

Syllabus zu:

Modulnummer: B141, B142, I160, W124

Modulname: Wahlpflichtfach 1, 2 (Betriebswirtschaftslehre)
Wahlpflichtfach 1 (Wirtschaftsinformatik)
Wahlpflichtfach 1 (Angewandte Informatik)
Wahlpflichtfach 1 (Wirtschaftsingenieurwesen)

Reaktive Programmierung

ECTS 6

Workload: 180 Std. (Kontaktstudium: 60 Std., Selbststudium: 120 Std.)

Prüfungssemester 5

Englischer Name Reactive programming

Modulverantwortlicher: ???

Dozenten:	Johannes Brauer, Christopher Karow
E-Mail:	brauer@nordakademie.de , c.karow@nordakademie.de
Telefon:	0152 55732132 , ???
Sprechzeiten:	nach Vereinbarung
Vorlesungszeiten:	gemäß Stundenplan

Positionierung im Studiengang	<p>Es werden verschiedene Aspekte der Programmierung reaktiver Systeme behandelt, die in den software-bezogenen Pflichtmoduln kaum betrachtet werden. Reaktive Systeme sind ereignis-getriebene Anwendungen, die eine kontinuierliche Interaktion mit ihrer Umgebung bewerkstelligen und dabei Aufgaben wie die Aktualisierung des Anwendungszustands und der Anzeige von Daten erledigen müssen. Die interaktivste Komponente einer Anwendung ist gewöhnlich die Benutzungsoberfläche, die auf verschiedene Ereignisse wie Mausklicks, Tastatureingaben oder die Betätigung von Schaltflächen reagieren muss. Die Programmierung von Benutzungsoberflächen (insbesondere Web-Oberflächen) stellt ein Schwerpunktthema des Moduls dar.</p> <p>Reaktive Software-Systeme sind durch eine Reihe von Herausforderungen geprägt:</p> <ul style="list-style-type: none">▪ Sie sind inhärent nebenläufig. Sie müssen z.B. auf asynchron auftretende Ereignisse verschiedener Herkunft reagieren, oder müssen kontinuierlich verändernde Daten darstellen können. Die Synchronisation solcher, nebenläufiger Berechnungen ist komplex.▪ Ihr Zustand ändert sich mit dem Zeitablauf und häufig bestehen Abhängigkeiten zwischen Daten. Wenn ein Datum sich ändert, müssen alle abhängigen Daten aktualisiert werden. Dies „per Hand“ zu bewerkstelligen ist mühsam und fehleranfällig.▪ Die Kontrollstruktur ist invertiert: Die Anwendung steuert sich nicht selbst, sondern die Ermittlung der nächsten auszuführenden Berechnung wird durch externe Ereignisse oder Systeme bestimmt. Eine häufig anzutreffende Lösung ist die Bereitstellung von Routinen, sog. Rückruffunktionen (<i>callback functions</i>), durch die Anwendung, die beim Auftreten bestimmter Ereignisse aktiviert werden und in der Regel zustandsändernde Operationen ausführen. Man spricht hier gern von der „Callback-Hölle“, weil viele isolierte Programmfragmente dieselben Daten verändern.
-------------------------------	--

	Die geschilderten Probleme legen nahe, funktionale Programmiertechniken in Betracht zu ziehen.
Qualifikations- und Modulziele	Die Teilnehmer*Innen werden mit aktuellen Techniken der Reaktiven Programmierung vertraut gemacht und erhalten Gelegenheit anhand selbst gewählter Themen bestimmte Techniken zu vertiefen, aufzubereiten und angemessen zu präsentieren.
Lehr- und Lernmethoden:	Die Veranstaltung wird in Form - eines einführenden Vorlesungsteils durch die Dozenten und - von Referaten der Teilnehmer*Innen durchgeführt. Die Ausarbeitung eines Referats und seine Präsentation stellt eine Prüfungsvorleistung dar.
Lerninhalte	<ul style="list-style-type: none"> - Grundlagen reaktiver Systeme - Vergleich mit Observer-Pattern - Bedeutung des funktionalen Programmierparadigmas - Funktional-reaktive Programmierung unter Verwendung verschiedener Programmiersprachen, Bibliotheken und Frameworks, z. B. <ul style="list-style-type: none"> - Javascript/TypeScript (React, Vue, rxJs, Angular/NGRX) - ClojureScript (Reagent, re-frame) - Kotlin (RxKotlin, Coroutines) - Java (RxJava, Spring Reactor) - Scala (Akka) - Haskell (Fran, Yampa) - Racket (FrTime) <p>Das endgültige Themenspektrum wird zu Beginn der Veranstaltung festgelegt und kann auch auf Vorschläge der Teilnehmer hin ergänzt werden. Ausgewählte Themen werden dann durch Referate der Teilnehmer und (möglicherweise) durch externe Vortragende behandelt.</p>
Prüfungsvorleistung	Ausarbeitung eines Referats und seine Präsentation
Leistungsbewertung	Hausarbeit
Erforderliche Vorkenntnisse	Kenntnisse der objektorientierten und der funktionalen Programmierung
Vorbereitung	-
Literatur/Lehrmaterial	<p>Bainomugisha, E., Lombide Carreton, A., Van Cutsem, T., Mostinckx, S., De Meuter, W. 2013. A Survey on reactive programming. ACM Comput. Surv. 45, 4, Article 52 (August 2013), 34 pages.</p> <p>Stephen Blackheath Anthony Jones. 2016. Functional Reactive Programming. Manning Publications.</p> <p>Rambabu Posa. 2018. Scala Reactive Programming. Packt Publishing.</p> <p>Mario Arias, Rivu Chakraborty. 2018. Functional Kotlin. Packt Publishing.</p> <p>Roland Kuhn, Brian Hanafee, Jamie Allen. 2017. Reactive Design Patterns. 2017. Manning Publications Co.</p>
Verflechtung mit der betrieblichen Praxis	<p>Im Zuge einer Entwicklung weg von monolithischen Software-Systemen auf dedizierten Servern hin zu einer web-basierten Micro-Services-Architektur in Cloud-Umgebungen (z.B. OpenShift, AWS, Microsoft Azure) erhält die reaktive Programmierung einen immer höheren Stellenwert. Hierbei spielt vor allem die zunehmende „Containerisierung“ eine große Rolle, da Programme immer seltener mit garantierten Ressourcen (wie z.B. RAM und CPU) ausgeführt werden. Anstatt dessen teilen sich mehrere Container den vorhandenen Ressourcen-Pool, was die Effizienzanforderungen an Software erhöht.</p> <p>Bei reaktiver Programmierung können keine Aussagen über den Ausführungszeitpunkt getroffen werden, da allein der Scheduler entscheidet, wann welcher Code in welchem Thread ausgeführt wird. Dies ermöglicht reaktiver Software äußerst ressourcensparend zu arbeiten. Muss das Programm auf das Ergebnis von länger laufenden Anfragen bei Datenbanken oder anderen Micro-Services (z.B. HTTP) warten, kann während der Wartezeit teile anderer Anfragen bearbeitet werden. Sobald das Ergebnis vorliegt, wird die ursprüngliche</p>

	<p>Verarbeitung reaktiv fortgesetzt. Das Programm blockiert also keine Ressourcen sondern ist in der Lage, durchgängig auf Eingaben (z.B. vom Frontend) zu reagieren. Dies hat zur Folge, dass bei selben Ressourcenlimits, ein reaktives Programm meistens mehr Anfragen handhaben kann, als ein klassisches Programm.</p> <p>In der Praxis wird der steigende Stellenwert vor allem durch die immer größer werdende Zahl reaktiver Frameworks in den verschiedensten Programmiersprachen deutlich. Verbreitete Vertreter hierfür sind Spring Reactor (Java/Kotlin), rxJs (JavaScript/TypeScript), react (JavaScript) und rxJava. Neue Features bestehender Frameworks zielen oft auf die Kompatibilität mit reaktiven Anwendungen ab. Dazu zählt z.B. reaktives Caching oder reaktive Datenbanktreiber.</p>
Vorlesungs- und Prüfungssprache	deutsch
Anschlussmodule	-