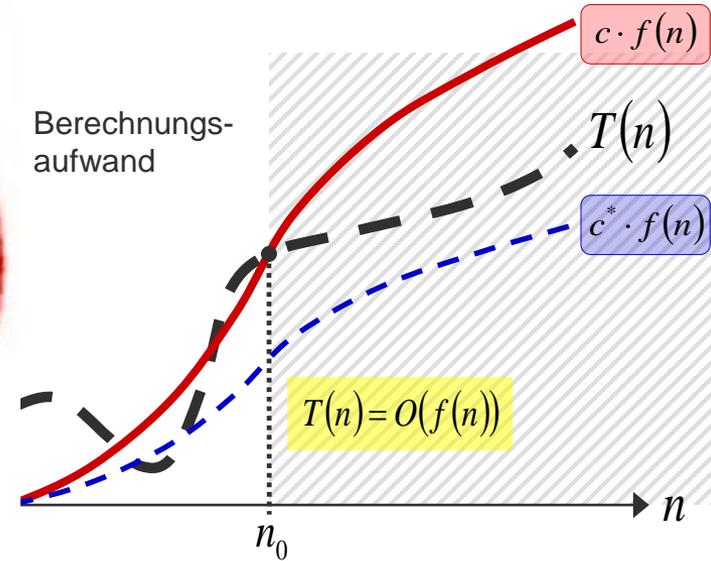


$\langle \text{Zahl} \rangle ::= \langle \text{Ziffer} \rangle \mid \langle \text{Ziffer} \rangle \langle \text{Op} \rangle \langle \text{Ziffer} \rangle$   
 $\langle \text{Ziffer} \rangle ::= "0" \mid "1"$   
 $\langle \text{Op} \rangle ::= "*" \mid "/" \mid "+" \mid "-"$



## Informatik II – SoSe 2016



## Curriculum Vitae – Hendrik Lensch

1993 – 1999 Universität Erlangen-Nürnberg (Dipl.-Inf.)



1999 – 2003 Max-Planck-Institut für Informatik (Dr.-Ing.)



2004 – 2006 Visiting Assistant Professor  
Stanford University



2006 – 2008 Nachwuchsgruppenleiter



2009 – 2011 Professor für Medieninformatik  
Universität Ulm



seit Okt. 2011 Professor für Computergrafik  
Universität Tübingen





---

## Und Sie ????

Informatik Bachelor

Medieninformatik Bachelor

Bioinformatik Bachelor

Medizininformatik Bachelor

Kognitionswissenschaftlicher

Lehramt-Studierende

Nebenfächler

Andere (welche ?)



---

# Diese Woche: Einführung

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



## Was bedeutet eigentlich Lehren?

„ **Lehren**‘ heißt natürlich **nicht**  
‘Vermittlung von Tatsachenwissen‘ –  
dies blieb ein relativ nebensächlicher Aspekt – ,  
sondern bedeutete vielmehr :  
**Anleitung zum Denken und Lesen** sowie  
**Ermutigung zum Reflektieren.**“

(P.B. Medawar. Ratschläge für einen jungen Wissenschaftler. piper, 1984)



---

# Ziel der Vorlesung: Lernen zu Programmieren

- Anleitung zum Programmieren...
- Fokus liegt auf den Programmiersprachen Java & C++
- Was ist überhaupt ein Programm?
- Daten & Instruktionen werden zu einem Programm.
- Schlechte versus guter Programmierstil
- Objektorientiertes programmieren
- Programmdesign



---

# Ablauf des Semesters (Änderungen vorbehalten)

Di 12.04. Intro, Eclipse, Java 101  
Do 14.04. Algorithmenkonstruktion  
Di 19.04. Datentypen  
Do 21.04. Programmsteuerung  
Di 26.04. Arrays I  
Do 28.04. Arrays II  
Di 03.05. Unterprogramme I  
Do 05.05 -- Himmelfahrt --  
Di 10.05. Unterprogramme II  
Do 12.05. Objekt-Orientierte Programmierung I  
Di 17.05. -- Pfingstpause --  
Do 19.05. -- Pfingstpause --  
Di 24.05. Objekt-Orientierte Programmierung II  
Do 26.05. -- Fronleichnam --  
Di 31.05. Objekt-Orientierte Programmierung III  
Do 02.06. Objekt-Orientierte Programmierung IV

Do 09.06. Dynamische Strukturen I  
Di 14.06. Dynamische Strukturen II  
Do 16.06. Komplexität I  
Di 21.06. Komplexität II  
Do 23.06. Suchen & Sortieren I  
Di 28.06. Suchen & Sortieren II  
Do 30.06. Suchen & Sortieren III  
Di 05.07. C++ - Einführung  
Do 07.07. Arrays & Pointer  
Di 12.07. Funktionen  
Do 14.07. OOP  
Di 19.07. Templates & STL  
Do 21.07. Wdh. & Ausblick



---

# ORGANISATORISCHES

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



---

## Dozenten und Organisatoren



- Dozent der Vorlesung

Prof. **Hendrik Lensch**, Computergrafik

Raum: Sand 14, C404

email: `hendrik.lensch@uni-tuebingen.de`

- Übungsleiter und Organisatoren

Dipl.-Inf. **Fabian Groh**

email: `fabian.groh@uni-tuebingen.de`

MSc Inf. **Sebastian Herholz**

email: `sebastian.herholz@uni-tuebingen.de`

MSc Math. **Patrick Wieschollek**

email: `patrick.wieschollek@uni-tuebingen.de`

Raum: Sand 14, C404, C430, C429, C409

---



---

## TutorInnen

- |                        |  |
|------------------------|--|
| 1. Fabian Bauer,       | fabian.bauer@student.uni-tuebingen.de      |
| 2. Verena Biener,      | verena.biener@student.uni-tuebingen.de     |
| 3. Alexander Blöck,    | alexander.bloeck@student.uni-tuebingen.de  |
| 4. Lucas Brandenburg,  | lucas.brandenburg@student.uni-tuebingen.de |
| 5. Dominik Heinrich,   | dominik.heinrich@student.uni-tuebingen.de  |
| 6. Finn Ickler,        | finn.ickler@student.uni-tuebingen.de       |
| 7. Annkatrin Jung,     | annkatrin.jung@student.uni-tuebingen.de    |
| 8. Vanessa Kirchner,   | vanessa.kirchner@student.uni-tuebingen.de  |
| 9. Daniel Merling,     | daniel.merling@student.uni-tuebingen.de    |
| 10. Jan-Oliver Nick,   | jan-oliver.nick@student.uni-tuebingen.de   |
| 11. Sanzenbacher Paul, | paul.sanzenbacher@student.uni-tuebingen.de |
| 12. Marco Piechotta,   | marco.piechotta@student.uni-tuebingen.de   |
| 13. Weibke Ringels,    | wiebke.ringels@student.uni-tuebingen.de    |
| 14. Ture Sayer,        | ture.sayer@student.uni-tuebingen.de        |
| 15. Mihael Simonic,    | mihael.simonic@student.uni-tuebingen.de    |
| 16. Benjamin Steinert, | benjamin.steinert@student.uni-tuebingen.de |



---

## Webseiten

Veranstaltungsseite – Arbeitsgruppe Computergrafik

<http://www.graphics.uni-tuebingen.de>

Abgabesystem – InfoMark (Anmeldung bis Donnerstag!)

<https://info2.informatik.uni-tuebingen.de>

**Demo**



---

## InfoMark – Materialien und Veranstaltungsunterlagen

- Alle Vorlesungsmaterialien und Informationen zur **Vorlesung** und den **Übungen** werden wir über den **InfoMark-Server** verwalten.
- **Materialien**
  - Vorlesungsfolien
  - Übungsblätter
  - Zusatzmaterialien / Handouts
  - Wiki
  - Forumsbereich: <https://forum-ps.informatik.uni-tuebingen.de/>
- Auch zum Abgeben der Übungsblätter **(immer elektronisch im InfoMark)!**
- **Übungsblätter** und auch **Programmierübungsaufgaben** werden immer im InfoMark **elektronisch veröffentlicht.**



---

# Organisation und Prüfung

- SWS: 4 V + 4 Ü (9 ECTS)

## Prüfungen:

- Klausur: Di. 26.07., 14-16 Uhr, N5, N6, N7
- Nachklausur: Do. 06.10., 14-16 Uhr, N6

## Übungen:

- min. 50% für Zulassung zur Prüfung
- min. 70% für Bonus von 0.3 Noten
- min. 80% für Bonus von 0.6 Noten



Voraussetzung:  
Bestehen der Klausur



---

## 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) sind **klausurrelevant**.



---

## Zusammenarbeit und Abgabe...

- Abgabe der Übungsblätter **individuell (jeder einzeln)**
- **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 einer 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 & C++**
  - Verstehen und Anwenden der elementaren Programmierkenntnisse in **Java & C++**

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

 V **DI 14 – 16 (N6)**  
**DO 14 – 16 (N6)**

 T $\ddot{U}$  - Tafelübung

 HD - Helpdesk



---

# Erstsemesterbetreuung

- spezielle Übungsslots
  - Annkatrin Jung: Mo 18-20, N9
  - Lucas Brandenburg: Mo 14-16, A6G07
  - es wird jeweils ein weiterer Termin in diesen Übungen vereinbart



---

## Organisation und Durchführung

- Alle **Materialien zu den Übungen** werden über die Plattform **InfoMark** bereit gestellt.
- Die „Tafelübungen“ werden in Tutorien durchgeführt.
- Es werden ca. 16 Tafelübungen mit je maximal 30 Teilnehmer/innen angeboten.
- **Einschreiben** für die Übungen:
  - Bitte melden Sie sich bis Donnerstag (bis 11:59) im **InfoMark** an!
  - Dann wählen Sie Ihre Übungstermine:

<https://info2.informatik.uni-tuebingen.de>



---

# Bearbeitung der Übungsblätter

- **Übungsblätter mit Programmieraufgaben** werden über das **InfoMark** ausgegeben und abgegeben.
  - Bearbeitungszeit ist typischerweise 1 Woche.
- Die Abgabe von schriftlichen Aufgaben und Programmen erfolgt über das **InfoMark**
  - Dort werden die Programme **automatisch analysiert und getestet**
  - Zwingend: **striktes Einhalten der vorgeschriebenen Abgabemodalitäten**



---

# Übungsblätter

- Bearbeitung
  - Die Blätter werden grundsätzlich in **individuell** bearbeitet
  - **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 eine Übungsgruppe aus dem Tutorium seine Lösung 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 / Rechenzentrum

- Freie Übungsform (vgl. Mathe-Rechenzentrum)
- Unterstützung durch Tutoren
  - bei Installationsproblemen
  - bei Fragen zu Übungen
  - bei der Abgabe in InfoMark
  - beim Programmieren
- Später: weitere Übungsaufgaben
- Termine:
  - **Mo. 12-16 N4**
  - **Mi. 14-16 N2** ← **startet morgen**

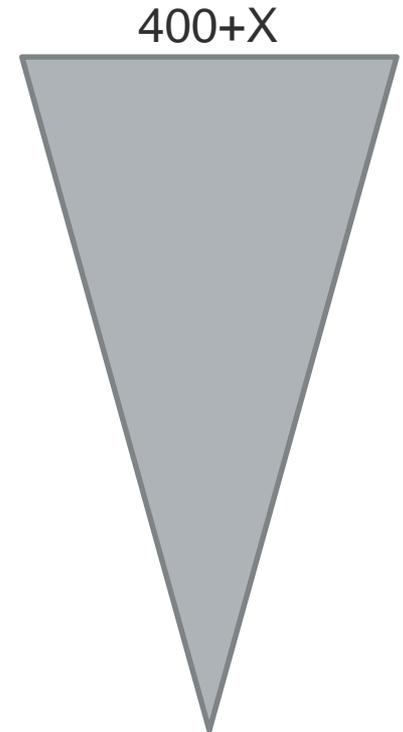


---

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

Bitte an folgende Reihenfolge halten:

1. Tutor / Tutorin Ihrer Übungsgruppe
2. Forum
3. Übungsleiter: Fabian, Patrick, Sebastian
4. Dozent: Hendrik Lensch



(... 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 ?
- Informatik & Mathematik
- Informatik ist systematische Informationsverarbeitung
- Informatik strukturiert (Gedanken, Systeme, die Welt)
- Informatik ist (fast) überall



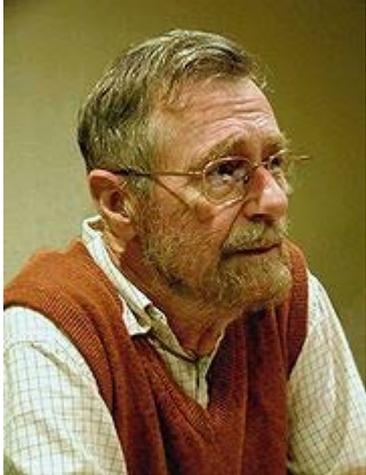
# Warum Informatik?

---





## Warum Informatik?



*„Was mich an der Informatik immer so angezogen hat, war, dass das, was konzeptuell schön zusammen passte, in der Praxis sofort hervorragend nutzbar war.“*

Edsger Wybe Dijkstra (1930 – 2002)  
(Niederländischer Informatiker und Erfinder der formalen Algorithmen)

*„...weil ich zu faul war zum Rechnen“*

Konrad Zuse (1910 – 1995)

auf die Frage, warum er die Rechenmaschine erfunden hat



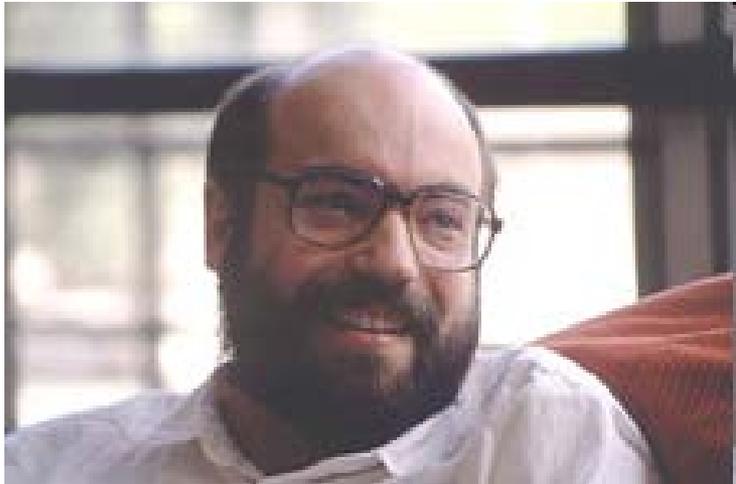


## Visionen und Entwicklungen der Informatik ...

*„Die viereckigen Kisten, die wir heute mit uns ‘rumtragen, werden im Jahr 2010 verschwunden sein. Computer befinden sich dann in Brillengläsern oder Kontaktlinsen, sind Teil von Hemden oder Sakkos.“ (1999)*

Ray Kurzweil (1948\*)

(Computerwissenschaftler, Softwareentwickler, Futurologe)



*„The most profound technologies are those that disappear. They weave themselves into fabric of everyday life until they are indistinguishable from it.“ (1991)*

Mark Weiser (1952-1999)

(amerikanischer Computerwissenschaftler @ XEROX PARC, Wegbereiter des *Ubiquitous Computing*; der Computer im 21. Jahrhundert)



---

# Informatik und Informationsverarbeitung

## Was ist Informatik?

### **Informatik** ...

(engl. *Computer Science*; auch *Computer Engineering*)

- ... ist **mehr als Programmieren!**
- ... befasst sich mit **allen Aspekten der Darstellung, Speicherung, Übertragung und maschinellen Verarbeitung** von Information



# Informatik und Informationsverarbeitung

## Definitionen

- a) Informatik [lat.]: **Wissenschaft**, die sich mit der grundsätzlichen  
Verfahrensweise der **Informationsverarbeitung** und allgemeinen Methoden der  
**Anwendung** solcher Verfahren (...) befasst
- b) Informationsverarbeitung: Aufnahme und Weiterleitung von **Informationen** an  
eine zentrale Speicher- und Auswerteeinrichtung; beim **Lebewesen** z.B. im  
Gehirn, im **technischen Bereich** mittels der Datenverarbeitung (Kybernetik)

(dtv Brockhaus Lexikon. München, 1986)



---

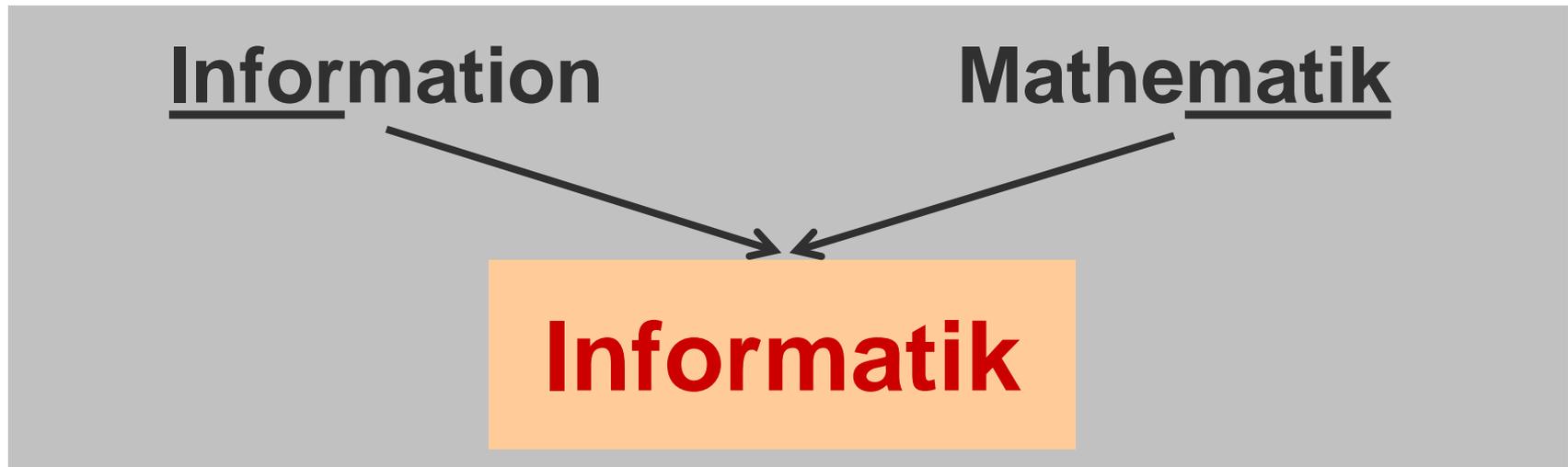
# Informatik ist Informationsverarbeitung

- Entwicklung von Formalismen und Modellen zur maschinellen Informationsverarbeitung, z.B.
  - Algorithmen und Datenstrukturen
  - Untersuchung abstrakter Maschinen- und Berechnungsmodelle (Berechenbarkeit, Dynamikverhalten, Komplexität, Automaten, ...) Programmier- und Maschinensprachen
- Informatik ist eine **Grundlagen- und Formal-Wissenschaft**
- Eine **Wissenschaft** verwendet allgemein anerkannte Techniken und Methoden traditioneller Disziplinen, beispielsweise  
Mathematik, Physik (Naturwissenschaften), Ingenieurwissenschaften, ...  
... im **Gegensatz** zur ausschließlichen Kunstfertigkeit oder Handwerklichkeit!



# Informatik als Informationsverarbeitende Mathematik

- Logik, Zahlentheorie, ...
- Informations- und Codierungstheorie



- Abgrenzung zur **Mathematik**:
- **Informatik** betrachtet **Strukturen** (oder Objekte) und deren **Konstruktion** aus elementaren Grundstrukturen  
(→ Kompositions- / Zerlegungsprinzip);
- Lösungen müssen konstruierbar (d.h. **programmier-** und **berechenbar**) sein!

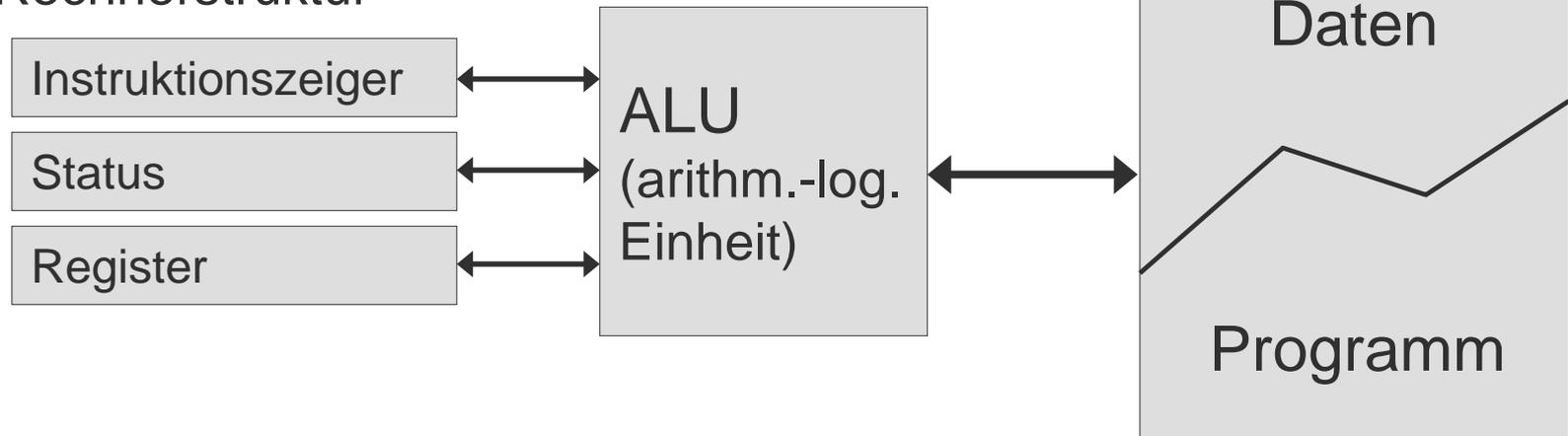


- Programmstruktur (Ausschnitt aus Java-Programm)

```
sum = 0;
for (int i = 0; i < number; i++) {
    sum = sum + points[i]
}

average = sum / number;
for (int i = 0; i < number; i++) {
    diff[i] = points[i] - average;
}
```

- Rechnerstruktur





---

# Konstruktion von Informatiksystemen

- **Modellierungen der realen Anwendungsumgebung**

- Identifikation relevanter Konzepte
- Festlegung eines geeigneten Abstraktionsniveaus
- Auswahl und Konstruktion von Darstellungsformen

- **Strukturierung des Problems**

- **Vertikal:** Abstraktionsebenen (d.h. Schichtenbildung)
- **Horizontal:** Systemkomponenten (d.h. Zerlegung / Modularisierung)

- **Konstruktion der Lösung**

- Schrittweise Verfeinerung des Lösungskonzepts
  - Am Anfang steht die Idee wie ein Aufgabe gelöst werden kann.
- Programmierung der Komponenten
- Integration und Systemrealisierung



# Informatik ist Überall

- Ingenieurwesen
- Wirtschaft
- Datenverarbeitung
- Biologie
- Umwelttechnik
- Robotik
- Mobilität (Automotive)
- Virtual Reality
- Entertainment
- ...

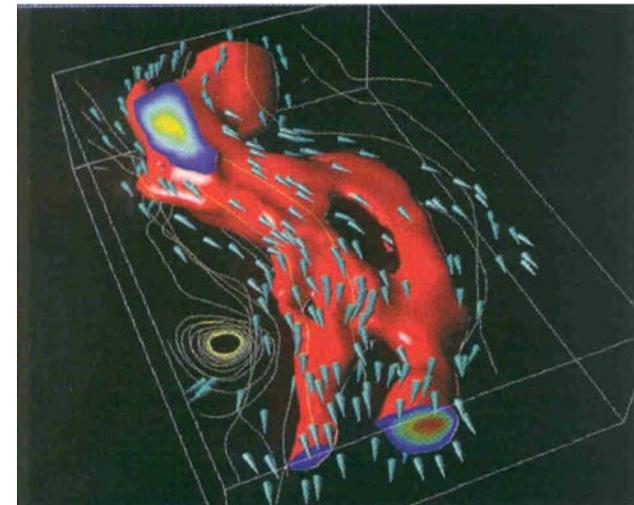
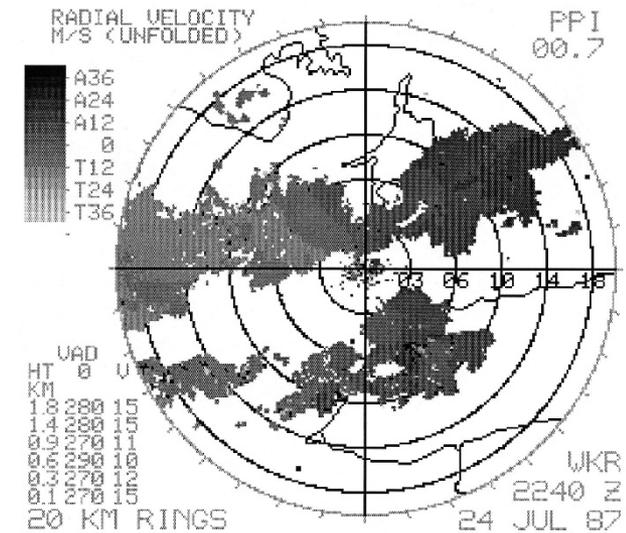




# Simulation technischer, physikalischer und biologischer Vorgänge

## Durchführung komplexer Berechnungen in verschiedenen Bereichen

- Strömungsmechanik: Konstruktion von Flugzeugen und Fahrzeugteilen
- Statik: Simulationen im Hoch- und Tiefbau
- Demografie: Planungen im Bereich öffentlicher Verwaltungen
- Biologie: Bioinformatik (z.B. Populationsentwicklungen, genetische Prozesse)
- Meteorologie: Wetterbeobachtungen und -prognosen



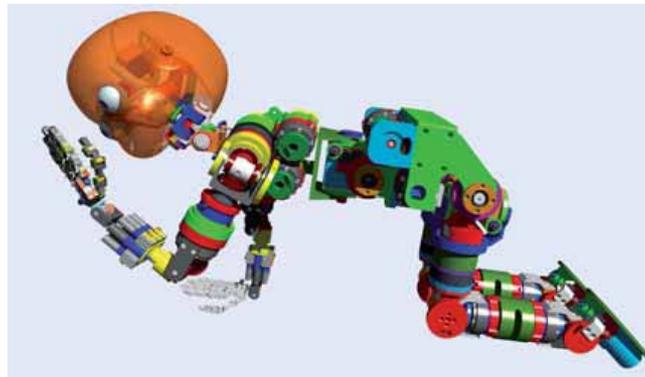


## Steuerung komplexer Systeme

- Roboter, Navigationssysteme, Autopiloten, Herzschrittmacher, ...
- Charakteristisch: Direkte Kopplung von **Software-Systemen mit physikalischen** Vorgängen (senso-motorische Kopplung)
- Sensoren      Erfassen von Mess- und Umgebungsdaten
- Aktuatoren    Umsetzung der physikalischen Aktion



Nasa Marsrobooter „Spirit“



Europäischer Forschungsrobooter iCub



Roboterhand RAPHaEL



# Sicherheit und Komfort – Fahrzeugassistenten

## Eingebettete Systeme (embedded systems)

- Prozesssteuerung durch enge Kopplung von Hard- u. Software
- Realisierung wichtiger Funktionen durch Software
- E/E-Komponenten (elektronisch / elektrische Komponenten) im modernen KfZ bieten hohen Funktionsumfang (10 Mio. Codezeilen)
- Vernetzung von E/E-Komponenten erhöht Komplexität
- Bis zu 90% der Fahrzeuginnovationen durch elektronisch beeinflusste Neuheiten



Bordcomputer



Navigation



Abstand-  
regelung



Motor



Außen-  
spiegel



Airbag



---

## Informatik: Gesellschaftswissenschaft

- Alle Bereiche menschlichen (Zusammen–)Lebens vertrauen heute auf computergestützte Technologie
  - Arbeitswelt, Beziehungen, Finanzen, Katastrophenschutz, Kommunikation, Kriminalistik, Kunst, Medizin, Nachrichtenwesen, Politik, Unterhaltung, Verkehr, Wissenschaft, ...
  - Google, Facebook, LinkedIn, Online Rollenspiele, etc.
- ⇒ Informatiker verändern die Gesellschaft....
- ... Sie können (gesellschaftliche) Probleme lösen —
  - ... aber auch ganz neue schaffen !



---

# Die Berufsvereinigung der Informatik in Deutschland

Gesellschaft  
für Informatik



<http://www.gi-ev.de/themen/was-ist-informatik.html>



# Abschlüsse



Informatik (BSc, MSc, BSc Edu.)



Medieninformatik (BSc, MSc)



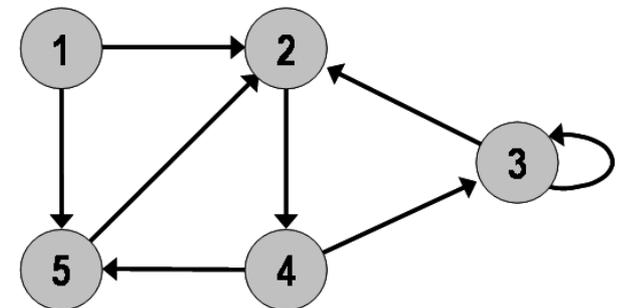
Kognitionswissenschaften (BSc, MSc)



Bioinformatik (BSc, MSc)

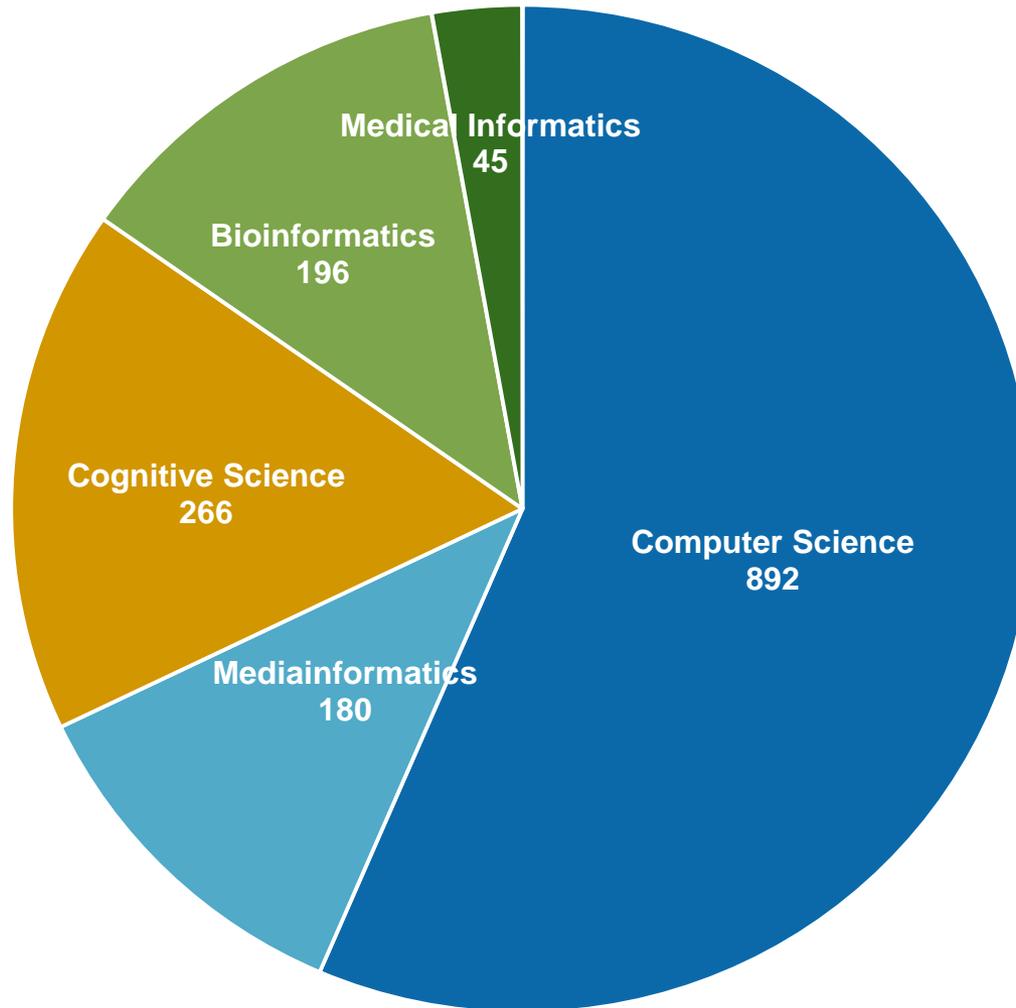


Medizininformatik (BSc, MSc)





# Studierendenzahlen



2015



# Informatik in Tübingen

## *Bioinformatics & Medical Informatics*

- Daniel Huson
- Oliver Kohlbacher
- *Andrei Lupas*
- Kay Nieselt

## *Vision & Cognition*

- *Michael Black*
- Martin Butz
- *Martin Giese*
- Volker Franz
- Enkelejda Kasneci
- Alexandra Kirsch
- Hendrik Lensch
- *Bettina Rolke*
- Andreas Schilling
- Felix Wichmann
- Andreas Zell

## *Software & Systems Engineering*

- Oliver Bringmann
- Torsten Grust
- Wolfgang Kuchlin
- Michael Menth
- Klaus Ostermann
- Wolfgang Rosenstiel
- Thomas Walter

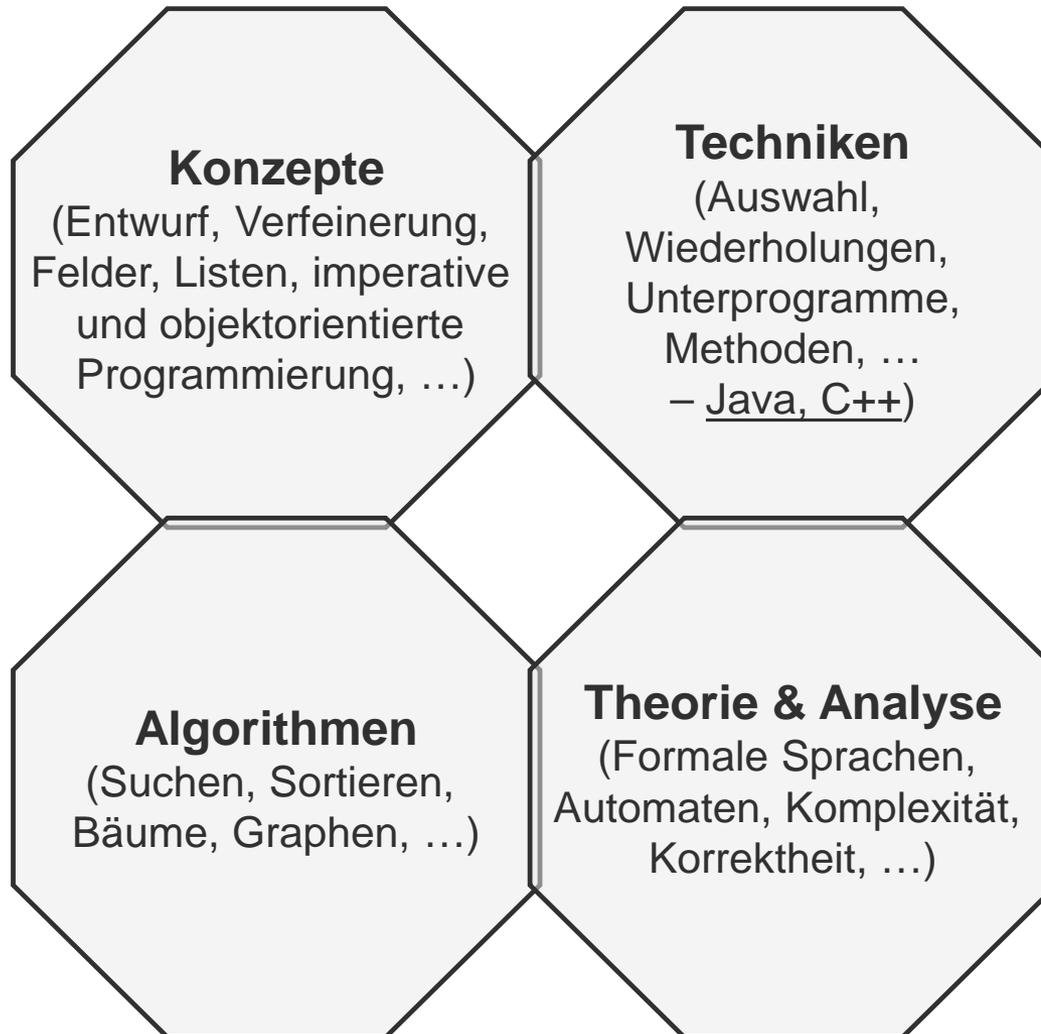
## *Theory*

- Britta Dorn, Peter Hauck, Michael Kaufmann, Klaus-Jörn Lange, Ulrike v. Luxburg, Peter Schroeder-Heister



# Inhalte der Veranstaltung

---





---

# DER ERSTE ALGORITHMUS



---

## Vom Problem zum Algorithmus

- Ziel des Algorithmus:

Beantworte die Frage:

**Was ist das Durchschnittsalter aller Studierenden im Raum?**

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

$N$  = Anzahl der Studierenden

$a_i$  = Alter von Person

- 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

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

- 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

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
```



---

# Übungsblatt 0

- Installieren
  - Java
  - Eclipse
- Registrieren im InfoMark
- Abgabe im InfoMark



---

## Ankündigung

# Vorlesung - Logik (Prof. Küchlin) beginnt erst nächste Woche



---

## Ausblick

- Nicht vergessen: **Registrieren beim InfoMark**

<https://info2.informatik.uni-tuebingen.de>

- Erster Helpdesk: morgen: Mittwoch 14-16 im Raum N2
- Erster Übungen ab Montag 18.04.2016
  - Zuordnung wird über InfoMark bekanntgegeben
- Nächste VLs:
  - Algorithmenbegriff
  - Von der Idee zum Algorithmus und zum Java Programm



---

# Ursprung der Folien

- Folieninhalte stammen aus der Vorlesung “Praktische Informatik I” an der Universität Ulm entnommen.

© H. Lensch, H. Neumann, E. Ohlebusch, M. Reichert,  
A. Lanz, R. Pryss, Universität Ulm, 2011