



Programmierung und Deskriptive Statistik

BSc Psychologie WiSe 2023/24


Belinda Fleischmann

Inhalte basieren auf Programmierung und Deskriptive Statistik von Dirk Ostwald, lizenziert unter CC BY-NC-SA 4.0

Herzlich willkommen!

(1) Einführung


Homepage der Abteilung für Methodenlehre I





OTTO VON GUERICKE
UNIVERSITÄT
MAGDEBURG


INSTITUT FÜR PSYCHOLOGIE

Sitemap Impressum Kontakt

Suchbegriff 


INSTITUT | STUDIUM | FORSCHUNG | PERSONEN

DIREKTLINKS 


[Home](#) > [Institut](#) > [Abteilungen des Ins...](#) > [Methodenlehre I: Experimentelle und Neuro...](#) > [Forschung](#) | [Lehre](#) | [CBBS Imaging Platform](#) | [Team](#)

Methodenlehre I: Experimentelle und Neurowissenschaftliche Psychologie

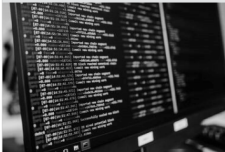
Forschung




Lehre




CBBS Imaging Platform



Team






Kontakt

Abteilungsleitung


[Prof. Dr. Dirk Ostwald](#)

 dirk.ostwald@ovgu.de

Tel.: + 49 391 67 57370

Abteilungsassistentz

[Birgit Müller](#)

 birgit.mueller@ovgu.de

Tel.: +49 391 67 58464

Anschrift

Otto-von-Guericke-Universität
Magdeburg
Institut für Psychologie
Universitätsplatz 2
Gebäude 24
39106 Magdeburg

[Anfahrt](#)

Formalia

Grundbegriffe der Informatik

R und Visual Studio Code

Formalia

Grundbegriffe der Informatik

R und Visual Studio Code

Ziele der Veranstaltung

Programmierung und Deskriptive Statistik (Modul C1)

- Einführung in die datenanalytische Programmierung
- Einführung in die Auswertung deskriptiver Statistiken mit R in Visual Studio Code

Design, Analyse, Dokumentation (Modul C2)

- Einführung in das Design quantitativer empirischer Studien
- Einführung in die Analyse und Dokumentation empirischer Daten

Empfohlene Vorbereitung:

Vorkurs "Grundlagen der Mathematik und Informatik"

TODO: [insert new link](#)


- Termine: Mittwochs in Raum G05-307
 - Gruppe 1 um 11-13 Uhr
 - Gruppe 2 um 13-15 Uhr
- Kursmaterialien (Folien, Videos, RMarkdown Code) auf der [Kurswebseite](#)
- Code auf [Github](#)
- Ankündigungen über die [Moodleseite](#)
- Vorherige Iteration des Kurses [PDS, WS 2023/2023](#)
- Empfohlene Vorbereitung: "[Grundlagen der Mathematik](#)"
- Q&A im [Mattermost-Channel](#)
 - Einmalige Registrierung zum Team "bsc-psy-2023" über [hier](#)
- Leistungsnachweis: Unbenotet, Multiple Choice, digitales Format
 - Teil 1 vor der Weihnachtspause
 - Teil 2 am Semesterende

TODO: insert new links

Termine

Datum	Einheit	Thema
11.10.23	Einführung	(1) Einführung
18.10.23	R Grundlagen	(2) R und Visual Studio Code
25.10.23	R Grundlagen	(2) R und Visual Studio Code
01.11.23	R Grundlagen	(3) Vektoren
08.11.23	R Grundlagen	(4) Matrizen und Arrays
15.11.23	R Grundlagen	(5) Listen und Dataframes
22.11.23	R Grundlagen	(6) Datenmanagement
30.11.23	R Grundlagen	(7) Häufigkeitsverteilungen
06.12.23	R Grundlagen	(8) Verteilungsfunktionen und Quantile
13.12.23	Deskriptive Statistik	(9) Maße der zentralen Tendenz
20.12.23	Deskriptive Statistik	(10) Maße der Datenvariabilität
Weihnachtspause		
10.01.24	Deskriptive Statistik	(11) Anwendungsbeispiel (Deskriptive Statistik)
17.01.24	Inferenzstatistik	(12) Anwendungsbeispiel (Parameterschätzung, Konfidenzintervalle)
24.01.24	Inferenzstatistik	(13) Anwendungsbeispiel (Hypothesentest)


Webseite des Kurses (Folien, Videos)




OTTO VON GUERICKE
UNIVERSITÄT
MAGDEBURG

INSTITUT FÜR PSYCHOLOGIE

Sitemap Impressum Kontakt

Suchbegriff 

INSTITUT | STUDIUM | FORSCHUNG | PERSONEN

DIREKTLINKS 

Home > Methodenlehre I: E... > Lehre > Wintersemester 2023 > Programmierung und Deskriptive Statistik

Programmierung und Deskriptive Statistik

Dieser Kurs gibt eine Einführung in die datenanalytische Programmierung und die Auswertung deskriptiver Statistiken mit **R** in **RStudio**.

Nach der [Studien- und Prüfungsordnung für den BSc Psychologie \(06/2020\)](#) und dem [Modulhandbuch für den BSc Psychologie \(09/2020\)](#) entspricht dieser Kurs dem Modul *CT Computergestützte Datenanalyse*.

Der RMarkdown Code der Vorlesungsfolien ist auf [github](#) und hier verfügbar:

- [R Markdown Code](#)

Als weiterführende Literatur zur Programmierung mit R werden empfohlen:


- Sauer, S. (2019) *Moderne Datenanalyse mit R*
- Wickham, H. (2019) *Advanced R*
- Cotton, R. (2013) *Learning R*
- Murrell, P. (2021) *R Graphics*

Als Einstieg in die Deskriptive Statistik bieten sich an

- Henze, N. (2016) *Deskriptive Statistik*
- Fahrmeier et al. (2016) *Statistik, Kapitel 1-3*

Vorlesungseinheiten

(0) Grundbegriffe der Informatik



(7) Grundbeg...

(1) Einführung

Letzte Änderung: 10.10.2022 - Ansprechpartner: [Webmaster](#)

**Kontakt****Abteilungsleitung**

>Prof. Dr. Dirk Ostwald

dirk.ostwald@ovgu.de

Tel.: + 49 391 67 57370

Abteilungsassistentz

>Birgit Müller

birgit.mueller@ovgu.de

Tel.: +49 391 67 58464

AnschriftOtto-von-Guericke-Universität
Magdeburg
Institut für Psychologie
Universitätsplatz 2
Gebäude 24
39106 Magdeburg

>Anfahrt

TODO: Links und Screenshots aktualisieren

Git-repository des Kurses (Folien, Source Codes)

The screenshot shows a GitHub repository page for 'belindamef/progr-und-deskr-stat-23'. The repository is public and has 7 commits. The file list includes: 1_Einfuehrung, doc, .gitignore, Header.tex, README.md, R_common.R, and Referenzen.bib. The README.md file is selected, showing the title 'Programmierung und Deskriptive Statistik (WiSe 2022/2023)' and a table of contents under 'Inhaltsverzeichnis'.

Repository Details:

- Repository: belindamef/progr-und-deskr-stat-23
- Public
- 7 commits
- 19 seconds ago

File List:

File	Commit	Time
1_Einfuehrung	Add (1) Einfuehrung	19 seconds ago
doc	Add Abbildungen fue README	9 minutes ago
.gitignore	Add .gitignore	1 hour ago
Header.tex	Add common files	34 minutes ago
README.md	Update README	10 minutes ago
R_common.R	Add common files	34 minutes ago
Referenzen.bib	Add common files	34 minutes ago

README.md Content:

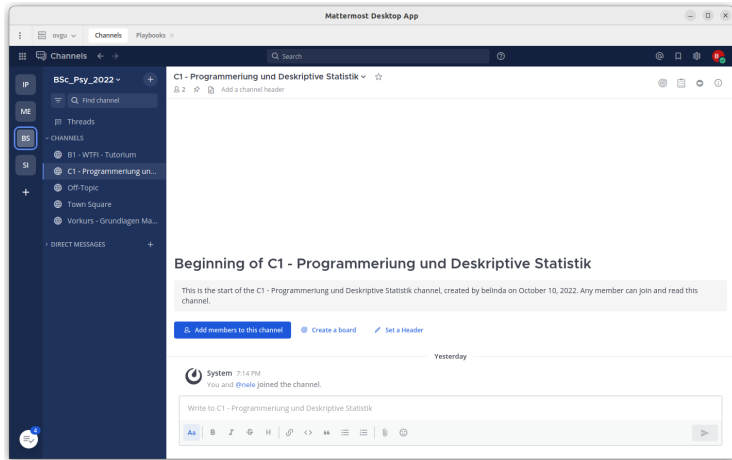
Programmierung und Deskriptive Statistik (WiSe 2022/2023)

Inhaltsverzeichnis

- Allgemeine Informationen
- Anleitung, wie man beitragen kann
 - Schritt 1: Repository kopieren (Fork)
 - Wichtige Eigenschaften eines Forks:

TODO: Links und Screenshots aktualisieren

Mattermost-Team BSc Psych 2023



TODO: Links und Screenshots aktualisieren

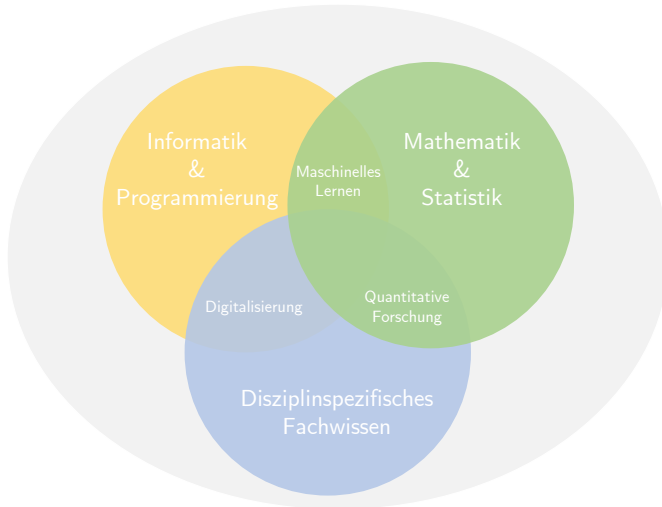
Q & A

Formalia

Grundbegriffe der Informatik

R und Visual Studio Code

Datenwissenschaften



Formalia

Grundbegriffe der Informatik

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

R und Visual Studio Code

Formalia

Grundbegriffe der Informatik

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

R und Visual Studio Code

- 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.

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)

Programmiersprachen Trends

PYPL Index (Stand: September 2023)

Worldwide, Sept 2023 :

Rank	Change	Language	Share	1-year trend
1		Python	27.99 %	+0.1 %
2		Java	15.9 %	-1.1 %
3		JavaScript	9.36 %	-0.1 %
4		C#	6.67 %	-0.4 %
5		C/C++	6.54 %	+0.3 %
6		PHP	4.91 %	-0.4 %
7		R	4.4 %	+0.2 %
8		TypeScript	3.04 %	+0.2 %
9	↑↑	Swift	2.64 %	+0.6 %
10		Objective-C	2.15 %	+0.1 %
11	↑↑	Rust	2.12 %	+0.5 %
12	↓↓↓	Go	2.0 %	-0.1 %
13	↓	Kotlin	1.78 %	-0.0 %
14		Matlab	1.58 %	+0.1 %
15		Ruby	1.05 %	-0.1 %

- Popularity of Programming Language
- Basierend auf Googlesearchanfragen zu Programmiersprachentutorials

- Dokumentation aller Schritte von Rohdaten bis zur Datenvisualisierung.
- Reproduktion wissenschaftlicher Ergebnisse durch Dritte.
- Essentieller Teil wissenschaftlicher Publikationen.
- Essentieller Teil täglicher wissenschaftlicher Arbeit.

- 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).

Formalia

Grundbegriffe der Informatik

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

R und Visual Studio Code

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.

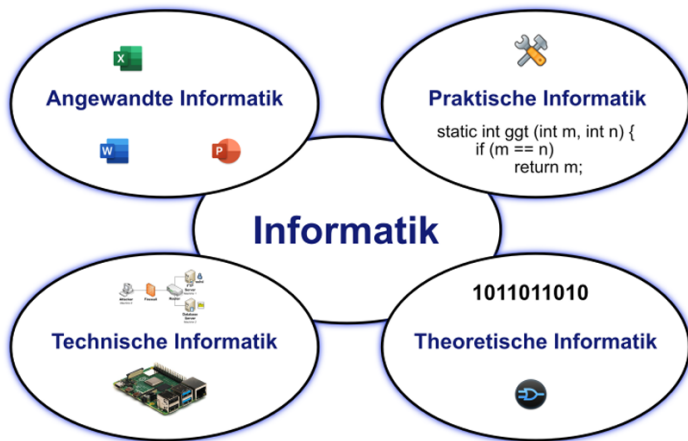
Wikipedia

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”, $o \leftarrow -, x = c(1,2,3)$)
 - Durchführung (Kochvorgang, Zusammenbau, R Skript laufen lassen)



Hattenhauer (2020) Informatik

Angewandte Informatik

- Anwendungssoftware, **Human-Computer-Interaction**, Informatik und Gesellschaft

Technische Informatik

- Mikroprozessortechnik, Rechnerarchitektur, Netzwerktechnik

Praktische Informatik

- **Programmierung**, **Algorithmen**, Datenbanken

Theoretische Informatik

- Automatentheorie, Berechenbarkeitstheorie, Komplexitätstheorie

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

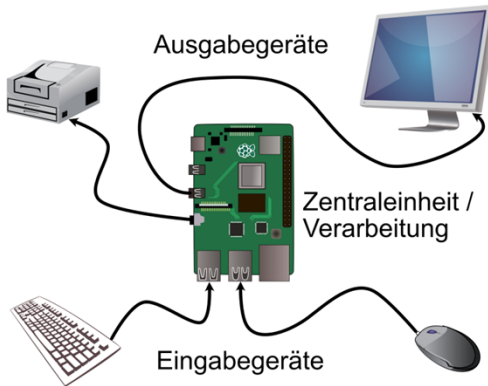
Formalia

Grundbegriffe der Informatik

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

R und Visual Studio Code

Hardwarekomponenten eines Computers



Hattenhauer (2020) Informatik

Zentraleinheit eines Computers (Hauptplatine, Motherboard, Mainboard)

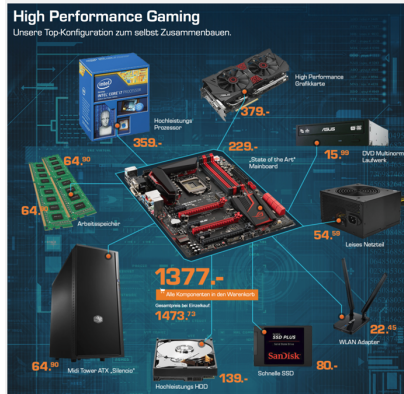


Figure 1: Hattenhauer (2020) Informatik

Hattenhauer (2020) Informatik

Zentraleinheit eines Computers (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)

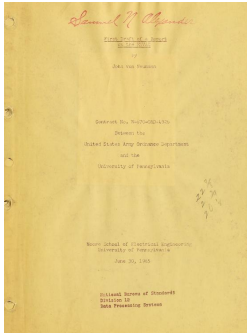
- Temporärer, flüchtiger Arbeitsspeicher des Systems
- Begrenzt, z.B. 16 GB

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

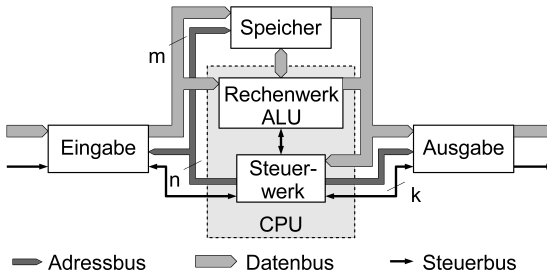


John von Neumann (1945) First Draft of a Report on the EDVAC (Electronic Discrete Variable Automatic Computer)

“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

- 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.



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

Formalia

Grundbegriffe der Informatik

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

R und Visual Studio Code

Vom Realwertproblem zum Programm

Realwertproblem

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

Problemspezifikation

- Genaue sprachliche Fassung des Realweltproblems.
- z.B. Methodenteil einer wissenschaftlichen Publikation.

Algorithmus

- Folge von Anweisungen zur Lösung des Problems.
- z.B. 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

- *Fintheit*. Die Anweisungsfolge muss 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.

Eine 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,...

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

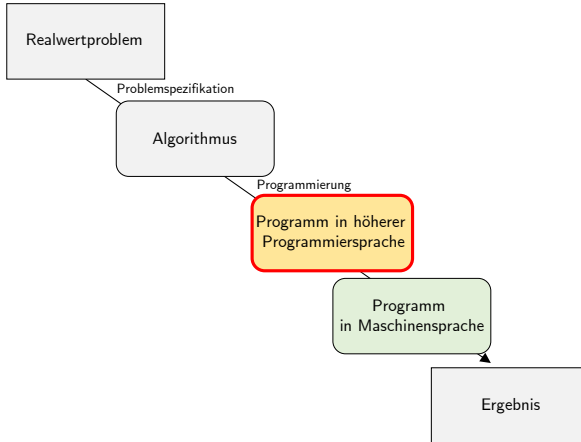
3. Generation (3GL)

- 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 ,
- ... per se 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.

Formalia

Grundbegriffe der Informatik

R und Visual Studio Code