

Lehrende/r	Dr. Sebastian Küpper	Modulverantwortliche/r	Dr. Sebastian Küpper
	Dauer des Moduls ein Semester	ECTS 10	Workload 300 Stunden
			Häufigkeit in jedem Semester
Lehrveranstaltung(en)	Moderne Programmier Techniken und -methoden		
Detaillierter Zeitaufwand	Bearbeitung des Lehrveranstaltungstextes: 150 Stunden Bearbeitung der Übungs- und Einsendeaufgaben: 75 Stunden Nachbearbeitung und Klausur- bzw. Prüfungsvorbereitung: 75 Stunden		
Qualifikationsziele	Die Studierenden verstehen die Ideen der interfacebasierter Programmierung und sind in der Lage, die interfacebasierte Programmierung anzuwenden. Sie wissen, was man unter Design by contract versteht und warum sie es anwenden sollten. Sie kennen die Vor- und Nachteile verschiedener Sprachen zur Umsetzung von Design by Contract. Sie wissen um die Nützlichkeit der Verifikation und können einfache Programmeigenschaften modellieren und grundlegende Verifikationstechniken nachvollziehen. Sie kennen die wichtigsten Entwurfsmuster und können sie bei der Software-Entwicklung verwenden. Sie wissen, was man unter Refactoring versteht und kennen verschiedene Refaktorisierungswerkzeuge und die Vorbedingungen für ihre Anwendbarkeit. Sie wissen, was man unter Metaprogrammierung versteht und können Vor- und Nachteile der Metaprogrammierung beurteilen. Sie wissen, wie die agilen Methoden wie das Extreme Programming funktionieren und welche Lösungen sie für die Probleme der Software-Entwicklung bieten.		
Inhalte	<p>Trotz eines zunehmenden Bewusstseins für die Notwendigkeit von vorbereitenden Tätigkeiten ist die Programmierung immer noch die Kernaktivität der Softwareentwicklung. Zwar hat die Einführung neuer Programmiersprachen wie Java durchaus Produktivitätssteigerungen bewirkt, jedoch ist die Fortentwicklung dieser Sprachen viel zu schwerfällig, um mit den ständig wachsenden Ansprüchen an Funktionalität und Umfang schritthalten zu können. Stattdessen haben sich im Kontext der objektorientierten Programmierung eine ganze Reihe von Techniken und Methoden entwickelt, mit deren Hilfe sich - auf der Basis existierender Programmiersprachen - die Softwareentwicklung effizienter gestalten lässt. Einige dieser Programmier Techniken und -methoden werden in dieser Lehrveranstaltung vorgestellt.</p> <p>Die Themen der Lehrveranstaltung sind:</p> <ul style="list-style-type: none"> - Interfacebasierte Programmierung - Design by contract - Modellierung zustandsbasierter Systeme - Verifikation via CTL Model Checking. - Entwurfsmuster (Design patterns) - Refactoring - Metaprogrammierung - Agile Softwareentwicklung (Extreme Programming) <p>Ergänzende Literatur:</p> <p>E.Gamma et al. Design Patterns (Addison-Wesley 1995). M. Fowler: Refactoring: Improving the Design of Existing Code (Addison-Wesley 1999), Refactoring (deutsche Version) (mitp Professional 31.10.2019) K. Beck, C. Andres: Extreme Programming Explained. Embrace Change 2. Ausgabe (Addison Wesley 2004).</p>		
Inhaltliche Voraussetzung	Es werden Programmierkenntnisse in Java vorausgesetzt, wie sie z. B. durch das Modul 63611 "Einführung in die objektorientierte Programmierung" vermittelt werden. Zusätzlich ist ein durch Praxis erworbenes Gefühl für die objektorientierte Programmierung Voraussetzung, um die Bedeutung der in der Lehrveranstaltung geschilderten Probleme und Lösungen abschätzen zu können. Weiterhin wird vorausgesetzt, dass Sie grundlegende Kenntnisse der Theoretischen		

Informatik, beispielsweise durch das Modul 63912 „Grundlagen der Theoretischen Informatik“ mitbringen.

Lehr- und
Betreuungsformen

Lehrveranstaltungsmaterial

internetgestütztes Diskussionsforum

Einsendeaufgaben mit Korrektur und/oder Musterlösung

Anmerkung

Verwendung des Moduls M.Sc. Data Science

M.Sc. Informatik

M.Sc. Praktische Informatik

M.Sc. Wirtschaftsinformatik

Prüfungsformen

Art der Prüfungsleistung

Voraussetzung

Prüfung

benotete zweistündige

Eine Zulassung zur Klausur erfolgt, wenn in 5 von 7 Einsendeaufgaben mindestens ein Punkt erreicht wurde.

Stellenwert der Note s. PO

Prüfungsklausur