Module im Bereich: Informatik I - Pflicht

- Algorithmen und Datenstrukturen
- Datenbanken 1
- Einführung in die Informatik
- Modellierung
- Software Engineering & IT-Projektmanagement
- Software Engineering & IT-Projektmanagement (dual)

Algorithmen und Datenstrukturen

Sommer

(engl. Algorithms and Data Structures)

Modulnummer: FIN-INF-110361

Kürzel	CP	Semester	Fachsem.	Dauer	Sprache	Niveau
AuD	10	Sommer	ab 2.	1 Semester	deutsch	Bachelor

Link zum LSF:	LSF
Zugang beschränkt:	false
Verantwortung:	Christian Rössl
Dozent:in:	Christian Rössl
Lehrveranstaltungen:	In Deutsch / In German • Vorlesung *Algorithmen und Datenstrukturen* • Übung *Algorithmen und Datenstrukturen* • Tutorien *Algorithmen und Datenstrukturen* In Englisch / In English • Lecture *Algorithms and Data Structures* • Exercise *Algorithms and Data Structures* • Tutorial *Algorithms and Data Structures*
Verwendbarkeit:	- B.Sc. INF: Informatik I - Pflicht - B.Sc. INF: Kernmodule (SPO 2027) - B.Sc. CV: Informatik I - Pflicht - B.Sc. CV: Kernmodule (SPO 2027) - B.Sc. INGINF: Informatik I - Pflicht - B.Sc. INGINF: Kernmodule (SPO 2027) - B.Sc. WIF: Gestalten - Pflicht - B.Sc. WIF: Kernmodule (SPO 2027) - B.Sc. INF (bilingual): Informatik I - Pflicht - B.Sc. INF (bilingual): Kernmodule (SPO 2027)

Angestrebte Lernergebnisse:
Die Studierenden

- kennen und verstehen die wichtigsten grundlegenden Datenstrukturen und Algorithmen auf diesen
- können diese anwenden und beurteilen, welche für eine bestimmte Anwendung (besser) geeignet sind
- verstehen vorgegebene und erstellen eigene Implementierungen

Inhalt:

- Listen
- Bäume
- Balancierte Suchbäume
- Heaps
- Hashverfahren
- Graphen
- Dynamische Programmierung
- Suche in Texten
- Entwurf und Analyse von Algorithmen

Arbeitsaufwand:
106h Präsenz + 194h selbstständige Arbeit

Prüfungsvorleistungen:	Studien-/Prüfungsleistungen:	Lehrform / SWS:
Um an der Klausur teilnehmen zu können, sind Prüfungsvorleistungen erforderlich. Details werden in der ersten Vorlesung bekannt gegeben.	Klausur 120 Minuten	• Vorlesung (4 SWS) • Übung (2 SWS) • Tutorium (1 SWS)

Voraussetzungen nach Prüfungsordnung:	Empfohlene Voraussetzungen:

keine	• Programmierkenntnisse in Java, insbesondere Objektorientierung und Abstrakte Datentypen • Grundlegendes Verständnis von Algorithmen, insbesondere Verständnis von Rekursion • Verständnis von asymptotischer Abschätzung des Aufwands von Algorithmen und deren formale Beschreibung (z.B. Groß-O-Notation) • Verständnis grundlegender Beweistechniken (z.B. vollständige Induktion)

Medienformen:	**Literatur:**
	• Saake & Sattler: Algorithmen und Datenstrukturen • Goodrich, Tamassia & Goldwasser: Data Structures and Algorithms in Java • Sedgewick & Wayne: Algorithms • Cormen, Leiserson, Rivest, Stein: Introduction to Algorithms

Hinweise:
Die Abgabe des **ersten Übungsblatts** erfolgt ggf. schon in der ersten Semesterwoche, d.h. die Aufgaben können vorher bearbeitet werden! Dazu ist eine rechtzeitige Anmeldung auf https://aud.vc.cs.ovgu.de erforderlich. Diese ist bereits vor Semesterbeginn möglich.

Datenbanken 1

Winter

(engl. Database Concepts)

Modulnummer: FIN-INF-100391

Kürzel	CP	Semester	Fachsem.	Dauer	Sprache	Niveau
DB1	5	Winter	ab 1.	1 Semester	deutsch	Bachelor

Link zum LSF:	LSF
Zugang beschränkt:	
Verantwortung:	Gunter Saake
Dozent:in:	Gunter Saake
Lehrveranstaltungen:	• Vorlesung *Datenbanken 1* • Übung *Datenbanken 1*
Verwendbarkeit:	- B.Sc. INF: Informatik I - Pflicht - B.Sc. INF: Kernmodule (SPO 2027) - B.Sc. CV: Informatik I - Pflicht - B.Sc. CV: Kernmodule (SPO 2027) - B.Sc. INGINF: Informatik I - Pflicht - B.Sc. INGINF: Kernmodule (SPO 2027) - B.Sc. WIF: Anwenden - Pflicht - B.Sc. WIF: Kernmodule (SPO 2027) - B.Sc. INF (bilingual): Informatik I - Pflicht - B.Sc. INF (bilingual): Kernmodule (SPO 2027)

Angestrebte Lernergebnisse:
Die Studierenden ...

- verstehen die Grundkonzepte von Datenbanksystemen
- können relationale Datenbanken entwerfen
- kennen relationale Datenbanksprachen
- können Datenbankanwendungen entwickeln

Inhalt:

- Eigenschaften von Datenbanksystemen
- Architekturen
- Konzeptueller Entwurf einer relationalen Datenbank
- Relationales Datenbankmodell
- Abbildung ER-Schema auf Relationen
- Datenbanksprachen (Relationenalgebra, SQL)
- Formale Entwurfskriterien und Normalisierungstheorie
- Anwendungsprogrammierung
- Weitere Datenbankkonzepte wie Sichten, Trigger, Rechtevergabe

Arbeitsaufwand:
56 h Präsenz + 94 h selbstständige Arbeit

Prüfungsvorleistungen:	Studien-/Prüfungsleistungen:	Lehrform / SWS:
Erfolgreicher Besuch der Übung	Schriftliche Prüfung (120 min)	• 2 SWS Vorlesung • 2 SWS Übung

Voraussetzungen nach Prüfungsordnung:	Empfohlene Voraussetzungen:
Kann nicht gleichzeitig mit dem Modul "Database Concepts" besucht werden	

Medienformen:	Literatur:
	Datenbanken - Konzepte und Sprachen. Gunter Saake, Kai-Uwe Sattler, Andreas Heuer. März 2018, ISBN 978-3-95845-777-5; Mitp-Verlag; Auflage: 6

Hinweise:

Einführung in die Informatik

Winter

(engl. Introduction to Computer Science)

Modulnummer: FIN-INF-110360

Kürzel	CP	Semester	Fachsem.	Dauer	Sprache	Niveau
Einf. INF	10	Winter	ab 1.	1 Semester	deutsch	Bachelor

Link zum LSF:	LSF
Zugang beschränkt:	false
Verantwortung:	Christian Rössl
Dozent:in:	Christian Rössl
Lehrveranstaltungen:	• Vorlesung *Einführung in die Informatik* • Übung *Einführung in die Informatik* • Tutorium *Einführung in die Informatik*
Verwendbarkeit:	- B.Sc. INF: Informatik I - Pflicht - B.Sc. INF: Kernmodule (SPO 2027) - B.Sc. CV: Informatik I - Pflicht - B.Sc. CV: Kernmodule (SPO 2027) - B.Sc. INGINF: Informatik I - Pflicht - B.Sc. INGINF: Kernmodule (SPO 2027) - B.Sc. WIF: Gestalten - Pflicht - B.Sc. WIF: Kernmodule (SPO 2027) - B.Sc. INF (bilingual): Informatik I - Pflicht - B.Sc. INF (bilingual): Kernmodule (SPO 2027)

Angestrebte Lernergebnisse:
Die Studierenden

- verstehen Grundkonzepte von Algorithmen, insbesondere auch Rekursion
- können Algortihmen und einfache Datenstrukturen in Java implementieren
- verstehen Grundlagen verschiedener Programmierparadigmen
- können Eigenschaften von Algorithmen und Datenstrukturen formal beschreiben (z.B. Korrektheit, Aufwand, ADT) und können grundlegende Beweistechniken anwenden

- wenden theoretische und praktische Inhalte in Übungen und Tutorien an

Inhalt:

- Algorithmische Grundkonzepte
- Grundkonzepte in Java
- Funktionen
- Objektorientierte Programmierung
- Programmierparadigmen
- Suchen und Sortieren
- Korrektheit von Algorithmen
- Aufwand von Algorithmen
- Abstrake Datentypen und Einführung in grundlegende Datenstrukturen
- Berechenbarkeit und Entscheidbarkeit

Arbeitsaufwand:

- 98 h Präsenzzeit
- 202 h selbstständige Arbeit

Prüfungsvorleistungen:	Studien-/Prüfungsleistungen:	Lehrform / SWS:
Um an der Klausur teilnehmen zu können, sind Prüfungsvorleistungen erforderlich. Details werden in der ersten Vorlesung bekannt gegeben.	Klausur 120 Minuten	• Vorlesung (4 SWS) • Übung (2SWS) • Tutorium (1 SWS)

Voraussetzungen nach Prüfungsordnung:	Empfohlene Voraussetzungen:
keine	

Medienformen:	Literatur:
	Überwiegend: • Saake & Sattler: Algorithmen und Datenstrukturen Ergänzend: • Goodrich, Tamassia & Goldwasser: Data Structures and Algorithms in Java • Sedgewick & Wayne: Algorithms • Cormen, Leiserson, Rivest & Stein: Introduction to Algorithms

Hinweise:

Modellierung

Winter

(engl. Modeling)

Modulnummer: FIN-INF-110127

Kürzel	CP	Semester	Fachsem.	Dauer	Sprache	Niveau
Mod	5	Winter	ab 1.	1 Semester	deutsch	Bachelor

Link zum LSF:	LSF
Zugang beschränkt:	
Verantwortung:	Dr.-Ing. Daniel Staegemann
Dozent:in:	Dr.-Ing. Daniel Staegemann
Lehrveranstaltungen:	• Vorlesung *Modellierung* • Übung *Modellierung*
Verwendbarkeit:	- B.Sc. INF: Informatik I - Pflicht - B.Sc. CV: Informatik I - Pflicht - B.Sc. INGINF: Informatik I - Pflicht - B.Sc. WIF: Gestalten - Pflicht - B.Sc. INF (bilingual): Informatik I - Pflicht

Angestrebte Lernergebnisse:
Studierende ...

- können Systeme, Daten und Prozesse mithilfe verschiedener Modellierungstechniken abbilden.
- können im Rahmen von Informations- und Kommunikationssystemen Lösungen gestalten, darstellen und kommunizieren.
- können komplexe, gestalterische Sachverhalte abstrahieren und analytisch denken.
- können Konzepte für Informatiklösungen und computergestützte Lösungen so entwerfen und darstellen, dass diese hierauf aufbauend implementiert werden können.
- können die von ihnen erworbenen Fähigkeiten der Abstraktion und Darstellung, besonders in Bezug auf die Abbildung von Prozessen, einerseits zur Erfüllung von Aufgaben an der Schnittstelle von Informatik, Wirtschaftsinformatik und Wirtschaftswissenschaften, aber

auch im Kontext von Fachgebieten außerhalb der Informatik einbringen.
- können die Darstellung ihrer Konzepte und Ideen an die Bedürfnisse unterschiedlicher Stakeholder anpassen und somit in zahlreichen Kontexten und über verschiedene Disziplinen hinweg einen wertvollen Beitrag zum Gelingen verschiedenartigster Projekte leisten.

Inhalt:

- Modellierungstheorie: Von der Diskurswelt zu formalisierten Informationsmodellen
- Prozesse, Workflows und Geschäftsprozesse
- Meta-Modelle, Referenzmodellierung
- Grundsätze ordnungsmäßiger Modellierung
- Grundlagen XML Fachkonzeptuelle
- Modellierung mit höheren Petri-Netzen, der Entity Relationship-Methode und der BPMN
- Objektorientierte Modellierung mit UML
- Umsetzung konkreter Aufgabenstellungen

Arbeitsaufwand:

- Präsenzzeiten: 28 SWS Vorlesung + 28 SWS Übung
- Selbstständiges Arbeiten: 54 h Vor- und Nachbereitung der Vorlesung + 40 h Entwicklung von Modellen für die Übung

Prüfungsvorleistungen:	Studien-/Prüfungsleistungen:	Lehrform / SWS:
	Benotete und unbenotete schriftliche Prüfung, 120 Min.	• Vorlesung (2 SWS) • Übung (2 SWS)

Voraussetzungen nach Prüfungsordnung:	Empfohlene Voraussetzungen:
keine	

Medienformen:	Literatur:
	Kecher, C. (2011): UML 2 – Das umfassende Handbuch. 4. Aufl. Reisig, W. (1998): Systementwurf mit Netzen. Berlin u. a.

Hinweise:

Software Engineering & IT-Projektmanagement

Winter

(engl. Software Engineering & IT-Projectmanagement)

Modulnummer: FIN-INF-110476

Kürzel	CP	Semester	Fachsem.	Dauer	Sprache	Niveau
SE&IT-PM	5	Winter	ab 3.	1 Semester	deutsch	Bachelor

Link zum LSF:	LSF
Zugang beschränkt:	false
Verantwortung:	Dr. Thomas Wilde
Dozent:in:	Dr. Thomas Wilde
Lehrveranstaltungen:	• Vorlesung *Software Engineering & IT-Projektmanagement* • Übung *Software Engineering & IT-Projektmanagement*
Verwendbarkeit:	- B.Sc. INF: Informatik I - Pflicht - B.Sc. INF: Kernmodule (SPO 2027) - B.Sc. CV: Informatik I - Pflicht - B.Sc. CV: Kernmodule (SPO 2027) - B.Sc. INGINF: Informatik I - Pflicht - B.Sc. INGINF: Kernmodule (SPO 2027) - B.Sc. WIF: Gestalten und Anwenden - Wahlpflicht - B.Sc. WIF: Kernmodule (SPO 2027) - B.Sc. INF (bilingual): Informatik I - Pflicht - B.Sc. INF (bilingual): Kernmodule (SPO 2027)

Angestrebte Lernergebnisse:
Studierende, die den Kurs erfolgreich absolviert haben ...

- ... verstehen den Begriff "Software Engineering", seine Bedeutung und die Notwendigkeit strukturierter Softwareentwicklung.
- ... kennen verschiedene Prozessmodelle, deren Vor- und Nachteile sowie die Kernaktivitäten des Software Engineerings.
- ... wenden Testmethoden und Refactoring an, um Softwarequalität zu sichern und Fehler zu vermeiden.
- ... verstehen und nutzen agile Methoden wie Scrum und Kanban und vergleichen sie mit traditionellen Ansätzen.
- ... setzen Versionskontrolle, IDEs und CI/CD-Pipelines effektiv ein und verstehen Continuous Integration & Deployment.
- ... schreiben lesbaren, wartbaren und effizienten Code nach bewährten Prinzipien und Konventionen.
- ... schätzen Aufwand und Ressourcen ein, planen Projekte und überwachen deren Fortschritt.
- ... kennen Architekturkonzepte und Design-Patterns und wenden diese in Softwareprojekten an.
- ... arbeiten effektiv im Team, präsentieren Ideen klar und dokumentieren ihre Arbeit.

Inhalt:

- Software Engineering - Was ist das und wozu wird es gebraucht?
- Prozessmodelle: sequentielle & iterativ,
- Prozessaktivitäten: Spezifikation, Entwicklung, Validierung, Evolution,
- Qualitätssicherung: Testint & Refactoring,
- Agile Softwareentwicklung,
- Tools & Werkzeuge,
- Clean-Coding / Code-Conventions,
- Continuous Integration, Continuous Deployment, Continuous Delivery,
- IT-Projektmanagement: Schätzung & Prozessplanung,
- Softwarearchitektur,
- Design-Patterns,
- Praktische Beispiele.

Arbeitsaufwand:

- Präsenzzeit: 28h Vorlesung + 28h Übung
- 94h selbstständige Arbeit

Prüfungsvorleistungen:	Studien-/Prüfungsleistungen:	Lehrform / SWS:

• Um an der Prüfung teilnehmen zu können, sind Prüfungsvorleistungen erforderlich. Details werden in der ersten Vorlesung bekannt gegeben.	• Prüfung: schriftliche Klausur, 120 Minuten; • Schein (SPO bis 2022): Bearbeitung der Aufgaben & Teilnahme an den Übungen.	• Vorlesung (2 SWS) Übung (2 SWS)

Voraussetzungen nach Prüfungsordnung:	Empfohlene Voraussetzungen:
keine	Erfolgreicher Abschluss von: • Einführung in die Informatik, • Algorithmen und Datenstrukturen.

Medienformen:	Literatur:
	• Ian Sommerville. *Software Engineering*, 10. Auflage. Pearson, 2016 • Erich Gamma, Richard Helm, Ralph E. Johnson, John Vlissides: *Design Patterns. Elements of Reusable Object-Oriented Software*, Addison Wesley, 1997 • Robert Martin: *Clean Code: A Handbook of Agile Software Craftsmanship*, Prentice Hall, 2008 Für alle Lehrbücher gilt, dass die Auflage und ggf. die Wahl zwischen englischem Original oder deutscher Übersetzung für uns irrelevant ist. Jede ältere Auflage ist in Ordnung.

Hinweise:

Software Engineering & IT-Projektmanagement (dual)

Winter

(engl. Software Engineering & IT-Projectmanagement)

Modulnummer: FIN-INF-1104762

Kürzel	CP	Semester	Fachsem.	Dauer	Sprache	Niveau
SE&IT-PM	5	Winter	ab 3.	1 Semester	deutsch	Bachelor

Link zum LSF:	LSF
Zugang beschränkt:	
Verantwortung:	Dr. Thomas Wilde
Dozent:in:	Dr. Thomas Wilde
Lehrveranstaltungen:	• Vorlesung *Software Engineering & IT-Projektmanagement* • Übung *Software Engineering & IT-Projektmanagement*
Verwendbarkeit:	- B.Sc. INF: Informatik I - Pflicht - B.Sc. INF: Kernmodule (SPO 2027) - B.Sc. CV: Informatik I - Pflicht - B.Sc. CV: Kernmodule (SPO 2027) - B.Sc. INGINF: Informatik I - Pflicht - B.Sc. INGINF: Kernmodule (SPO 2027) - B.Sc. WIF: Gestalten und Anwenden - Wahlpflicht - B.Sc. WIF: Kernmodule (SPO 2027)

Angestrebte Lernergebnisse:
Studierende, die den Kurs erfolgreich absolviert haben ...

- ... verstehen den Begriff "Software Engineering", seine Bedeutung und die Notwendigkeit strukturierter Softwareentwicklung.

- ... kennen verschiedene Prozessmodelle, deren Vor- und Nachteile sowie die Kernaktivitäten des Software Engineerings.
- ... wenden Testmethoden und Refactoring an, um Softwarequalität zu sichern und Fehler zu vermeiden.
- ... verstehen und nutzen agile Methoden wie Scrum und Kanban und vergleichen sie mit traditionellen Ansätzen.
- ... setzen Versionskontrolle, IDEs und CI/CD-Pipelines effektiv ein und verstehen Continuous Integration & Deployment.
- ... schreiben lesbaren, wartbaren und effizienten Code nach bewährten Prinzipien und Konventionen.
- ... schätzen Aufwand und Ressourcen ein, planen Projekte und überwachen deren Fortschritt.
- ... kennen Architekturkonzepte und Design-Patterns und wenden diese in Softwareprojekten an.
- ... arbeiten effektiv im Team, präsentieren Ideen klar und dokumentieren ihre Arbeit.

Inhalt:

- Software Engineering - Was ist das und wozu wird es gebraucht?
- Prozessmodelle: sequentielle & iterativ,
- Prozessaktivitäten: Spezifikation, Entwicklung, Validierung, Evolution,
- Qualitätssicherung: Testint & Refactoring,
- Agile Softwareentwicklung,
- Tools & Werkzeuge,
- Clean-Coding / Code-Conventions,
- Continuous Integration, Continuous Deployment, Continuous Delivery,
- IT-Projektmanagement: Schätzung & Prozessplanung,
- Softwarearchitektur,
- Design-Patterns,
- Praktische Beispiele.

Arbeitsaufwand:

- Präsenzzeit: 28h Vorlesung + 28h Übung
- 94h selbstständige Arbeit

Prüfungsvorleistungen:	Studien-/Prüfungsleistungen:	Lehrform / SWS:

	• Prüfung: schriftliche Klausur, 120 Minuten; • Schein (SPO bis 2022): Bearbeitung der Aufgaben & Teilnahme an den Übungen.	Vorlesung (2 SWS) Übung (2 SWS)

Voraussetzungen nach Prüfungsordnung:	Empfohlene Voraussetzungen:
keine	Erfolgreicher Abschluss von: • Einführung in die Informatik, • Algorithmen und Datenstrukturen.

Medienformen:	Literatur:
	• Ian Sommerville. *Software Engineering*, 10. Auflage. Pearson, 2016 • Erich Gamma, Richard Helm, Ralph E. Johnson, John Vlissides: *Design Patterns. Elements of Reusable Object-Oriented Software*, Addison Wesley, 1997 • Robert Martin: *Clean Code: A Handbook of Agile Software Craftsmanship*, Prentice Hall, 2008 Für alle Lehrbücher gilt, dass die Auflage und ggf. die Wahl zwischen englischem Original oder deutscher Übersetzung für uns irrelevant ist. Jede ältere Auflage ist in Ordnung.

Hinweise: