehlsaddresse um 1 auf.
- Sprungbefehle erlauben eine Abweichung von der gespeicherten Reihenfolge
- Grundlegende Befehle sind
  - Arithmetische Befehle (z.B. Addition, Multiplikation)
  - Logische Vergleiche (z.B. logisches UND, logisches ODER)
  - Transportbefehle (z.B. Eingabewerk $\rightarrow$ Speicher, Speicher $\rightarrow$ Rechenwert)
- Alle Daten (z.b. Befehle, Adressen) werden binär codiert
- Binäre Enkodierung/Dekodierung geschieht durch geeignete Schaltwerke.

Von-Neumann-Architektur



Quelle: Wikipedia

- SISD System (single instruction stream, single data stream)
- Befehls- und Operandenfolge mit streng sequentieller Abarbeitung

Sequentielle Abarbeitung von Befehlen ist Grundprinzip der Progammierung

Datenanalyse und Programmierung

- Computergestützte Datenanalyse
- Informatik und Rechnerarchitektur
- **Algorithmen und Programme**
- Selbstkontrollfragen

Vom Realwertproblem zum Programm

Realwertproblem

- Das Problem, das mithilfe eines Computers gelöst werden soll.
- Bspw. Auswertung von Fragebogendaten einer psychologischen Studie.

Problemspezifikation

- Genaue sprachliche Fassung des Realweltproblems
- Bspw. der Methodenteil einer wissenschaftlichen Publikation.

Algorithmus

- Folge von Anweisungen zur Lösung des Problems.
- Bspw. Dateneinlesen, deskriptive Statistiken berechnen, T-Test durchführen.

Programm

- Ein Algorithmus, der von einem Computer ausgeführt werden kann.
- Eine in einer Programmiersprache verfasste Textdatei.

Definition (Algorithmus)

Ein *Algorithmus* ist eine Folge von Anweisungen, um aus gewissen Eingabedaten bestimmte Ausgabedaten herzuleiten, wobei folgende Bedingungen erfüllt sein müssen

- *Finitheit.* Die Anweisungsfolge ist in einem endlichen Text vollständig beschrieben sein.
- *Effektivität.* Jede Anweisung muss tatsächlich ausführbar sein.
- *Terminierung.* Der Algorithmus endet nach endlich vielen Anweisungen.
- *Determiniertheit.* Der Ablauf des Algorithmus ist zu jedem Punkt fest vorgeschrieben.

Wenn E die Menge der zulässigen Eingabedaten und A die Menge der zulässigen Ausgabedaten bezeichnet, dann ist ein Algorithmus eine Funktion

$$f : E \to A, e \mapsto f(e) \tag{1}$$

Umgekehrt heißen Funktionen, die durch einen Algorithmus beschrieben werden können, *berechenbare Funktionen*.

Bemerkung

- Effektivität sollte nicht mit Effizienz verwechselt werden.

Programmiersprache

- Bestimmt die Regeln, denen ein Programm gehorchen muss
- Definiert eine Syntax, also Vokabular und Programmaufbau
- Definiert Semantik, also die Bedeutung der erlaubten Anweisungen

```
1
2  // Quellcodebeispiel in C++
3
4  #include <cstdlib>
5  #include <iostream>
6
7  using namespace std;
8
9  int main(int argc, char *argv[])
10 {
11     int alter; // Variable vom Typ Integer
12
13     cout << "Wie alt bist du?";
14     cin >> alter;
15     cout << "Du bist " << alter << " Jahre alt" << endl;
16     getc();
17     return 0;
18 }
19
```

Quelle: Wikipedia

Maschinensprache

- Elementare Operationsbefehle (z.b. Speichern, Vergleichen, Addieren)
- Elementare Operationsbefehle werden als Binärzahlen kodiert

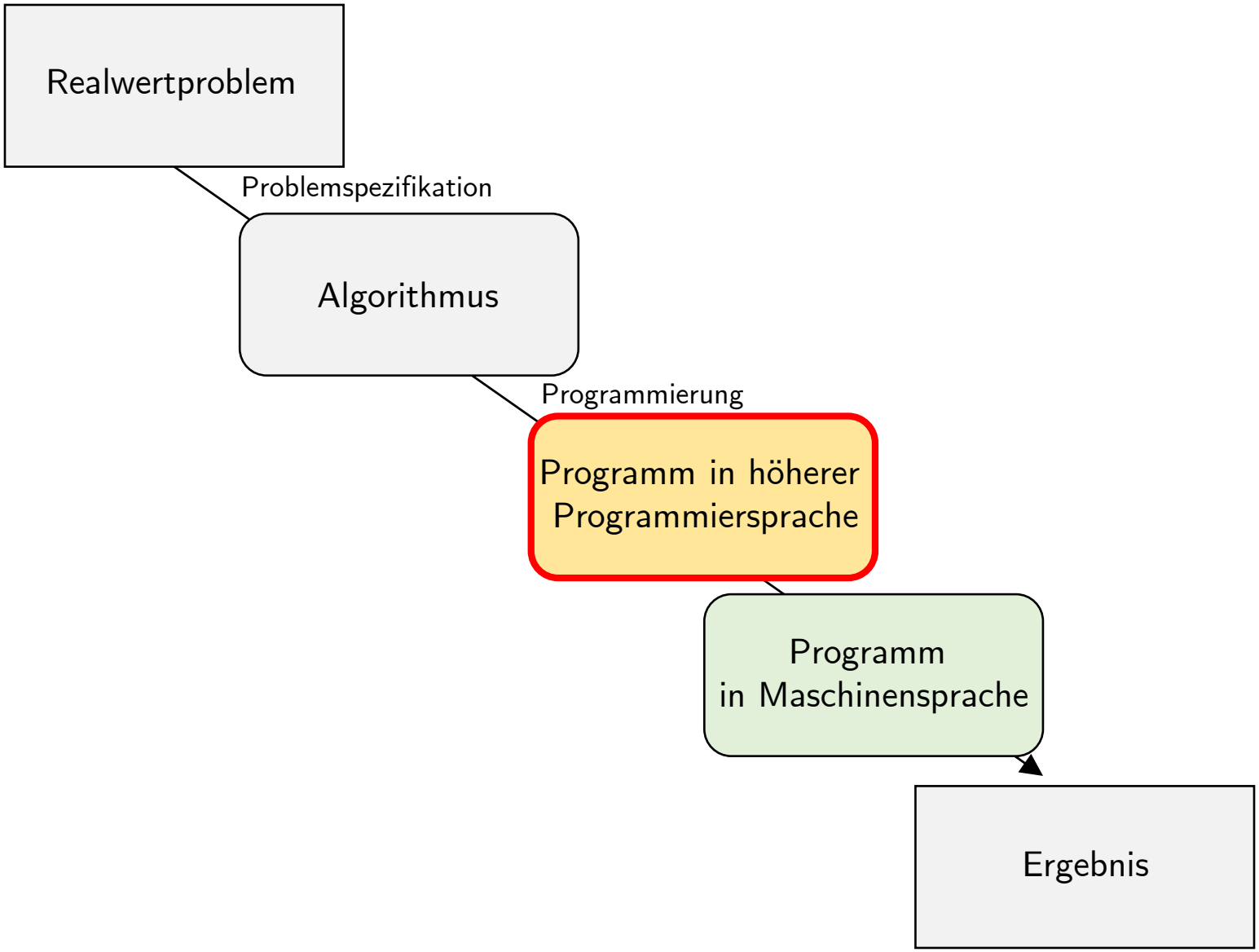
Addiere Inhalt R1 zu Inhalt R2	$\Rightarrow$ 1001 0010
Erhöhe Inhalt R um 1	$\Rightarrow$ 1001 0110
Übertrage Inhalt R1 nach R3	$\Rightarrow$ 0010 0011

- Programme in Maschinensprache heißen *Maschinenprogramme*
- De facto führt ein Computer nur Maschinenprogramme aus
- Für Menschen ist die Programmierung in Maschinensprache mühselig.

Höhere Programmiersprache

- An die menschliche Sprache angelehnte Wörter und Sätze
- Interpreter oder Compiler übersetzen Programme in Maschinensprache
- R, Python, Matlab, C++, Java, FORTRAN, COBOL,...

Algorithmen und Programme



Programmierparadigmen

Imperative Programmierung

Problemlösungsweg wird als Folge von Anweisungen (Befehlen) vorgegeben

Befehle verarbeiten Daten, die mithilfe von *Variablen* adressiert werden.

- Prozedurale imperative Programmierung
  - Daten und sie manipulierende Befehle werden separat behandelt
  - Prozeduren (Funktionen) als zentrales Strukturkonzept
- Objektorientierte imperative Programmierung
  - Daten und manipulierende Befehle werden als *Objekt* zusammengefasst
  - Objekte als zentrales Strukturkonzept

Deklarative Programmierung (hier nicht von Relevanz)

Beschreibung des Problems steht im Vordergrund.

Der Lösungsweg wird nach vorgegebenen Regeln vom Computer ermittelt.

Generationen von Programmiersprachen

1. Generation (1GL)
- Maschinensprachen
- 10110000 01100001 (in hexadezimaler Darstellung: B0 61)

2. Generation (2GL)
- Assemblersprachen ab 1950, erste Form der symbolischen Programmierung
- Bspw. “MOV Al, 61H” # Intel-Prozessor-spezifische Sprache

3. Generation (3GL)
- Höhere Programmiersprachen ab 1970 wie FORTRAN, C, C++, Java
- Programmierfreundlich, prozessor-unabhängig

4. Generation (4GL)
- Höhere Programmiersprachen ab 1980 wie Python, Matlab, R
- Codeoverhead Minimisierung, Automation, Flexibilität, Multiparadigmatisch

Kompilierte Programmiersprachen

- Gesamter Quellcode wird vor Ausführung in Maschinensprache übersetzt.
- Das Übersetzungsprogramm heißt *Compiler*.
- Der übersetzte Maschinencode wird vom Prozessor ausgeführt.
- Das ausführbare Programm wird nicht übersetzt und läuft schnell.
- Bei Änderungen des Quellcodes muss neu kompiliert werden.
- Beispiele für kompilierte Sprachen sind Java, C, C++.

Interpretierte Programmiersprachen

- Quellcode wird während der Ausführung in maschinennahe Sprache übersetzt.
- Das Ausführungsprogramm heißt *Interpreter*.
- Das Programm läuft aufgrund der Interpretation langsamer.
- Bei Änderungen des Quellcodes muss nicht neu interpretiert werden.
- Beispiele für interpretierte Sprachen sind Python und R.

Die Programmiersprache R ist

... eine imperative Programmiersprache ,

... perse objektorientiert, kann aber prozedural genutzt werden,

... eine höhere Programmiersprache der 4. Generation,

... eine interpretierte Sprache,

... auf die statistische Analyse von Daten zugeschnitten.

Datenanalyse und Programmierung

- Computergestützte Datenanalyse
- Informatik und Rechnerarchitektur
- Algorithmen und Programme
- **Selbstkontrollfragen**

Selbstkontrollfragen

1. Geben Sie die typische Struktur einer computergestützten Datenanalyse wieder.
2. Erläutern Sie den Begriff "Datenanalyseskript".
3. Definieren Sie den Begriff "Informatik".
4. Erläutern Sie die Akronyme CPU, RAM, SSD, und GPU.
5. Nennen Sie wesentliche Aspekte der Von-Neumann Rechnerarchitektur.
6. Definieren Sie den Begriff des Algorithmus.
7. Erläutern Sie den Zusammenhang von Algorithmen und Programmen.
8. Was bezeichnen die Syntax und Semantik einer Programmiersprache?
9. Differenzieren Sie die Begriffe "Maschinensprache" und "höhere Programmiersprache".
10. Erläutern Sie die Prinzipien der prozeduralen und objektorientierten imperativen Programmierung.
11. Skizzieren Sie die Entwicklung der Programmiersprachen der ersten bis vierten Generation.
12. Grenzen Sie die Begriffe der kompilierten und interpretierten Programmiersprache voneinander ab.

Literatur

Cotton, R. (2013). *Learning R*. O'Reilly, Beijing ; Sebastopol, CA, first edition edition.

Hattenhauer, R. (2020). *Informatik*. Pearson, second edition.

Sauer, S. (2019). *Moderne Datenanalyse mit R: Daten einlesen, aufbereiten, visualisieren, modellieren und kommunizieren*. FOM-Edition. Springer Fachmedien Wiesbaden, Wiesbaden.

Wickham, H. (2019). *Advanced R, Second Edition*. CRC Press.