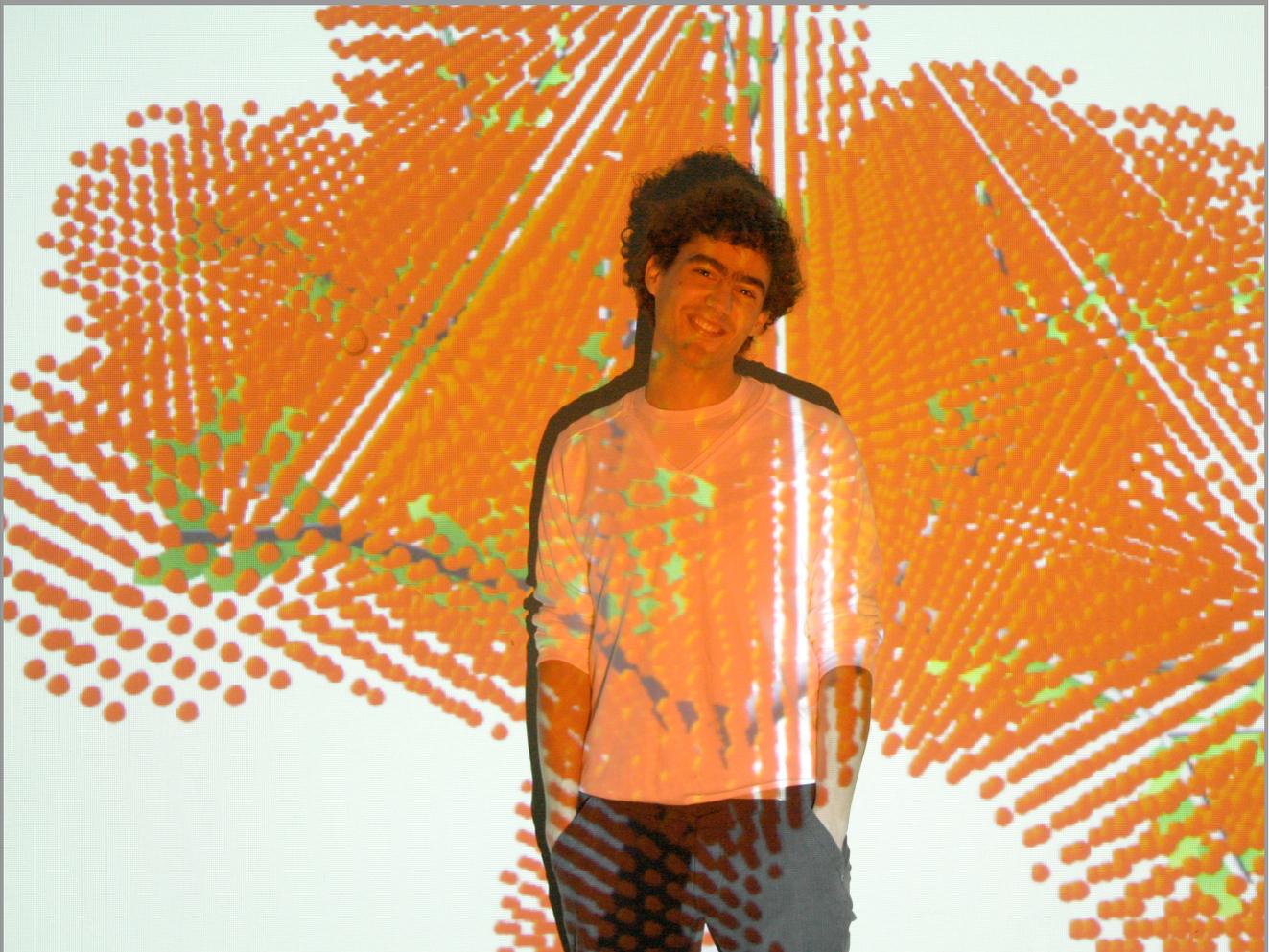


# Modulhandbuch Informatik (M.Sc.)

Sommersemester 2013  
Langfassung  
Stand: 15.02.2013

Fakultät für Informatik



Modul-ID	Modul	LP
IN4INALGVG	Algorithmen zur Visualisierung von Graphen	5
IN4INBI	Bioinformatics	6
IN4INAG	Algorithmische Geometrie	5
IN4INNWANA	Algorithmische Methoden zur Netzwerkanalyse	5
IN4INDAA	Design und Analyse von Algorithmen	10
IN4INGALA	Graphenalgorithmen und Lineare Algebra Hand in Hand	5
IN4INKO	Kombinatorische Optimierung	5
IN4INMPAR	Modelle der Parallelverarbeitung	5
IN4INNWA	Netzwerkalgorithmen	10
IN4INPAN	Parallele Algorithmen	5
IN4INGALGOP	Praktikum Algorithmentechnik	6
IN4INRAN	Randomisierte Algorithmen	5
IN4INALGTS	Seminar Algorithmentechnik	4
<b>VF 3: Kryptographie und Sicherheit</b>		
IN4INFKRYP	Fortgeschrittene Themen der Kryptographie	9
IN4INKT	Komplexitätstheorie	6
IN4INNTP	Netzwerk-Sicherheit – Theorie und Praxis	9
IN4INPAK	Praktische Aspekte der Kryptographie	6
IN4INSLP	Praktikum: Sicherheitslabor	6
IN4INSIS	Seminar: Sichere IT-Systeme	3
IN4INSICH	Sicherheit (Stammmodul)	6
IN4INTAK	Theoretische Aspekte der Kryptographie	6
<b>VF 4: Betriebssysteme</b>		
IN4INAKBP	Ausgewählte Kapitel der Betriebssystemprogrammierung	9
IN4INEBS	Energiebewusste Systeme	6
IN4INPSEI	Multi-Server Systeme	9
IN4INRS	Rechnerstrukturen (Stammmodul)	6
IN4INSEMBS	Seminar Betriebssysteme	3
<b>VF 5: Parallelverarbeitung</b>		
IN4INACA	Advanced Computer-Architecture	10
IN4INALGZELL	Algorithmen in Zellularautomaten	5
IN4INBI	Bioinformatics	6
IN4INMPAR	Modelle der Parallelverarbeitung	5
IN4INPAN	Parallele Algorithmen	5
IN4INPARRP	Parallelrechner und Parallelprogrammierung	4
IN4INPV	Parallelverarbeitung	9
IN4INRAN	Randomisierte Algorithmen	5
IN4INRS	Rechnerstrukturen (Stammmodul)	6
IN4INZFS	Seminar Zellularautomaten und diskrete komplexe Systeme für Fortgeschrittene	4
<b>VF 6: Softwaretechnik und Übersetzerbau</b>		
IN4INAKBP	Ausgewählte Kapitel der Betriebssystemprogrammierung	9
IN4INFON	Fortgeschrittene Objektorientierung	5
IN4INSLP	Praktikum: Sicherheitslabor	6
IN4INPSQ	Praktikum Software-Qualität auf Cloud-Grossrechner IBM z10	6
IN4INSQEP	Praktikum Software Quality Engineering mit Eclipse	6
IN4INPMKP	Praxis der Multikern-Programmierung	6
IN4INPFM	Projektgruppe Formale Methode der Softwareentwicklung	24
IN4INPMSE	Projektgruppe Modellgetriebene Softwareentwicklung	24
IN4INPGSE	Projektgruppe Softwareentwicklung auf dem Cloud-Großrechner IBM z10	24
IN4INSIS	Seminar: Sichere IT-Systeme	3
IN4INSEMSS	Seminar Software-Systeme	3
IN4INSEMSWT	Seminar Softwaretechnik	3
IN4INSWM	Software-Methodik	9
IN4INSWs	Software-Systeme	9
IN4INSWT2	Softwaretechnik II	6
IN4INSPt	Sprachtechnologien	10
IN4INCOMP1	Sprachtechnologie und Compiler	8

**Modul: Fortgeschrittene Objektorientierung [IN4INFON]**

**Koordination:** G. Snelting  
**Studiengang:** Informatik (M.Sc.)  
**Fach:** VF 6: Softwaretechnik und Übersetzerbau

<b>ECTS-Punkte</b>	<b>Zyklus</b>	<b>Dauer</b>
5	Jedes 2. Semester, Sommersemester	1

**Lehrveranstaltungen im Modul**

Nr.	Lehrveranstaltung	SWS V/Ü/T	Sem.	LP	Lehrveranstaltungs- verantwortliche
24665	Fortgeschrittene Objektorientierung (S. 449)	2/2	S	5	G. Snelting

**Erfolgskontrolle**

Die Erfolgskontrolle erfolgt in Form einer mündlichen Prüfung im Umfang von i.d.R. 15 Minuten gemäß § 4 Abs. 2 Nr. 2 SPO.

Die Modulnote ist die Note der mündlichen Prüfung.

**Bedingungen**

Keine.

**Empfehlungen**

Gute Java-Kenntnisse

**Lernziele**

Die Teilnehmer kennen Grundlagen verschiedener objektorientierter Sprachen (z.B. Java, C#, Smalltalk, Scala) Die Teilnehmer kennen Verhalten, Implementierung, Semantik und softwaretechnische Nutzung von Vererbung und dynamischer Bindung. Die Teilnehmer kennen innovative objektorientierte Sprachkonzepte (zB Generizität, Aspekte, Traits). Die Teilnehmer kennen theoretische Grundlagen (z.B. Typsysteme), softwaretechnische Werkzeuge (zB Refaktorisierung) und Verfahren zur Analyse von objektorientierten Programmen (z.B. Points-to Analyse). Die Teilnehmer haben einen Überblick über aktuelle Forschung im Bereich objektorientierter Programmierung.

**Inhalt**

- Verhalten und Semantik von dynamischer Bindung
- Implementierung von Einfach- und Mehrfachvererbung
- Generizität, Refaktorisierung
- Traits und Mixins, Virtuelle Klassen
- Cardelli-Typsystem
- Palsberg-Schwartzbach Typinferenz
- Call-Graph Analysen, Points-to Analysen
- operationale Semantik, Typsicherheit
- Bytecode, JVM, Bytecode Verifier, dynamische Compilierung

**Anmerkungen**

Dies ist keine Veranstaltung zur objektorientierten Softwareentwicklung! Vielmehr werden Kenntnisse in objektorientierter Softwaretechnik (z.B. Java, UML, Design Patterns) vorausgesetzt.

## Lehrveranstaltung: Fortgeschrittene Objektorientierung [24665]

**Koordinatoren:** G. Snelting  
**Teil folgender Module:** Fortgeschrittene Objektorientierung (S. 201)[IN4INFON]

ECTS-Punkte	SWS	Semester	Sprache
5	2/2	Sommersemester	de

### Erfolgskontrolle

Die Erfolgskontrolle wird in der Modulbeschreibung erläutert.

### Bedingungen

Dies ist keine Veranstaltung zur objektorientierten Softwareentwicklung! Vielmehr werden Kenntnisse in objektorientierter Softwaretechnik (z.B. Java, UML, Design Patterns) vorausgesetzt.

### Empfehlungen

Gute Java-Kenntnisse

### Lernziele

Die Teilnehmer kennen Grundlagen verschiedener objektorientierter Sprachen (z.B. Java, C#, Smalltalk, Scala)  
 Die Teilnehmer kennen Verhalten, Implementierung, Semantik und softwaretechnische Nutzung von Vererbung und dynamischer Bindung. Die Teilnehmer kennen innovative objektorientierte Sprachkonzepte (z.B. Generizität, Aspekte, Traits). Die Teilnehmer kennen theoretische Grundlagen (z.B. Typsysteme), softwaretechnische Werkzeuge (z.B. Refaktorisierung) und Verfahren zur Analyse von objektorientierten Programmen (z.B. Points-to Analyse). Die Teilnehmer haben einen Überblick über aktuelle Forschung im Bereich objektorientierter Programmierung.

### Inhalt

- Verhalten und Semantik von dynamischer Bindung
- Implementierung von Einfach- und Mehrfachvererbung
- Generizität, Refaktorisierung
- Traits und Mixins, Virtuelle Klassen
- Cardelli-Typsystem
- Palsberg-Schwartzbach Typinferenz
- Call-Graph Analysen, Points-to Analysen
- operationale Semantik, Typsicherheit
- Bytecode, JVM, Bytecode Verifier, dynamische Compilierung