

Studiengang	Informatik-Diplom, Wirtschaftsinformatik-Diplom		
Modultitel	Nebenläufige Programmierung: Tasks und Threads		
Modulnummer	CS-PS-NPTT-06	01.04.07	
Art des Moduls (Pflicht-, Wahlpflicht- oder Wahlmodul)	Wahl- oder Wahlpflichtmodul Hauptstudium		
Modul-Verantwortlicher	<i>Prof. Dr. Jürgen F. H. Winkler</i>		
Leistungspunkte (ECTS credits)	9		
Arbeitsaufwand (work load):	Lehr- und Lernformen	SWS	Studentischer Arbeitsaufwand in Stunden
	Vorlesung	4	60
	Vor- und Nachbereitung der Vorlesung		45
	Übung zur Vorlesung	2	30
	Vor- und Nachbereitung der Übung		45
	Selbststudium		30
	Prüfung einschließlich Vorbereitung		60
	Lehrform(en) (V, Ü, S, P)	4V + 2Ü	
Häufigkeit des Angebots (Zyklus)	jährlich im Sommersemester		
Dauer des Moduls	1 Semester		
Voraussetzung für die Zulassung zum Modul	Vordiplom Informatik und Modul Programmiersprachen empfohlen		
Verwendbarkeit (Voraussetzung wofür)	ID, MD, WMD, BID, WID, IG, IM und andere Interessenten		
Voraussetzung für die Vergabe von Leistungspunkten (Prüfungsform)	Bearbeitung von Übungsblättern Prüfung (mündlich oder schriftlich)		
Inhalte	<ul style="list-style-type: none"> • Grundlegenden Phänomene der Nebenläufigkeit: mehrere TOCs (thread of control), zeitlich überlappende Zugriffe zu Objekten müssen koordiniert werden • nebenläufige Programmierung in Java und Ada • Implementierungsaspekte: welche Hardware-Unterstützung gibt es und welche weiteren Vorkehrungen müssen im BS / Laufzeitsystem getroffen werden? • Grundzüge der verteilten Systeme • Da das Verstehen nebenläufiger Abläufe ungewohnter und auch schwieriger ist als das sequentieller Abläufe, ist es sinnvoll, sich während des Studiums mit diesem Paradigma vertraut zu machen. Insbesondere, da Nebenläufigkeit in vielen Anwendungen in zunehmendem Maße vorkommt 		
(Qualifikations)Ziele	<ul style="list-style-type: none"> • Kenntnis der grundlegenden Besonderheiten und Probleme der Nebenläufigkeit • Kenntnis der wesentlichen Sprachelemente der NL Programmierung der Sprachen Ada und Java • Programmtechnische Realisierung einfacher Aufgabenstellungen mittels nebenläufiger Programmierung • Kenntnis der prinzipiellen Implementierung der NL-Sprachelemente • Kenntnis der Grundprinzipien der verteilten Systeme 		