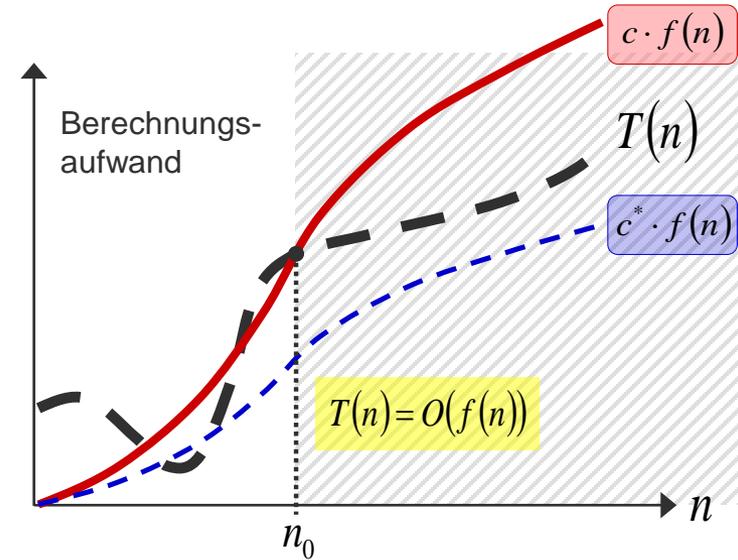
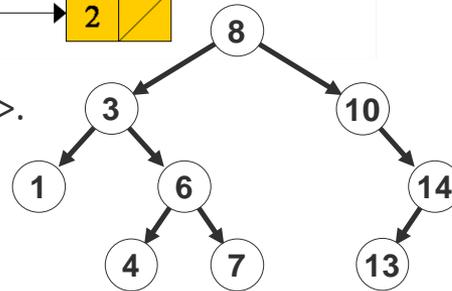


$\langle \text{Zahl} \rangle ::= \langle \text{Ziffer} \rangle \mid \langle \text{Ziffer} \rangle \langle \text{Zahl} \rangle.$
 $\langle \text{Ziffer} \rangle ::= "0" \mid "1".$
 $\langle \text{Op} \rangle ::= "*" \mid "/" \mid "+" \mid "-".$



Informatik II – SoSe 2018



Ein wenig über mich:

Curriculum Vitae ++

Ausbildung

- Abitur am Gymnasium in Bad Königshofen i. Gr. (1995)
 - Auslandsaufenthalt in Cape Town, SA (1992-1993)
- Studium der Informatik an der Universität Würzburg (1995-2001)
 - Nebenfach Mathematik (Vordiplom)
 - Nebenfach Psychologie (Diplom)
 - Diplom 2001.
 - Auslandsaufenthalt in Urbana-Champaign, IL (visiting scholar, 1999-2000)
- PhD Studium in Informatik an der University of Illinois at Urbana-Champaign, IL (2002-2004).
- Habilitation & Privatdozent an der Universität Würzburg (2010)

Wissenschaftliche Arbeit

- Systeme
 - Learning Classifier Systems
 - Schema lernende Systeme (ACS2 ..)
 - XCS / XCSF system (datamining, regression learning)
 - Artificial Neural Networks
 - Growing Neural Gas (temporal learning)
 - Hebbian learning
 - Rekurrente ANNs (z.B. RNNPBs)
 - Generative ANNs
 - Second-order ANNs
 - Bayesian Information Processing & Free Energy-based Inference
 - Cognitive models (behavior, attention, abstraction, conceptual thinking)
- Aktuelle wissenschaftliche Arbeiten:
 - Kognitionspsychologisch:
 - Virtual Realities und Augmented Realities
 - Multisensorische Integration
 - Antizipation
 - Verhalten, Aufmerksamkeit & Kognition
 - Modellierungen dazu
 - Künstliche Intelligenz:
 - Intelligente, adaptive Roboter (in Simulationsumgebungen) – Lernen, Kontrolle & Entscheidungsfindung.
 - Wahrnehmung des eigenen Körpers im Raum und Informationsintegration
 - Abstraktion ausgehend von sensomotorischen Erfahrungen
 - Super Mario Projekt

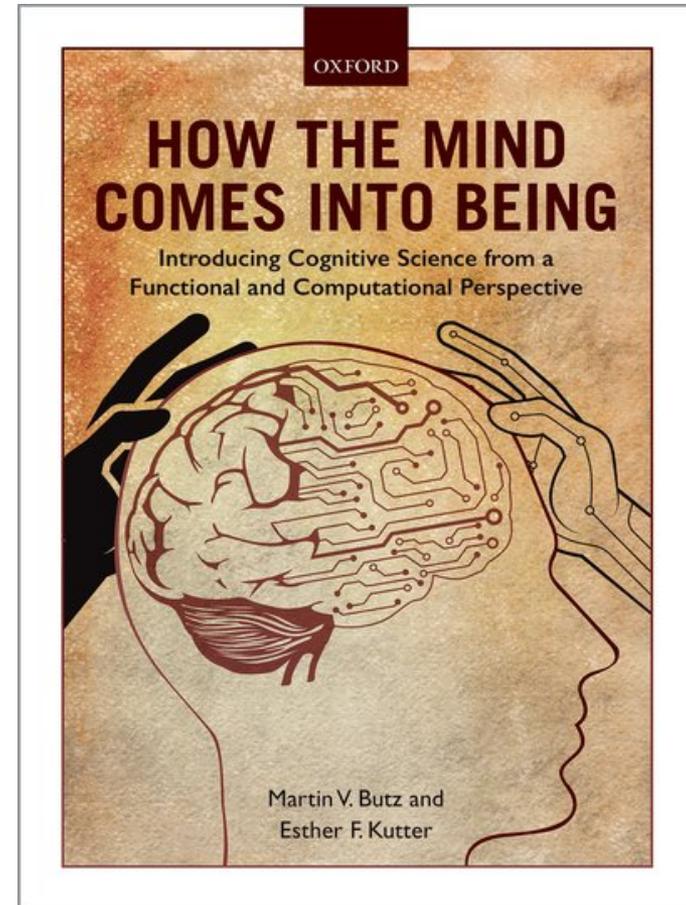


How the Mind Comes into Being

Introducing Cognitive Science from a Functional and Computational Perspective

- Provides an interdisciplinary perspective, helping the reader to develop an understanding of how the mind works that goes beyond disciplinary boundaries
- Adopts a computational approach, helping the reader to understand the mind on a functional level, in contrast to purely conceptual, verbalized levels
- Includes exercises and examples, helping the reader to consolidate the covered material and encouraging them to think 'outside of the box'

➤ Published in February 2017





Ein wenig über meinen Lehrstuhl: Lehrstuhl für Kognitive Modellierung

Mitarbeiter

- Dr. Anna Belardinelli
- Dr. Johannes Lohmann
- Dr. Sebastian Otte



- Stephan Ehrenfeld
- Dania Humaidan
- Danil Koryakin
- Fabian Schrodtt

Forschungsschwerpunkte

- Wie wir unseren eigenen Körper erkennen & kontrollieren
 - Multimodale Modelle
 - Probabilistisch & Neuronal
 - Maschinelles Lernen
 - Psychologische Experimente: Rubber Hand Illusion, Virtual Reality
- Wie wir andere Objekte und Entitäten erkennen und mit diesen interagieren
 - Antizipation und Zielorientiertheit
 - Auflösung von Redundanzen
 - Zielorientierte Suche nach Informationen
 - Psychologische Experimente: Eye-Tracking & Motion Tracking während der Objektinteraktion
- Wie wir Lernen
 - Körpererkennung und –kontrolle
 - Objekterkennung und –manipulation
 - Abstraktere Kognition



Und Sie ????

Informatik Bachelor

Medieninformatik Bachelor

Bioinformatik Bachelor

Medizininformatik Bachelor

Kognitionswissenschaftlicher

Lehramt-Studierende (NWT und andere)

Nebenfächler

Andere (welche ?)



Heute: Einführung

- Was wird hier gelehrt?
- Organisatorisches
- Literatur
- Allgemeine Hinweise
- Motivation und Einordnung
- Anwendungen der Informatik
- Erste Programmbeispiele



Lernen zu Programmieren und „algorithmisch“ zu Denken

- Anleitung zum Programmieren...
- Fokus liegt auf der Programmiersprache Java
- Was ist überhaupt ein Programm?
- Daten & Instruktionen werden zu einem Programm
- Schlechter versus guter Programmierstil
- Objektorientiertes Programmieren
- Programmdesign



Ablauf des Semesters (Änderungen vorbehalten)

Do 19.04. Intro, Eclipse, Java

Di 24.04. Algorithmenkonstruktion &
Programmsteuerung

Do 26.04. Datentypen

Do 03.05. Arrays I

Di. 08.05. Arrays II

Do 10.05. -- Himmelfahrt --

Di. 15.05 Unterprogramme I

Do 17.05 Unterprogramme II

Di 06.06. -- Pfingstpause --

Do 08.06. -- Pfingstpause --

Di 29.05. Unterprogramme III

Do 31.05. -- Fronleichnam --

Di 05.06. Objekt-Orientierte Programmierung I

Do 07.06. Objekt-Orientierte Programmierung II

Di 12.06. Objekt-Orientierte Programmierung III

Do 14.06. Objekt-Orientierte Programmierung IV

Di 19.06. Rekursion I

Do 21.06. Rekursion II

Di 26.06. Dynamische Strukturen I

Do 28.06. Dynamische Strukturen II

Di 03.07. Komplexität I

Do 05.07 Generische Programmierung

Di 10.07. Suchen & Sortieren I

Do 12.07. Suchen & Sortieren II

Di 17.07. Suchen & Sortieren III

Do 19.07. Wdh. & Ausblick

Di 24.07 Klausur

Di 09.10 Nachklausur



ORGANISATORISCHES

- Personen
- Organisation der Übungen & Übungsgruppen
- Abgabe von Übungen
- Benotung



Dozenten und Organisatoren

- Dozent der Vorlesung

Prof. **Martin Butz**, Kognitive Modellierung

Raum: Sand 14, C415

email: martin.butz@uni-tuebingen.de



- Übungsleiter

Dr. **Sebastian Otte**

email: sebastian.otte@uni-tuebingen.de

Raum: Sand 14, C420



TutorInnen

1	Yvette Bodry	yvette.bodry@student.uni-tuebingen.de
2	Cedric Breuning	cedric.breuning@student.uni-tuebingen.de
3	Dominik Heinrich	dominik.heinrich@student.uni-tuebingen.de
4	Katrin Kunz	katrin.kunz@student.uni-tuebingen.de
5	Maximilian Oberhoff	maximilian.oberhoff@student.uni-tuebingen.de
6	Gabriel Paradzik	gabriel.paradzik@student.uni-tuebingen.de
7	Marco Piechotta	marco.piechotta@student.uni-tuebingen.de
8	Andreas Sauter	andreas.sauter@student.uni-tuebingen.de
9	Jonas Schäfer	jonas.schaefer@student.uni-tuebingen.de
10	Jonathan Schmidt	jonathan.schmidt@student.uni-tuebingen.de
11	Jakob Stoll	jakob-samuel.stoll@student.uni-tuebingen.de
12	Ingeborg Wenger	ingeborg.wenger@student.uni-tuebingen.de
13	Stefan Zukunft	stefan.zukunft@student.uni-tuebingen.de
14	Luca Dreiling	luca.dreiling@student.uni-tuebingen.de
15	Marco Kinkel	marco.kinkel@student.uni-tuebingen.de
16	Jan-Niklas Boehm	jan-niklas.boehm@student.uni-tuebingen.de
17	Bjoern Bamberg	bjoern.bamberg@student.uni-tuebingen.de
18	Steffen Pohl	steffen.pohl@student.uni-tuebingen.de



Webseiten

Lehrstuhlseite – Arbeitsgruppe Kognitive Modellierung

<http://cm.inf.uni-tuebingen.de>

Abgabesystem – InfoMark (Anmeldung ab Dienstag!)

<https://info-cg.informatik.uni-tuebingen.de>



InfoMark – Materialien und Veranstaltungsunterlagen

- Alle Vorlesungsmaterialien und Informationen zur **Vorlesung** und den **Übungen** werden wir über den **InfoMark-Server** verwalten.
- **Materialien**
 - Vorlesungsfolien
 - Übungsblätter & Programmieraufgaben
 - Zusatzmaterialien / Handouts
 - Wiki
- Auch zum Abgeben der Übungsblätter & Programmieraufgaben (**immer elektronisch im InfoMark**)!
- Übungsblätter und auch **Programmierübungsaufgaben** werden immer im InfoMark **elektronisch veröffentlicht**.



Organisation und Prüfung

- SWS: 4 V + 2 Ü (9 ECTS)

Prüfungen:

- Klausur: Di. 24.07., 14-16 Uhr, N3, N6, N7
- Nachklausur: Do. 09.10., 14-16 Uhr, Neue Aula / Hörsäle 09 & 10

Übungen:

- min. 50% für Zulassung zur Prüfung

Note:

- Wenn Klausur bestanden, dann wird die Note berechnet aus:
 - $\text{Maximum}\{\%Klausur; 0.7 * \%Klausur + 0.3 * \%Papierübungen\}$
- Sonst: 5 (nicht bestanden – zweite Chance in Nachklausur)



Grundsätzliches:

- Die Übungen sind integraler Bestandteil der Veranstaltung.
- Die Themen der Vorlesung sollen dort anhand **praktischer Beispiele** eingeübt und vertieft werden.
- In dieser Veranstaltung werden die **Grundlagen** für viele weitere Details für das spätere Informatikstudium gelegt.
- Alle Inhalte der Veranstaltung (Vorlesung, Übungen, Programmierübungen) sind **klausurrelevant**.



Zusammenarbeit und Abgabe...

- Abgabe der Übungsblätter zu zweit.
- **Abgaben dürfen nicht identisch / klar einfach kopiert sein.**
- **Plagiarismus** wird geahndet!
 - Wir werden **alle** Abgaben auf Ähnlichkeit prüfen.
 - Alle beteiligten Personen bekommen **NULL Punkte**.
 - **Wiederholer** werden von der Veranstaltung ausgeschlossen.
- **Logins** dürfen nur von der **zugehörigen Person** genutzt werden.
 - **Passwörter werden nicht veröffentlicht.**



Übungen – Ziele und Methodologie

- Vertiefen und Einüben des **Vorlesungsstoffes**
- Erlangen **praktischer Fertigkeiten** im Umgang mit der Programmiersprache und Anwenden der **Konzepte** aus der Vorlesung:
 - Analyse und Strukturierung von Problemstellungen
 - Entwurf von Algorithmen
 - Entwicklung und Verstehen von Algorithmen
 - Erstellung von Programmen
 - Programmieren in **Java** Verstehen und Anwenden der elementaren Programmier-kenntnisse in **Java**

Zeit	MO	DI	MI	DO	FR
08 - 10			TÜ		TÜ
10 - 12				TÜ	TÜ
12 - 14	TÜ	TÜ	Tutoren- treffen	TÜ	TÜ
14 - 16	TÜ	VL		VL	
16 - 18	TÜ	TÜ			
18 - 20	TÜ				



V **DI 14 – 16 (N07)**
DO 14 – 16 (N07)



TÜ – Übung



HD – Helpdesk (Computerpools Morgenstelle –
 Donnerstag 26.4. erster Termin)



Organisation und Durchführung

- Alle **Materialien zu den Übungen** werden über die Plattform **InfoMark** bereit gestellt.
- Tafelübungen werden in Tutorien durchgeführt.
- Es werden ca. 18 Tafelübungen mit je maximal 25 Teilnehmer/innen angeboten.
- **Einschreiben** für die Übungen:
 - Bitte melden Sie sich **nächsten DIENSTAG** im **InfoMark** an
(siehe dann auch infos auf Lehrstuhlwebseiten)!
 - Dann wählen Sie Ihre Übungstermine:
<https://info-cg.informatik.uni-tuebingen.de>



Bearbeitung der Übungsblätter

- Übungsblätter mit Programmieraufgaben werden über das **InfoMark** abgegeben.
 - Bearbeitungszeit ist typischerweise 1 Woche.
- Die Abgabe von schriftlichen Aufgaben und Programmen erfolgt über das **InfoMark**
 - **Bitte halten Sie die vorgeschriebenen Abgabemodalitäten ein!**



Übungsblätter

- Bearbeitung
 - Die Blätter sollten alleine oder zu zweit bearbeitet werden.
 - **Besprechung** der Aufgaben und der Musterlösung wird in den Tutorien stattfinden.
- Tafelübungen
 - Es wird eine **aktive Teilnahme an den Übungen und in den Tutorien** erwartet.
 - Es stellt immer einzelne Personen aus dem Tutorium die Lösungen vor.
 - Musterlösungen werden vom Tutor vorgestellt.



Abgabe der Lösung

- Die **Abgabe einer Lösung** bedeutet,
 - dass man diese verstanden hat und
 - in der Lage ist, diese auch in der Tafelübung zu präsentieren.
- **Punkte** für eine Aufgabe werden dann erreicht (und somit angerechnet), wenn
 - die Aufgabe erfolgreich bearbeitet und **rechtzeitig** abgegeben wurde,
 - die Abgabe im richtigen Format erfolgt,
 - man sich im Tutorium **aktiv beteiligt** und seine Lösung ggf. präsentiert.



Helpdesk

- Freie Übungsform (vgl. Mathe-Rechenzentrum) mit Unterstützung
- Unterstützung durch **2 Tutoren & KommilitonInnen**
 - bei Installationsproblemen
 - bei Fragen zu Übungen
 - bei der Abgabe in InfoMark
 - beim Programmieren / bei Programmieraufgaben
 - Lassen Sie sich helfen!
 - Kommen Sie um zu helfen!
 - Helfen Sie sich gegenseitig!
- Termine & Ort: **Immer Montags 15-19 & Donnerstags 16-20**
 - **Beginn**: Donnerstag, 26.04.2018
 - **Ort**: Rechnerpools an der Morgenstelle
(C-Bau, Raum N03 & C-Bau, Raum H33)

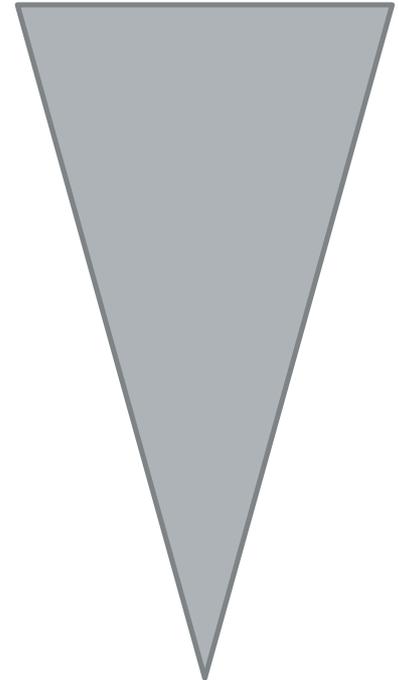


Bei Problemen / Fragen / Sorgen / Vorschlägen / Kritik etc...

Bitte an folgende Reihenfolge halten:

1. Forum
2. Tutor / Tutorin Ihrer Übungsgruppe
3. Übungsleiter: Sebastian Otte
4. Dozent: Martin Butz

400+X



(... sonst könnte die Reaktionszeit unnötig lang werden)



LITERATUR





Informatik allgemein, Programmieren, Algorithmen

Allgemeine Einführungen in Konzepte der Informatik

- [1] *Als Übersichts- und Nachschlagewerk mit Java Beispielen:*
H.-P. Gumm, M. Sommer (2010) Einführung in die Informatik, 10. Aufl., Oldenburg Verlag, München
- [2] *Zugängliche Einführung (allerdings mit Modula 2 Beispielen):*
H.-J. Appelrath, I. Megaard (2000) Skriptum Informatik – eine konventionelle Einführung, 5. Aufl., Teubner Verlag, Stuttgart.
- [3] *Sehr gut zugängliche Einführung in die Theoretische Informatik:*
U. Schöning (2008). Ideen der Informatik: Grundlegende Modelle und Konzepte der Theoretischen Informatik. Oldenbourg Wissenschaftsverlag, München.
- [4] Etwas schwerer zugänglich:
H.-J. Appelrath, D. Boles, V. Claus, I. Wegener (2002)
Starthilfe Informatik. 2. Aufl., Teubner Verlag, Stuttgart.
- [5] Relativ schwer zugänglich und somit eher später als Fachliteratur:
M. Broy (2012) Informatik, 3. überarbeitete Aufl. Springer, Heidelberg

Übersichten und Kompendien

- [1] V. Claus, A. Schwill (2003) Schüler-Duden Informatik. Duden Verlag, Mannheim
- [2] P. Bartke, H.-D. Burkhard, R. Buttke, W. Coy, L. Engelmann, A. Kalenberg, H.-J. Laabs (2006). Duden Informatik - Gymnasiale Oberstufe: Schülerbuch mit CD-ROM: Duden. Sekundarstufe II. Duden Schulbuch.
- [3] Internet (!)



Programmieren mit Java

- [1] D.J. Barnes & M. Kölling (2013/2009). Java lernen mit BlueJ: Eine Einführung in die objektorientierte Programmierung. 5./4. Aufl., Pearson Studium.
- [2] D.J. Eck (1996-2007) Introduction to Programming using Java, 5th edition. SoHo Books, Creative Commons Attribution-Share, San Francisco (vollständiger Abdruck und weitere Materialien verfügbar auf der Webseite <http://math.hws.edu/javanotes>; das Buch ist verfügbar unter <http://math.hws.edu/eck/cs124/downloads/javanotes5.pdf>)
- [3] *W. Küchlin ist Professor hier im Fachbereich:*
W. Küchlin, A. Weber (2002) Einführung in die Informatik – Objektorientiert mit Java, 2., überarbeitete u. erweiterte Auflage. Springer, Berlin
- [4] J. Bloch (2008). Effective Java: A Programming Language Guide (Java Series). 2nd Ed., Addison-Wesley Longman, Amsterdam.
- [5] B. Eckel (2006) Thinking in Java: The definitive introduction to object-oriented programming in the language of the world wide web, 4th edition. Prentice Hall, Upper Saddle River, NJ



Programmieren mit Java

- [6] *Großes Buch mit vielen Beispielen:*
D. Bell & M. Parr (2010) Java for students, 6th edition.
Prentice-Hall, Upper-Saddle River, NJ
- [7] *Nicht so technisch mit guten Zusammenfassungen:*
P. Pepper (2007) Programmieren lernen – Eine grundlegende
Einführung mit Java, 3. Auflage. Springer Verlag, Berlin
- [8] *Einführung etwas langatmig – sehr umfangreich:*
D. Ratz, J. Scheffler, D. Seese, J. Wiesenberger (2011) Grundkurs
Programmieren in Java, 6., überarbeitete Auflage.
Carl Hanser Verlag, München
- [9] *Lustige Einführung die aber auf verwirrend sein kann*
K. Sierra, B. Bates (2007) Java von Kopf bis Fuß, 1. Auflage, 2.
korr. Nachdruck. O'Reilly, Köln
- [10] *Nachschlagewerk für Erfahrenere*
R. Liguori, P. Liguori (2008) Java kurz & gut, 1. Auflage, O'Reilly, Köln



Algorithmen und Datenstrukturen

- [1] T.H. Cormen, C.E. Leiserson, R.L. Rivest (2001) Introduction to algorithms. MIT Press, Cambridge
- [2] T.H. Cormen, C.E. Leiserson, R.L. Rivest, C. Stein (2009) Algorithmen – Eine Einführung, 3. überarbeitete u. erweiterte Auflage. Oldenbourg Verlag, München (Übersetzung von [1])
- [3] T. Ottmann, P. Widmayer (1996) Algorithmen und Datenstrukturen, 3. Aufl. Spektrum Akad. Verlag, Heidelberg
- [4] R. Sedgewick, K. Wayne (2011) Algorithms, 4th Edition. Addison-Wesley, Reading
- [5] N. Wirth (1998) Algorithmen und Datenstrukturen, 3. Aufl. B.G. Teubner, Stuttgart



MOTIVATION

- Warum Informatik ?
- Inhalte
- Der Erste Algorithmus

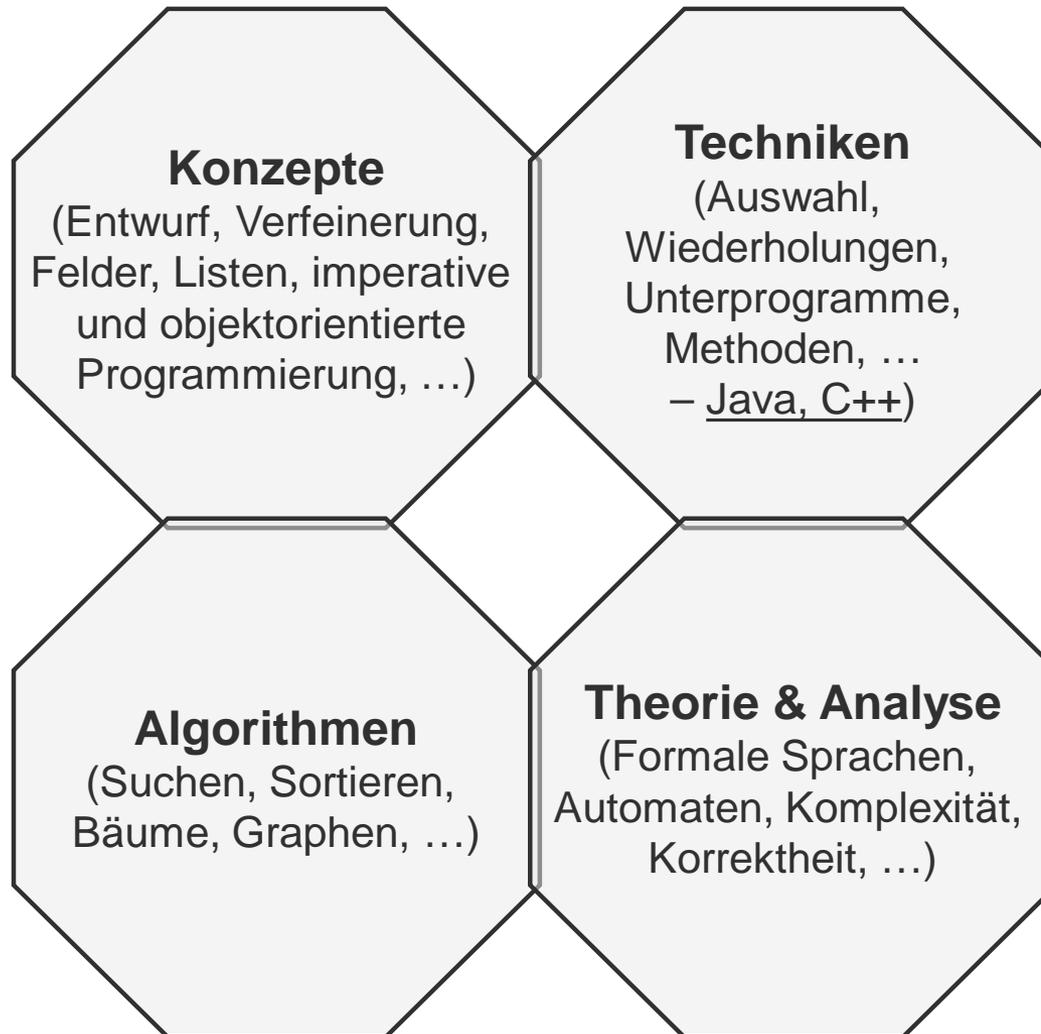


Warum Informatik?





Inhalte der Veranstaltung





DER ERSTE ALGORITHMUS



Vom Problem zum Algorithmus

- Ziel des Algorithmus:

Beantworte die Frage:

Was ist das Durchschnittsalter aller Studierenden im Raum?

- Wie gehen wir vor?



Mathematische Lösung

- Summiere das Alter auf
- Teile durch die Anzahl der Personen

$$\bar{a} = \frac{1}{N} \sum_N a_i$$

N = Anzahl der Studierenden

a_i = Alter von Person i



Durchschnittsalter – Ansatz 1

- Mathematische Formel

$$\bar{a} = \frac{1}{N} \sum_N a_i$$

N = Anzahl der Studierenden

a_i = Alter von Person i

- Überführen in iterative Update-Regel

$$\bar{a}_1 = a_1, \quad \bar{a}_2 = \frac{\bar{a}_1 + a_2}{2}, \quad \bar{a}_3 = \frac{2 \cdot \bar{a}_2 + a_3}{3}$$

$$\bar{a}_{n+1} = \frac{n \cdot \bar{a}_n + a_{n+1}}{n+1}$$



Durchschnittsalter – Ansatz 2

- Mathematische Formel

$$\bar{a} = \frac{1}{N} \sum_N a_i$$

- Zwei Teilaufgaben:

- bestimme $A = \sum_N a_i$

- berechne

$$\bar{a} = \frac{A}{N}$$



Vom Problem zum Algorithmus

- Ziel des Algorithmus war:

Beantworte die Frage:

Was ist das Durchschnittsalter aller Studierenden im Raum?

1. Ansatz:

Iterative Durchschnittsbildung (Update in jedem Schritt)

2. Ansatz:

Gesamalter (also eine iterative Summenbildung mit Update in jedem Schritt)
Quotientenbildung

3. Ansatz:

Stochastisch (approximativ)

Auswahl einer zufälligen Stichprobe – mittleres Alter der Stichprobe als Approximation



Vom Problem zum Algorithmus

Gesuchte Antwort auf die Frage:

Wie viele Studierende sind im Raum?



Vom Problem zum Algorithmus

Gesuchte Antwort auf die Frage:

Wie viele Studierende sind im Raum?

1. Lösung: **sequentiell**

- einer nach dem anderen

$$N = \sum_N 1$$

... langsam



Vom Problem zum Algorithmus

Gesuchte Antwort auf die Frage:

Wie viele Studierende sind im Raum?

1. Lösung: **sequentiell**

- einer nach dem anderen

$$N = \sum_N 1$$

2. Lösung: (teilweise) **parallel**

- Einer nach dem anderen, aber alle Reihen parallel
- Zusammenfügen der Zwischenergebnisse
- **Teile und herrsche (Divide-and-Conquer)**



Mittelwert von drei Zahlen

```
import java.util.Scanner; // Utility Klasse zum Einlesen von Werten

public class Average{
    public static void main(String[] args) {
        double avg;
        double v1, v2, v3;

        Scanner scanner = new Scanner(System.in);
        System.out.print("Wert_1: ");
        v1 = scanner.nextDouble();
        System.out.print("Wert_2: ");
        v2 = scanner.nextDouble();
        System.out.print("Wert_3: ");
        v3 = scanner.nextDouble();
        scanner.close();

        avg = (v1 + v2 + v3 ) / 3;

        System.out.println("Mittelwert von 3 Zahlen: " + avg);

    } // end main
} // end class Average
```



Ausblick

- Registrieren bei **InfoMark** am Dienstag

<https://info-cg.informatik.uni-tuebingen.de>

- **Erster Helpdesk**: Donnerstag (26.04) 16-20 Uhr in den Rechnerpools an der Morgenstelle
(C-Bau, Raum N03 & C-Bau, Raum H33)
- **Erste Übungsgruppen ab Donnerstag 03.05.2018**
 - Zuordnung zu den Gruppen wird über InfoMark bekanntgegeben (nächste Woche)
- Nächste VLs:
 - Algorithmenbegriff
 - Von der Idee zum Algorithmus und zum Java Programm