

Programmierung II (W3WI_109)

Programming II

FORMALE ANGABEN ZUM MODUL

MODULNUMMER	VERORTUNG IM STUDIENVERLAUF	MODULDAUER (SEMESTER)	MODULVERANTWORTUNG	SPRACHE
W3WI_109	1. Studienjahr	1	Professor Dr. Dietmar Ratz	Deutsch/Englisch

INGESETZTE LEHRFORMEN

Vorlesung, Übung, Laborübung, Case Study

INGESETZTE PRÜFUNGSFORMEN

PRÜFUNGSLEISTUNG	PRÜFUNGSUMFANG (IN MINUTEN)	BENOTUNG
Klausur oder Portfolio	120	ja

WORKLOAD UND ECTS-LEISTUNGSPUNKTE

WORKLOAD INSGESAMT (IN H)	DAVON PRÄSENZZEIT (IN H)	DAVON SELBSTSTUDIUM (IN H)	ECTS-LEISTUNGSPUNKTE
150	60	90	5

QUALIFIKATIONSZIELE UND KOMPETENZEN

FACHKOMPETENZ

Die Studierenden kennen fortgeschrittene Konzepte objektorientierter Programmiersprachen. Sie besitzen Kenntnisse über wichtige Algorithmen und Datenstrukturen sowie über Methoden zur Beurteilung der Effizienz und Qualität von Algorithmen.

METHODENKOMPETENZ

Die Studierenden können fortgeschrittene Konzepte der Objektorientierung anwenden und autonom mittlere bis größere lauffähige Programme implementieren und testen. Sie sind in der Lage, Algorithmen in verschiedenen Darstellungsarten zu verstehen und ihre Effizienz bzw. Qualität zu beurteilen, aber auch selbstständig Algorithmen und dazu erforderliche Datenstrukturen zu entwickeln und zu implementieren.

PERSONALE UND SOZIALE KOMPETENZ

Die Studierenden können eigenständig Algorithmen und Lösungsverfahren erarbeiten. Sie können stichhaltig und sachgemessen über Konzepte und eigene Algorithmen und deren Implementierungen und die damit verbundenen Probleme argumentieren, eigene Umsetzungen plausibel darstellen und eventuelle Fehler nachvollziehbar gegenüber anderen begründen.

ÜBERGREIFENDE HANDLUNGSKOMPETENZ

Die Studierenden können unter Einsatz der Programmiersprache komplexe praktische Probleme modellieren, algorithmisch behandeln und in anwenderfreundliche und effiziente Lösungen umsetzen. Sie können praktische Problemstellungen analysieren und bekannte Algorithmen und Datenstrukturen effizienzorientiert darauf anwenden und falls notwendig an die Problemstellungen anpassen.

LERNEINHEITEN UND INHALTE

LEHR- UND LERNEINHEITEN	PRÄSENZZEIT	SELBSTSTUDIUM
Fortgeschrittene Programmierung	30	45

Fortgeschrittene objektorientierte Konzepte: Generische Interfaces und Klassen, Nutzung der Klassenbibliothek.

Aufbau grafischer Oberflächen: Layout, typische Komponenten für grafische

Benutzungsschnittstellen, Ereignisbehandlung. Fortgeschrittene Programmiermethodik:

Parallele Programmierung mit Threads, Synchronisations- und Kommunikationskonzepte, Ein- und Ausgabe über Streams.

LERNEINHEITEN UND INHALTE

LEHR- UND LERNEINHEITEN

Algorithmen und Datenstrukturen

PRÄSENZZEIT

30

SELBSTSTUDIUM

45

Algorithmen: Beschreibung und Analyse. Datentypen: elementare, strukturierte, objektorientierte und generische Datentypen. Datenstrukturen: lineare Listen mit Feldstruktur, einfach und doppelt verkettete Listen, Bäume, Stapel und Schlangen mit ihren Grundoperationen für Einfügen, Löschen etc. Abstrakte Datentypen: Collections und Iteratoren. Suchverfahren, Sortierverfahren, Teile-und-Herrsche-Paradigma, Backtracking-Algorithmen.

BESONDERHEITEN

-

VORAUSSETZUNGEN

-

LITERATUR

Falls Java als Programmiersprache eingesetzt wird:

- Deck, K.-G. und Neuendorf, H.: Grundkurs Programmieren für Wirtschaftsinformatiker. Vieweg, Wiesbaden.
- Ottmann, T.; Widmayer, P.: Algorithmen und Datenstrukturen, Spektrum Akademischer Verlag; Heidelberg (Neckar).
- Ratz, D., Scheffler, J., Seese, D. und Wiesenberger, J.: Grundkurs Programmieren in Java, Hanser, München.
- Saake, G., Sattler, K.: Algorithmen und Datenstrukturen: Eine Einführung mit Java. dpunkt, Heidelberg.
- Sierra, K., Bates, B., Schulten, L., Buchholz, E.: Java von Kopf bis Fuß, O'Reilly, Köln.
- Sedgewick, R., Wayne K.: Algorithmen: Algorithmen und Datenstrukturen. Pearson Studium, München.

Falls andere Programmiersprachen eingesetzt werden, wird die empfohlene Literatur in der Vorlesung bekannt gegeben.