

Masterprogramm „Management Information Systems“ (MIS)

Für das Masterprogramm „Management Information Systems“ sind gemäß der aktualisierten Aufnahmeordnung vom 19. Oktober 2022 für die Aufnahme folgende Zulassungskriterien in § 1 Absatz 1 festgelegt:

A) Fachliche Voraussetzungen

a. Ein **erster berufsqualifizierender Hochschulabschluss** in einem der folgenden Studiengänge:

- Wirtschaftsinformatik
- Informatik
- Betriebswirtschaftslehre
- Wirtschaftswissenschaft
- Digitale Medien
- Wirtschaftsingenieurwesen

oder in einem Studiengang, der keine wesentlichen Unterschiede in Inhalt, Umfang und Anforderungen zu den vorgenannten erkennen lässt, mit Studienleistungen im Umfang von mindestens 180 Leistungspunkten (Credit Points = CP) nach dem European Credit Transfer and Accumulation System (ECTS), oder Leistungen, die keine wesentlichen Unterschiede in Inhalt, Umfang und Anforderungen zu jenen erkennen lassen.

b. Der Nachweis von mindestens **12 CP in den Bereichen praktische Informatik** und **12 CP betriebswirtschaftliche Grundlagen**, die in einem vorherigen Studium erbracht worden sind.

B) Sprachliche Voraussetzungen

c. Zum **Zeitpunkt der Bewerbung** müssen **Englisch-Sprachkenntnisse**, die mindestens dem **Niveau C1** des Gemeinsamen Europäischen Referenzrahmens für Sprachen (GER) entsprechen, nachgewiesen werden. Der Sprachnachweis kann entfallen, wenn Bewerberinnen und Bewerber ihre Hochschulzugangsberechtigung oder den letzten Hochschulabschluss in englischer Sprache erworben haben.

Wie können die Fachkenntnisse nachgewiesen werden?

Zu den geforderten fachlichen Voraussetzungen zählen jeweils 12 CP aus den Bereichen praktische Informatik und betriebswirtschaftliche Grundlagen. Werden diese Voraussetzung nicht über das Fachcurriculum des Erststudiums erfüllt, so besteht die Möglichkeit, dass diese über **Zusatzleistungen im Rahmen des Erststudiums** erbracht werden. Studierenden der Universität Bremen stehen zum Erwerb dieser Studienanteile folgende Module zur Verfügung:

Erwerb 12 CP praktische Informatik

Am Fachbereich 3 Mathematik/Informatik können Studierende diese CP über den erfolgreichen Abschluss folgender Module erwerben:

Option 1: Wintersemester & Sommersemester

	Wintersemester	Sommersemester
Variante 1 (aus dem Vollfach Informatik bzw. Wirtschaftsinformatik)	Modul Praktische Informatik 1 (reduziert auf 6 CP)	Modul Praktische Informatik 2 (6 CP)
Variante 2 (aus dem Komplementärfach Informatik bzw. Digitale Medien)	Modul Praktische Informatik 1: LV „Grundlagen der Programmierung“ (6 CP)	Modul Praktische Informatik 2: LV „Objektorientierte Programmierung“ (3 CP) und (!) „Algorithmen und Datenstrukturen“ (6 CP)

Das Lehrangebot beider Varianten startet jährlich im WiSe und wird im SoSe abgeschlossen. Die entsprechenden Modulbeschreibungen finden Sie in den Anlagen 1 und 2.

Option 2: Sommersemester & Wintersemester

Sommersemester	Wintersemester
(aus Voll- und Komplementärfach Informatik:) LV „Datenbankgrundlagen und Modellierung“ (VAK 03-IBGP-DBM)	(aus dem Komplementärfach Informatik:) LV „Grundlagen der Programmierung“ (VAK 03-DMB-MI-21-GDP)

Die Modulbeschreibungen für Option 2 finden Sie in der Anlage 3.

(Rückfragen: Studienzentrum FB 3 Informatik, sz@uni-bremen.de)

Erwerb 12 CP betriebswirtschaftliche Grundlagen

Am Fachbereich 7 Wirtschaftswissenschaft können Studierende die erforderlichen CP über den erfolgreichen Abschluss der folgenden Module erwerben, diese sind teilweise auch für den Bereich der Fachergänzenden Studien der Universität Bremen freigeschaltet. Angebote aus dem Bereich der eGeneral Studies können in jedem Semester über das ZMML absolviert werden.

Angebote aus dem FB 7 (je 6 CP):

Angebote im Wintersemester	Angebote im Sommersemester
Rechnungswesen und Abschluss	Marketing
Personal und Organisation	Finanzierung und Investition
Unternehmensbesteuerung	Wertschöpfungsprozesse
International Business (WiSe ab Studienjahr 2024/25)	<i>International Business</i> (<i>letztmalig im SoSe 2024</i>)

Die aktuellen Modulbeschreibungen können dem Modulhandbuch zu den Bachelorstudiengängen BWL& WiWi entnommen werden:

<https://www.uni-bremen.de/wiwi/studium/downloads>

eGeneral Studies Angebote (je 3 CP):

Einführung in die Betriebswirtschaftslehre
Nachhaltiges Management
Projektmanagement

Beschreibungen zu diesen Fächern finden Sie auf der Internetseite

<https://oncourse.uni-bremen.de/>

(Rückfragen: Studienzentrum FB 7 stzwiwi@uni-bremen.de)

Anlagen:

- Anlage 1 – Praktische Informatik Variante 1
- Anlage 2 – Praktische Informatik Variante 2
- Anlage 3 – Praktische Informatik Option 2 (SoSe & WiSe)

Anlage 1

Variante 1: aus VF Informatik (12 vs. 15CP):

- Praktische Informatik 1 (9CP, Reduktion auf 6CP möglich)
 - PI1: Imperative Programmierung und Objektorientierung
- Praktische Informatik 2 (6CP)
 - PI2: Algorithmen und Datenstrukturen

Praktische Informatik 1							Modulkennziffer:		
<i>Practical Computer Science 1</i>							IBGP-PI1		
Bachelor									
Pflicht <input checked="" type="checkbox"/>									
Wahlpflicht <input type="checkbox"/>									
Studienabschnitt: Grundlagen Praktische und Technische Informatik									
Anzahl der SWS	V	UE	K	S	Prak.	Proj.	Σ	Kreditpunkte: 9	Turnus angeboten in jedem WiSe
	4	0	0	0	4	0	8		
Formale Voraussetzungen: -									
Inhaltliche Voraussetzungen: -									
Vorgesehenes Semester: 1. Semester									
Sprache: Deutsch									
Ziele: <ul style="list-style-type: none"> • Grundlegende Informatikkonzepte wiedergeben und erklären können. • Konzepte einer imperativen Programmiersprache kennen, verstehen und anwenden können. • Anschauliche Sachverhalte im Modell der Objektorientierung ausdrücken können. • Einfache Algorithmen entwickeln und in Java umsetzen können. • Einfache in Java realisierte Algorithmen systematisch testen können. • Probleme in Teilprobleme zerlegen und diese Strukturierung mit Mitteln von Java umsetzen und aussagekräftig dokumentieren können. • Formale Syntaxbeschreibungen verstehen können. • Eine einfache Entwicklungsumgebung nutzen können. • LaTeX zur Erstellung einfacher Dokumente nutzen können. • Versionsverwaltungssysteme einsetzen können. • In Gruppen Probleme analysieren und gemeinsam Lösungsstrategien entwickeln und präsentieren können. <p>Die Vorlesungen Praktische Informatik 1 und 2 vermitteln essenzielles Grundwissen und Basisfähigkeiten, deren Beherrschung für nahezu jede vertiefte Beschäftigung mit Informatik – sowohl in der industriellen Anwendung, als auch in der Forschung – Voraussetzung ist.</p>									

Inhalte:

1. Basiswissen: von Neumannsche Rechnerorganisation – Grundlagen der Rechnerarchitektur – Programm und Prozess – Programmiersprachen – Compiler, Assembler, Loader, Linker, Interpreter, Laufzeitumgebungen, Betriebssysteme – Grafische Benutzungsschnittstellen
2. Datenstrukturen: Information und ihre Repräsentation – Datentypen und Typanalyse – Elementare und zusammengesetzte Datentypen – rekursive Datentypen – Kanonische Operationen auf den eingeführten Datenstrukturen
3. Programmierparadigmen: (1) Imperative und funktionale Programmierung, (2) Objektorientierte (imperative) Programmierung, (3) Sequenzielle Programme versus nebenläufige Programme
4. Grundkomponenten imperativer Programmiersprachen: Schnittstellen und Ein-/Ausgabe, Variablen und Zuweisungen, Kontrollstrukturen, Blöcke, Funktionen, Rekursion
5. Syntax und Semantik imperativer Programmiersprachen: Syntax und Methoden der Syntax-Spezifikation, reguläre Ausdrücke, (erweiterte) Backus-Naur-Form (E)BNF
6. Prinzipien der objektorientierten Programmierung: Geheimnisprinzip – Methoden – Operationen – Objekte – Klassen – Botschaften – Ereignisverarbeitung – Attribute – Vererbung – Polymorphismus – Überladung – Generische Datentypen
7. Umsetzung der Punkte 2.-6. mit Java – Illustration anhand einfacher Algorithmen
8. Programmdokumentation und zugehörige Hilfswerkzeuge, z.B. JavaDoc
9. Testen von Programmen und zugehörige Hilfswerkzeuge, z.B. JUnit
10. Grundlagen der Netzwerkkommunikation: IP-Adressen, DNS, TCP, UDP
11. Grundkonzepte der Entwicklung graphischer Oberflächen

Programmier-Praktikum: Programmentwicklung in Java – Realisierung einzelner, überschaubarer Programmieraufgaben

Lehrveranstaltung(en):

- 03-IBGP-PI1 Praktische Informatik 1: Imperative Programmierung und Objektorientierung

Unterlagen (Skripte, Literatur, Programme usw.):

- David J. Barnes, Michael Kölling: Java lernen mit BlueJ - Objects first - Eine Einführung in Java. Aktuelle Auflage. Pearson Studium.

Weitere Informationen (Beispielprogramme, Musterlösungen, im WWW verfügbare Literatur) sind auf der Web-Seite der Veranstaltung zu finden.

Form der Prüfung:

KP; PL1: 70%, PL2: 30%; Portfolio, Klausur

Arbeitsaufwand	Präsenz	112 h
	Übungsbetrieb/Prüfungsvorbereitung	158 h
	Summe	270 h

Lehrende:

Dr. T. Röfer, N.N.

Verantwortlich:

Prof. Dr. U. Bormann

Praktische Informatik 2 <i>Practical Computer Science 2</i>							Modulkennziffer: IBGP-PI2		
Bachelor Pflicht <input checked="" type="checkbox"/> Wahlpflicht <input type="checkbox"/>									
Studienabschnitt: Grundlagen Praktische und Technische Informatik									
Anzahl der SWS	V 2	UE 2	K 0	S 0	Prak. 0	Proj. 0	Σ 4	Kreditpunkte: 6	Turnus angeboten in jedem SoSe
Formale Voraussetzungen: -									
Inhaltliche Voraussetzungen: Praktische Informatik 1									
Vorgesehenes Semester: 2. Semester									
Sprache: Deutsch									
<p>Ziele:</p> <ul style="list-style-type: none"> • Typische Datenstrukturen identifizieren und problemadäquat einsetzen können. • Datenstrukturen und Algorithmen in Java umsetzen können. • Wesentliche Algorithmen der Informatik erklären, anwenden und modifizieren können. • Algorithmische Alternativen bezüglich der Eignung für ein Problem beurteilen können. • Grundbegriffe der formalen Verifikation erläutern können. • Die Komplexität von einfachen Algorithmen analysieren können. • Eine komplexe Entwicklungsumgebung nutzen können. • Generische und funktionale Konzepte in eigenen Programmen einsetzen können. • In Gruppen Probleme analysieren und gemeinsam Lösungsstrategien entwickeln und präsentieren können. <p>Die Vorlesungen Praktische Informatik 1 und 2 vermitteln essenzielles Grundwissen und Basisfähigkeiten, deren Beherrschung für nahezu jede vertiefte Beschäftigung mit Informatik – sowohl in der industriellen Anwendung, als auch in der Forschung – Voraussetzung ist.</p>									
<p>Inhalte:</p> <ol style="list-style-type: none"> 1. Algorithmen: Begriff des Algorithmus – Beschreibung von Algorithmen – Algorithmische Umsetzung kanonischer Operationen auf Datenstrukturen – Grundlegende Strategien: Greedy, Divide-and-Conquer, Backtracking, dynamische Programmierung 2. Komplexität von Algorithmen – $O(n)$-Notation und asymptotische Analyse 3. Suchen und Sortieren auf Arrays: Binäre Suche – Quicksort und weitere Sortieralgorithmen – Komplexitätsvergleiche 4. Mengen – Multimengen – Relationen – Funktionen: Datenstrukturen und Algorithmen zur Realisierung kanonischer Operationen (z.B. Mengenalgebra) 5. Listen – Stapel – Warteschlangen: Datenstrukturen zur Realisierung (Arrays versus Verkettung und dynamische Speicherallokation für Elemente), Algorithmen zur Realisierung kanonischer Operationen (Listentraversalion, Anfügen, Einfügen, Löschen, Suchen, Stack-Operationen, FIFO-Warteschlangenoperationen) 6. Bäume: Binäre Bäume, AVL-Bäume, Rot-Schwarz-Bäume, B-Bäume – Suchen, Einfügen, Löschen, Traversalion 7. Hashing: Hash-Array, Hashfunktion, Hash Buckets, offenes Hashing 8. Graphen: ungerichtete, gerichtete, gewichtete Graphen – Repräsentation durch Knoten- und Kantenlisten, durch Adjazenzmatrizen, Adjazenzlisten – Algorithmen auf Graphen: Breitensuche, Tiefensuche, kürzeste Wege auf gewichteten Graphen: Dijkstras Algorithmus, minimal aufspannende Bäume: Algorithmen von Prim et al. und Kruskal 9. Spezifikation von Programmen: Vor- und Nachbedingungen – Invarianten 10. Verifikation: Partielle und totale Korrektheit sequenzieller Programme – Formale Verifikation, z.B. Hoare Logik (Pre-/Postconditions) – Eigenschaftsbeweis durch Strukturelle Induktion <p>Lehrveranstaltung(en):</p> <ul style="list-style-type: none"> • 03-IBGP-PI2 Praktische Informatik 2: Algorithmen und Datenstrukturen 									

Unterlagen (Skripte, Literatur, Programme usw.):

- G. Saake und K.-U. Sattler: Algorithmen und Datenstrukturen. dpunkt.verlag, Heidelberg (2004)
- R. Schiedermeier: Programmieren mit Java. Pearson, München (2005)

Weitere Informationen (Beispielprogramme, Musterlösungen, im WWW verfügbare Literatur) sind auf der Web-Seite der Veranstaltung zu finden.

Form der Prüfung:

KP, PL1: 70%, PL2: 30%, Portfolio, Klausur

Arbeitsaufwand	Präsenz	56 h
	Übungsbetrieb/Prüfungsvorbereitung	124 h
	Summe	180 h

Lehrende:
Dr. T. Röfer, N.N.

Verantwortlich:
Prof. Dr. U. Bormann

Anlage 2

- Variante 2: aus KF Informatik (15CP):**
- Praktische Informatik 1 (6CP):
 - Grundlagen der Programmierung (GdP, 6CP)
 - Praktische Informatik 2 (9CP)
 - Objektorientierte Programmierung (OOP, 3CP)
 - Algorithmen und Datenstrukturen (AuD, 6CP)

Praktische Informatik 1							Modulkennziffer:	
<i>Practical Computer Science 1</i>							KINF-P1	
Bachelor								
Pflicht <input checked="" type="checkbox"/>								
Wahlpflicht <input type="checkbox"/>								
Studienabschnitt: (Komplementärfach Informatik)								
Anzahl der SWS	V	UE	K	S	Prak.	Proj.	Σ	Kreditpunkte: 6
	2	2	0	0	0	0	4	Turnus angeboten in jedem WiSe
Formale Voraussetzungen: -								
Inhaltliche Voraussetzungen: -								
Vorgesehenes Semester: ab 1. Semester								
Sprache: Deutsch								
Ziele: Die Studierenden								
<ul style="list-style-type: none"> • beherrschen grundlegende Konzepte des imperativen und objektorientierten Programmierens • können graphisch-interaktive Programme in der Programmierumgebung Processing, welche auf der aktuell weit verbreiteten Programmiersprache JAVA basiert, entwickeln • können selbstständig kreative Ideen in Entwurfskonzepte und Programme überführen und dabei auch Medien wie Bild und Ton angemessen einbetten • können spezifische Probleme in Teilprobleme zerlegen und diese Strukturierung mit Mitteln von Processing/Java umsetzen und aussagekräftig dokumentieren • beherrschen die Erstellung und Bearbeitung größerer, komplexer Programme mit einem Team von mehreren Personen • verstehen typische Denkweisen der Informatik, um in interdisziplinären Projekten mit Informatikern kommunizieren zu können • sind in der Lage, ihr Vorgehen im größeren Kontext der Informatik einzuordnen 								
Inhalte: .								
Grundlagen der Programmierung:								
<ul style="list-style-type: none"> • Variablen • Bedingte Anweisungen • Schleifen • Mathematische Formeln in Programmen • Funktionen und Rekursion • Verwendung von Objekten und Klassen, Grundlagen der Vererbung • Arrays (veränderlicher Größe und mehrerer Dimensionen) • Kommentare in Programmen 								
Die Programmierumgebung Processing:								
<ul style="list-style-type: none"> • Grafik und Interaktion • Einbettung von Medien (Bild, Ton, Video) • Verwendung von Zufallsfunktionen (Perlin-Noise und lineare Zufallsverteilungen) • Methoden des Debuggings 								

Inhalte 2: .

Ausgewählte Aspekte der Informatik:

- Grundlagen des maschinellen Rechnens
- Grundlagen der Rechnerarchitektur
- Programm und Prozess
- Programmierparadigmen und Programmiersprachen (inkl. Einordnung von Processing/JAVA)
- Zusammenhänge und Funktion von Compiler, Assembler, Loader, Linker, Interpreter, Laufzeitumgebung
- Begriff des Algorithmus

Lehrveranstaltung(en)

03-B-MI-21 Grundlagen der Programmierung

Unterlagen (Skripte, Literatur, Programme usw.):

- Daniel Shiffman: "Learning Processing: A Beginner's Guide to Programming Images, Animation, and Interaction", The Morgan Kaufmann Series in Computer Graphics, 2015. Quellen im Internet:
 - <http://learningprocessing.com> (Beispiele und Videos zum Buch)
 - <https://processing.org> (Referenz und Tutorials)

Form der Prüfung:

KP; PL1: xx%, PL2: xx%; Portfolio, Fachgespräch

Arbeitsaufwand	Präsenz	56 h
	Übungsbetrieb/Prüfungsvorbereitung	124 h
	Summe	180 h

Lehrende:
Dr. Tim Laue

Verantwortlich:
Dr. Tim Laue

Praktische Informatik 2 <i>Practical Computer Science 2</i>							Modulkennziffer: KINF-P2		
Bachelor Pflicht <input checked="" type="checkbox"/> Wahlpflicht <input type="checkbox"/>									
Studienabschnitt: (Komplementärfach Informatik)									
Anzahl der SWS	V	UE	K	S	Prak.	Proj.	Σ	Kreditpunkte: 9	Turnus angeboten in jedem SoSe
	0	0	6	0	0	0	6		
Formale Voraussetzungen: -									
Inhaltliche Voraussetzungen: Grundlagen der Programmierung									
Vorgesehenes Semester: 2. Semester									
Sprache: Deutsch									
Ziele: <ul style="list-style-type: none"> • Grundlegende Konzepte der objektorientierten Programmierung kennen, verstehen und anwenden können. • Anschauliche Sachverhalte im Modell der Objektorientierung ausdrücken können. • Eine einfache Entwicklungsumgebung nutzen können. • LaTeX zur Erstellung einfacher Dokumente nutzen können. • Versionsverwaltungssysteme verstehen und einsetzen können. • Datenstrukturen und Algorithmen in Java umsetzen können • Fehler unter Einsatz eines einfachen Debuggers finden können. • Einfache Komponententests zur Qualitätssicherung erstellen und durchführen können. • Ein Softwaredokumentationswerkzeug verwenden können. • Typische Datenstrukturen identifizieren und problemadäquat einsetzen können. • Wesentliche Algorithmen der Informatik erklären, anwenden und modifizieren können. • Algorithmische Alternativen bezüglich der Eignung für ein Problem beurteilen können. • Die Komplexität von einfachen Algorithmen analysieren können. • In Gruppen Probleme analysieren und gemeinsam Lösungsstrategien entwickeln und präsentieren können. 									

Inhalte:

1. Prinzipien der objektorientierten Programmierung: Geheimnisprinzip – Methoden – Operationen – Objekte – Klassen – Botschaften – Ereignisverarbeitung – Attribute – Vererbung – Polymorphismus – Überladung – Generische Datentypen – Interfaces
2. Datenstrukturen: Information und ihre Repräsentation – Datentypen und Typanalyse – Elementare und zusammengesetzte Datentypen – rekursive Datentypen
3. Fehlervermeidung: Exceptions
4. Dokumentation von Klassen, Methoden und Attributen
5. Automatisierte Komponententests
6. Fehlersuche (Debugging): Breakpoint – schrittweise Ausführung – Stacktrace
7. Umsetzung der Punkte 1.-6. mit Java, Javadoc und JUnit
8. Algorithmen: Begriff des Algorithmus – Beschreibung von Algorithmen – Algorithmische Umsetzung kanonischer Operationen auf Datenstrukturen – Grundlegende Strategien: Greedy, Divide-and-Conquer, Backtracking, dynamische Programmierung, zufallsgesteuerte Algorithmen, genetische Algorithmen, heuristische Algorithmen, probabilistische Algorithmen
9. Komplexität von Algorithmen – $O(n)$ -Notation und asymptotische Analyse
10. Suchen und Sortieren auf Arrays: Binäre Suche – Quicksort und weitere Sortieralgorithmen – Komplexitätsvergleiche
11. Listen – Stapel – Warteschlangen: Datenstrukturen zur Realisierung (Arrays versus Verkettung und dynamische Speicherallokation für Elemente), Algorithmen zur Realisierung kanonischer Operationen (Listentraversion, Anfügen, Einfügen, Löschen, Suchen, Stack-Operationen, FIFO-Warteschlangenoperationen)
12. Bäume: Binäre Bäume, AVL-Bäume, Rot-Schwarz-Bäume, B-Bäume – Suchen, Einfügen, Löschen, Traversion
13. Hashing: Hash-Array, Hashfunktion, Hash Buckets, offenes Hashing
14. Graphen: ungerichtete, gerichtete, gewichtete Graphen – Repräsentation durch Knoten- und Kantenlisten, durch Adjazenzmatrizen, Adjazenzlisten – Algorithmen auf Graphen: Breitensuche, Tiefensuche, kürzeste Wege auf gewichteten Graphen: Dijkstras Algorithmus, minimal aufspannende Bäume: Algorithmen von Prim et al. und Kruskal

Im Rahmen des Übungsbetriebes werden \LaTeX und Versionskontrolle mittels Git eingeführt und verwendet.

Lehrveranstaltung(en):

- 03-B-MI-22.1 Objektorientierte Programmierung [OOP] (3 CP)
- 03-B-MI-22.2 Algorithmen und Datenstrukturen [AuD] (6 CP)

Unterlagen (Skripte, Literatur, Programme usw.):

- David J. Barnes, Michael Kölling: Java lernen mit BlueJ - Objects first - Eine Einführung in Java. Aktuelle Auflage. Pearson Studium.
- Christian Ullenboom: Java ist auch eine Insel. Aktuelle Auflage. Rheinwerk Computing.
- Thomas Ottmann, Peter Widmayer: Algorithmen und Datenstrukturen. Aktuelle Auflage, Spektrum Akademischer Verlag.
- Robert Sedgewick, Robert Wayne: Algorithmen. Aktuelle Auflage. Pearson Studium.
- Markus von Rimscha: Algorithmen kompakt und verständlich. Aktuelle Auflage. Springer Vieweg.

Form der Prüfung:

KP; PL1: 30%, PL2: 55%, PL3: 15% ; Klausur, Portfolio, Fachgespräch

Arbeitsaufwand	Präsenz	84 h
	Übungsbetrieb/Prüfungsvorbereitung	186 h
	Summe	270 h
Lehrende: Dr. K. Hölscher	Verantwortlich: Dr. K. Hölscher	

03-IBG-DBM	Datenbankgrundlagen & Modellierung
	<i>Foundations of Data Bases & Modeling</i>

Lehrform (teaching format) / **SWS** (hours per week): 2VL + 2UE

Kreditpunkte (credit points): 6

Turnus (frequency): i.d.R. jedes SoSe

Inhaltliche Voraussetzungen (content-related prior knowledge/skills): KEINE

Sprache (language): Deutsch

Lehrende (teaching staff): AG Datenbanken (Prof. Dr. Sebastian Maneth)

Studiengang (degree program)	Module	Semester
Informatik (Bachelor VF)	IBGP-DBM	2. Sem.
Informatik (Bachelor KF)	KINF-G1/G2/G3	ab 2. Sem.
Digitale Medien (Bachelor)	DMB-MI-8	ab 4. Sem.
Wirtschaftsinformatik (Bachelor)	INF-3	2. Sem.
Systems Engineering (Bachelor)	IBGP-DBM	2. Sem.
Mathematik (Bachelor VF)	Anwendungsfach Informatik	ab 2. Sem.
ProMat (Master)	Informatikwerkzeuge	ab 1.Sem.
Zertifikatsstudium DiMePäd	DM in Lernumgebungen	ab 1.Sem.

Lernergebnisse:

- Mit relationalen Datenbanken umgehen. Insbesondere elementare relationale Datenbankschemata und Datenmanipulationsanweisungen verstehen, formulieren und verwenden.
- Den Aufbau von Datenbankabfragen kennen und häufig auftretende Anfragen selbstständig formulieren.
- UML Diagramme erstellen für statische Aspekte (Klassendiagramme) als auch für dynamische Aspekte (Aktivitäts-, Zustands- und Sequenzdiagramme)
- UML-Modelle (mit Klassen, Assoziationen, elementaren Attributtypen, gängigen Multiplizitäten und üblichen Vererbungsstrukturen) in relationale Datenbankschemata transformieren. Insbesondere Schlüssel- und Fremdschlüsselbeziehungen erkennen und nutzen.
- Grundideen und Begriffe des relationalen Entwurfs verstehen (Abhängigkeiten zwischen Attributen, Schlüssel, Schlüsselkandidaten, Normalformen, Gütekriterien).

Learning Outcome:

Inhalte:

- Konzepte der Structured Query Language (SQL) • Schemadefinition
- Datendefinition
- Datenbankabfragen
- UML Modellierung
- Relationaler Datenbankentwurf

Modul 03-INF-BA-KINF-P1: Praktische Informatik 1

Modulgruppenzuordnung:

- Pflicht

Empfohlene inhaltliche Voraussetzungen:

keine

Lerninhalte:

Grundlagen der Programmierung:

- Variablen
- Bedingte Anweisungen
- Schleifen
- Mathematische Formeln in Programmen
- Funktionen und Rekursion
- Verwendung von Objekten und Klassen, Grundlagen der Vererbung
- Arrays (veränderlicher Größe und mehrerer Dimensionen)
- Kommentare in Programmen

Die Programmierumgebung Processing:

- Grafik und Interaktion
- Einbettung von Medien (Bild, Ton, Video)
- Verwendung von Zufallsfunktionen (Perlin-Noise und lineare Zufallsverteilungen)
- Methoden des Debuggings

Ausgewählte Aspekte der Informatik:

- Grundlagen des maschinellen Rechnens
- Grundlagen der Rechnerarchitektur
- Programm und Prozess
- Programmierparadigmen und Programmiersprachen (inkl. Einordnung von Processing/JAVA)
- Zusammenhänge und Funktion von Compiler, Assembler, Loader, Linker, Interpreter, Laufzeitumgebung
- Begriff des Algorithmus

Lernergebnisse / Kompetenzen:

Die Studierenden

- beherrschen grundlegende Konzepte des imperativen und objektorientierten Programmierens
- können graphisch-interaktive Programme in der Programmierumgebung Processing, welche auf der aktuell weit verbreiteten Programmiersprache JAVA basiert, entwickeln
- können selbstständig kreative Ideen in Entwurfskonzepte und Programme überführen und dabei auch Medien wie Bild und Ton angemessen einbetten
- können spezifische Probleme in Teilprobleme zerlegen und diese Strukturierung mit Mitteln von Processing/Java umsetzen und aussagekräftig dokumentieren
- beherrschen die Erstellung und Bearbeitung größerer, komplexer Programme mit einem Team von mehreren Personen
- verstehen typische Denkweisen der Informatik, um in interdisziplinären Projekten mit Informatikern kommunizieren zu können
- sind in der Lage, ihr Vorgehen im größeren Kontext der Informatik einzuordnen

Workloadberechnung:

56 h SWS / Präsenzzeit / Arbeitsstunden

124 h Selbstlernstudium

Bestehen Auswahlmöglichkeiten von Lehrveranstaltungen im Modul?

nein

Unterrichtsprache(n):

Deutsch

Modulverantwortliche(r):

Dr. Tim Laue

Häufigkeit:

Wintersemester, jährlich

Dauer:

1 Semester

Modul gültig seit / Modul gültig bis:

SoSe 24 / -

ECTS-Punkte / Arbeitsaufwand:

6 / 180 Stunden

Modulprüfungen**Modulprüfung:** Modulprüfung**Prüfungstyp:** Kombinationsprüfung**Prüfungsform:**

Bekanntgabe zu Beginn des Semesters

Die Prüfung ist unbenotet?

nein

Anzahl Prüfungsleistungen / Studienleistungen / Prüfungsvorleistungen:

2 / - / -

Prüfungssprache(n):

Deutsch

Beschreibung:

PL1: Portfolio, PL2: Fachgespräch

Lehrveranstaltungen des Moduls**Lehrveranstaltung:** Grundlagen der Programmierung**Häufigkeit:**

Wintersemester, jährlich

Gibt es parallele Veranstaltungen?

nein

SWS:

4

Dozent*in:

Dr. Tim Laue

Unterrichtsprache(n):

Deutsch

Literatur:

- Daniel Shiffman: "Learning Processing: A Beginner's Guide to Programming Images, Animation, and Interaction", The Morgan Kaufmann Series in Computer Graphics, 2015. Quellen im Internet:
 - <http://learningprocessing.com> (Beispiele und Videos zum Buch)
 - <https://processing.org> (Referenz und Tutorials)

Lehrform(en):

Vorlesung mit Übung

Zugeordnete Modulprüfung:

Modulprüfung