65002 Grundlagen der Informatik 2

Modulverantwortliche/r Dr. Sebastian Küpper

Dauer des Moduls	ECTS	Workload	Häufigkeit
ein Semester	10	300 Stunden	in jedem Semester

Lehrveranstaltung(en) Grundlagen der Informatik 2

Detaillierter Zeitaufwand
Lektionen: 150 Stunden
Einsendearbeiten: 75 Stunden
Prüfungsvorbereitung: 75 Stunden

Qualifikationsziele
Nach Abschluss der Lehrveranstaltung sollen die Kursteilnehmer informatische Problemstellungen mit den Konzepten der Objektorientierung modellieren und ihre Softwarelösungen unter Berücksichtigung üblicher Praktiken der Objektorientierung umsetzen können. Darüber hinaus ist es das Ziel der Veranstaltung, algorithmisches Denken zu lernen und zu vertiefen, so dass die Teilnehmer in der Lage sind, für eine gegebene Aufgabe geeignete Datenstrukturen und Algorithmen zu entwickeln oder auszuwählen. Das Prinzip der Rekursion soll verstanden werden und somit bei entsprechenden Problemen eingesetzt werden können. Mit der Strategie der dynamischen Programmierung sollen optimierte Lösungsverfahren aus initialen rekursiven Lösungen gewonnen werden. Algorithmen und objektorientiertes Design sollen in der vorgestellten Beispielssprache C# zuverlässig umgesetzt werden können.
Schließlich sollen die Kursteilnehmer in die Lage versetzt werden, die Komplexität und Entscheidbarkeit eines Problems abzuschätzen und gegebenenfalls formal nachzuweisen. Der Umgang mit endlichen Automaten als Modellierungswerkzeug mit beschränkter Ausdrucksmächtigkeit aber günstigen algorithmischen Eigenschaften soll zuverlässig beherrscht werden.

Inhalte
Die Veranstaltung ist in drei Teile geteilt. Im ersten Teil wird, basierend auf der Einführung in die imperative Programmierung mit C# in der Veranstaltung Grundlagen der Informatik 1, in die objektorientierte Programmierung eingeführt. Grundlegende Strukturbegriffe wie Objekt, Klasse oder Schnittstelle werden eingeführt, eingeordnet und in C# praktisch eingesetzt. Wichtige Aspekte der Objektorientierung wie Vererbung, Überschreiben, Verdecken und Überladen von Methoden, dynamische Bindung und das Geheimnisprinzip werden theoretisch und praktisch erarbeitet.

Im zweiten Teil werden Algorithmen und Datenstrukturen mit einem besonderen Schwerpunkt auf Rekursion und dynamische Programmierung besprochen, um grundlegende Techniken für den Entwurf effizienter Algorithmen zu erwerben. Ergänzend zu den Darstellungen zu Rechnernetzen aus Grundlagen der Informatik 1 wird ein kurzer Überblick über Grundprinzipien der Kryptographie gegeben. Alle algorithmischen Überlegungen werden praktisch mit C# erprobt.

Aus der Perspektive der Grenzen der Algorithmik wird im dritten Teil der Lehrveranstaltung die Berechenbarkeits- und Komplexitätstheorie motiviert. In diesem Teil werden zunächst Beweistechniken besprochen, um nachzuweisen, dass ein Problem unentscheidbar oder NP-hart ist, bevor mithilfe der Chomsky-Hierarchie die Möglichkeiten eingeschränkter Rechenmodelle erläutert werden. Ein besonderer Schwerpunkt wird hierbei auf endliche Automaten für reguläre Sprachen gelegt. Die theoretischen Überlegungen werden in enger Anlehnung an die Programmiersprache C# dargelegt.

Inhaltliche Voraussetzung
Grundlegende Programmierkenntnisse der imperativen Programmierung und zur Funktionsweise von Computern, insbesondere der Codierung (z. B. aus dem Kurs Grundlagen der Informatik 1) werden vorausgesetzt. Darüber hinaus werden Schulkenntnisse zur Mathematik vorausgesetzt.

Zur erfolgreichen Teilnahme am Kurs ist ein Zugang zu einem PC mit einem modernen Betriebssystem – Windows, MacOS oder Linux – für die Programmierpraxis und ein

	Internetzugang für die Teilnahme an Diskussionsforen, zur Übermittlung von Einsendearbeiten, zum Herunterladen von Programmierprojekten und zur Kommunikation mit der Lehrveranstaltungsbetreuung erforderlich.
Lehr- und Betreuungsformen	Lehrveranstaltungsmaterial Einsendeaufgaben mit Korrektur und/oder Musterlösung Internetgestütztes Diskussionsforum Studientag/e Zusatzmaterial Fachmentorielle Betreuung Lehrvideos Betreuung und Beratung durch Lehrende Online-Tutorium
Anmerkung	Das Modul wird mit einer Portfolioprüfung abgeschlossen. Die Portfolioprüfung besteht aus einer digitalen Teilleistung (digital durchgeführte Programmieraufgabe und Multiple-Choice-Fragen zu Konzepten der Objektorientierung und Algorithmik) während des Semesters, bei der bis zu 50 Prozentpunkte erreicht werden können, und einer Klausur am Ende des Semesters, bei der ebenfalls bis zu 50 Prozentpunkte erreicht werden können. Die Anmeldung zu beiden Teilen der Portfolioprüfung erfolgt mit der Anmeldung zur Klausur.
Formale Voraussetzung	keine

Verwendung des Moduls	B.Sc. Mathematisch-technische Softwareentwicklung B.Sc. Wirtschaftsinformatik

Prüfungsformen	Art der Prüfungsleistung	Voraussetzung
Prüfung	benotete Portfolioprüfung (s. Anmerkung)	keine
Stellenwert der Note s. PO		