

<b>Modulbezeichnung</b>	<b>Patterns und Frameworks</b>
<b>Modulbezeichnung (eng.)</b>	Patterns und Frameworks
<b>Semester (Häufigkeit)</b>	5 (jedes Wintersemester)
<b>ECTS-Punkte (Dauer)</b>	5 (1 Semester)
<b>Art</b>	Pflichtmodul
<b>Studentische Arbeitsbelastung</b>	26 h Kontaktzeit + 124 h Selbststudium
<b>Voraussetzungen (laut BPO)</b>	Für die erfolgreiche Teilnahme sind die Vorkenntnisse aus den Lehrveranstaltungen GP1, GP2, Internet-Server-Programmierung, Softwaretechnik, Betriebssysteme 1, Mensch-ComputerKommunikation und Datenbanken zwingend erforderlich.
<b>Empf. Voraussetzungen</b>	
<b>Verwendbarkeit</b>	BOMI, BOWI, MOMI
<b>Prüfungsform und -dauer</b>	Referat 30 min in Form eines Kolloquiums mit Vorstellung des Projekts und Diskussion
<b>Lehr- und Lernmethoden</b>	Multimedial aufbereitetes Online-Studienmodul zum Selbststudium mit zeitlich parallel laufender Hausarbeit sowie Präsenzphasen.
<b>Modulverantwortliche(r)</b>	N. Streekmann
<p><b>Qualifikationsziele</b> Die Studierenden können</p> <ul style="list-style-type: none"> <li>• ein Softwareprojekt mit vorgegebenen Anforderungen in der Programmiersprache Java selbstständig implementieren.</li> <li>• die vorgestellten Entwurfsmuster in der Anwendungsentwicklung erkennen und selbst bewusst einsetzen.</li> <li>• eine adäquate Softwarearchitektur unter Berücksichtigung der vorgestellten Architekturmuster und Frameworks entwerfen.</li> <li>• die vorgestellten Frameworks in einem Projekt gezielt einsetzen.</li> <li>• synchrone und asynchrone Kommunikation jeweils mittels verschiedener Ansätze (RMI/Web Services, Sockets/Web Sockets) in einer verteilten Java-Anwendung umsetzen und diesbezüglich Vor- und Nachteile erörtern.</li> <li>• komplexe Verarbeitungsprozesse in Java aufteilen, effizient parallelisieren und synchronisieren. Sie können diesbezüglich Vor- und Nachteile unterschiedlicher Ansätze erörtern.</li> </ul>	

## Lehrinhalte

### Einleitung

- Ziele und Struktur des Moduls
- Objektorientierung und UML-Klassendiagramm als Wiederholung
- Softwareprojekt als Prüfungsleistung Entwurfsmuster
- Erzeugungsmuster (Singleton, Fabrikmethode und Dependency Injection)
- Strukturmuster (Kompositum, Proxy, Adapter und Fassade)
- Verhaltensmuster (Beobachter, Strategie) Architekturmuster
- Schichtenarchitektur und MVC/MVVM
- Komponenten-basierte Architektur
- Microservices Verteilte Programmierung
- Synchrone und asynchrone Kommunikation im verteilten System
- Remote Method Invocation (RMI)
- SOAP Web Services
- REST Web Services
- Objekt-Relationales Mapping (ORM) und Spring Data
- Sockets und Web Sockets Frameworks
- Desktop-App mit JavaFX
- Mobile-App mit Android
- Web-App mit Angular Nebenläufige Programmierung
- Nebenläufigkeit und Parallelität
- Threads in Java
- Synchronisation von Threads
- Thread Pooling und Futures

## Literatur

Joshua Bloch: Effective Java - Best Practices für die Java-Plattform, dpunkt, 3. Aufl., 2018. Matthias Geirhos: Entwurfsmuster - Das umfassende Handbuch, Rheinwerk Computing, 2015. Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides: Entwurfsmuster - Elemente wiederverwendbarer objektorientierter Software, Addison-Wesley, 6. Aufl., 2010. Michael Inden: Der Java-Profi - Persistenzlösungen und REST Services, Datenaustauschformate, Datenbankentwicklung und verteilte Anwendungen, dpunkt, 2016. Michael Inden: Der Weg zum Java-Profi - Konzepte und Techniken für die professionelle Java-Entwicklung, dpunkt, 4. Aufl., 2017. Veikko Krypczyk, Olena Bochkor: Handbuch für Softwareentwickler: Das Standardwerk zu professionellem Software Engineering, Rheinwerk Computing, 2018. Christoph Kecher, Alexander Salvanos, Ralf Hoffmann-Elbern: UML 2.5 - Das umfassende Handbuch, Rheinwerk Computing, 6. Auflage, 2017. Bernhard Lahres, Gregor Rayman, Stefan Strich: Objektorientierte Programmierung - Das umfassende Handbuch. Prinzipien guter Objektorientierung auf den Punkt erklärt, Rheinwerk Computing, 4. Aufl., 2018. Robert Martin: Clean Code - Refactoring, Patterns, Testen und Techniken für sauberen Code, mitp, 2009. Robert Martin: Clean Architecture - Gute Softwarearchitekturen - Das Praxis-Handbuch für professionelles Softwaredesign. Regeln und Paradigmen für effiziente Softwarestrukturierung, mitp, 2018. Bernd Oestereich, Axel Scheithauer: Analyse und Design mit der UML 2.5 - Objektorientierte Softwareentwicklung, De Gruyter Oldenbourg, 11. Auflage, 2013. Kai Spichale: API-Design: Praxishandbuch für Java- und Webservice- Entwickler, dpunkt, 2016.

## Lehrveranstaltungen

Dozenten/-innen	Titel der Lehrveranstaltung	SWS
N. Streekmann	Pattern und Frameworks	4