

### Compilerbau

Kurzzeichen:	M_ComBau
Durchführungszeitraum:	FS 2017 - FS 2020
ECTS-Punkte:	4
Lernziele:	<ul style="list-style-type: none"><li>• Die Konzepte und Techniken der Compiler und Laufzeitsysteme für moderne objektorientierte Programmiersprachen verstehen.</li><li>• Eigene vollständige Compiler und Laufzeitsysteme für objektorientierte Programmiersprachen auf aktuellen Prozessor-Plattformen entwickeln können.</li><li>• Die Syntax und Semantik von modernen Programmiersprachen genauestens verstehen und auch spezifizieren können.</li></ul>
Verantwortliche Person:	Luc Bläser
Empfohlene Module:	.Net Technologien Betriebssysteme 1 Objektorientierte Programmierung Software-Engineering 2
Zusätzlich vorausgesetzte Kenntnisse:	keine
Modultyp:	Standard-Modul für Informatik (Bachelor 05) (Empfohlenes Semester: 6) Standard-Modul für Informatik (Bachelor 11) (Empfohlenes Semester: 6) Standard-Modul für Informatik (Bachelor 14) (Empfohlenes Semester: 6) Standard-Modul für Software Engineering (Bachelor 14)

### ECTS-Punkte pro Kategorie

Kategorie:	<b>Informatik (Bachelor 05)</b> Aufbau Informatik / 4 Punkte <b>Informatik (Bachelor 11)</b> Aufbau Informatik / 4 Punkte <b>Informatik (Bachelor 14)</b> Grundlagen Informatik und Aufbau Informatik / 4 Punkte Software Engineering (Bachelor 14)
------------	---

### Modulbewertung

Bewertungsart:	Note von 1 - 6
----------------	----------------

### Leistungsbewertung

Während des Semesters:	Bewertung des zu entwickelnden Compilers und Laufzeitsystems gemäss Übungsaufgaben. Die Bewertung setzt sich aus definierten Zwischen-Reviews und einer Schlussbewertung zusammen.
------------------------	--

Bewertungsart:	Note von 1 - 6
----------------	----------------

Bemerkungen:	<b>ACHTUNG: DIESES MODUL ERFORDERT VIEL AUFWAND IM SEMESTER! SEHR GUTE PROGRAMMIERFÄHIGKEITEN UND GENÜGEND ZEITRESERVEN WERDEN VORAUSSGESETZT.</b>
--------------	--

### Kurse in diesem Modul

#### Compilerbau

Kürzel:	ComBau
Lernziele:	Siehe oben
Plan und Lerninhalt:	<ul style="list-style-type: none"><li>• Programmiersprachspezifikation: Syntax und Semantik</li><li>• Architektur von Compilern und Laufzeitsystemen</li><li>• Lexer/Scanner</li><li>• Parser (LL vs. LR), inkl. Diskussion von Generator-Tools</li><li>• Zwischendarstellung (Symbol Table, Abstract Syntax Tree)</li><li>• Semantic Analysis: Symbolauflösung, Type Checking, Semantic Rules</li><li>• Code Generierung (für Zwischensprache und Intel 64 Prozessor)</li><li>• Zwischensprachen (stack-basiert)</li><li>• Virtuelle Maschine mit Loader und Interpreter</li></ul>

- Just-In-Time Compiler (JITer) für Intel 64 Prozessoren
- Elementare Laufzeitunterstützung: Stack und Heap Management
- OO-Laufzeitunterstützung: Vererbung und Type Tests, Virtual Tables und Dynamic Dispatch
- Garbage Collector: Mark And Sweep, Root Set, Diskussion fortgeschrittener GCs (Copy Collector, Incremental GC, Generational/Partitoned GC)
- Statische Analyse und Optimierung: Datenfluss-Analyse (je nach verfügbaren Vorlesungswochen im Semester)

Im Rahmen der Übungen entwickeln die Teilnehmer einen Compiler und eine Virtual Machine für eine C#-ähnliche Programmiersprache, inkl. Unterstützung der Objekt-Orientierung und Implementation eines GCs und Just-In-Time-Compilers.

Benötigte Technologien: .NET/C#, Intel 64 Assembler, Unmanaged .NET (evtl. auch C/C++)

Kursart:

Vorlesung mit 2 Lektionen pro Woche

Uebung mit 2 Lektionen pro Woche

Übergangsregelungen:

Compilerbau, Compilerbau, Compilerbau und Programmiermodelle

Beschreibung erzeugt: 2019-03-01 11:56:58  
Letzte Modüländerung: 2015-12-09 16:52:55  
Modul-Id: 28223  
Status: aktiviert