

Herzlich
Willkommen!



Worum geht's?

um

Eine Einführung in die Informatik

Was passiert heute? |

Grundsätzliches

Organisatorisches

Inhaltliches



Die Tutoren ("Bremser")

- sind drei alte Hasen und zwölf engagierte Drittsemester.
- betreuen die Tutorien und die Office Hours und das Forum und und und.
- stehen bei Fragen jederzeit zur Verfügung.
- sind einfach super.

Matthias Adams
Tom Baumeister
Matthias Cosler
Christina Eimer
Lennart Haas
Anna Hartung
Lukas Schäfer
Frederik Schmitt
Sanny Schmitt
Johannes Schulz
Julian Siber

Der Assistent

- ist Felix Freiberger

Der Dozent

- ist Holger Hermanns

Der Joker

- ist Gereon Fox



Gregory Stock
Benedict Strube
Anna Wilhelm
Nathalie Zeller



Das dCMS

- ist ein Kursverwaltungssystem.
- ist unser und Ihr Portal zur Bereitstellung und zum Abrufen aller wichtigen Informationen.
- wurde von Markus Bauer und Marc Jose entwickelt, auf der Basis eines Systems von Sigurd Schneider.
- ist inzwischen stabil.
- ist hier:

<https://dcms.cs.uni-saarland.de>

Anmeldung vor Donnerstag, 23:59 Uhr

dCMS

Hauptseite

Informationen ▾

Anmeldung

Persönliche Statusseite

Login ▾



Programmierung 1

Holger Hermanns

News

Aktuell gibt es keine Neuigkeiten

Inhalt der Vorlesung

Die Vorlesung führt in grundlegende Konzepte der Programmierung anhand der funktionalen Programmiersprache SML ein. Genauer geht es dabei um Datenstrukturen und Algorithmen, Programmierkonstrukte und Programmiertechniken, sowie die Struktur und die Realisierung von Programmiersprachen.

Publikum

Diese Vorlesung richtet sich an Studienanfänger der Informatik, Bioinformatik, Medieninformatik, und Computerlinguistik. Auch Wirtschaftsinformatiker und einige Studierende mit Nebenfach Informatik hören diese Vorlesung. Es werden keine besonderen Kenntnisse vorausgesetzt, ausser der Freude am mathematischen Denken. Insbesondere werden keinerlei Programmierkenntnisse erwartet.

Die Tutorien ("Übungen")

- sind wöchentlich am Dienstag im Zeitraum 8-14 Uhr.
- beginnen am 8. November (leider leider).
- dienen zum Üben, Vertiefen und Hinterfragen des Stoffs,
und zum Testen Ihres Leistungsstandes.
- brauchen keine Laptops.

Ihnen wird ein spezifisches Tutorium
und ein spezifischer Tutor zugeteilt werden.

Auswahl der Zeitslots vor Donnerstag, 23:59 Uhr

"NOT OK" bedeutet: nachweisbare universitäre Veranstaltung

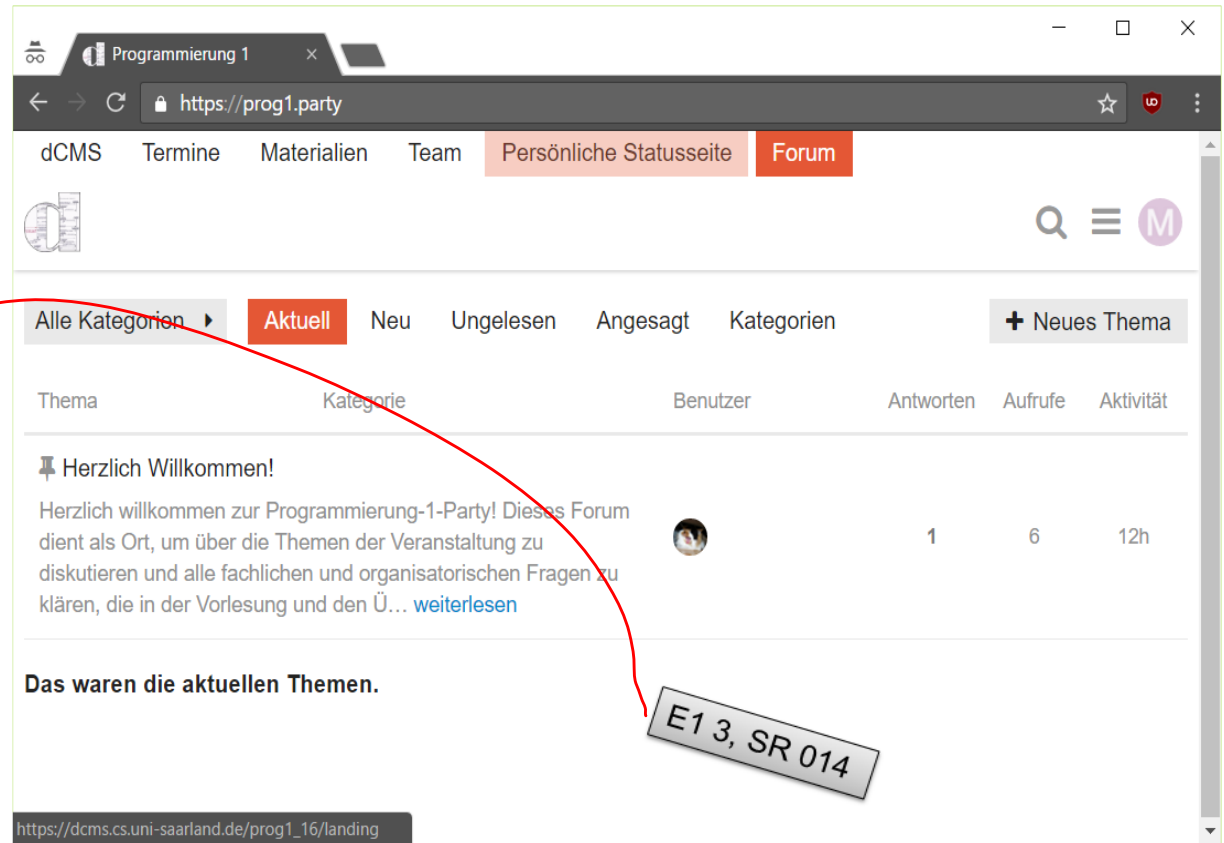
Die Office Hours

- werden täglich angeboten, außer Dienstags

Montag	10-12 Uhr	E1 1, SR 206
Mittwoch	12-14 Uhr	E1 3, SR 014
Donnerstag	14-16 Uhr	E1 3, SR 014
Freitag	14-16 Uhr	E1 3, SR 014

- dort treffen Sie Tutoren und Kommilitonen und können mit diesen inhaltliche Fragen klären

Dasselbe gibt es auch 24/7 im Cyberspace!



Das Forum

Die Übungsblätter

- werden wöchentlich online im dCMS bereitgestellt,
in aller Regel vor Donnerstag morgen.
- sind Ihre Basis, um selbsttätig Kompetenz im Stoff zu erwerben.

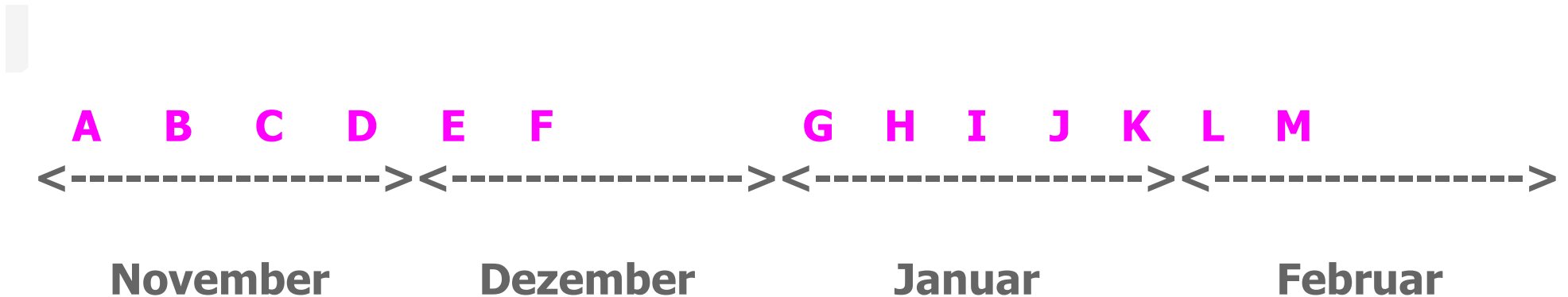
Insgesamt gibt es 13 Übungsblätter,

durchbuchstabiert von A bis M,

in der Regel über den Stoff der 2 letzten Vorlesungen.

Die Minitests

- gibt es wöchentlich, zu Beginn der Tutorien, ab dem 8. November.
- testen Ihren Leistungsstand in Bezug auf den aktuellen Stoff.
- dauern jeweils eine Viertelstunde.
- werden bepunktet: 0-15 Punkte.



Die Klausuren

- gibt es zweimal zwei:

Mittelklausur:	Samstag, 7. Januar	ab 10:00 Uhr
Nachmittelklausur:	Samstag, 18. Februar	ab 10:00 Uhr

Endklausur:	Freitag, 3. März	ab 10:00 Uhr
Nachendklausur:	Dienstag, 11. April	ab 10:00 Uhr

- dauern 90 Minuten.

**Das Modul 'Programmierung 1' wird bestanden,
indem beide Klausuren
oder die jeweiligen Nachklausuren bestanden werden.**

Die Zulassungsbedingungen zu den Klausurterminen

Sie sind zur **Mittelklausur** zugelassen.

Um zur **Endklausur** zugelassen zu sein, müssen Sie
in der Mittelklausur mindestens 25% der erreichbaren Punkte erlangt haben,
sowie in den 13 Minitests
nach Streichung der beiden schlechtesten Summanden
mindestens eine Summe von
88 Punkten (von maximal 165 Punkten) erlangt haben.

Sie sind zu dem **Nachklausuren** zugelassen,
sofern Sie zur Endklausur zugelassen sind.

Achtung: Dies gilt insbesondere auch für die Nachmittelklausur.

Die Note

- ergibt sich aus den beiden Klausurergebnissen.

- ist eine der folgenden Zahlen:

5,0

- erniedrigt sich (sofern möglich) um eine Stufe,
sofern in den 13 Minitests

4,0

nach Streichung der beiden

3,7

schlechtesten Summanden

3,3

mindestens eine Summe von

3,0

150 Punkten (von maximal 165)

2,7

erlangt wurde.

2,3

2,0

1,7

1,3

1,0



Die Vorlesungen

- finden an folgenden Tagen statt:

Dienstags 14-16 Uhr
 außer am 1. November

Donnerstags 10-12 Uhr

Der kNobelpreis

ist eine super Sache!

- Je eine kNobel-Aufgabe erscheint parallel zum Übungsblatt.
- Das sind knifflige Knobeleien.
- Aufgaben stammen vom Team oder von externen Knoblern.
- Aufgaben werden bewertet.
- High-Score-Liste auf der kNobel-Seite im dCMS.
- Manche Aufgabe enthalten eine Überraschung.
- Zu jeder Aufgabe gibt es ein unglaubliches Nobel-Tutorium
meistens Donnerstags, 16:00

Der beste Studierende erhält den phänomenalen kNobelpreis, der nach der letzten Aufgabe in der Vorlesung überreicht wird.



Ihre Ziele:

Der Zweck dieses Kurses ist, daß Sie lernen

- die Grundkonzepte der Informatik
formal zu beschreiben und anzuwenden,
- Programme zu erklären und zu analysieren,
- Strukturierte Problemlösungen zu entwerfen,
- die Korrektheit von Programmen zu beweisen,
- das Gelernte auf neue Probleme anzuwenden.

Es geht nicht darum, Wissen anzuhäufen,

sondern darum, Kompetenz

im Beschreiben, Erklären, Analysieren,

Argumentieren, Beweisen, und Anwenden des Wissen
zu erlangen.

Dies ist nicht immer einfach.

Es erfordert Üben, Üben, Üben.

Was ist das?



Was ist das?

Der
Schwamm



Was ist das?

Der
Dschungel

(Version
Saarland)

Wer sind Sie?

Saar ?

Weiteste Anreise?

Mathe inner Schule?

Vorkurs?

Informatik? Eingebette Systeme? WiInf? Alles Übrige?

Erstsemester?

if ?

Wiener-Prozess zweiter Ordnung mit der cadlag-Eigenschaft ?

Adjunktion ?

Abduktion ?

wohlfundiert ?

Oh ?

Programmierung Eine Einführung in die Informatik

Vorlesung¹

Holger Hermanns

basierend auf dem Buch
von

Gert Smolka

Programmierung Eine Einführung in die Informatik

Vorlesung¹

Holger Hermanns

basierend auf dem Buch
von

Gert Smolka

<http://www.ps.uni-sb.de/prog-buch/leseprobe.pdf>



Kapitel 1

Schnellkurs

alles Wörter

unser erstes Programm:

Bezeichner werden an Werte gebunden

val x = 4 * 7 + 3

val y = x * (x - 29)

Konstanten bestimmter, fester Wert

Schlüsselwörter strukturieren das Programm

Operatoren verknüpfen Werte

$x = 31$
 $y = 62$

Wir betrachten
Programme als
implementierte Mathematik

Wir benutzen eine Programmiersprache
als Labor um mit
implementierter Mathematik

Experimente machen zu können

Unsere

Programmiersprache: Standard ML andere folgen später im Studium

Was ist das?

"value"
 $val \langle \text{Bezeichner} \rangle = \langle \text{Ausdruck} \rangle$

Eine Deklaration

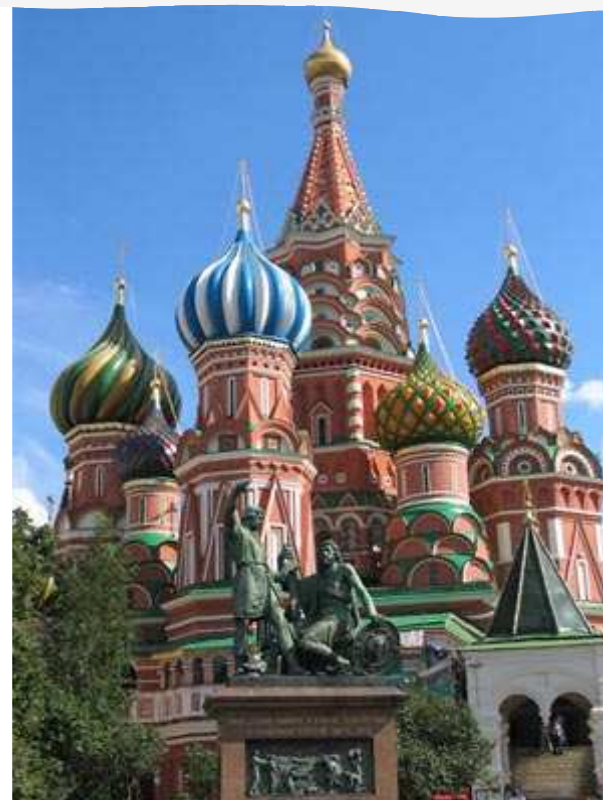
```
val x = 4 * 7 + 3  
val y = x * (x-29)
```

Ein **Programm** ist eine Folge von Deklarationen

Bezeichner
Konstanten
Schlüsselwörter
Operatoren

Was ist das?
Moscow ML

Moskau



```
ML
Enter 'quit();' to quit.
-
val x = 3000 * 8000 ;
> val x = 24000000 : int
- fun quadrt x = if x = 0 then x else quadrat (x - 1) ;
! Toplevel input:
! fun quadrt x = if x = 0 then x else quadrat (x - 1) ;
!
! Syntax error.
- fun quadrt x = if x = 0 then x else quadrat (x - 1) ;
! Toplevel input:
! fun quadrt x = if x = 0 then x else quadrat (x - 1) ;
!
! Unbound value identifier: quadrat
- fun quadrat x = if x = 0 then x else quadrat (x - 1) ;
> val quadrat = fn : int -> int
quadrat 25 ;
> val it = 0 : int
it + 1;
> val it = 1 : int
~it
;
> val it = ~1 : int
(it * it - it) div it ;
> val it = ~2 : int
```

andere Optionen: SML-NJ
Alice

Ein Interpreter der Programmiersprache ML

oder: unser Labor

Feinheiten

Klammersparregeln → Anhang

Das Semikolon: ;

signalisiert dem Interpreter
"Los jetzt!"

Die Tilde: ~

Negationsoperator für Zahlen: ~ 7

Subtraktionsoperator: $x - 29$

Der Ergebnisbezeichner: *it*

Das können Sie sich sparen:

`val it =`

Die Fehlermeldungen:

vorerst weitgehend unverständlich

! Unbound value identifier: val

Prozeduren

implementieren Funktionen,
die durch Gleichungen
beschrieben werden

$$\text{quadrat}(x) = x \cdot x$$

Prozedurdeklaration

`fun quadrat (x:int) = x*x`

Name

Argumentvariable

Typ

Rumpf

Argumentmuster

Prozedur

Funktion

Algorithmus

Eine Funktion kann durch verschiedene Prozeduren berechnet werden

fun quadrat (x:int) = x*x

oder ✓

fun quadrat' (y:int) = y*(y-1)+y

Eine Prozedur fixiert eine Berechnungsvorschrift

Algorithmus

Abu Abdullah Muhammad bin Musa al-Khwarizm

(ca. 770 - 840)

- + schrieb das erste Buch über das Rechnen mit Dezimalzahlen
- + Namensgeber für das Wort Algorithmus
- + führte die Ziffer 0 in das arabische Zahlensystem (und damit in alle modernen Zahlensysteme) ein.
- + der Begriff Algebra wurde aus dem Titel seines Buches Al-Dschabr abgeleitet.

Das Buch wurde mehrfach ins Lateinische übersetzt und hatte großen Einfluss auf die arabische (und dann auch europäische) Entwicklung der Mathematik.

