

26 Patterns and Frameworks	
Patterns and Frameworks	
Semester	5
Dauer (Semester)	einsemestrig
Credit Points	5
Pflicht/ Wahlpflicht	Pflicht
Häufigkeit des Angebotes/ Verwendbarkeit	Jedes Semester, nach Bedarf der VFH-Hochschulen
Modulverantwortliche(r)	Prof. Dr. Jens Ehlers
Lerngebiet	Informatik Softwareentwicklung
Teilnahmevoraussetzungen	Der erfolgreiche Abschluss der Module Grundlagen der Programmierung I+II, Softwaretechnik und Datenbanken wird empfohlen.
Lernergebnisse	<p>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>
Prüfungsvorleistung	Hausarbeit/Projekt/Übung
Medien-/ Lernform	Multimedial aufbereitetes Online-Studienmodul zum Selbststudium mit zeitlich parallel laufender Online-Betreuung (E-Mail, Foren, Chat, Webkonferenzen, Einsendeaufgaben u. a.) sowie ggf. Präsenzphasen
Arbeitsaufwand	Prüfung: 30 Minuten Selbststudium: 111,5 h Betreutes Lernen: 26,5 h Vorbereitung PVL: 12 h
Präsenzinhalte	Inhaltliche Klärung; Vorstellung Lösungskonzept des Projekts

Prüfungsform	mündliche Prüfung/ Referat (30 min.)
Literatur	<p>Joshua Bloch: Effective Java - Best Practices für die Java-Plattform, dpunkt, 3. Aufl., 2018.</p> <p>Matthias Geirhos: Entwurfsmuster - Das umfassende Handbuch, Rheinwerk Computing, 2015.</p> <p>Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides: Entwurfsmuster - Elemente wiederverwendbarer objektorientierter Software, Addison-Wesley, 6. Aufl., 2010.</p> <p>Michael Inden: Der Java-Profi - Persistenzlösungen und REST-Services, Datenaustauschformate, Datenbankentwicklung und verteilte Anwendungen, dpunkt, 2016.</p> <p>Michael Inden: Der Weg zum Java-Profi - Konzepte und Techniken für die professionelle Java-Entwicklung, dpunkt, 4. Aufl., 2017.</p> <p>Veikko Krypczyk, Olena Bochkor: Handbuch für Softwareentwickler: Das Standardwerk zu professionellem Software Engineering, Rheinwerk Computing, 2018.</p> <p>Christoph Kecher, Alexander Salvanos, Ralf Hoffmann-Elbern: UML 2.5 - Das umfassende Handbuch, Rheinwerk Computing, 6. Auflage, 2017.</p> <p>Bernhard Lahres, Gregor Rayman, Stefan Strich: Objektorientierte Programmierung - Das umfassende Handbuch. Prinzipien guter Objektorientierung auf den Punkt erklärt, Rheinwerk Computing, 4. Aufl., 2018.</p> <p>Robert Martin: Clean Code - Refactoring, Patterns, Testen und Techniken für sauberen Code, mitp, 2009.</p> <p>Robert Martin: Clean Architecture - Gute Softwarearchitekturen - Das Praxis-Handbuch für professionelles Softwaredesign. Regeln und Paradigmen für effiziente Softwarestrukturierung, mitp, 2018.</p> <p>Bernd Oestereich, Axel Scheithauer: Analyse und Design mit der UML 2.5 - Objektorientierte Softwareentwicklung, De Gruyter Oldenbourg, 11. Auflage, 2013.</p> <p>Kai Spichale: API-Design: Praxishandbuch für Java- und Webservice-Entwickler, dpunkt, 2016.</p>
weitere Hinweise	Dieses Modul wird auf Deutsch angeboten

Studieninhalte	
Einleitung	<ul style="list-style-type: none"> <li>• Ziele und Struktur des Moduls</li> <li>• Objektorientierung und UML-Klassendiagramm als Wiederholung</li> <li>• Softwareprojekt als Prüfungsleistung</li> </ul>
Entwurfsmuster	<ul style="list-style-type: none"> <li>• Erzeugungsmuster (Singleton, Fabrikmethode und Dependency Injection)</li> </ul>

- 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