

# Vorlesungsverzeichnis

Master of Science - Geoinformation und  
Visualisierung  
Prüfungsversion Wintersemester 2010/11

Wintersemester 2020/21

# Inhaltsverzeichnis






<b>Abkürzungsverzeichnis</b>	<b>4</b>
<b>Geoinformation I</b>	<b>5</b>
<b>Geovisualisierung I</b>	<b>5</b>
<b>Vertiefung Geo- und Umweltwissenschaften</b>	<b>5</b>
<b>Informatik für Naturwissenschaftler und Prozesse I</b>	<b>5</b>
83809 U - Grundlagen der Programmierung (Rechnerübung)	5
83810 VU - Grundlagen der Programmierung	5
<b>Geoinformation II</b>	<b>7</b>
<b>Geovisualisierung II</b>	<b>7</b>
<b>Raumanalysen mit GIS</b>	<b>7</b>
<b>GIS-Adaption</b>	<b>7</b>
<b>Informatik für Naturwissenschaftler und Prozesse II</b>	<b>7</b>
<b>Geoinformation III</b>	<b>7</b>
<b>GIS-Projektmanagement</b>	<b>7</b>
<b>Vertiefung Informatik</b>	<b>7</b>
83801 U - Theoretische Informatik I: Modellierungskonzepte - Automaten und formale Sprachen	8
83802 TU - Theoretische Informatik I: Modellierungskonzepte - Automaten und formale Sprachen	8
83803 V - Theoretische Informatik I: Modellierungskonzepte - Automaten und formale Sprachen	9
83812 VU - Grundlagen der Informationsverarbeitung	9
83820 S - Boolesche Funktionen und binäre Schaltungen	10
83833 VU - Chipentwurf	11
83837 VU - Grundlagen der Betriebssysteme und Rechnernetze	13
83839 VU - Inferenz-Methoden	14
83849 VU - Multimedia-Technologie	14
83851 PJ - Software Engineering I	15
83852 VU - Software Engineering I	16
83854 VU - Compiler und Programmtransformation	17
83905 VU - Maschinelles Lernen & Intelligente Datenanalyse II	18
83906 S - Parallel Computing - Praktikum	18
83909 VU - Pervasive Computing	18
83916 VP - Declarative Modeling	19
83920 PR - Declarative Problem Solving and Optimization	20
83921 VU - Declarative Problem Solving and Optimization	20
83922 VU - Didaktik der Informatik II	21
83935 VU - Leistungsanalyse: Messen, Modellieren, Simulation	21
83936 PJ - Leistungsanalyse: Messen, Modellieren, Simulation	21

<b>Studienprojekt</b>	<b>21</b>
<b>Fakultativ</b>	<b>22</b>
<b>Fakultative Lehrveranstaltungen.....</b>	<b>22</b>
<b>Glossar</b>	<b>23</b>

# Abkürzungsverzeichnis

## Veranstaltungsarten

AG	Arbeitsgruppe
B	Blockveranstaltung
BL	Blockseminar
DF	diverse Formen
EX	Exkursion
FP	Forschungspraktikum
FS	Forschungsseminar
FU	Fortgeschrittenenübung
GK	Grundkurs
HS	Hauptseminar
KL	Kolloquium
KU	Kurs
LK	Lektürekurs
OS	Oberseminar
P	Projektseminar
PJ	Projekt
PR	Praktikum
PU	Praktische Übung
RE	Repetitorium
RV	Ringvorlesung
S	Seminar
S1	Seminar/Praktikum
S2	Seminar/Projekt
S3	Schulpraktische Studien
S4	Schulpraktische Übungen
SK	Seminar/Kolloquium
SU	Seminar/Übung
TU	Tutorium
U	Übung
V	Vorlesung
VE	Vorlesung/Exkursion
VP	Vorlesung/Praktikum
VS	Vorlesung/Seminar
VU	Vorlesung/Übung
WS	Workshop

N.N.	Noch keine Angaben
n.V.	Nach Vereinbarung
LP	Leistungspunkte
SWS	Semesterwochenstunden
	Belegung über PULS
	Prüfungsleistung
	Prüfungsnebenleistung
	Studienleistung
	sonstige Leistungserfassung

## Veranstaltungsrhythmen

wöch.	wöchentlich
14t.	14-täglich
Einzel	Einzeltermin
Block	Block
BlockSa	Block (inkl. Sa)
BlockSaSo	Block (inkl. Sa, So)

## Andere

# Vorlesungsverzeichnis

## Geoinformation I

Für dieses Modul werden aktuell keine Lehrveranstaltungen angeboten

## Geovisualisierung I

Für dieses Modul werden aktuell keine Lehrveranstaltungen angeboten

## Vertiefung Geo- und Umweltwissenschaften

Für dieses Modul werden aktuell keine Lehrveranstaltungen angeboten

## Informatik für Naturwissenschaftler und Prozesse I

### 83809 U - Grundlagen der Programmierung (Rechnerübung)

Gruppe	Art	Tag	Zeit	Rhythmus	Veranstaltungsort	1.Termin	Lehrkraft
1	U	Mi	08:00 - 10:00	wöch.	Online.Veransta	04.11.2020	Dr. Henning Bordihn
2	U	Mi	10:00 - 12:00	wöch.	Online.Veransta	04.11.2020	Dr. Henning Bordihn
Bitte beachten Sie die geänderte Zeit von zuvor 12-14 Uhr auf 10-12 Uhr!							
3	U	Mi	14:00 - 16:00	wöch.	Online.Veransta	04.11.2020	Dr. Henning Bordihn
4	U	Mi	16:00 - 18:00	wöch.	Online.Veransta	04.11.2020	Dr. Henning Bordihn
5	U	Fr	10:00 - 12:00	wöch.	Online.Veransta	06.11.2020	Dr. Henning Bordihn
6	U	Fr	12:00 - 14:00	wöch.	Online.Veransta	06.11.2020	Dr. Henning Bordihn

### Kommentar

Die in der Vorlesung und den Übungen behandelten Konzepte werden im Computerlabor exemplarisch realisiert. Dabei wird der Umgang mit dem Betriebssystem UNIX/Linux und der Programmiersprache Python erlernt.

### Leistungsnachweis

In der Rechnerübung zum Modul Grundlagen der Programmierung gibt es eine Prüfungsnebenleistung (PNL) zum Abschluss des Moduls (Verbuchung der Leistungspunkte). Die Zulassung zur Prüfung erfolgt unabhängig von dieser PNL. Die PNL wird durch eine Testatleistung im Computerlabor (45-60 Minuten während einer der Rechnerübungen) erbracht und gilt als bestanden, wenn mindestens 50% der geforderten Testatleistung erzielt wurden.

### 83810 VU - Grundlagen der Programmierung

Gruppe	Art	Tag	Zeit	Rhythmus	Veranstaltungsort	1.Termin	Lehrkraft
Alle	VO	Mi	14:00 - 15:00	Einzel	3.06.H03	28.10.2020	Dr. Henning Bordihn
Es ist eine der Vorbesprechungen wahrzunehmen! Es gibt 110 Plätze pro Vorbesprechung.							
Alle	VO	Mi	16:00 - 17:00	Einzel	3.06.H03	28.10.2020	Dr. Henning Bordihn
Es ist eine der Vorbesprechungen wahrzunehmen! Es gibt 110 Plätze pro Vorbesprechung.							
Alle	VO	Do	09:00 - 10:00	Einzel	3.06.H03	29.10.2020	Dr. Henning Bordihn
Es ist eine der Vorbesprechungen wahrzunehmen! Es gibt 110 Plätze pro Vorbesprechung.							
Alle	VO	Do	11:00 - 12:00	Einzel	3.06.H03	29.10.2020	Dr. Henning Bordihn
Es ist eine der Vorbesprechungen wahrzunehmen! Es gibt 110 Plätze pro Vorbesprechung.							
Alle	V	N.N.	N.N.	wöch.	N.N.	N.N.	Dr. Henning Bordihn
1	U	Do	10:00 - 12:00	wöch.	Online.Veransta	05.11.2020	Dr. Henning Bordihn
2	U	Do	12:00 - 14:00	wöch.	Online.Veransta	05.11.2020	Dr. Henning Bordihn
3	U	Do	14:00 - 16:00	wöch.	Online.Veransta	05.11.2020	Dr. Henning Bordihn
4	U	Do	16:00 - 18:00	wöch.	Online.Veransta	05.11.2020	Dr. Henning Bordihn
5	U	Fr	14:00 - 16:00	wöch.	Online.Veransta	06.11.2020	Dr. Henning Bordihn
6	U	Fr	16:00 - 18:00	wöch.	Online.Veransta	06.11.2020	Dr. Henning Bordihn

#### Kommentar

- **Grundbegriffe der Informatik**
  - Hardware, Software, Programm, Prozess, Betriebssystem, Netzwerk
- **Einführung in UNIX/Linux**
  - Prozesskonzept
  - Dateisystem, Rechteverwaltung
  - Shell, Systemvariablen, Kommandosubstitution, Ein- und Ausgabeströme
  - Einige UNIX-Werkzeuge
- **Mathematische Grundlagen**
  - Relationen, Funktionen, Operationen
  - mathematische Aussagen und Beweise
- **Vom Problem zum Algorithmus**
  - Algorithmenbegriff
  - Modellbildung/Abstraktion und Verfeinerung
  - Graphen und ihre Repräsentation
  - Pseudocode, Variablen, Kontrollstrukturen, grundlegende Datentypen
  - Brute-Force-Algorithmen
  - Komplexität und andere Gütekriterien
  - Grenzen des algorithmisch Machbaren
- **Vom Algorithmus zum Programm**
  - Imperative Programmierung
  - Prozedurale Programmierung, Funktionen, Parameter, Aufruf-Stack
  - Rekursion
  - Objektorientierte Programmierung
  - Funktionale Programmierung
  - Programmierung mit Python
  - Ausblick auf logische Programmierung
- **Vom Programm zum Prozess**
  - Interpreter *versus* Compiler
  - Assembler
- **Algorithmen**
  - einfache numerische Algorithmen
  - Algorithmen auf Graphen, vor allem Breiten- und Tiefensuche
  - u.v.m.

#### Leistungsnachweis

Am Schluß der Vorlesung wird eine benotete Klausur (120 Minuten, ohne Unterlagen) angeboten.

Prüfungsnebenleistung (PNL): Für die Zulassung zur Prüfung müssen Übungsaufgaben (Moodle) selbstständig bearbeitet werden. Die PNL gilt als bestanden, wenn mindestens 60% der Aufgaben erfolgreich bearbeitet worden sind.

#### Bemerkung

Die Einzeltermine am 28.10. und 29.10. sind Auftaktveranstaltungen, die als Präsenzveranstaltungen im Hörsaal H03+H04 durchgeführt werden.

Jede(r) soll **genau einen** der vier Termine wahrnehmen. Wichtige inhaltliche und organisatorische Hinweise werden gegeben. Anschließend können Fragen gestellt werden.

Die Teilnahme wird dringend empfohlen.

**Bitte registrieren Sie sich für einen der Einzeltermine (Vorbesprechung) in der Terminumfrage unter diesem Link:**

<https://terminplaner4.dfn.de/LQ1IVJS4KCALqcKL>



#### **Kurzkommentar**

**Einschreibeschlüssel Moodle: GdP20**

**Bitte beachten Sie, dass die Zuordnung zu den Übungsgruppen und Gruppen der Rechnerübungen ausschließlich über eine Registrierung in Moodle erfolgt. PULS ist hier nicht maßgeblich.**

#### **Geoinformation II**

Für dieses Modul werden aktuell keine Lehrveranstaltungen angeboten

#### **Geovisualisierung II**

Für dieses Modul werden aktuell keine Lehrveranstaltungen angeboten

#### **Raumanalysen mit GIS**

Für dieses Modul werden aktuell keine Lehrveranstaltungen angeboten

#### **GIS-Adaption**

Für dieses Modul werden aktuell keine Lehrveranstaltungen angeboten

#### **Informatik für Naturwissenschaftler und Prozesse II**

Für dieses Modul werden aktuell keine Lehrveranstaltungen angeboten

#### **Geoinformation III**

Für dieses Modul werden aktuell keine Lehrveranstaltungen angeboten

#### **GIS-Projektmanagement**

Für dieses Modul werden aktuell keine Lehrveranstaltungen angeboten

#### **Vertiefung Informatik**

83801 U - Theoretische Informatik I: Modellierungskonzepte - Automaten und formale Sprachen							
Gruppe	Art	Tag	Zeit	Rhythmus	Veranstaltungsort	1.Termin	Lehrkraft
1	U	Di	16:00 - 18:00	wöch.	3.04.0.04	03.11.2020	Prof. Dr. Christoph Kreitz, Tom Kranz
Bitte beachten Sie dass die Übungen grundsätzlich online stattfinden. Sofern die Teilnehmerzahl kleiner oder gleich der Raumkapazität ist wird ggf in den Präsenzbetrieb gewechselt.							
2	U	Do	12:00 - 14:00	wöch.	3.04.0.04	05.11.2020	Prof. Dr. Christoph Kreitz, Tom Kranz
Bitte beachten Sie dass die Übungen grundsätzlich online stattfinden. Sofern die Teilnehmerzahl kleiner oder gleich der Raumkapazität ist wird ggf in den Präsenzbetrieb gewechselt.							
2	U	Do	12:00 - 14:00	wöch.	Online.Veranstalt	05.11.2020	Prof. Dr. Christoph Kreitz, Tom Kranz
Bitte beachten Sie dass die Übungen grundsätzlich online stattfinden. Sofern die Teilnehmerzahl kleiner oder gleich der Raumkapazität ist wird ggf in den Präsenzbetrieb gewechselt.							
3	U	Di	16:00 - 18:00	wöch.	3.04.1.02	03.11.2020	Dr. rer. nat. Sebastian Böhne
Für Lehramtsstudierende. Bitte beachten Sie dass die Übungen grundsätzlich online stattfinden. Sofern die Teilnehmerzahl kleiner oder gleich der Raumkapazität ist wird ggf in den Präsenzbetrieb gewechselt.							
3	U	Di	16:00 - 18:00	wöch.	Online.Veranstalt	03.11.2020	Dr. rer. nat. Sebastian Böhne
Für Lehramtsstudierende. Bitte beachten Sie dass die Übungen grundsätzlich online stattfinden. Sofern die Teilnehmerzahl kleiner oder gleich der Raumkapazität ist wird ggf in den Präsenzbetrieb gewechselt.							
4	U	Fr	12:00 - 14:00	wöch.	3.04.1.02	06.11.2020	Dr. rer. nat. Sebastian Böhne
Für Lehramtsstudierende. Bitte beachten Sie dass die Übungen grundsätzlich online stattfinden. Sofern die Teilnehmerzahl kleiner oder gleich der Raumkapazität ist wird ggf in den Präsenzbetrieb gewechselt.							
4	U	Fr	12:00 - 14:00	wöch.	Online.Veranstalt	06.11.2020	Dr. rer. nat. Sebastian Böhne
Für Lehramtsstudierende. Bitte beachten Sie dass die Übungen grundsätzlich online stattfinden. Sofern die Teilnehmerzahl kleiner oder gleich der Raumkapazität ist wird ggf in den Präsenzbetrieb gewechselt.							
Kommentar							
Die Theoretische Informatik beschäftigt sich mit den grundlegenden Fragestellungen der Informatik. Hierzu werden Computer- und Automatenmodelle idealisiert und mathematisch untersucht. Die Automatentheorie und die Theorie der formalen Sprachen (Thema des ersten Semesters) ist grundlegend für die Entwicklung von Programmiersprachen und Compilern. Sie untersucht, mit welchen Techniken welche Arten von Sprachen effizient analysiert werden können. Die Berechenbarkeitstheorie (Thema des vierten Semesters) befasst sich mit den prinzipiellen Grenzen des Berechenbaren und der Relation zwischen verschiedenen Computer- und Programmiermodellen. Die Komplexitätstheorie (Thema des vierten Semesters) untersucht Effizienz von Algorithmen im Hinblick auf Platz- und Zeitbedarf und kümmert sich insbesondere um die Frage, wie effizient man bestimmte Probleme lösen kann. Die Veranstaltung ist prinzipiell für Studenten des ersten Semesters geeignet, setzt jedoch ein gutes Verständnis mathematischer Konzepte und Methoden voraus. Für einige Studenten ist es daher sinnvoller, zunächst an den entsprechenden Mathematikveranstaltungen teilzunehmen und die theoretische Informatik erst im dritten Semester zu belegen.							
Voraussetzung							
Die Veranstaltung ist prinzipiell für Studenten des ersten Semesters geeignet, setzt jedoch ein gutes Verständnis mathematischer Konzepte und Methoden voraus. Für die meisten Studenten ist daher die Teilnahme an dem Mathematik Brueckenkurs dringend zu empfehlen.							
Literatur							
J. Hopcroft, R. Motwani, J. Ullman: Einführung in die Automatentheorie, Formale Sprachen und Komplexitätstheorie, Pearson 2002 Michael Sipser: Introduction to the Theory of Computation. 2. Auflage, PWS 2005							
Leistungsnachweis							
Bearbeitung von Hausaufgaben (Voraussetzung fuer Klausurzulassung). Klausur zu Beginn des vorlesungsfreien Zeitraums (i.d.R. Freitag nach Vorlesungsende)							

83802 TU - Theoretische Informatik I: Modellierungskonzepte - Automaten und formale Sprachen							
Gruppe	Art	Tag	Zeit	Rhythmus	Veranstaltungsort	1.Termin	Lehrkraft
1	TU	Mi	10:00 - 12:00	wöch.	Online.Veranstalt	04.11.2020	Prof. Dr. Christoph Kreitz



**Kommentar**

Die Theoretische Informatik beschäftigt sich mit den grundlegenden Fragestellungen der Informatik. Hierzu werden Computer- und Automatenmodelle idealisiert und mathematisch untersucht. Die Automatentheorie und die Theorie der formalen Sprachen (Thema des ersten Semesters) ist grundlegend für die Entwicklung von Programmiersprachen und Compilern. Sie untersucht, mit welchen Techniken welche Arten von Sprachen effizient analysiert werden können. Die Berechenbarkeitstheorie (Thema des vierten Semesters) befasst sich mit den prinzipiellen Grenzen des Berechenbaren und der Relation zwischen verschiedenen Computer- und Programmiermodellen. Die Komplexitätstheorie (Thema des vierten Semesters) untersucht Effizienz von Algorithmen im Hinblick auf Platz- und Zeitbedarf und kümmert sich insbesondere um die Frage, wie effizient man bestimmte Probleme lösen kann. Die Veranstaltung ist prinzipiell für Studenten des ersten Semesters geeignet, setzt jedoch ein gutes Verständnis mathematischer Konzepte und Methoden voraus. Für einige Studenten ist es daher sinnvoller, zunächst an den entsprechenden Mathematikveranstaltungen teilzunehmen und die theoretische Informatik erst im dritten Semester zu belegen.

**Voraussetzung**

Die Veranstaltung ist prinzipiell für Studenten des ersten Semesters geeignet, setzt jedoch ein gutes Verständnis mathematischer Konzepte und Methoden voraus. Für die meisten Studenten ist daher die Teilnahme an dem Mathematik Brueckenkurs dringend zu empfehlen.

**Literatur**

J. Hopcroft, R. Motwani, J. Ullman: Einführung in die Automatentheorie, Formale Sprachen und Komplexitätstheorie, Pearson 2002  
Michael Sipser: Introduction to the Theory of Computation. 2. Auflage, PWS 2005

**Leistungsnachweis**

Bearbeitung von Hausaufgaben (Voraussetzung fuer Klausurzulassung). Klausur zu Beginn des vorlesungsfreien Zeitraums (i.d.R. Freitag nach Vorlesungsende)

**83803 V - Theoretische Informatik I: Modellierungskonzepte - Automaten und formale Sprachen**

Gruppe	Art	Tag	Zeit	Rhythmus	Veranstaltungsort	1.Termin	Lehrkraft
1	V	N.N.	N.N.	wöch.	N.N.	N.N.	Prof. Dr. Christoph Kreitz

**Kommentar**

Die Theoretische Informatik beschäftigt sich mit den grundlegenden Fragestellungen der Informatik. Hierzu werden Computer- und Automatenmodelle idealisiert und mathematisch untersucht. Die Automatentheorie und die Theorie der formalen Sprachen (Thema des ersten Semesters) ist grundlegend für die Entwicklung von Programmiersprachen und Compilern. Sie untersucht, mit welchen Techniken welche Arten von Sprachen effizient analysiert werden können. Die Berechenbarkeitstheorie (Thema des vierten Semesters) befasst sich mit den prinzipiellen Grenzen des Berechenbaren und der Relation zwischen verschiedenen Computer- und Programmiermodellen. Die Komplexitätstheorie (Thema des vierten Semesters) untersucht Effizienz von Algorithmen im Hinblick auf Platz- und Zeitbedarf und kümmert sich insbesondere um die Frage, wie effizient man bestimmte Probleme lösen kann. Die Veranstaltung ist prinzipiell für Studenten des ersten Semesters geeignet, setzt jedoch ein gutes Verständnis mathematischer Konzepte und Methoden voraus. Für einige Studenten ist es daher sinnvoller, zunächst an den entsprechenden Mathematikveranstaltungen teilzunehmen und die theoretische Informatik erst im dritten Semester zu belegen.

**Voraussetzung**

Die Veranstaltung ist prinzipiell für Studenten des ersten Semesters geeignet, setzt jedoch ein gutes Verständnis mathematischer Konzepte und Methoden voraus. Für die meisten Studenten ist daher die Teilnahme an dem Mathematik Brueckenkurs dringend zu empfehlen.

**Literatur**

J. Hopcroft, R. Motwani, J. Ullman: Einführung in die Automatentheorie, Formale Sprachen und Komplexitätstheorie, Pearson 2002  
Michael Sipser: Introduction to the Theory of Computation. 2. Auflage, PWS 2005

**Leistungsnachweis**

Bearbeitung von Hausaufgaben (Voraussetzung fuer Klausurzulassung). Klausur zu Beginn des vorlesungsfreien Zeitraums (i.d.R. Freitag nach Vorlesungsende)

**83812 VU - Grundlagen der Informationsverarbeitung**

Gruppe	Art	Tag	Zeit	Rhythmus	Veranstaltungsort	1.Termin	Lehrkraft
Alle	TU	Mi	10:00 - 12:00	wöch.	Online.Veranstalt	04.11.2020	Prof. Dr.-Ing. Ulrike Lucke
Alle	V	N.N.	N.N.	wöch.	N.N.	N.N.	Prof. Dr.-Ing. Ulrike Lucke
1	U	Di	08:00 - 10:00	wöch.	3.06.H05	03.11.2020	Petra Vogel
2	U	Mo	10:00 - 12:00	wöch.	Online.Veranstalt	02.11.2020	Dr. rer. nat. Sebastian Böhne
3	U	Mi	08:00 - 10:00	wöch.	Online.Veranstalt	04.11.2020	Dr. rer. nat. Sebastian Böhne

## Bemerkung

Weitere Informationen zur Lehrveranstaltung und deren Durchführung im WS 2020/2021 finden Sie in Moodle unter folgendem Link:

[Grundlagen der Informationsverarbeitung](#)

## 83820 S - Boolesche Funktionen und binäre Schaltungen

Gruppe	Art	Tag	Zeit	Rhythmus	Veranstaltungsort	1.Termin	Lehrkraft
1	U	Do	16:00 - 18:00	wöch.	Online.Veranstalt	05.11.2020	Prof. Dr. Michael Gössel

## Kommentar

Digitale Schaltungen sind zunehmend in allen Lebensbereichen präsent und von Wichtigkeit. Aus meiner Sicht ist es dabei von grosser Bedeutung, dass Sie tief verstehen, was Sie entwerfen möchten, die Struktur vorgeben und bewerten können und nicht nur einen Synthesetool anwenden, um ein international konkurrenzfähiges Produkt zu erhalten. Dabei ist die Patentliteratur zu berücksichtigen.

Das Seminar soll ihnen dabei helfen, zu erkennen, wo Ihre Stärken und Möglichkeiten beim künftigen Entwerfen von konkurrenzfähigen digitalen Schaltungen liegen.

## Literatur

Allen Teilnehmerinnen und Teilnehmern wird ein Manuskript im pdf-Format bereitgestellt. Für die Vorträge werden Literaturhinweise einschliesslich zu berücksichtigender Patente gegeben, wobei die Literatur durch die Vortragenden eigenständig ergänzt werden. Das Finden geeigneter relevanter Quellen soll von Ihnen als normaler Teil Ihrer künftigen Arbeit erkannt und geübt werden.

## Leistungsnachweis

Von den Teilnehmerinnen und Teilnehmern, die sich für 3 Leistungspunkte eingetragen haben, wird ein Vortrag gehalten, der bewertet wird und dessen Zensur 70% der Bewertung ergibt.

Zu jeder Veranstaltung ist eine Zusammenfassung von 1/2 bis 1 Seite zu schreiben. Die Bewertung dieser Zusammenfassungen ergibt 30 % der gesamten Bewertung.

Von den Teilnehmern und Teilnehmerinnen, die sich für 6 Leistungspunkte eingeschrieben haben, wird ein Vortrag gehalten, der bewertet wird dessen Zensur mit 40 % in die Bewertung eingeht. Weiterhin wird eine Ausarbeitung zu einem Thema des Seminars angefertigt, deren Zensur mit 40 % in die Bewertung eingeht.

Zu jeder Veranstaltung ist eine Zusammenfassung von 1/2 bis 1 Seite zu schreiben. Die Bewertung dieser Zusammenfassungen ergibt 20% der Zensur.

Zu jedem Vortrag ist eine ausführliche Konsultation per Skype mit dem Seminarleiter vorgesehen, ehe der Vortrag gehalten wird.

## Lerninhalte

Das Seminar beschäftigt sich mit grundlegenden Eigenschaften Boolescher Funktion, binären Schaltung und Anwendungen. Folgenden Themen sollen als Vorlesung durch den Seminarleiter und als Vorträge der Teilnehmer und Teilnehmerinnen behandelt werden:

1. Digitalisierung analoger Werte

2. Boolesche Funktionen. Klassen Boolescher Funktionen (lineare Funktionen, monotone Funktionen, Selbstduale Funktionen, bezüglich Superposition abgeschlossene Klassen.

2. Darstellung Boolescher Funktionen. Normalformen wie disjunktive Normalform, konjunktive Normalform, antivalente Normalform, Implikanten, Primimplikanten, Minimierung. Entscheidungsdiagramme

4. Realisierungen Boolescher Schaltungen. verschiedene Basen, CMOS-Realisierungen, Fehler in binären Schaltungen, Fehlermodelle

Schaltungen zur Fehlererkennung, Scan-Pfad zum Testen Boolescher Schaltungen. Fehlertoleranz.

5. Spezielle Schaltungen wie verschiedene Addierer, Zähler, Decoder u. a.

6. Datenkompression und Codierung. Lineare Datenkompression und linear rückgekoppelte Schieberegister. Linear rückgekoppelte Schieberegister zu Polynommultiplikation und Polynomdivision, Zusammenhang zu zyklischen Codes. WOM-Code, Gray-Code zur Codierung arithmetischer Werte.

## Zielgruppe

Bachelor- und Master-Studenten und Studentinnen, die Grundkenntnisse zu digitalen Systemen und Booleschen Funktionen haben und die in einer Welt digitaler Schaltungen diese Schaltungen verstehen und konkurrenzfähig mitentwickeln wollen und die motiviert zu eigener, schöpferischer Arbeit sind.

83833 VU - Chipentwurf							
Gruppe	Art	Tag	Zeit	Rhythmus	Veranstaltungsort	1.Termin	Lehrkraft
1	U	Di	12:00 - 14:00	wöch.	Online.Veranstalt	03.11.2020	Prof. Dr. Milos Krstic
Übung durch Anselm Breitenreiter							
1	V	N.N.	N.N.	wöch.	N.N.	N.N.	Prof. Dr. Milos Krstic

## Kommentar

Wie geplant am Dienstag, den 3. November von 12:00 bis 14:00 Uhr werden wir die ersten synchronen Online-Übungen haben. Die Übungen werden von Anselm Breitenreiter und Oliver Schrape (in cc) zur Verfügung gestellt. Unten finden Sie die Details der Zoom-Sitzung.

Die Vorlesungen werden im Allgemeinen im asynchronen Modus über Moodle gehalten.

Die Einführungsvorlesung wird über Zoom außergewöhnlich synchron sein, so dass wir uns kennenlernen. Die Einführungsvorlesung findet am Donnerstag, den 5. November von 17:00 bis 18:30 Uhr statt. Unten haben Sie auch den entsprechenden Zoom-Link.

Der Kurs soll in deutscher Sprache mit Folien in englischer Sprache abgehalten werden. Wenn es jedoch Studenten gibt, die die Vorlesungen nicht in deutscher Sprache verfolgen können, werden wir die Alternativen in der Einführungsvorlesung diskutieren.

Im Allgemeinen werden die Kommunikation und das Hochladen von Vorlesungen und Übungen in Moodle durchgeführt. Bitte benutzen Sie den folgenden Link, um sich anzumelden:

<https://moodle2.uni-potsdam.de/course/view.php?id=26760>

Wenn Sie Fragen haben, zögern Sie nicht, uns über Puls, Moodle oder E-Mail zu kontaktieren.

Mit freundlichen Grüßen,

Milos Krstic, Anselm Breitenreiter, Oliver Schrape

Anselm Breitenreiter is inviting you to a scheduled Zoom meeting.

Topic: Chip Design 1<sup>st</sup> Lab Meeting

Time: Nov 3, 2020 12:00 PM Paris

Join Zoom Meeting

<https://uni-potsdam.zoom.us/j/63541183865>

Meeting ID: 635 4118 3865

Passcode: 45160752

One tap mobile

+496950502596,,63541183865#,,,,,,0#,,45160752# Germany

+496971049922,,63541183865#,,,,,,0#,,45160752# Germany

### Dial by your location

+49 695 050 2596 Germany

+49 69 7104 9922 Germany

+49 30 5679 5800 Germany

Meeting ID: 635 4118 3865

Passcode: 45160752

**Voraussetzung**

Grundlagen der Informationsverarbeitung

**Leistungsnachweis**

Die Leistungsbewertung erfolgt basierend auf Projektergebnissen und mündlichen Prüfung

**Lerninhalte**

Beim Design eingebetteter Systeme ist das Zusammenspiel von Software- und Hardwarekomponenten sehr wichtig. Die Grundlage beim Hardwareentwurf ist das Verständnis von diversen Schaltungskonzepten und Designmethodiken. In diesem Zusammenhang soll die Lehrveranstaltung das Konzept von synchronen und asynchronen Designs vorstellen. Aufbauend darauf soll der Entwicklungsprozess von Hardware-Systemen anhand von ASIC- und FPGA-Implementierungen exemplarisch nachvollzogen werden. Ziel dieses Kurses ist das Verständnis der Studenten in Bezug auf Hardwaredesign zu erweitern.

Hier ist die detaillierte Liste der Themen:

- Einführung, VLSI Design
- Hardwareentwurfprinzipien
- Advanced VHDL für Logiksynthese
- Asynchrone Designmethoden
- ASIC Designflow (Logiksynthese, Layout, Verification)
- FPGA Design
- Chip Fertigungsprozess und Chiptest

Diese Veranstaltung beinhaltet Beispiele aus der Praxis und erfordert eine aktive Arbeit der Studenten. Durch diesen Kurs werden die Studierenden in der Lage sein, den Chip zu entwerfen, der tatsächlich im IHP produziert wird. Sie würden auch einen Überblick über die Chip-Produktion bekommen. Schließlich können die Studierenden den tatsächlich produzierten Chip testen.

83837 VU - Grundlagen der Betriebssysteme und Rechnernetze							
Gruppe	Art	Tag	Zeit	Rhythmus	Veranstaltungsort	1.Termin	Lehrkraft
Alle	V	Mo	16:00 - 18:00	wöch.	3.06.H05	02.11.2020	Prof. Dr. Bettina Schnor
Alle	V	Fr	10:00 - 12:00	14t.	3.06.H01	06.11.2020	Prof. Dr. Bettina Schnor
1	U	Fr	10:00 - 12:00	14t.	3.04.0.04	13.11.2020	Petra Vogel
2	U	Fr	10:00 - 12:00	14t.	3.04.0.04	13.11.2020	Petra Vogel
3	U	Fr	10:00 - 12:00	14t.	3.06.S27	13.11.2020	Petra Vogel
4	U	Fr	16:00 - 18:00	14t.	3.01.H10	13.11.2020	Petra Vogel
4	U	Fr	16:00 - 18:00	Einzel	3.06.H03	22.01.2021	Petra Vogel

**Kommentar**

Die Vorlesung gibt eine Einführung in die Grundlagen der Betriebssysteme und Rechnernetze. Im ersten Teil der Vorlesung werden die Grundlagen von Betriebssystemen vermittelt: Adressräume, Speicherverwaltung, Organisation des Dateisystems, Prozessverwaltung, Nebenläufigkeit, Koordination/Synchronisation und Verklebungen. Exemplarisch werden die Betriebssysteme UNIX und Windows besprochen. Im zweiten Teil der Vorlesung werden die Grundlagen der Rechnerkommunikation vermittelt. Dazu werden anhand des ISO-Referenzmodell OSI die verschiedenen Schichten von Kommunikationsarchitekturen besprochen. Als konkretes Beispiel wird die Internetarchitektur mit den Internetprotokollen TCP, UDP und IP vorgestellt.

**Voraussetzung**

Grundlagen aus der technischen Informatik wie sie zum Beispiel im Modul Grundlagen der Informationsverarbeitung vermittelt werden.

**Leistungsnachweis**

Es müssen 50% der Hausaufgabenpunkte erreicht werden, um zur Klausur zugelassen zu werden. Die Note ergibt sich aus der Klausurnote.

**Bemerkung**

Weitere Informationen zur Lehrveranstaltung und deren Durchführung im WS 2020/2021 finden Sie in Moodle unter folgendem Link:

[Grundlagen Betriebssysteme und Rechnernetze](#)

und auf der Webseite zur Lehrveranstaltung [Webseite zur Vorlesung/Übung GBR WS 20/21](#)

83839 VU - Inferenz-Methoden							
Gruppe	Art	Tag	Zeit	Rhythmus	Veranstaltungsort	1.Termin	Lehrkraft
1	VU	Mi	08:00 - 10:00	wöch.	Online.Voranstalt	04.11.2020	Prof. Dr. Christoph Kreitz
1	V	N.N.	N.N.	wöch.	N.N.	N.N.	Prof. Dr. Christoph Kreitz
<b>Kommentar</b>							
Logisches Schließen steht im Zentrum allen intelligenten Handelns. Die Fähigkeit, logische Schlüsse zu ziehen, ist die Voraussetzung für das Lösen von Problemen, für das Planen von Aktionen, für kognitives Verständnis und damit im Endeffekt für jede Form des wissenschaftlichen Fortschritts. Inferenzmethoden, die automatische Verarbeitung von Wissen mittels logischer Schlüsse, sind daher eine der Schlüsseltechniken der Künstlichen Intelligenz. Speziell spielen sie bei Expertensystemen, intelligenten Agenten, logischen Programmiersprachen, der Verifikation und Synthese von Programmen, und in vielen weiteren Anwendungen eine fundamentale Rolle. Im Modul werden die wichtigsten Konzepte des automatischen Schließens vorgestellt und demonstriert. Themenschwerpunkte sind * Prädikatenlogik und formale Kalküle (Seqüenzen und Tableauxverfahren) * Die Konnektionsmethode und ihre Beziehung zu Resolution und deren Verfeinerungen * Unifikation; Optimierungstechniken; Spezielle Verfahren für Aussagenlogik * Einbau von Theorien, insbesondere Gleichheit, Induktion und Termersetzung * Behandlung von Modallogik; konstruktive Logik, lineare Logik; Logik höherer Stufe Optional: Seminar zur Vertiefung aktueller Themen im Anschluss die Veranstaltung (P2)							
<b>Voraussetzung</b>							
Logikvorkenntnisse sind erforderlich							
<b>Literatur</b>							
*Wolfgang Bibel: Deduktion -- Automatisierung der Logik, R.Oldenbourg, 1992 *Lincoln Wallen: Automated deduction in nonclassical logics, MIT Press, 1990 *Literatur zu den einzelnen Themengruppen wird separat vorgestellt.							
<b>Leistungsnachweis</b>							
Muendliche Pruefung. Klausur bei mehr als 15 Teilnehmern							

83849 VU - Multimedia-Technologie							
Gruppe	Art	Tag	Zeit	Rhythmus	Veranstaltungsort	1.Termin	Lehrkraft
Alle	V	N.N.	N.N.	wöch.	N.N.	N.N.	Prof. Dr.-Ing. Ulrike Lucke
1	U	Do	14:00 - 16:00	wöch.	Online.Veranstatt	05.11.2020	Axel Wiepke
<b>Links:</b>							
begleitender Moodle-Kurs		<a href="https://moodle2.uni-potsdam.de/course/view.php?id=26558">https://moodle2.uni-potsdam.de/course/view.php?id=26558</a>					
<b>Kommentar</b>							
<p>Die Vorlesung Multimedia-Technologie vermittelt einen Einstieg in die Grundlagen, Verfahren, Komponenten und Systeme multimedialer Datenverarbeitung. Das beginnt bei der digitalen Speicherung von Informationen, geht über die Übertragung und Verarbeitung dieser Daten und endet mit Ansätzen zur Darstellung von bzw. Interaktion mit digitalen Medien. Das schließt sowohl statische (z.B. Grafik, Text) als auch dynamische (z.B. Audio, Video) Medientypen ein. Die Vorlesung behandelt theoretisches Überblickswissen, das in den Übungen praktisch erprobt und auf verschiedene Anwendungsfälle übertragen wird. Begleitend wird im Rahmen einer Praxisaufgabe ein eigenes Multimediaangebot entwickelt. Es wird ein grundlegendes Verständnis für die digitale Informationsverarbeitung vorausgesetzt. Daher eignet sich die Veranstaltung insbesondere für Studienanfänger der Informatik sowie als Neben-/Beifach.</p>							
<b>Voraussetzung</b>							
grundlegendes Verständnis für die digitale Informationsverarbeitung							
<b>Literatur</b>							
wird in der Lehrveranstaltung bekannt gegeben							
<b>Leistungsnachweis</b>							
wird in der Lehrveranstaltung bekannt gegeben							



83851 PJ - Software Engineering I							
Gruppe	Art	Tag	Zeit	Rhythmus	Veranstaltungsort	1.Termin	Lehrkraft
1	PJ	Sa	12:00 - 14:00	Einzel	3.06.H04	07.11.2020	Dr. Henning Bordihn
1	PJ	Sa	12:00 - 14:00	Einzel	3.06.H03	07.11.2020	Dr. Henning Bordihn
1	PJ	Sa	14:00 - 16:00	Einzel	3.06.H04	07.11.2020	Dr. Henning Bordihn
1	PJ	Sa	14:00 - 16:00	Einzel	3.06.H03	07.11.2020	Dr. Henning Bordihn
1	PJ	Fr	10:00 - 12:00	wöch.	3.04.0.02	20.11.2020	Dr. Henning Bordihn
<b>Kommentar</b>							
<p><b>**Kursbeschreibung**:</b> Diese Veranstaltung vermittelt theoretische Grundlagen und praktische Kompetenzen der Modellierung und der Softwareentwicklung. Neben traditionellen Ansätzen wird das moderne Paradigma der modellgetriebenen Softwareentwicklung behandelt. Bei der Modellierung werden verschiedene Beschreibungsformen für das Verhalten und die Struktur eines Softwaresystems oder seiner Komponenten betrachtet. Ferner werden die Kernphasen des Software-Entwicklungsprozesses betrachtet, von der Anforderungsanalyse über den Entwurf und die Implementierung bis zum Testen. Die modellgetriebene Softwareentwicklung umfasst Methoden und Techniken, die Software weitgehend automatisiert aus geeigneten Modellen generieren. Der in diesem Zusammenhang bedeutsamen Verifikation von kritischen Systemeigenschaften auf Modellebene wird durch die Behandlung von Methoden des Model Checking Rechnung getragen. Die Konzepte werden anhand von Anwendungsbeispielen und Werkzeugen demonstriert und geübt. Ausgewählte Aspekte werden vertieft und in einem Projekt angewendet. Zu den Inhalten der Lehrveranstaltung gehören unter anderem: - Kernphasen der Softwareentwicklung und Vorgehensmodelle - Anforderungsanalyse und -spezifikation - Verhaltensmodellierung/Prozessmodellierung - Qualitätssicherung auf Modellebene insbesondere durch Model Checking - Strukturmodellierung, objektorientierte Modellierung - Software-Architekturen und Design-Patterns - Objektorientierte Implementierung von Entwürfen - Verifikation und Validierung mit dem Schwerpunkt auf Testen - Modellierungsmethodik, Metamodellierung</p>							
<b>Voraussetzung</b>							
<p>Grundlagen der Programmierung.</p> <p>Ferner werden die Kenntnisse aus den Kursen Theoretische Informatik I (bzw. Modellierungskonzepte der Informatik) und Praxis der Programmierung erwartet.</p>							
<b>Literatur</b>							
<p>Ian Sommerville. Software Engineering, Ninth Edition. Pearson, 2011, 0-13-705346-0 Ian Sommerville, Perdita Stevens. Software Engineering: AND Using UML, Software Engineering with Objects and Components. Pearson Education, 2007, 1-4058-9258-7 Helmut Balzert. Lehrbuch der Software-Technik (Band 1): Software-Entwicklung. Spektrum Akademischer Verlag Heidelberg, 1996, 3-8274-0042-2 Perdita Stevens. Using UML: Software Engineering with Objects and Components. Pearson Education, 2005, 0-3212-6967-5 M. Hitz, G. Kappel, E. Kapsammer, W. Retschitzegger. UML@Work. Objektorientierte Modellierung mit UML 2. dpunkt.verlag, Heidelberg, 2005, 3-89864-261-5 Edmund M. Clarke Jr., Orna Grumberg, Doron A. Peled. Model Checking. MIT Press, 2000, 978-0-262-03270-4 Mary Poppendieck, Tom Poppendieck. Lean Software Development: An Agile Toolkit. Addison-Wesley Professional, 2003, 0-3211-5078-3</p>							
<b>Leistungsnachweis</b>							
<p>Am Ende des Semesters werden 6 benotete Leistungspunkte vergeben, wobei die Zensur folgendermaßen als gewichtetes Mittel vergeben wird: * Klausur: 70%, * Projekt: 30%. Die Klausur muss bestanden werden.</p>							
<b>Bemerkung</b>							
<p>Im Wintersemester 2020/21 wird der Kurs in deutscher Sprache angeboten. Voraussetzung für die Teilnahme im Masterprogramm sind:</p> <p>1) ausreichende Deutschkenntnisse,</p> <p>2) dass kein Kurs mit ähnlichen Inhalten im Bachelorsrudium besucht worden ist, der zum Zugang zu Masterstudium berechtigte.</p>							

83852 VU - Software Engineering I							
Gruppe	Art	Tag	Zeit	Rhythmus	Veranstaltungsort	1.Termin	Lehrkraft
Alle	V	Mo	14:00 - 16:00	wöch.	3.06.H05	02.11.2020	Dr. Henning Bordihn
Der Vorlesungstermin am Montag wird alternativ zur Vorlesung am Mittwoch angeboten. Der Vorlesungstermin am Montag entfällt, sobald die Raumkapazität des Mittwochstermins genügt. Bitte halten Sie sich deshalb den Mittwochstermin in jedem Fall frei!							
Alle	V	Mi	14:00 - 16:00	wöch.	3.06.H05	04.11.2020	Dr. Henning Bordihn
1	U	Mo	12:00 - 14:00	wöch.	3.06.H04	09.11.2020	Dr. Henning Bordihn
2	U	Fr	08:00 - 10:00	wöch.	3.04.0.04	06.11.2020	Dr. Henning Bordihn
3	U	Fr	12:00 - 14:00	wöch.	3.06.H03	06.11.2020	Dr. Henning Bordihn
3	U	Fr	12:00 - 14:00	Einzel	3.01.H10	04.12.2020	Dr. Henning Bordihn
3	U	Fr	12:00 - 14:00	wöch.	3.06.H04	11.12.2020	Dr. Henning Bordihn
3	U	Fr	12:00 - 14:00	Einzel	3.01.H10	08.01.2021	Dr. Henning Bordihn
4	U	Fr	12:00 - 14:00	wöch.	3.04.0.04	06.11.2020	Dr. Henning Bordihn
Für Lehramtsstudierende.							
5	U	Fr	14:00 - 16:00	wöch.	3.04.0.04	06.11.2020	N.N.
<b>Kommentar</b>							
<p><b>**Kursbeschreibung**:</b> Diese Veranstaltung vermittelt theoretische Grundlagen und praktische Kompetenzen der Modellierung und der Softwareentwicklung. Neben traditionellen Ansätzen wird das moderne Paradigma der modellgetriebenen Softwareentwicklung behandelt. Bei der Modellierung werden verschiedene Beschreibungsformen für das Verhalten und die Struktur eines Softwaresystems oder seiner Komponenten betrachtet. Ferner werden die Kernphasen des Software-Entwicklungsprozesses betrachtet, von der Anforderungsanalyse über den Entwurf und die Implementierung bis zum Testen. Die modellgetriebene Softwareentwicklung umfasst Methoden und Techniken, die Software weitgehend automatisiert aus geeigneten Modellen generieren. Der in diesem Zusammenhang bedeutsamen Verifikation von kritischen Systemeigenschaften auf Modellebene wird durch die Behandlung von Methoden des Model Checking Rechnung getragen. Die Konzepte werden anhand von Anwendungsbeispielen und Werkzeugen demonstriert und geübt. Ausgewählte Aspekte werden vertieft und in einem Projekt angewendet. Zu den Inhalten der Lehrveranstaltung gehören unter anderem: - Kernphasen der Softwareentwicklung und Vorgehensmodelle - Anforderungsanalyse und -spezifikation - Verhaltensmodellierung/Prozessmodellierung - Qualitätssicherung auf Modellebene insbesondere durch Model Checking - Strukturmodellierung, objektorientierte Modellierung - Software-Architekturen und Design-Patterns - Objektorientierte Implementierung von Entwürfen - Verifikation und Validierung mit dem Schwerpunkt auf Testen - Modellierungsmethodik, Metamodellierung</p>							
<b>Voraussetzung</b>							
Grundlagen der Programmierung.							
Ferner werden die Kenntnisse aus den Kursen Theoretische Informatik I (bzw. Modellierungskonzepte der Informatik) und Praxis der Programmierung erwartet.							
<b>Literatur</b>							
<p>Ian Sommerville. Software Engineering, Ninth Edition. Pearson, 2011, 0-13-705346-0 Ian Sommerville, Perdita Stevens. Software Engineering: AND Using UML, Software Engineering with Objects and Components. Pearson Education, 2007, 1-4058-9258-7 Helmut Balzert. Lehrbuch der Software-Technik (Band 1): Software-Entwicklung. Spektrum Akademischer Verlag Heidelberg, 1996, 3-8274-0042-2 Perdita Stevens. Using UML: Software Engineering with Objects and Components. Pearson Education, 2005, 0-3212-6967-5 M. Hitz, G. Kappel, E. Kapsammer, W. Retschitzegger. UML@Work. Objektorientierte Modellierung mit UML 2. dpunkt.verlag, Heidelberg, 2005, 3-89864-261-5 Edmund M. Clarke Jr., Orna Grumberg, Doron A. Peled. Model Checking. MIT Press, 2000, 978-0-262-03270-4 Mary Poppendieck, Tom Poppendieck. Lean Software Development: An Agile Toolkit. Addison-Wesley Professional, 2003, 0-3211-5078-3</p>							
<b>Leistungsnachweis</b>							
Am Ende des Semesters werden 6 benotete Leistungspunkte vergeben, wobei die Zensur folgendermaßen als gewichtetes Mittel vergeben wird: * Klausur: 70%, * Projekt: 30%. Die Klausur muss bestanden werden.							

### Bemerkung

Bitte beachten Sie, dass dieser Kurs zu einem Pflichtmodul im Bachelorprogramm gehört

und deshalb in deutscher Sprache gelehrt wird. Masterstudenten können unter zwei Voraussetzungen teilnehmen:

- 1) Sie sind in der Lage, den Kurs in deutscher Sprache zu verfolgen und auch in den Projektgruppen auf deutsch mitzuarbeiten.
- 2) Sie hatten keinen vergleichbaren Kurs in Ihrem Bachelorstudium.

**Bitte registrieren Sie sich für einen der alternativen Vorlesungstermine unter folgendem Link:**

<https://terminplaner4.dfn.de/9EhISX8bjA2ZNn4U>

### Kurzkommentar

**Einschreibeschlüssel Moodle: SoEn20**

**Bitte registrieren Sie sich für eine der Übungsgruppen im Moodlekurs.**

**Die Zuordnung der Übungsgruppen erfolgt ausschließlich über Moodle.**

**PULS ist hierfür nicht maßgeblich.**

### 83854 VU - Compiler und Programmtransformation

Gruppe	Art	Tag	Zeit	Rhythmus	Veranstaltungsort	1.Termin	Lehrkraft
1	V	Do	08:00 - 10:00	wöch.	3.04.0.04	05.11.2020	Dr. Henning Bordihn
1	U	Do	10:00 - 12:00	wöch.	3.04.0.04	05.11.2020	Dr. Henning Bordihn

### Kommentar

In modernen Ansätzen der Softwareentwicklung greift man immer häufiger auf spezialisierte Hochsprachen zurück, die auf Aspekte der jeweiligen Anwendungsdomäne oder eine möglichst einfache Kommunikation mit Auftraggebern oder potenziellen Kunden zugeschnitten sind. Dies erlaubt eine Beschreibung der Systeme, die von technischen Details abstrahiert und ihre Validierung erleichtert.

Einen größeren Mehrwert erzielt man, wenn die Hochsprachen-Beschreibungen automatisch in ausführbare Sprachen transformiert werden können. Dieser Prozess wird auch von klassischen Compilern ausgeführt, die Beschreibungen z.B. in universellen Programmiersprachen in Maschinencode übersetzen.

In diesem Kurs werden die Methoden und Techniken des klassischen Compilerbaus behandelt. Darauf aufbauend wird dann erläutert, was bei der Definition von domänenspezifischen Hochsprachen zu beachten ist und wie die klassischen Compiler-Techniken auf die Transformation dieser Sprachen in andere Programmiersprachen angewendet werden können.

Die in der Vorlesung vorgestellten Techniken werden sowohl theoretisch beleuchtet als auch anhand konkreter Anwendungsszenarien im Rahmen der Übungen erprobt. Geeignete Softwaretools werden vorgestellt, von denen ausgewählte in zwei kleinen Projekten eingesetzt werden.

### Voraussetzung

Theoretische Informatik 1 (bzw. Modellierungskonzepte der Informatik) und Praxis der Programmierung

### Leistungsnachweis

Prüfungsnebenleistung (für den Abschluss des Moduls): zwei Projekte, die je zu mindestens 50% erfüllt werden müssen

Prüfungsleistung: mündliche Prüfung 30 Minuten

83905 VU - Maschinelles Lernen & Intelligente Datenanalyse II							
Gruppe	Art	Tag	Zeit	Rhythmus	Veranstaltungsort	1.Termin	Lehrkraft
Alle	V	Do	10:00 - 12:00	wöch.	Online.Veransta	05.11.2020	Prof. Dr. Tobias Scheffer
1	U	Mi	12:00 - 14:00	wöch.	Online.Veransta	04.11.2020	Prof. Dr. Tobias Scheffer
2	U	Do	12:00 - 14:00	wöch.	Online.Veransta	05.11.2020	Prof. Dr. Tobias Scheffer
3	U	Fr	12:00 - 14:00	wöch.	Online.Veransta	06.11.2020	Prof. Dr. Tobias Scheffer
<b>Kommentar</b>							
<p>Aufbauend auf der Vorlesung Intelligente Datenanalyse beschäftigt sich die Veranstaltung vertiefend mit Algorithmen, die aus Daten lernen können. Algorithmen des maschinellen Lernens gewinnen aus Daten Modelle, mit denen sich dann Vorhersagen über das beobachtete System treffen lassen. Anwendungen für Datenanalyse-Verfahren erstrecken sich von der Vorhersage von Kreditrisiken über die Auswertung astronomischer Daten bis zu persönlichen Musikempfehlungen. Die Veranstaltung setzt sich aus einem Vorlesungs- und einem Projektteil zusammen. Der Vorlesungsteil vermittelt das notwendige Wissen über Datenanalyse sowie über Matlab. Im Projektteil werden anwendungsnahe Aufgaben eigenständig bearbeitet.</p>							
<b>Voraussetzung</b>							
Intelligente Datenanalyse							
<b>Leistungsnachweis</b>							
Projektaufgabe und mündliche Prüfung							

83906 S - Parallel Computing - Praktikum							
Gruppe	Art	Tag	Zeit	Rhythmus	Veranstaltungsort	1.Termin	Lehrkraft
1	S	Mi	16:00 - 18:00	wöch.	Online.Veransta	04.11.2020	Prof. Dr. Bettina Schnor
<b>Kommentar</b>							
<p>In dem Seminar sollen aktuelle Themen aus dem Bereich High Performance Computing bearbeitet werden. Dabei wird ein Schwerpunkt auf selbstständiges praktisches Arbeiten gelegt. Die Themen werden in Kooperation mit wissenschaftlichen Einrichtungen wie beispielsweise dem GeoForschungszentrum (GFZ) und dem Potsdamer Institut fuer Klimafolgenforschung (PIK) bzw. Anwendern aus den naturwissenschaftlichen Instituten der Universität Potsdam gestellt.</p>							
<b>Voraussetzung</b>							
Vorlesung Paralleles Rechnen							
<b>Leistungsnachweis</b>							
<p>Erfolgreicher Vortrag und eine Ausarbeitung werden verlangt. Da die Arbeitsgruppen in den Kooperationseinrichtungen i.d.R. international besetzt sind, ist die Arbeitssprache Englisch. Die Sprache von Abschlussvortrag und Ausarbeitung werden im Einzelfall festgelegt.</p>							
<b>Bemerkung</b>							
deutsch/englisch							
<p>Weitere Informationen zur Lehrveranstaltung finden Sie hier <a href="https://www.cs.uni-potsdam.de/bs/teaching/courses.html">https://www.cs.uni-potsdam.de/bs/teaching/courses.html</a></p>							

83909 VU - Pervasive Computing							
Gruppe	Art	Tag	Zeit	Rhythmus	Veranstaltungsort	1.Termin	Lehrkraft
1	U	Mi	12:00 - 14:00	wöch.	Online.Veranstalt	04.11.2020	Dr. Raphael Zender
1	V	N.N.	N.N.	wöch.	N.N.	N.N.	Dr. Raphael Zender
Links:							
Moodle-Kurs			<a href="https://moodle2.uni-potsdam.de/course/view.php?id=25623">https://moodle2.uni-potsdam.de/course/view.php?id=25623</a>				
Kommentar							
Der Kurs bietet einen tiefgehenden Einblick in das hochaktuelle Forschungsfeld Pervasive Computing. Neben den grundlegenden Herausforderungen intelligenter Umgebungen werden aktuelle Entwicklungen zu Themen wie mobile Netze, Kontextbewusstsein, intuitive Mensch-Maschine-Kommunikation, Sicherheit & Vertraulichkeit u.v.m. behandelt sowie anhand von Fallstudien aus den Bereichen Pervasive Learning und Pervasive Games diskutiert.							
Voraussetzung							
Grundkenntnisse von Rechnernetzen							
Leistungsnachweis							
Klausur							

**Bemerkung**

Die Vorlesung wird wöchentlich als **Aufzeichnung** in Moodle zur Verfügung gestellt. Der Mittwoch-Termin dient der Diskussion der Vorlesungsinhalte und wird **online** durchgeführt.

**Zusätzlich** findet jede zweite Woche nach der Diskussion eine Übungsveranstaltung am Mittwoch-Termin statt - ebenfalls online. Für die Übung ist die selbstständige Bearbeitung eines zuvor veröffentlichten Übungsblattes erforderlich. Darüber hinaus wird das Erbringen einer Selbstlernleistung erwartet. Details dazu finden sich im Moodle-Kurs.

Der **erste Termin der Lehrveranstaltung ist Mittwoch, der 04.11.2020 um 12 Uhr**. Alle Interessenten sollten an diesem Termin teilnehmen, da hier der weitere Ablauf des Semesters besprochen wird! Details zur Teilnahme finden sich im Moodle-Kurs.

**Zum Moodle-Kurs** : <https://moodle2.uni-potsdam.de/course/view.php?id=25623>

83916 VP - Declarative Modeling							
Gruppe	Art	Tag	Zeit	Rhythmus	Veranstaltungsort	1.Termin	Lehrkraft
Alle	V	N.N.	N.N.	wöch.	N.N.	N.N.	Javier Romero Davila, Prof. Dr. Torsten Schaub, Etienne Tignon
1	U	Mo	12:00 - 14:00	wöch.	Online.Veranstalt	04.01.2021	Javier Romero Davila, Francois Laferriere, Prof. Dr. Torsten Schaub, Etienne Tignon
1	PR	N.N.	N.N.	wöch.	N.N.	N.N.	Francois Laferriere, Javier Romero Davila, Prof. Dr. Torsten Schaub, Etienne Tignon

**Links:**

moodle	<a href="https://moodle2.uni-potsdam.de/course/view.php?id=27164">https://moodle2.uni-potsdam.de/course/view.php?id=27164</a>
potassco	<a href="https://potassco.org">https://potassco.org</a>
user guide	<a href="https://github.com/potassco/guide/releases">https://github.com/potassco/guide/releases</a>

**Kommentar**

The goal of this course is to learn and experience advanced modeling and implementation techniques in the area of declarative problem solving, more precisely, answer set programming (ASP); it is conceived as a continuation of the course on Declarative Problem Solving and Optimization.

**Voraussetzung**

Either previous or simultaneous course on Declarative Problem Solving and Optimization.

**Literatur**

- Answer Set Solving in Practice by Martin Gebser, Roland Kaminski, Benjamin Kaufmann, and Torsten Schaub. Synthesis Lectures on Artificial Intelligence and Machine Learning, Morgan and Claypool
- Potassco User Guide by the Potassco team, <https://github.com/potassco/guide/releases>
- Answer Set Programming by Vladimir Lifschitz. Springer
- Knowledge Representation, Reasoning, and the Design of Intelligent Agents: The Answer-Set Programming Approach by Michael Gelfond and Yulia Kahl. Cambridge University Press

**Leistungsnachweis**

Implementation, documentation, presentation.

**Lerninhalte**

- Motivation
- Sophisticated modeling
- Multi-shot solving
- Theory solving
- Heuristic-driven solving
- Systems
- Preferences and Optimization
- Applications

**Zielgruppe**

MSc students who want to deepen their practical knowledge in declarative problem solving, more precisely, answer set programming (ASP)

83920 PR - Declarative Problem Solving and Optimization							
Gruppe	Art	Tag	Zeit	Rhythmus	Veranstaltungsort	1.Termin	Lehrkraft
1	PR	N.N.	N.N.	wöch.	N.N.	N.N.	Francois Laferriere, Javier Romero Davila, Prof. Dr. Torsten Schaub, Etienne Tignon
Kommentar							
Suite of usually practical assignments accompanying the course							
Leistungsnachweis							
Successfully accomplished assignments							

83921 VU - Declarative Problem Solving and Optimization							
Gruppe	Art	Tag	Zeit	Rhythmus	Veranstaltungsort	1.Termin	Lehrkraft
Alle	V	N.N.	N.N.	wöch.	N.N.	N.N.	Prof. Dr. Torsten Schaub
1	U	Mo	12:00 - 14:00	wöch.	Online.Veranstalt	02.11.2020	Javier Romero Davila, Francois Laferriere, Prof. Dr. Torsten Schaub, Etienne Tignon
Links:							
moodle			<a href="https://moodle2.uni-potsdam.de/course/view.php?id=22378">https://moodle2.uni-potsdam.de/course/view.php?id=22378</a>				
potassco			<a href="https://potassco.org">https://potassco.org</a>				
user guide			<a href="https://github.com/potassco/guide/releases">https://github.com/potassco/guide/releases</a>				
Kommentar							
<p>Answer Set Programming (ASP) is a prime approach to declarative problem solving. Although initially tailored to modeling problems in the area of Knowledge Representation and Reasoning (KRR), its attractive combination of a rich yet simple modeling language with high-performance solving capacities has sparked interests in academia and industry way beyond KRR. This course presents a detailed introduction to ASP, aiming at using ASP languages and systems for solving application problems. Starting from the essential formal foundations, it introduces ASP's solving technology, modeling language and methodology, while illustrating the overall solving process by practical examples.</p>							
Voraussetzung							
Motivation.							
Literatur							
<ul style="list-style-type: none"><li>• Answer Set Solving in Practice by Martin Gebser, Roland Kaminski, Benjamin Kaufmann, and Torsten Schaub. Synthesis Lectures on Artificial Intelligence and Machine Learning, Morgan and Claypool</li><li>• Potassco User Guide by the Potassco team, <a href="https://github.com/potassco/guide/releases">https://github.com/potassco/guide/releases</a></li><li>• Answer Set Programming by Vladimir Lifschitz. Springer</li><li>• Knowledge Representation, Reasoning, and the Design of Intelligent Agents: The Answer-Set Programming Approach by Michael Gelfond and Yulia Kahl. Cambridge University Press</li></ul>							
Leistungsnachweis							
Marked exam and assignments							



### Bemerkung

Offline communication is conducted primarily via the associated moodle page.

Announcements are also made through the email list of registered students in puls.

Questions can be address to [asp@lists.cs.uni-potsdam.de](mailto:asp@lists.cs.uni-potsdam.de)

A tutorial introduction to answer set programming, used in the projects, is given separately.

### Lerninhalte

- Motivation
- Introduction
- Modeling
- Language
- Grounding
- Foundations
- Solving
- Advanced modeling

### Kurzkommentar

We start Monday Nov 9 - cu

### Zielgruppe

This is an introductory lecture for MSc students with varying backgrounds.

### 83922 VU - Didaktik der Informatik II

Gruppe	Art	Tag	Zeit	Rhythmus	Veranstaltungsort	1.Termin	Lehrkraft
Alle	V	Mo	14:00 - 16:00	wöch.	Online.Veranstalt	02.11.2020	Prof. Dr. Andreas Schwill
1	U	Do	14:00 - 16:00	Einzel	3.04.1.02	05.11.2020	Christian Hoffmann, Prof. Dr. Andreas Schwill
1	U	Do	18:00 - 20:00	wöch.	3.04.1.02	12.11.2020	Prof. Dr. Andreas Schwill

### Leistungsnachweis

Prüfungsgespräch im Umfang von 15-20 Minuten oder Projektarbeit

### 83935 VU - Leistungsanalyse: Messen, Modellieren, Simulation

Gruppe	Art	Tag	Zeit	Rhythmus	Veranstaltungsort	1.Termin	Lehrkraft
1	VU	Do	14:00 - 16:00	wöch.	3.06.H01	05.11.2020	Prof. Dr. Bettina Schnor
1	VU	Do	14:00 - 16:00	wöch.	3.06.H02	10.12.2020	Prof. Dr. Bettina Schnor
1	VU	Do	14:00 - 16:00	Einzel	3.06.H01	11.02.2021	Prof. Dr. Bettina Schnor

### Leistungsnachweis

Projekthausaufgabe mit Zwischenberichten und Abschlussvortrag

### Bemerkung

Weitere Informationen zur Lehrveranstaltung finden Sie hier: <https://www.cs.uni-potsdam.de/bs/teaching/courses.html>

### 83936 PJ - Leistungsanalyse: Messen, Modellieren, Simulation

Gruppe	Art	Tag	Zeit	Rhythmus	Veranstaltungsort	1.Termin	Lehrkraft
1	PJ	N.N.	N.N.	wöch.	N.N.	N.N.	Prof. Dr. Bettina Schnor

### Leistungsnachweis

Projekthausaufgabe mit Zwischenberichten und Abschlussvortrag

### Studienprojekt

Für dieses Modul werden aktuell keine Lehrveranstaltungen angeboten

#### Fakultativ

Für dieses Modul werden aktuell keine Lehrveranstaltungen angeboten

## Fakultative Lehrveranstaltungen

# Glossar

Die folgenden Begriffserklärungen zu Prüfungsleistung, Prüfungsnebenleistung und Studienleistung gelten im Bezug auf Lehrveranstaltungen für alle Ordnungen, die seit dem WiSe 2013/14 in Kraft getreten sind.

<b>Prüfungsleistung</b>	Prüfungsleistungen sind benotete Leistungen innerhalb eines Moduls. Aus der Benotung der Prüfungsleistung(en) bildet sich die Modulnote, die in die Gesamtnote des Studiengangs eingeht. Handelt es sich um eine unbenotete Prüfungsleistung, so muss dieses ausdrücklich („unbenotet“) in der Modulbeschreibung der fachspezifischen Ordnung geregelt sein. Weitere Informationen, auch zu den Anmeldeöglichkeiten von Prüfungsleistungen, finden Sie unter anderem in der <a href="#">Kommentierung der BaMa-O</a>
<b>Prüfungsnebenleistung</b>	Prüfungsnebenleistungen sind für den Abschluss eines Moduls relevante Leistungen, die – soweit sie vorgesehen sind – in der Modulbeschreibung der fachspezifischen Ordnung beschrieben sind. Prüfungsnebenleistungen sind immer unbenotet und werden lediglich mit "bestanden" bzw. "nicht bestanden" bewertet. Die Modulbeschreibung regelt, ob die Prüfungsnebenleistung eine Teilnahmevoraussetzung für eine Modulprüfung oder eine Abschlussvoraussetzung für ein ganzes Modul ist. Als Teilnahmevoraussetzung für eine Modulprüfung muss die Prüfungsnebenleistung erfolgreich vor der Anmeldung bzw. Teilnahme an der Modulprüfung erbracht worden sein. Auch für Erbringung einer Prüfungsnebenleistungen wird eine Anmeldung vorausgesetzt. Diese fällt immer mit der Belegung der Lehrveranstaltung zusammen, da Prüfungsnebenleistung im Rahmen einer Lehrveranstaltungen absolviert werden. Sieht also Ihre fachspezifische Ordnung Prüfungsnebenleistungen bei Lehrveranstaltungen vor, sind diese Lehrveranstaltungen zwingend zu belegen, um die Prüfungsnebenleistung absolvieren zu können.
<b>Studienleistung</b>	Als Studienleistung werden Leistungen bezeichnet, die weder Prüfungsleistungen noch Prüfungsnebenleistungen sind.



Quelle: Karla Pirze

# Impressum

## Herausgeber

Am Neuen Palais 10  
14469 Potsdam

Telefon: +49 331/977-0

Fax: +49 331/972163

E-mail: [presse@uni-potsdam.de](mailto:presse@uni-potsdam.de)

Internet: [www.uni-potsdam.de](http://www.uni-potsdam.de)

## Umsatzsteueridentifikationsnummer

DE138408327

## Layout und Gestaltung

[jung-design.net](http://jung-design.net)

## Druck

16.2.2021

## Rechtsform und gesetzliche Vertretung

Die Universität Potsdam ist eine Körperschaft des Öffentlichen Rechts. Sie wird gesetzlich vertreten durch Prof. Oliver Günther, Ph.D., Präsident der Universität Potsdam, Am Neuen Palais 10, 14469 Potsdam.

## Zuständige Aufsichtsbehörde

Ministerium für Wissenschaft, Forschung und Kultur des Landes Brandenburg  
Dortustr. 36  
14467 Potsdam

## Inhaltliche Verantwortlichkeit i. S. v. § 5 TMG und § 55 Abs. 2 RStV

Referat für Presse- und Öffentlichkeitsarbeit  
Referatsleiterin und Sprecherin der Universität  
Silke Engel

Am Neuen Palais 10

14469 Potsdam

Telefon: +49 331/977-1474

Fax: +49 331/977-1130

E-mail: [presse@uni-potsdam.de](mailto:presse@uni-potsdam.de)

Die einzelnen Fakultäten, Institute und Einrichtungen der Universität Potsdam sind für die Inhalte und Informationen ihrer Lehrveranstaltungen zuständig.



[puls.uni-potsdam.de](http://puls.uni-potsdam.de)

