

Gültig für 2022.HS

<b>Modulbezeichnung: Einführung Java-Programmierung</b>	
Modulkürzel	w.BA.XX.2JAVAPG.XX
Beschreibung des Moduls	In diesem Modul werden die Grundlagen für das prozedurale und objektorientierte Programmieren mit Java vermittelt. Die Studierenden werden in der algorithmischen und der objektorientierten Denkweise geschult und üben deren Anwendung. Hauptziel ist es anhand der erlernten Programmiersprache ein Verständnis für die Möglichkeiten und Herausforderungen der Softwareentwicklung zu erlangen.
Studiengang und Vertiefungsrichtung	<ul style="list-style-type: none"> <li>§ Betriebsökonomie - Vertiefung in Accounting, Controlling, Auditing</li> <li>§ Betriebsökonomie - Vertiefung in Banking and Finance</li> <li>§ Betriebsökonomie - Vertiefung in Banking and Finance (FLEX)</li> <li>§ Betriebsökonomie - Vertiefung in Banking and Finance (PIE)</li> <li>§ Betriebsökonomie - Vertiefung in Behavioral Design</li> <li>§ Betriebsökonomie - Vertiefung in Economics and Politics</li> <li>§ Betriebsökonomie - Vertiefung in General Management</li> <li>§ Betriebsökonomie - Vertiefung in General Management (Flex)</li> <li>§ Betriebsökonomie - Vertiefung in Marketing</li> <li>§ Betriebsökonomie - Vertiefung in Risk and Insurance</li> </ul>
Rechtliche Grundlagen	Studienordnung BSc vom 29.01.2009, Anhang zur Studienordnung für die Bachelorstudiengänge Betriebsökonomie, Wirtschaftsinformatik und Wirtschaftsrecht erstmals beschlossen am 12.05.2009
Modulkategorie	<b>Modultyp:</b> Wahlpflichtmodul <span style="float: right;"><b>Studienabschnitt:</b> Hauptstudium</span>
ECTS	3
Verantwortliche OE	W Institut für Wirtschaftsinformatik
Modulverantwortung	David Grünert (grud)
Modulverantwortung Stellvertretung	-
Spezifische Vorkenntnisse	-
Beitrag des Moduls für Learning Goals des Studiengangs (durch das Modul betroffene)	<ul style="list-style-type: none"> <li>§ Fachkompetenz</li> <li>§ Methodenkompetenz</li> <li>§ Sozialkompetenz</li> <li>§ Selbstkompetenz</li> </ul>
Beitrag des Moduls für Learning Objectives des Studiengangs	<p>Fachkompetenz</p> <ul style="list-style-type: none"> <li>§ Theorie- &amp; Praxisrelevante Fachinhalte wissen &amp; verstehen</li> <li>§ Theorie- &amp; Praxisrelevante Fachinhalte anwenden, analysieren und verknüpfen</li> <li>§ Theorie- &amp; Praxisrelevante Fachinhalte evaluieren</li> </ul> <p>Methodenkompetenz</p> <ul style="list-style-type: none"> <li>§ Problemlösung &amp; Kritisches Denken</li> <li>§ Wissenschaftliche Methoden</li> <li>§ Arbeitsmethoden, -techniken &amp; -verfahren</li> <li>§ Nutzung von Informationen</li> <li>§ Kreativität &amp; Innovation</li> </ul> <p>Sozialkompetenz</p> <ul style="list-style-type: none"> <li>§ Schriftliche Kommunikation</li> <li>§ Mündliche Kommunikation</li> <li>§ Interkulturalität &amp; Perspektivenübernahme</li> </ul> <p>Selbstkompetenz</p> <ul style="list-style-type: none"> <li>§ Selbstmanagement &amp; Selbstreflexion</li> <li>§ Lernen &amp; Veränderung</li> </ul>
Lernziele des Moduls	<p>Die Studierenden...</p> <ul style="list-style-type: none"> <li>§ verfügen über die Grundlagen im Bereich prozeduraler Programmierung.</li> <li>§ verfügen über die Grundlagen im Bereich objektorientierter Programmierung.</li> <li>§ beurteilen und lesen Anforderungen an Software.</li> <li>§ entwerfen zu Anforderungen Algorithmen.</li> <li>§ übersetzen Algorithmen in ein Java-Programm.</li> <li>§ können selbständig ihre Fähigkeiten zum Programmieren in Java erweitern.</li> <li>§ können sich einfacher in eine andere Programmiersprache einarbeiten.</li> </ul>

Inhalte des Moduls	§ Datenstrukturen: Variablen, Listen, Arrays § Prozedurales Programmieren: Variablen, Zuweisungen, Operationen, Bedingungen, Schleifen, Methoden, Rückgabewerte, Argumente § Objektorientiertes Programmieren: Klassen, Klassenattribute, Klassenmethoden, Objekte, Zeiger, Assoziationen. § Programmieren reaktiver Software: Kommandozeilen, GUI § Toolkenntnis: Programmierumgebung (Eclipse)		
Verknüpfung zu anderen Modulen	-		
Unterrichtsmethoden	§ Anwendungsaufgaben § Übungen § Problemorientierter Unterricht § Projektarbeit § Online-Kurs ohne Präsenzunterricht	<b>Eingesetzte Sozialformen:</b> Einzelarbeit	
Digitale Lernressourcen	§ Reader § Lehrvideos § Übungs- und Anwendungsaufgaben (inkl. Lösungen) § (Multiple-Choice)-Tests		
Unterrichtsgliederung	<b>Kontaktstudium</b>	<b>Begleitetes Selbststudium</b>	<b>Autonomes Selbststudium</b>
Grossklasse	-	-	
Kleinklasse	-	-	
Gruppenunterricht	-	-	
Praktikum	-	90 h	
Seminar	-	-	
<b>Total</b>	<b>0 h</b>	<b>90 h</b>	<b>0 h</b>
Leistungsnachweise			
<b>Modulendprüfung</b>	<b>Form</b>	<b>Dauer (Min.)</b>	<b>Gewichtung</b>
Schriftliche Prüfung	Spezifizierte Unterlagen	60	70,00 %
Hilfsmittel	kein Taschenrechner	mit Diktionär	
<b>Andere</b>	<b>Bewertung</b>	<b>Dauer (Min.)</b>	<b>Gewichtung</b>
Drei Programmierprojekte (Praxisaufgaben). Abgabetermine gemäss Semesterprogramm. Jedes Projekt zählt zu 10%	Note	-	30,00 %
Theorieaufgaben. Bedingung: Mindestens 80% der Punkte erhalten. Abgabetermine gemäss Semesterprogramm	Pass/Fail	-	-
Programmieraufgaben. Bedingung: Mindestens 80% der Punkte erhalten. Abgabetermine gemäss Semesterprogramm	Pass/Fail	-	-
Präsenzverpflichtung im Kontaktstudium	Zwingende Präsenzzeit: Keine		
Unterrichts- und Prüfungssprache	Deutsch		
Pfichtliteratur	-		
Ergänzende Literatur	-		
Bemerkungen	<ul style="list-style-type: none"> <li>Das Modul ist als reiner Online-Kurs konzipiert und wird ohne Präsenzunterricht durchgeführt. Der Kurs umfasst 44 Lektionen bestehend aus einem Theorieteil und einem Aufgabenteil. Die Theorie wird mittels Videos, Text und interaktiven Beispielen vermittelt. Der Aufgabenteil umfasst sowohl Theorie- wie auch Programmieraufgaben.</li> <li>Während dem Kurs werden drei Projekte (Praxisaufgaben) realisiert und eingereicht. Diese Projekte bilden zusammen mit der schriftlichen Prüfung den notenrelevanten Teil des Leistungsnachweises. Darüber hinaus müssen bei den Theorie- und Programmieraufgaben 80% der Punkte erreicht werden. Details zum Leistungsnachweis werden im Semesterplan kommuniziert.</li> </ul>		