

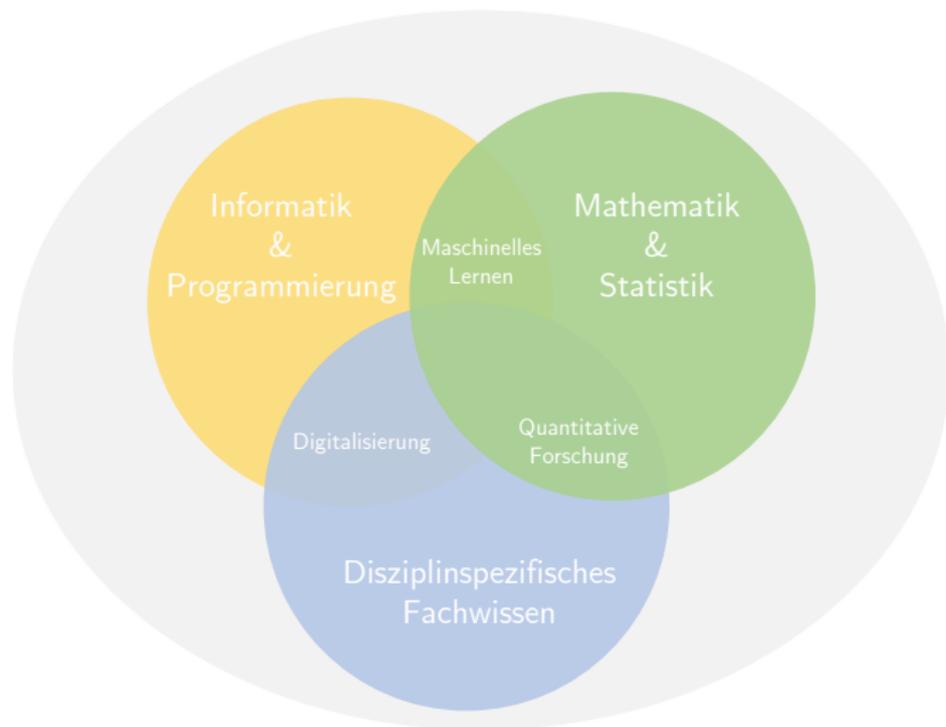


Grundlagen der Mathematik und Informatik

BSc Psychologie WiSe 2021/22

Prof. Dr. Dirk Ostwald

(7) Grundbegriffe der Informatik



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

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

- Wissenschaftliche Daten liegen heutzutage als digitale Daten vor.
- Digitale Daten werden mit Hilfe eines Computers analysiert.
- Zur Analyse von digitalen Daten schreibt man Computerprogramme.
- Diese Computerprogramme heißen Datenanalyseskripte.

Struktur computergestützter Datenanalyse

1. Einlesen und Bereinigen eines digitalen Datensatzes
2. Berechnung und Visualisierung deskriptiver Statistiken
3. Probabilistische Modellierung und Inferenz
4. Dokumentation und Präsentation der Ergebnisse

Typische Werkzeuge zur Analyse psychologischer Daten

- **R** (frei, Datenwissenschaft, Statistik, Psychologie)
- **Python** (frei, Datenwissenschaft, Anwendung)
- **Matlab** (kommerziell, Engineering, Neuroimaging)

Altmodisch

- **SPSS** (kommerziell, Sozialwissenschaften, Psychologie)
- **JMP** (kommerziell, Biologie, Psychologie)
- **STATA** (kommerziell, Wirtschaftswissenschaften)

PYPL Index September 2021

Worldwide, Sept 2021 compared to a year ago:

Rank	Change	Language	Share	Trend
1		Python	29.48 %	-2.4 %
2		Java	17.18 %	+0.7 %
3		JavaScript	9.14 %	+0.8 %
4		C#	6.94 %	+0.6 %
5		PHP	6.49 %	+0.4 %
6		C/C++	6.49 %	+0.9 %
7		R	3.59 %	-0.5 %
8	↑↑↑↑	TypeScript	2.18 %	+0.3 %
9		Swift	2.1 %	-0.4 %
10	↓↓↓	Objective-C	2.06 %	-0.6 %
11	↑	Kotlin	1.91 %	+0.3 %
12	↓↓↓	Matlab	1.74 %	-0.1 %

- Popularity of Programming Language
- Googlesearchanfragen zu Programmiersprachentutorials

Datenanalyseskripte

- Dokumentation aller Schritte vom Rohdatum zur Datenvisualisierung.
- Reproduktion wissenschaftlicher Ergebnisse durch Dritte.
- Essentieller Teil wissenschaftlicher Publikationen.
- Essentieller Teil täglicher wissenschaftlicher Arbeit.

Zusammenfassung

- Die Digitalisierung betrifft insbesondere auch die Wissenschaft.
- Forschungsdatenmanagement ist eine akute Herausforderung.
- Programmierung als zentrales Handwerkszeug wissenschaftlicher Arbeit.
- Informatikkenntnisse sind in der Arbeitswelt unverzichtbar.
- Dies gilt auch für Psychotherapeut:innen (z.B. Online-Intervention).

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

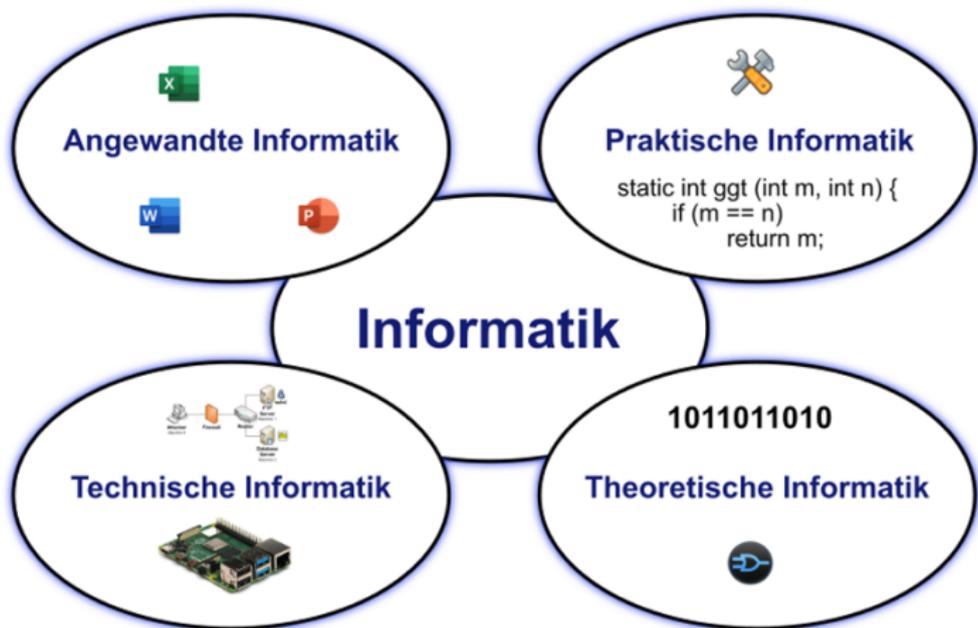
Computer

- 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 die 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 die Psychologie

Maschinelles Lernen und Künstliche Intelligenz

- Datenanalyse aus Sicht der Informatik

Computervisualistik

- Bilderkennung und Bildsynthese, Virtuelle Realität, Augmented Reality

Computerlinguistik

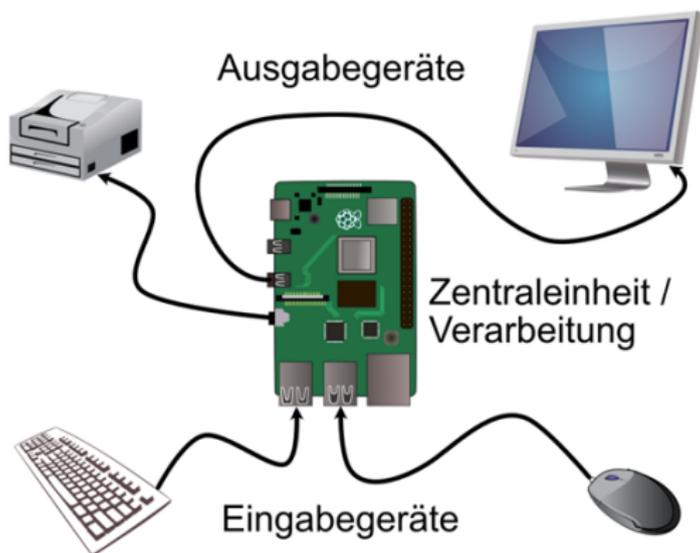
- Spracherkennung und Sprachsynthese

Bioinformatik

- Lebenswissenschaften, Genomik, Bildgebende Verfahren der Medizin

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

Hardwarekomponenten eines Computers



Hattenhauer (2020) Informatik

Zentraleinheit eines Computers

High Performance Gaming
Unsere Top-Konfiguration zum selbst Zusammenbauen.

High Performance Grafikkarte 379.-

Hochleistungsprozessor 359.-

DVD Multiformat Laufwerk 15.99

State of the Art Mainboard 229.-

Arbeitsspeicher 64.90

Mid Tower ATX „Silencio“ 64.90

Hochleistungs HDD 139.-

SanDisk schnelle SSD 80.-

LAN Netzteil 54.99

WLAN Adapter 22.95

1377.-
Alle Komponenten in den Warenkorb

1473.73
inkl. Versand

Hattenhauer (2020) Informatik

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

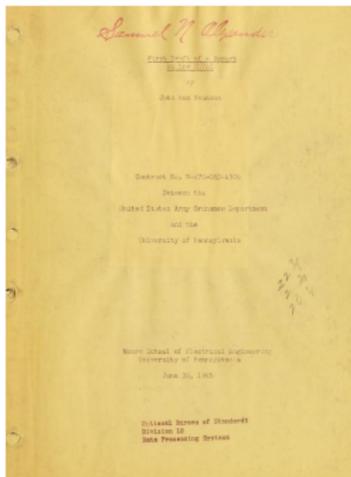
Massenspeicher

- 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



John von Neumann (1945) First Draft of a Report on the EDVAC (Electronic Discrete Variable Automatic Computer)

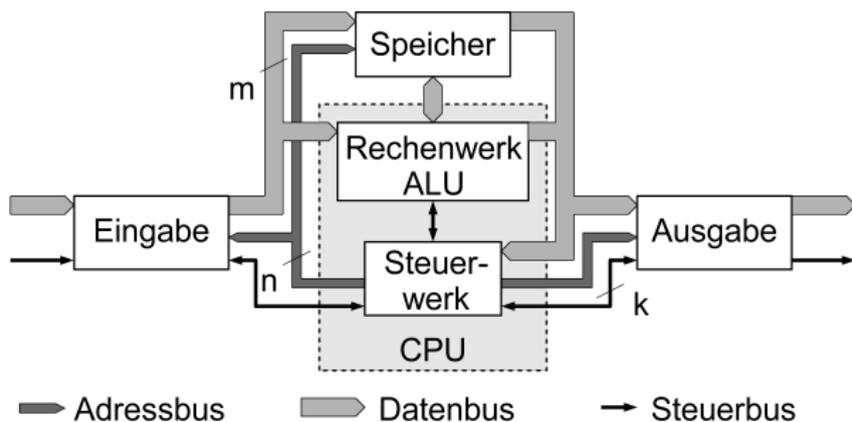
“The orders which are received by CC [Central Control] come from M [Memory], i.e. from the same place where the numerical material is stored.” (Sec. 14.0)

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 (adressierte) 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 Befehlsadresse 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 → Speicher, Speicher → 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

⇒ Daten und Programme können in den Speicher geladen werden.

⇒ Sequentielle Abarbeitung von Befehlen ist Grundprinzip der Programmierung

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

Vom Realwertproblem zum Programm

Realwertproblem

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

Problemspezifikation

- Genaue sprachliche Fassung des Realwertproblems.
- Beispiel: Methodenteil einer wissenschaftlichen Publikation.

Algorithmus

- Folge von Anweisungen zur Lösung des Problems.
- Beispiel: 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

- *Finithheit*. 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

$$f : E \rightarrow A, e \mapsto f(e) \quad (1)$$

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

```
#if [ -z "$USER_NAME" -o -z "$USER_TYPE" -o -z "$GROUP" ]
if [ -z "$USER_NAME" -o -z "$USER_TYPE" ]
then
#     echo "Please set the user name, type and group"
    echo "Please set the user name and type"
    exit 1
fi

# generate a random password
# -y: include special characters
# -n: include numbers
# -l: one generated passwords per line
#pwgen -y 15 -n 5 -l
echo "Propositions for random passwords to use in next step:"
pwgen -s -n -l 15 5

# add the user
# requires password to be given via input
adduser --firstuid 1000 --lastuid 9999 --no-create-home ${USER_NAME}
```

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, . . .

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

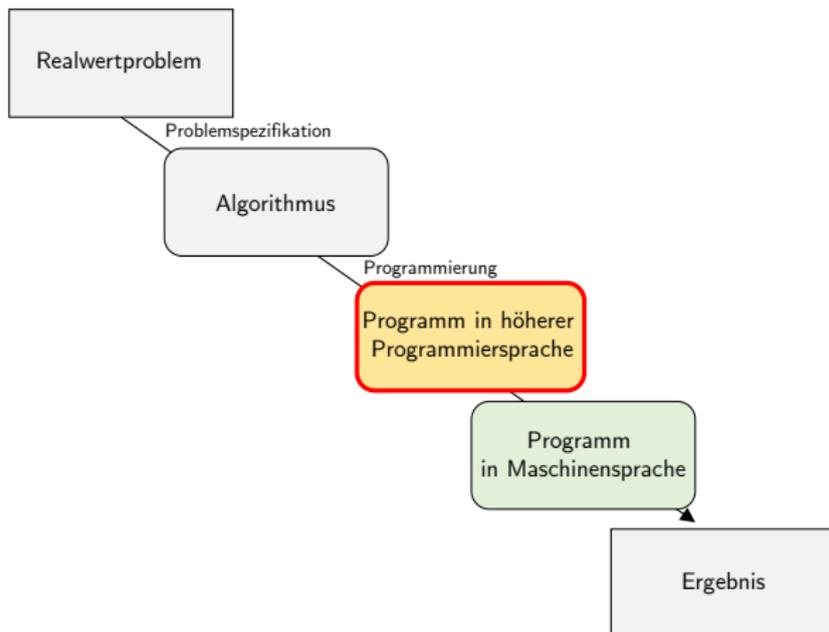
3. Generation (3GL)

- 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

4GL Programmierung



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) bilden das zentrale Strukturkonzept.
 - **Objektorientierte imperative Programmierung**
 - Daten und manipulierende Befehle werden als *Objekte* zusammengefasst.
 - Objekte bilden das zentrale Strukturkonzept.
- Praktisch liegen oft Mischformen vor.

Kompilierte Programmiersprachen

- Gesamter Quellcode wird *vor der 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
 - Informatik
 - 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. Skizzieren Sie Prinzipien der prozeduralen und objektorientierten imperativen Programmierung.
11. Skizzieren Sie die Entwicklung der Programmiersprachen der ersten bis vierten Generation.
12. Differenzieren Sie die Begriffe der kompilierten und der interpretierten Programmiersprachen.