fiw.thws

Fakultät Informatik und Wirtschaftsinformatik

Technische Hochschule Würzburg-Schweinfurt

Modulhandbuch

Bachelor Software Engineering (B. Eng.)

fiw.thws
Fakultät Informatik und Wirtschaftsinformatik
Technische Hochschule Würzburg-Schweinfurt

Inhalt

1. Semester

fiw.thws
Fakultät Informatik und Wirtschaftsinformatik
Technische Hochschule Würzburg-Schweinfurt

Modul: 1510600
Algebra

Modulprofil

Prüfungsnummer
1510600

Dauer
1 Semester

Häufigkeit des Angebots
Jedes Wintersemester

SWS
4

ECTS-Credits (CP)
5.0

Workload
Angeleitete Studienzeit:
Synchrone Teilnahme: 1 Std.
Asynchrone Teilnahme: 3 Std.

Selbststudienzeit: 90 Std.

Gesamt: 150 Std.

Lehrveranstaltungsart(en)
Seminaristischer Unterricht, Übung

Lehrsprache
Deutsch

Organisation

Modulverantwortung
Prof. Dr. Andreas Keller

Dozierende
Prof. Dr. Andreas Keller

Verwendbarkeit	BSED
Studiensemester	1. Semester
Art des Moduls	Pflichtmodul

Verpflichtende Voraussetzungen gemäß SPO
keine

Empfohlene Voraussetzungen
Beherrschung der Schulmathematik

Inhalte
Allgemeine Grundlagen:
- Körper der reelle Zahlen
- Prinzip der vollständige Induktion
- Einführung in den Körper der komplexe Zahlen

Lineare Algebra:
- Vektorräume (lineare Unabhängigkeit, Basis und Dimension)
- Matrizen (Rechnen mit Matrizen, Spur und Determinante, Rang einer Matrix)
- Lineare Gleichungssysteme
- Gaußscher Algorithmus
- Lineare Abbildungen
- Eigenwerte und Eigenvektoren
- Diagonalisierung

Prüfung

Verpflichtende Voraussetzung gemäß SPO für die Teilnahme an der Prüfung
Keine

Art der Prüfung
Schriftliche Prüfung (sP) gemäß § 23 APO

Dauer/Form der Prüfung
90 Minuten

Die konkrete Festlegung der abzuleistenden Prüfung erfolgt im Studienplan

Prüfungssprache
Deutsch

Voraussetzung für die Vergabe von Leistungspunkten
Keine

Lernergebnisse

- Die Studierenden kennen grundlegende mathematische Konzepte und Begriffe, die in Informatik und Technik Anwendung finden.
- Die Studierenden verstehen die Bedeutung mathematischer Methoden und deren Rolle in der Entwicklung von Informatikanwendungen.
- Die Studierenden wenden mathematische Techniken an, um praktische Probleme in der Informatik und Technik zu lösen.
- Die Studierenden analysieren mathematische Probleme, um geeignete Lösungsstrategien zu identifizieren und zu entwickeln.
- Die Studierenden bewerten verschiedene Lösungsansätze und deren Effektivität im Kontext spezifischer mathematischer Herausforderungen.
- Die Studierenden erstellen komplexe mathematische Modelle, die in realen Anwendungen der Informatik und Technik genutzt werden können.

Literatur

- Gramlich, Günter: Lineare Algebra – Eine Einführung; Fachbuchverlag Leipzig im Carl Hanser Verlag 2021
- Hartmann, Peter: Mathematik für Informatiker; Vieweg + Teubner, Wiesbaden 2019
- Papula, Lothar: Mathematik für Ingenieure und Naturwissenschaftler 1 und 2; Vieweg + Teubner; Wiesbaden 2024
- Schubert, Matthias: Mathematik für Informatiker; Vieweg + Teubner, Wiesbaden 2012
- Strang, Gilbert: Lineare Algebra; Springer-Verlag, Berlin/Heidelberg/ New York 2003

fiw.thws
Fakultät Informatik und Wirtschaftsinformatik
Technische Hochschule Würzburg-Schweinfurt

Modul: 1510400
Datenbanken

Modulprofil

Prüfungsnummer
1510400

Dauer
1 Semester

Häufigkeit des Angebots
Jedes Wintersemester

SWS
4

ECTS-Credits (CP)
5.0

Workload
Angeleitete Studienzeit:
Synchrone Teilnahme: 1 Std.
Asynchrone Teilnahme: 3 Std.

Selbststudienzeit: 90 Std.

Gesamt: 150 Std.

Lehrveranstaltungsart(en)
Seminaristischer Unterricht, Übung

Lehrsprache
Deutsch

Organisation

Modulverantwortung
Michael Rott

Dozierende
Michael Rott

Verwendbarkeit	BSED
Studiensemester	1. Semester
Art des Moduls	Pflichtmodul

Verpflichtende Voraussetzungen gemäß SPO
keine

Empfohlene Voraussetzungen
Keine

Inhalte
Das Modul vermittelt die grundlegenden Konzepte und Techniken der Datenbankentwicklung. Es werden das relationale Datenmodell und die Relationen-Algebra als theoretische Grundlagen vorgestellt. Ein Schwerpunkt liegt auf der Datenbankmodellierung, insbesondere der Erstellung von Entity-Relationship-Modellen (ER-Modelle) und deren Überführung in relationale Schemata unter Berücksichtigung von Normalformen. Einführung in die Sprache SQL, einschließlich der Datenmanipulation, Datenabfrage sowie der Definition von Schemata und der Transaktionsverwaltung. In praktischen Übungen und semesterbegleitenden Projekten wird die Datenbankentwicklung und -administration geübt.

Prüfung

Verpflichtende Voraussetzung gemäß SPO für die Teilnahme an der Prüfung
Keine

Art der Prüfung
Sonstige Prüfung (soP) gemäß §§ 26, 27 APO

Dauer/Form der Prüfung
Mündliche Prüfung

Die konkrete Festlegung der abzuleistenden Prüfung erfolgt im Studienplan

Prüfungssprache
Deutsch

Voraussetzung für die Vergabe von Leistungspunkten
Keine

Lernergebnisse

- Die Studierenden können grundlegende Konzepte der Datenpersistenz und die Unterschiede zwischen persistenten und nicht-persistenten Daten erläutern.
- Die Studierenden können die zentralen Begriffe der relationalen Datenbanken, wie Relation, Primärschlüssel, Fremdschlüssel und Normalisierung, definieren.
- Die Studierenden verstehen die Relationale Algebra und können einfache Operationen darauf anwenden.
- Die Studierenden können den Zusammenhang zwischen konzeptioneller, logischer und physischer Datenmodellierung erklären und deren Bedeutung für die Datenbankentwicklung begründen.
- Die Studierenden sind in der Lage, Entity-Relationship-Modelle (ERM) für gegebene Anwendungsfälle zu erstellen und diese in relationale Schemata zu überführen.
- Die Studierenden können SQL-Abfragen zur Datenmanipulation (DML) und Schema-Definition (DDL) formulieren und ausführen.
- Die Studierenden können bestehende Datenbankschemata analysieren und hinsichtlich Redundanz, Konsistenz und Normalformen bewerten.
- Die Studierenden sind in der Lage, fachliche Informationsbedarfe zu analysieren und daraus geeignete Datenstrukturen und Abfragen abzuleiten.

Literatur

- Michael Kofler (2024). Datenbanksysteme - Das umfassende Lehrbuch (2. Auflage). Bonn: Rheinwerk Verlag GmbH
- Kemper, A., & Eickler, A. (2015). Datenbanksysteme – Eine Einführung (10. Auflage). München: De Gruyter Oldenbourg Verlag
- Elmasri, R., & Navathe, S. B. (2015). Grundlagen von Datenbanksystemen (7. Auflage). München: Pearson Studium
- Garcia-Molina, H., Ullman, J. D., & Widom, J. (2013). Database Systems: The Complete Book (2nd ed.). Upper Saddle River, NJ: Pearson
- Saake, G., Sattler, K.-U., & Heuer, A. (2011). Datenbanken – Konzepte und Sprachen (3. Auflage). München: Pearson Studium

fiw.thws
Fakultät Informatik und Wirtschaftsinformatik
Technische Hochschule Würzburg-Schweinfurt

Modul: 1510500
Grundlagen Informatik

Modulprofil

Prüfungsnummer
1510500

Dauer
1 Semester

Häufigkeit des Angebots
Jedes Wintersemester

SWS
4

ECTS-Credits (CP)
5.0

Workload
Angeleitete Studienzeit:
Synchrone Teilnahme: 1 Std.
Asynchrone Teilnahme: 3 Std.

Selbststudienzeit: 90 Std.

Gesamt: 150 Std.

Lehrveranstaltungsart(en)
Seminaristischer Unterricht, Übung

Lehrsprache
Deutsch

Organisation

Modulverantwortung
Prof. Dr. Peter Braun

Dozierende
Prof. Dr. Peter Braun

Verwendbarkeit	BSED
Studiensemester	1. Semester
Art des Moduls	Pflichtmodul

Verpflichtende Voraussetzungen gemäß SPO
keine

Empfohlene Voraussetzungen
keine

Inhalte
Das Modul vermittelt die Grundlagen der Informatik für Studierende außerhalb der Kern-Informatik.

- Information, Informationsgehalt, Informationscodierung, Darstellung von Zahlen und Zeichen, Codierung von Text, Datumsangaben, Farbinformationen
- Binärarithmetik, Boole'sche Algebra und Logikgatter
- Modelle und Modellbildung als grundlegendes Prinzip in der Informatik, Abstraktion, Reduktion, Dekomposition, Aggregation
- Beschreibung von Datenstrukturen mit der erweiterten Backus-Naur-Form
- Modellierung dynamischer Systeme und ihre Beschreibung mit endlichen Automaten und Zustandsdiagrammen
- Formale Sprachen, reguläre Grammatiken und das Wortproblem
- Weitere Automatenmodelle: Moore und Mealy Automaten
- Der Begriff des Algorithmus, Berechenbarkeit, Halteproblem, Funktionsweise und Programmierung von Turing-Maschinen
- Grundlegende Algorithmen zum Suchen und Sortieren
- Geschichte der Hardwareentwicklung
- Aufbau und prinzipielle Arbeitsweise eines Computers und Mikroprozessors, Von-Neumann Architektur, Moore'sches Gesetz
- Aufbau und Funktionsweise des Internet und World Wide Web
- Einführung in die Sprachen HTML und Markdown
- Aufbau von verteilten Systemen, Client-Server, Peer-to-Peer, Blockchain, Git
- Geschichte der Künstlichen Intelligenz, Verfahren des maschinellen Lernens, Regression, Funktionsweise von neuronalen Netzen
- Datenschutz und Ethik in der Informatik

Prüfung

Verpflichtende Voraussetzung gemäß SPO für die Teilnahme an der Prüfung
Keine

Art der Prüfung
Sonstige Prüfung (soP) gemäß §§ 26, 27 APO

Dauer/Form der Prüfung
Mündliche Prüfung

Die konkrete Festlegung der abzuleistenden Prüfung erfolgt im Studienplan

Prüfungssprache
Deutsch

Voraussetzung für die Vergabe von Leistungspunkten
Keine

Lernergebnisse

- Die Studierenden erinnern grundlegende Begriffe der Informationsverarbeitung und können den Informationsgehalt von Nachrichten messen.
- Die Studierenden verstehen Codierungen von Daten und grundlegende Methoden zur Modellbildung innerhalb der Informatik.
- Die Studierenden wenden Methoden zur Beschreibung von Datenstrukturen an.
- Die Studierenden analysieren einfache dynamische Systeme und beschreiben diese mit Zustandsdiagrammen.
- Die Studierenden erstellen reguläre Grammatiken und lösen das Wortproblem mit Hilfe von endlichen Automaten.
- Die Studierenden analysieren einfache Problemstellungen und erstellen Lösungen mit Hilfe von Turing-Maschinen.
- Die Studierenden erinnern wichtige Punkte der Geschichte der Informatik.

Literatur

- Gumm, Heinz-Peter; Sommer, Manfred: Einführung in die Informatik, 10. Auflage, Oldenbourg, 2012.
- Ernst, Hartmut; Schmidt, Jochen; Beneken, Gerd: Grundkurs Informatik: Grundlagen und Konzepte für die erfolgreiche IT-Praxis, 8. Auflage, Springer Verlag, 2023

fiw.thws
Fakultät Informatik und Wirtschaftsinformatik
Technische Hochschule Würzburg-Schweinfurt

Modul: 1510100
IT-Projektmanagement und BWL

Modulprofil

Prüfungsnummer
1510100

Dauer
1 Semester

Häufigkeit des Angebots
Jedes Wintersemester

SWS
4

ECTS-Credits (CP)
5.0

Workload
Angeleitete Studienzeit:
Synchrone Teilnahme: 1 Std.
Asynchrone Teilnahme: 3 Std.

Selbststudienzeit: 90 Std.

Gesamt: 150 Std.

Lehrveranstaltungsart(en)
Seminaristischer Unterricht, Übung

Lehrsprache
Deutsch

Organisation

Modulverantwortung
Prof. Dr. Eva Wedlich

Dozierende
Prof. Dr. Eva Wedlich

Verwendbarkeit	BSED
Studiensemester	1. Semester
Art des Moduls	Pflichtmodul

Verpflichtende Voraussetzungen gemäß SPO
Keine

Empfohlene Voraussetzungen
Keine

Inhalte
Folgende Themen werden behandelt:
Einführung Projekt und Projektmanagement
- Projektorganisation
- Projektplanungsprozess
- Projektkalkulation
- Projektsteuerung und –überwachung
- Projektabschluss

Grundbegriffe der Betriebswirtschaftslehre:
- Der Betrieb
- Die betriebswirtschaftlichen Produktionsfaktoren
- Betriebswirtschaftliche Ziele
- Betriebswirtschaftliche Kennzahlen

Konstitutive Entscheidungen eines Betriebes:
- Standortwahl:
- Rechtsformen:

Betriebswirtschaftliche Funktionen:
- Beschaffung und Einkauf
- Marketing und Vertrieb

Prüfung

Verpflichtende Voraussetzung gemäß SPO für die Teilnahme an der Prüfung
Keine

Art der Prüfung
Schriftliche Prüfung (sP) gemäß § 23 APO

Dauer/Form der Prüfung
90 Minuten

Die konkrete Festlegung der abzuleistenden Prüfung erfolgt im Studienplan

Prüfungssprache
Deutsch

Voraussetzung für die Vergabe von Leistungspunkten
Keine

Lernergebnisse

- Die Studierenden können grundlegende Begriffe und Konzepte des IT-Projektmanagements sowie der Betriebswirtschaftslehre wiedergeben.
- Die Studierenden wenden Methoden zur Entwicklung von Projektplänen und zur Durchführung einer Standortanalyse an praktischen Beispielen an.
- Die Studierenden analysieren projektbezogene und betriebswirtschaftliche Daten.
- Die Studierenden beurteilen die Effizienz von Projektsteuerungsmechanismen und treffen fundierte Entscheidungen zur Projektkalkulation und -überwachung.
- Die Studierenden evaluieren alternative Standorte und strategische betriebswirtschaftliche Entscheidungen.
- Die Studierenden entwickeln eigenständig Lösungsstrategien für komplexe IT-Projektmanagementszenarien und betriebswirtschaftliche Herausforderungen.

Literatur

- Vahs, D.; Schäfer-Kunz, J.: Einführung in die Betriebswirtschaftslehre; 9. Auflage; Schäffer-Poeschel, Stuttgart, 2025
- Wöhe, G.: Einführung in die allgemeine Betriebswirtschaftslehre; 28. Auflage; Vahlen; München, 2023
- Hanke, D.: Die 10 wichtigsten Methoden im Projektmanagement: Der schnelle, verständliche und bewährte Einstieg ins klassische Projektmanagement, 2022
- Timinger, H.: Modernes Projektmanagement: Mit traditionellem, agilem und hybridem Vorgehen zum Erfolg, Wiley, 2024

fiw.thws
Fakultät Informatik und Wirtschaftsinformatik
Technische Hochschule Würzburg-Schweinfurt

Modul: 1510300
Introduction to Software Engineering

Modulprofil

Prüfungsnummer
1510300

Dauer
1 Semester

Häufigkeit des Angebots
Jedes Wintersemester

SWS
4

ECTS-Credits (CP)
5.0

Workload
Angeleitete Studienzeit:
Synchrone Teilnahme: 1 Std.
Asynchrone Teilnahme: 3 Std.

Selbststudienzeit: 90 Std.

Gesamt: 150 Std.

Lehrveranstaltungsart(en)
Seminaristischer Unterricht, Übung

Lehrsprache
Englisch

Organisation

Modulverantwortung
Prof. Dr.-Ing. Tobias Fertig

Dozierende
Prof. Dr. Isabel John,
Prof. Dr.-Ing. Tobias Fertig,
Prof. Dr.-Ing. Anne Heß

Verwendbarkeit	BSED
Studiensemester	1. Semester
Art des Moduls	Pflichtmodul

Verpflichtende Voraussetzungen gemäß SPO
keine

Empfohlene Voraussetzungen
keine

Inhalte
Software engineering covers all phases of software development, from the initial idea to the tested and delivered system. This module introduces the fundamental principles and concepts of software engineering, focusing on systematic development processes, requirements engineering, and basic modeling techniques. It provides the foundation for advanced modules on Analysis and Design and Software Quality.

Fundamentals of Software Engineering
- Objectives and principles of software engineering
- Characteristics and challenges of software projects
- Overview of software development activities and processes (agile vs. traditional)

Introduction to software (requirements) modeling with UML
- Introduction to UML Diagrams: Class Diagrams, Use Case Diagrams, Activity Diagrams, Sequence Diagrams

Software Development Processes & Team Collaboration
- Roles in software projects (e.g., Product Owner, Developer, Tester)
- Fundamentals of project organization and documentation
- Cost and benefit estimation in software projects
- Agile processes (e.g., Scrum)

Quality Aspects
- Introduction to software quality (transition to "Software Quality")

Prüfung

Verpflichtende Voraussetzung gemäß SPO für die Teilnahme an der Prüfung
Keine

Art der Prüfung
Sonstige Prüfung (soP) gemäß §§ 26, 27 APO

Dauer/Form der Prüfung
Schriftliche Prüfung, Portfolio

Die konkrete Festlegung der abzuleistenden Prüfung erfolgt im Studienplan

Prüfungssprache
Englisch

Voraussetzung für die Vergabe von Leistungspunkten
Keine

Lernergebnisse

- Students are able to explain the principles of software engineering and assess their relevance.
- Students compare and justify the application of different software development processes.
- Students collect, model, and specify systematically software requirements using UML.
- Students use UML diagrams to structure and analyze software systems.
- Students evaluate software projects in terms of feasibility, cost, and quality.
- Students understand fundamental quality assurance methods.

Literatur

- Sommerville, Ian: Software Engineering, Pearson, 2018
- Oestereich, Bernd: Analyse und Design mit UML 2.5, Oldenbourg, 2013/2020
- Rupp, Chris: UML glasklar, Hanser, 2012
- McLaughlin: Objektorientierte Analyse und Design von Kopf bis Fuß, O'Reilly, 2017
- Kecher, Christoph; Hoffmann-Elbern, Ralf; Will, Torsten T.: UML 2.5: Das umfassende Handbuch, Rheinwerk Computing, 2021

fiw.thws
Fakultät Informatik und Wirtschaftsinformatik
Technische Hochschule Würzburg-Schweinfurt

Modul: 1510200
Programmieren 1

Modulprofil

Prüfungsnummer
1510200

Dauer
1 Semester

Häufigkeit des Angebots
Jedes Wintersemester

SWS
4

ECTS-Credits (CP)
5.0

Workload
Angeleitete Studienzeit:
Synchrone Teilnahme: 1 Std.
Asynchrone Teilnahme: 3 Std.

Selbststudienzeit: 90 Std.

Gesamt: 150 Std.

Lehrveranstaltungsart(en)
Seminaristischer Unterricht, Übung

Lehrsprache
Deutsch

Organisation

Modulverantwortung
Prof. Dr. Tristan Wimmer

Dozierende
Prof. Dr. Tristan Wimmer

Verwendbarkeit	BSED
Studiensemester	1. Semester
Art des Moduls	Pflichtmodul

Verpflichtende Voraussetzungen gemäß SPO
Keine

Empfohlene Voraussetzungen
Keine

Inhalte
Dieses Modul zielt darauf ab, Studierenden die Grundlagen der Programmierung mithilfe der Programmiersprache Python beizubringen. Es stellt die Grundkonzepte von Programmiersprachen und Programmierparadigmen vor und schafft die Basis für weitere Module im Studiengang Software-Engineering.
Folgende Themen werden behandelt:

- Elementarer Datentypen, Datenstrukturen und Operatoren
- Kontrollstrukturen: Schleifen und bedingte Anweisungen
- Programmieren mit Funktionen
- Einführung in die objektorientierte Programmierung
- Einführung in das Konzept der Vererbung
- Einführung in das Exception Handling

Neben diesen Themen werden in diesem Modul die geeigneten Strukturierungsmöglichkeiten von Code, sowie Dokumentationsmöglichkeiten für einen sauberen und gut lesbaren Programmierstil, aufgezeigt. Desweiteren wird den Studierenden gezeigt, wie sie Problemen am besten begegnen und lösen.

Prüfung

Verpflichtende Voraussetzung gemäß SPO für die Teilnahme an der Prüfung
Keine

Art der Prüfung
Schriftliche Prüfung (sP) gemäß § 23 APO

Dauer/Form der Prüfung
90 Minuten

Die konkrete Festlegung der abzuleistenden Prüfung erfolgt im Studienplan

Prüfungssprache
Deutsch

Voraussetzung für die Vergabe von Leistungspunkten
Keine

Lernergebnisse

- Die Studierenden identifizieren und benennen die grundlegenden Datentypen, Datenstrukturen und Operatoren und wenden sie in der Programmiersprache Python an.
- Die Studierenden erläutern, wie Kontrollstrukturen wie Schleifen und bedingte Anweisungen den Ablauf von Programmen steuern und wie diese in Python implementiert werden.
- Die Studierenden schreiben einfache Python-Programme, die Funktionen und Parameterübergaben nutzen, um spezifische Aufgaben zu lösen und wenden dabei das Prinzip Divide & Conquer an.
- Die Studierenden verstehen objektorientierter Programmierung anzuwenden, um durch Kapselung die Struktur und Wartbarkeit eines Programms zu verbessern.
- Die Studierenden entwerfen und implementieren für eine spezifische Anforderung ein objektorientiertes Programm in Python und nutzen dabei die grundlegenden Prinzipien der Vererbung nutzt.
- Die Studierenden wenden Exception Handling für fehlerhafte Eingaben und Datentypinkompatibilität an.

Literatur
Häberlein, Tobias. Programmieren Mit Python: Eine Einführung in Die Prozedurale, Objektorientierte Und Funktionale Programmierung. 1st ed. 2024. Berlin, Heidelberg: Springer Berlin Heidelberg, 2024. https://doi.org/10.1007/978-3-662-68678-2.

2. Semester

fiw.thws
Fakultät Informatik und Wirtschaftsinformatik
Technische Hochschule Würzburg-Schweinfurt

Modul: 1511100 Algorithmen und Datenstrukturen

Modulprofil

Prüfungsnummer
1511100

Dauer
1 Semester

Häufigkeit des Angebots
Jedes Sommersemester

SWS
4

ECTS-Credits (CP)
5.0

Workload
Angeleitete Studienzeit:
Synchrone Teilnahme: 1 Std.
Asynchrone Teilnahme: 3 Std.

Selbststudienzeit: 90 Std.

Gesamt: 150 Std.

Lehrveranstaltungsart(en)
Seminaristischer Unterricht, Übung

Lehrsprache
Deutsch

Organisation

Modulverantwortung
Prof. Dr.-Ing. Tobias Fertig

Dozierende
Prof. Dr.-Ing. Tobias Fertig, Prof. Dr.-Ing. Sebastian Biedermann

Verwendbarkeit	BSED
Studiensemester	2. Semester
Art des Moduls	Pflichtmodul

Verpflichtende Voraussetzungen gemäß SPO
Keine

Empfohlene Voraussetzungen
Keine

Inhalte
Die Veranstaltung behandelt verschiedene komplexere Algorithmen und Datenstrukturen der Informatik in Theorie und praktischer Anwendung. Zur Implementierung der Lösungen können beliebige Programmiersprachen eingesetzt werden. Es werden exemplarisch die folgenden Themenschwerpunkte in Theorie und Praxis behandelt:

- Algorithmusbegriff, Datenstrukturen
- Stacks, Queues, Listen (mit Optimierungen)
- Graphen & verschiedene Algorithmen auf Graphen
- Verschiedene Bäume mit jeweiligen Vor- und Nachteilen
- Hashmaps und Sondierungsstrategien
- Monte-Carlo- und Las-Vegas-Algorithmen
- Evolutionäre Algorithmen
- Verschlüsselungsalgorithmen und Datenschutz
- Dezentrale Software und Blockchain-Datenstrukturen

Prüfung

Verpflichtende Voraussetzung gemäß SPO für die Teilnahme an der Prüfung
Keine

Art der Prüfung
Schriftliche Prüfung (sP) gemäß § 23 APO

Dauer/Form der Prüfung
90 Minuten

Die konkrete Festlegung der abzuleistenden Prüfung erfolgt im Studienplan

Prüfungssprache
Deutsch

Voraussetzung für die Vergabe von Leistungspunkten
Keine

Lernergebnisse

- Die Studierenden benennen und charakterisieren grundlegende Datenstrukturen und Algorithmen, einschließlich graph- und baumbasierter Verfahren.
- Die Studierenden erläutern typische Einsatzszenarien von Algorithmen anhand praxisnaher Beispiele.
- Die Studierenden wählen für gegebene Problemstellungen geeignete Datenstrukturen und Algorithmen aus.
- Die Studierenden analysieren die Leistung und Skalierbarkeit eingesetzter Algorithmen.
- Die Studierenden implementieren Algorithmen in einer Programmiersprache und testen deren Funktionalität.
- Die Studierenden bewerten verschiedene Lösungsansätze im Hinblick auf Effizienz und Anwendbarkeit.
- Die Studierenden entwickeln eigenständig algorithmische Lösungen für spezifische Anwendungsfälle.

Literatur

- Saake, Gunter; Sattler, Kai-Uwe: Algorithmen und Datenstrukturen, eine Einführung mit Java; 6. überarb. Aufl.; dpunkt-Verlag; Heidelberg, 2020
- Cormen, T., Leiseren, C., Riverest, R., Stein, C.: Algorithmen – Eine Einführung; 3. Aufl.; Oldenburg Verlag, 2010
- Fertig, Tobias; Schütz, Andreas. Blockchain the Comprehensive Guide. Rheinwerk Computing, 2024

fiw.thws
Fakultät Informatik und Wirtschaftsinformatik
Technische Hochschule Würzburg-Schweinfurt

Modul: 1510900
Analyse und Design

Modulprofil

Prüfungsnummer
1510900

Dauer
1 Semester

Häufigkeit des Angebots
Jedes Sommersemester

SWS
4

ECTS-Credits (CP)
5.0

Workload
Angeleitete Studienzeit:
Synchrone Teilnahme: 1 Std.
Asynchrone Teilnahme: 3 Std.

Selbststudienzeit: 90 Std.

Gesamt: 150 Std.

Lehrveranstaltungsart(en)
Seminaristischer Unterricht, Übung

Lehrsprache
Deutsch

Organisation

Modulverantwortung
Prof. Dr. Isabel John

Dozierende
Prof. Dr. Isabel John,
Prof. Dr.-Ing. Tobias Fertig,
Prof. Dr.-Ing. Anne Heß

Verwendbarkeit	BSED
Studiensemester	2. Semester
Art des Moduls	Pflichtmodul

Verpflichtende Voraussetzungen gemäß SPO
keine

Empfohlene Voraussetzungen
- Introduction to Software Engineering
- Programmieren 1

Inhalte
Dieses Modul baut auf den Grundlagen aus "Introduction to Software Engineering" auf und vermittelt Methoden und Techniken zur systematischen Analyse und dem strukturierten Entwurf von Softwaresystemen. Studierende lernen, komplexe Anforderungen zu analysieren, Softwarearchitekturen zu entwerfen und modellgestützte Entwicklungsansätze anzuwenden. Folgende Themen werden behandelt:

Anforderungsanalyse und Spezifikation
- Erhebung, Dokumentation und Validierung von Anforderungen
- Anwendung von Use Cases, User Stories und Szenarien
- Techniken zur Strukturierung und Priorisierung von Anforderungen

Grundlagen des Design
- SOLID-Prinzipien und Clean Code Design
- Design Patterns (z. B. Factory, Singleton, Observer)
- Modellierung und Entwurf von Softwarearchitekturen mit UML
- Datenmodellierung und Schnittstellenentwurf
- Detaillierte UML-Modellierung (z. B. Komponentendiagramme, Sequenzdiagramme, Deployment Diagramme)
- Einführung in Architekturmuster wie MVC und Standardarchitekturen (Schichten, Client-Server, Microservices)
- Abstraktionsebenen: Vom logischen zum physischen Design

Werkzeuge und Dokumentation
- Nutzung von CASE-Tools zur Modellierung
- Dokumentationstechniken für Architekturentscheidungen

Prüfung

Verpflichtende Voraussetzung gemäß SPO für die Teilnahme an der Prüfung
Keine

Art der Prüfung
Sonstige Prüfung (soP) gemäß §§ 26, 27 APO

Dauer/Form der Prüfung
Portfolio, Schriftliche Prüfung

Die konkrete Festlegung der abzuleistenden Prüfung erfolgt im Studienplan

Prüfungssprache
Deutsch

Voraussetzung für die Vergabe von Leistungspunkten
Keine

Lernergebnisse

- Die Studierenden kennen verschiedene Aspekte der Anforderungsanalyse
- Die Studierenden analysieren Anforderungen systematisch, strukturieren sie und dokumentieren diese.
- Die Studierenden konzipieren Softwarearchitekturen basierend auf Entwurfsmustern.
- Die Studierenden entwerfen Datenbank- und API-Schnittstellen für Anwendungen.
- Die Studierenden dokumentieren und präsentieren Architekturentscheidungen begründet.
- Die Studierenden kennen unterschiedliche Architekturmuster und wenden sie gezielt an.
- Die Studierenden beherrschen Methoden des objektorientierten Designs im Softwareentwurf.

Literatur

- Sommerville, Ian: Software Engineering, Pearson, 2018
- Oestereich, Bernd: Analyse und Design mit UML 2.5, Oldenbourg, 2013/2020
- Rupp, Chris: UML glasklar, Hanser, 2012
- McLaughlin: Objektorientierte Analyse und Design von Kopf bis Fuß, O'Reilly, 2017

fiw.thws
Fakultät Informatik und Wirtschaftsinformatik
Technische Hochschule Würzburg-Schweinfurt

Modul: 1511000
Netzwerke

Modulprofil

Prüfungsnummer
1511000

Dauer
1 Semester

Häufigkeit des Angebots
Jedes Sommersemester

SWS
4

ECTS-Credits (CP)
5.0

Workload
Angeleitete Studienzeit:
Synchrone Teilnahme: 1 Std.
Asynchrone Teilnahme: 3 Std.

Selbststudienzeit: 90 Std.

Gesamt: 150 Std.

Lehrveranstaltungsart(en)
Seminaristischer Unterricht, Übung

Lehrsprache
Englisch

Organisation

Modulverantwortung
Prof. Dr. Rolf Schillinger

Dozierende
Prof. Dr. Rolf Schillinger

Verwendbarkeit	BSED
Studiensemester	2. Semester
Art des Moduls	Pflichtmodul

Verpflichtende Voraussetzungen gemäß SPO
Keine

Empfohlene Voraussetzungen
Grundlagen Informatik

Inhalte
In this module, students gain a foundational understanding of networking technologies, their relevance for software engineering, and key security aspects of interconnected systems.
The course follows a layered approach, focusing on the four layers of the Internet Protocol Suite (TCP/IP) as the foundation of modern networking. Key topics include:

- Foundations: Basic networking concepts, ISO/OSI reference model as the conceptual grounding for modern networking
- TCP/IP concepts: Link layer and its protocols like Ethernet (IEEE 802.3) and Wi-Fi (IEEE 802.11), auxiliary protocols like (R)ARP, NDP, Internet layer with IPv4, IPv6, ICMP, IPsec, and routing basics (IGP, BGP), Transport layer with TCP and UDP, short introduction to QUIC, Application protocols with HTTP, HTTPS, DNS, and DHCP
- Networking devices, for example, NICs, hubs, switches, bridges, and routers
- Tools like Wireshark, network simulation via GNS3
- Mobile networking technologies such as 4G, LTE, and 5G
- Security considerations across all networking layers like ARP spoofing, IPSec, TLS attacks, firewalls, IDS and IPS

Prüfung

Verpflichtende Voraussetzung gemäß SPO für die Teilnahme an der Prüfung
Keine

Art der Prüfung
Schriftliche Prüfung (sP) gemäß § 23 APO

Dauer/Form der Prüfung
90 Minuten

Die konkrete Festlegung der abzuleistenden Prüfung erfolgt im Studienplan

Prüfungssprache
Englisch

Voraussetzung für die Vergabe von Leistungspunkten
Keine

Lernergebnisse

- Students will be able to describe and explain the principles of layered network architectures
- Students will be able to define and explain the key functions of each of TCP/IP's layers
- Students will be able to design a scalable best-practice internetworking architecture for a fictional global company, incorporating appropriate link layer protocols, network hardware, subnetting, and routing protocols
- Students will be able to construct and operate a medium-sized TCP/IP network in a virtual lab environment
- Students will be able to evaluate a given network's structure and assess its effectiveness in meeting specified functional or performance requirements

Literatur

- Tanenbaum, A. S., Wetherall, D. J., & Feamster, N. (2021). Computer networks (6th Global ed.). Pearson Education.
- Kurose, J. F., & Ross, K. W. (2021). Computer networking: A top-down approach (8th ed.). Pearson.
- Stevens, W. R., & Fall, K. R. (2011). TCP/IP illustrated, Volume 1: The protocols (2nd ed.). Addison-Wesley Professional.
- Stallings, W. (2017). Network security essentials: Applications and standards (6th ed.). Pearson.
- Grigorik, I. (2013). High performance browser networking: What every web developer should know about networking and web performance. O'Reilly Media.

fiw.thws
Fakultät Informatik und Wirtschaftsinformatik
Technische Hochschule Würzburg-Schweinfurt

Modul: 1510800
Programmieren 2

Modulprofil

Prüfungsnummer
1510800

Dauer
1 Semester

Häufigkeit des Angebots
Jedes Sommersemester

SWS
4

ECTS-Credits (CP)
5.0

Workload
Angeleitete Studienzeit:
Synchrone Teilnahme: 1 Std.
Asynchrone Teilnahme: 3 Std.

Selbststudienzeit: 90 Std.

Gesamt: 150 Std.

Lehrveranstaltungsart(en)
Seminaristischer Unterricht, Übung

Lehrsprache
Deutsch

Organisation

Modulverantwortung
Prof. Dr. Peter Braun

Dozierende
Prof. Dr. Peter Braun

Verwendbarkeit	BSED
Studiensemester	2. Semester
Art des Moduls	Pflichtmodul

Verpflichtende Voraussetzungen gemäß SPO
Keine

Empfohlene Voraussetzungen
Programmieren 1

Inhalte
Das Modul "Programmieren 2" baut auf den im ersten Semester erworbenen Kenntnissen in Python und objektorientierter Programmierung (OOP) auf und führt in die Programmiersprache Kotlin ein. Dabei werden sowohl die Sprachkonzepte von Kotlin als auch moderne Softwareentwicklungsprinzipien vermittelt. Die Schwerpunkte des Moduls sind:

- Syntax und Konzepte von Kotlin: Variablen, Kontrollstrukturen, Funktionen und Lambdas
- Objektorientierte Programmierung in Kotlin: Klassen, Objekte, Vererbung, Interfaces und Datenklassen
- Funktionale Programmieransätze: Higher-Order-Funktionen, Immutability und Collections-APIs
- Kotlin-spezifische Konzepte: Null-Safety, Extension Functions, Smart Casts und Coroutines
- Testgetriebene Entwicklung (TDD) mit Kotlin und Unit-Tests
- Fehlerbehandlung mit Exceptions und idiomatischen Ansätzen in Kotlin
- Einführung in moderne Softwareentwicklung mit Kotlin: Nutzung von Build-Tools (Gradle), Abhängigkeitsmanagement, einfache Anwendung von Design Patterns
- Projektarbeit: Umsetzung einer praxisnahen Anwendung unter Verwendung von Kotlin
- Praktische Übungen und Projekte begleiten die theoretischen Inhalte, sodass die Studierenden eigenständig Kotlin-Programme entwickeln und deren Qualität durch Tests und Code-Reviews verbessern können.

Prüfung

Verpflichtende Voraussetzung gemäß SPO für die Teilnahme an der Prüfung
Keine

Art der Prüfung
Schriftliche Prüfung (sP) gemäß § 23 APO

Dauer/Form der Prüfung
90 Minuten

Die konkrete Festlegung der abzuleistenden Prüfung erfolgt im Studienplan

Prüfungssprache
Deutsch

Voraussetzung für die Vergabe von Leistungspunkten
Keine

Lernergebnisse
Nach erfolgreichem Abschluss des Moduls sind die Studierenden in der Lage:

- Grundlegende und fortgeschrittene Kotlin-Sprachkonzepte zu verstehen und anzuwenden.
- Objektorientierte Prinzipien in Kotlin zu nutzen und eigene Klassenhierarchien zu entwerfen.
- Funktionale Programmierparadigmen in Kotlin anzuwenden, um sauberen und effizienten Code zu schreiben.
- Null-Sicherheit und Typsysteme von Kotlin zu verstehen und effektiv einzusetzen.
- Testgetriebene Entwicklung (TDD) und Unit-Testing mit Kotlin durchzuführen, um robuste Software zu entwickeln.
- Effiziente Fehlerbehandlung unter Nutzung idiomatischer Kotlin-Techniken zu implementieren.
- Kotlin-Projekte mit modernen Build-Tools (z. B. Gradle) zu verwalten und Abhängigkeiten sinnvoll einzubinden.
- Eine praxisnahe Softwarelösung in Kotlin zu konzipieren, zu implementieren und zu dokumentieren.
- Bestehenden Kotlin-Code zu analysieren und auf Qualität, Lesbarkeit und Effizienz zu überprüfen.
- Die Unterschiede und Gemeinsamkeiten zwischen Python und Kotlin zu reflektieren und für unterschiedliche Anwendungsfälle abzuwägen.

Literatur
- Kofler, Michael. Kotlin: Das Umfassende Handbuch. 1. Auflage. Bonn: Rheinwerk, 2021.
- Szwillus, Karl. Kotlin: Einstieg Und Praxis. 1. Auflage. Frechen: mitp, 2020.

fiw.thws
Fakultät Informatik und Wirtschaftsinformatik
Technische Hochschule Würzburg-Schweinfurt

Modul: 1510700
Projekt 1

Modulprofil

Prüfungsnummer
1510700

Dauer
1 Semester

Häufigkeit des Angebots
Jedes Sommersemester

SWS
1

ECTS-Credits (CP)
5.0

Workload
Angeleitete Studienzeit:
Synchrone Teilnahme: 1 Std.
Asynchrone Teilnahme: 0 Std.

Selbststudienzeit: 135 Std.

Gesamt: 150 Std.

Lehrveranstaltungsart(en)
Projekt

Lehrsprache
Deutsch

Organisation

Modulverantwortung
Prof. Dr. Peter Braun

Dozierende
Prof. Dr. Peter Braun,
Prof. Dr. Isabel John,
Michael Rott,
Prof. Dr. Rolf Schillinger,
Prof. Dr.-Ing. Tobias Fertig,
Prof. Dr. Frank-Michael Schleif,
Prof. Dr.-Ing. Sebastian Biedermann,
Prof. Dr. Tristan Wimmer,
Prof. Dr.-Ing. Anne Heß

Verwendbarkeit	BSED
Studiensemester	2. Semester
Art des Moduls	Pflichtmodul

Verpflichtende Voraussetzungen gemäß SPO
Keine

Empfohlene Voraussetzungen
Programmieren 1, IT-Projektmanagement und BWL, Introduction to Software Engineering, Grundlagen Informatik

Inhalte
Die Studierenden entwickeln in einer Gruppe mit drei bis vier Personen eine Konsolenanwendung gemäß vorgegebener Anforderungen. Die Anwendung soll einen nicht-trivialen Algorithmus enthalten (z. B. Such-, Sortier-, Graphen- oder Optimierungsalgorithmen) und mit Dateien arbeiten. Die Software besteht aus mehreren Modulen, jedoch ohne eine umfassende Softwarearchitektur. Zu den zentralen Themen gehören:

- Modularisierung des Codes
- Effiziente Implementierung eines Algorithmus
- Nutzung von Versionskontrolle (z. B. Git)
- Grundlegende Code-Dokumentation (README, Funktionskommentare)
- Einfache Tests zur Validierung des Algorithmus
- Einfache Aufgabenverwaltung und -verteilung im Team

Die Studierenden werden während der Projektbearbeitung kontinuierlich von einer Dozentin oder einem Dozenten betreut. Die Betreuung erfolgt durch regelmäßige Treffen, in denen der Arbeitsfortschritt besprochen wird, sowie durch begleitende Zwischenpräsentationen, in denen die Entwicklung des Projekts reflektiert und weiterentwickelt wird.

Prüfung

Verpflichtende Voraussetzung gemäß SPO für die Teilnahme an der Prüfung
Keine

Art der Prüfung
Sonstige Prüfung (soP) gemäß §§ 26, 27 APO

Dauer/Form der Prüfung
Präsentation, Dokumentation

Die konkrete Festlegung der abzuleistenden Prüfung erfolgt im Studienplan

Prüfungssprache
Deutsch

Voraussetzung für die Vergabe von Leistungspunkten
Keine

Lernergebnisse
Nach dem erfolgreichen Abschluss des Moduls haben die Studierenden die folgenden Kompetenzen:

- Die Studierenden erinnern die grundlegenden Konzepte der Softwareentwicklung und Versionskontrolle.
- Die Studierenden verstehen die Prinzipien modularer Softwareentwicklung und die Strukturierung von Projekten.
- Die Studierenden wenden Git und Versionskontrolle in einem Teamkontext an.
- Die Studierenden analysieren nicht-triviale Algorithmen hinsichtlich Laufzeit und Effizienz.
- Die Studierenden bewerten verschiedene Implementierungsansätze für eine gegebene Problemstellung.
- Die Studierenden erstellen eine kleine, modulare Konsolenanwendung nach vorgegebenen Anforderungen.
- Die Studierenden erstellen eine technische Präsentation zur Vorstellung ihrer Lösung.

Literatur
Wird in Abhängigkeit von den konkreten Projekten und verwendeten Technologie in der Veranstaltung bekannt gegeben.

fiw.thws
Fakultät Informatik und Wirtschaftsinformatik
Technische Hochschule Würzburg-Schweinfurt

Modul: 1511200
Stochastik

Modulprofil

Prüfungsnummer
1511200

Dauer
1 Semester

Häufigkeit des Angebots
Jedes Sommersemester

SWS
4

ECTS-Credits (CP)
5.0

Workload
Angeleitete Studienzeit:
Synchrone Teilnahme: 1 Std.
Asynchrone Teilnahme: 3 Std.

Selbststudienzeit: 90 Std.

Gesamt: 150 Std.

Lehrveranstaltungsart(en)
Seminaristischer Unterricht, Übung

Lehrsprache
Deutsch

Organisation

Modulverantwortung
Prof. Dr. Patrik Stilgenbauer

Dozierende
Prof. Dr. Patrik Stilgenbauer

Verwendbarkeit	BSED
Studiensemester	2. Semester
Art des Moduls	Pflichtmodul

Verpflichtende Voraussetzungen gemäß SPO
keine

Empfohlene Voraussetzungen
Algebra: Grundkenntnisse der linearen Algebra (insb. lineare Gleichungssysteme, Matrizenalgebra, Vektorräume, Skalarprodukt), Aussagen- und Mengenalgebra, Kombinatorik.
Programmieren I: Programmierlogik, Entwurf einfacher Algorithmen.
Schulkenntnisse aus der Analysis vorausgesetzt (Differential- und Integralrechnung).

Inhalte
- Deskriptive Statistik: Grundbegriffe, Häufigkeitsverteilungen, Lageparameter, Streuungsparameter, lineare Korrelations- und Regressionsrechnung.
- Wahrscheinlichkeitstheorie: Ergebnismenge, Ereignisse, Wahrscheinlichkeitsbegriff von Kolmogorow, bedingte Wahrscheinlichkeit und Unabhängigkeit, diskrete und stetige Zufallsvariablen, Erwartungswert und Varianz, Gesetz der großen Zahlen, Binomialverteilung, hypergeometrische Verteilung, Poissonverteilung, Exponentialverteilung, Normalverteilung, Summen von Zufallsvariablen, zentraler Grenzwertsatz.
- Schließende Statistik: Punkt- und Intervallschätzungen, Signifikanztests.
- Anwendung und Visualisierung stochastischer Methoden durch Programmierbeispiele in Python oder R.

Prüfung

Verpflichtende Voraussetzung gemäß SPO für die Teilnahme an der Prüfung
Keine

Art der Prüfung
Schriftliche Prüfung (sP) gemäß § 23 APO

Dauer/Form der Prüfung
90 Minuten

Die konkrete Festlegung der abzuleistenden Prüfung erfolgt im Studienplan

Prüfungssprache
Deutsch

Voraussetzung für die Vergabe von Leistungspunkten
Keine

Lernergebnisse
Nach Abschluss des Moduls sind die Studierenden in der Lage:

- grundlegende Konzepte und Begriffe der Wahrscheinlichkeitsrechnung und Statistik sicher anzuwenden,
- statistische Kennzahlen und Wahrscheinlichkeiten zu berechnen, zu interpretieren und zur Beschreibung von Daten einzusetzen,
- typische Wahrscheinlichkeitsverteilungen auszuwählen und auf geeignete, praxisnahe Problemstellungen anzuwenden,
- stochastische und statistische Fragestellungen mit logisch-strukturiertem Denken zu analysieren und lösungsorientiert zu bearbeiten,
- die Methoden als Grundlage für weiterführende Anwendungen in Softwareentwicklung, Data Science und Machine Learning zu nutzen.

Literatur

- Bamberg, G., Baur, F. und Krapp, M.: Statistik, De Gruyter Oldenbourg, 2022.
- Bourier, G.: Beschreibende Statistik, Springer Gabler, 2025.
- Bourier, G.: Wahrscheinlichkeitsrechnung und schließende Statistik, Springer Gabler, 2018.
- Dreiseitl, S.: Mathematik für Software Engineering, Springer Vieweg, 2018.
- Henze, N.: Stochastik für Einsteiger, Vieweg + Teubner, 2011.
- Kurt, N.: Stochastik für Informatiker, Springer Vieweg, 2020.
- Teschl, G. und Teschl, S.: Mathematik für Informatiker - Band 2 (Analysis und Stochastik), Springer Vieweg, 2014.

3. Semester

fiw.thws
Fakultät Informatik und Wirtschaftsinformatik
Technische Hochschule Würzburg-Schweinfurt

Modul: 1511600
Backend Systems

Modulprofil

Prüfungsnummer
1511600

Dauer
1 Semester

Häufigkeit des Angebots
Jedes Wintersemester

SWS
4

ECTS-Credits (CP)
5.0

Workload
Angeleitete Studienzeit:
Synchrone Teilnahme: 1 Std.
Asynchrone Teilnahme: 3 Std.

Selbststudienzeit: 90 Std.

Gesamt: 150 Std.

Lehrveranstaltungsart(en)
Seminaristischer Unterricht, Übung

Lehrsprache
Englisch

Organisation

Modulverantwortung
Prof. Dr. Peter Braun

Dozierende
Prof. Dr. Peter Braun

Verwendbarkeit	BSED
Studiensemester	3. Semester
Art des Moduls	Pflichtmodul

Verpflichtende Voraussetzungen gemäß SPO
None

Empfohlene Voraussetzungen
Programmieren 1 und Programmieren 2

Inhalte

- Introduction to distributed systems, client-server, and peer-to-peer systems.
- Software architectures for backend systems (3-tier, hexagonal, monolithic vs. micro-service, event-driven)
- Frameworks to implement backend systems (e.g. Spring)
- Advanced database techniques, scalability, replication, sharding, ORM-tools, query caching, CAP theorem
- Protocols for remote procedure call, for example, GraphQL and Google RPC.
- Basics of the HTTP protocol and application in the form of Web APIs.
- Comprehensive introduction to the REST architecture principle: resources, URLs, CRUD, hypermedia, caching, security.
- Configuration of Web servers (Apache), load balancer, and public caches (nginx)
- Testing of backend systems, performance testing using JMeter, monitoring and logging
- Security aspects of network protocol and backend systems

In the traditional degree programme, the lecturer provides or agrees with the topics of the practical examples for the examination. In the dual study programme, the lecturer consults with the company on a task, ensuring practical relevance and feedback from the company.

Prüfung

Verpflichtende Voraussetzung gemäß SPO für die Teilnahme an der Prüfung
Keine

Art der Prüfung
Sonstige Prüfung (soP) gemäß §§ 26, 27 APO

Dauer/Form der Prüfung
Portfolio

Die konkrete Festlegung der abzuleistenden Prüfung erfolgt im Studienplan

Prüfungssprache
Englisch

Voraussetzung für die Vergabe von Leistungspunkten
Keine

Lernergebnisse

- The students understand the fundamental concepts and differences of distributed systems, including their architecture and communication models.
- The students analyze various software architectures for backend systems and evaluate their suitability for different use cases.
- The students apply advanced database techniques such as replication and sharding to enhance data availability and performance.
- The students implement a backend system using a framework like Spring, following best practices for configuration, deployment, and security.
- The students compare different protocols for remote procedure calls, such as GraphQL and Google RPC, assessing their strengths and limitations.
- The students design RESTful APIs by applying the principles of the REST architecture, focusing on resources, URLs, CRUD operations, and security strategies.
- The students evaluate the security aspects of network protocols and backend systems, proposing improvements based on best practices.

Literatur

- Coulouris, J. Dollimore, and T. Kindberg, Distributed Systems: Concepts and Design (4th Edition) (International Computer Science). Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 2005.
- N. Biswas, Practical GraphQL: Learning Full-Stack GraphQL Development with Projects. Berkeley, CA: Apress, 2023.
- J. Webber, S. Parastatidis, und I. Robinson, REST in practice: hypermedia and systems architecture, 1. ed. in Theory in practice. Beijing Köln: O'Reilly, 2010.
- L. Richardson und M. Amundsen, RESTful Web APIs, First edition, Second release. Beijing Cambridge Farnham Köln Sebastopol Tokyo: O'Reilly, 2015.
- I. Dominte, Web API Development for the Absolute Beginner: A Step-by-step Approach to Learning the Fundamentals of Web API Development with .NET 7. Berkeley, CA: Apress, 2023.

fiw.thws
Fakultät Informatik und Wirtschaftsinformatik
Technische Hochschule Würzburg-Schweinfurt

Modul: 1511700
Data Science

Modulprofil

Prüfungsnummer
1511700

Dauer
1 Semester

Häufigkeit des Angebots
Jedes Wintersemester

SWS
4

ECTS-Credits (CP)
5.0

Workload
Angeleitete Studienzeit:
Synchrone Teilnahme: 1 Std.
Asynchrone Teilnahme: 3 Std.

Selbststudienzeit: 90 Std.

Gesamt: 150 Std.

Lehrveranstaltungsart(en)
Seminaristischer Unterricht, Übung

Lehrsprache
Englisch

Organisation

Modulverantwortung
Prof. Dr. Frank-Michael Schleif

Dozierende
Prof. Dr. Frank-Michael Schleif

Verwendbarkeit	BSED
Studiensemester	3. Semester
Art des Moduls	Pflichtmodul

Verpflichtende Voraussetzungen gemäß SPO
Keine

Empfohlene Voraussetzungen
Datenbanken, Programmieren 1, Introduction to Software Engineering, Programmieren 2

Inhalte
This module provides a foundational introduction to data science within software engineering, focusing on data-driven development, data management,
and practical analysis techniques. Students gain both theoretical knowledge and hands-on experience across the full data lifecycle.

Core Topics

Data Science & Data Literacy:
- Role of data in modern software systems
- Data-driven decision-making in engineering
- Data ethics, privacy, and compliance (e.g., GDPR)
- Data quality and bias awareness

Semi-Structured Data (XML & JSON):
- XML/JSON as flexible data formats
- Schema definitions (DTD, XML Schema)
- Querying techniques (XPath, XSLT, JSONPath)
- Real-world applications in software systems

Data Integration & Warehousing:
- Multidimensional modeling (star/snowflake schemas)
- ETL processes and data pipelines
- Data integration from relational sources and APIs
- OLAP and basic exposure to NoSQL/Big Data tools

Graph Databases & Networked Data:
- Basics of graph theory in data modeling
- Graph data structures and query languages (e.g., Cypher)
- Applications: recommendation engines, social networks

Practical Data-Driven Solutions:
- Design and implementation of software with integrated data flows
- Performance considerations in data handling
- Security, privacy, and compliance in data processing

Prüfung

Verpflichtende Voraussetzung gemäß SPO für die Teilnahme an der Prüfung
Keine

Art der Prüfung
Sonstige Prüfung (soP) gemäß §§ 26, 27 APO

Dauer/Form der Prüfung
Schriftliche Prüfung, Portfolio

Die konkrete Festlegung der abzuleistenden Prüfung erfolgt im Studienplan

Prüfungssprache
Englisch

Voraussetzung für die Vergabe von Leistungspunkten
Keine

Lernergebnisse
Upon successful completion, students will be able to:
Knowledge and Understanding:

- Explain core principles of data management and their application in software systems
- Describe key data models (relational, semi-structured, graph) and related technologies (XML, JSON, XPath, XSLT, Cypher)
- Understand data integration, ETL, warehousing, and NoSQL concepts
- Demonstrate awareness of data ethics, privacy, and regulatory frameworks

Skills and Competences:

- Design and query structured, semi-structured, and graph-based data
- Build ETL workflows and apply OLAP techniques for analytical processing
- Apply graph-based analysis for networked data
- Evaluate data quality, security, and compliance in practical contexts

Literatur

- Skiena, S.S.; The Data Science Design Manual, Springer, 2017
- Robinson, I; Graph Databases 2nd Ed.; O'Reilly Media; 2015
- Friesen, Jeff; Java XML and JSON; 2019
- Brian Knight, Professional Microsoft SQL Server 2014 Integration Services (Wrox Programmer to Programmer), Wrox, 2014
- Trevor Hastie, The Elements of Statistical Learning, Springer, 2009
- Ralph Kimball, Margy Ross, Warren Thornthwaite, Joy Mundy, Bob Becker: The Data Warehouse Lifecycle Toolkit, 2nd Edition, Wiley 2008

fiw.thws
Fakultät Informatik und Wirtschaftsinformatik
Technische Hochschule Würzburg-Schweinfurt

Modul: 1511800
Professional Skills

Modulprofil

Prüfungsnummer
1511800

Dauer
1 Semester

Häufigkeit des Angebots
Jedes Wintersemester

SWS
4

ECTS-Credits (CP)
5.0

Workload
Angeleitete Studienzeit:
Synchrone Teilnahme: 1 Std.
Asynchrone Teilnahme: 3 Std.

Selbststudienzeit: 90 Std.

Gesamt: 150 Std.

Lehrveranstaltungsart(en)
Seminaristischer Unterricht, Übung

Lehrsprache
Deutsch

Organisation

Modulverantwortung
Prof. Dr. Isabel John

Dozierende
Prof. Dr. Peter Braun,
Prof. Dr. Isabel John,
Prof. Dr.-Ing. Tobias Fertig,
Prof. Dr. Frank-Michael Schleif,
Prof. Dr.-Ing. Anne Heß

Verwendbarkeit	BSED
Studiensemester	3. Semester
Art des Moduls	Pflichtmodul

Verpflichtende Voraussetzungen gemäß SPO
keine

Empfohlene Voraussetzungen
keine

Inhalte
In diesem Modul werden den Studierenden theoretische und praktische Kenntnisse und Fähigkeiten vermittelt, die in einem professionellen Arbeitsumfeld in verschiedensten Bereichen Anwendung finden.
Dabei erlernen und erproben die Studierenden eine Reihe von Methoden, Techniken, und Tools, die in verschiedene Schwerpunktthemen eingeordnet sind. Dazu gehören

- Lern- und Arbeitstechniken
- Wissenschaftliches Arbeiten
- Zielgruppenorientierte fachliche Kommunikation sowie
- Arbeiten in (internationalen) Teams

Prüfung

Verpflichtende Voraussetzung gemäß SPO für die Teilnahme an der Prüfung
Keine

Art der Prüfung
Sonstige Prüfung (soP) gemäß §§ 26, 27 APO

Dauer/Form der Prüfung
Portfolio

Die konkrete Festlegung der abzuleistenden Prüfung erfolgt im Studienplan

Prüfungssprache
Deutsch

Voraussetzung für die Vergabe von Leistungspunkten
Keine

Lernergebnisse

- Die Studierenden wenden Methoden zur effektiven Planung und Strukturierung ihrer Arbeitsprozesse an
- Die Studierenden benennen die Grundprinzipien des wissenschaftlichen Arbeitens in der Informatik
- Die Studierenden führen eine Literaturrecherche durch und organisieren die Ergebnisse
- Die Studierenden beschreiben detailliert die Grundlagen zur Gestaltung effektiver wissenschaftlicher Kommunikation in Form von Texten, Präsentationen, Poster, Videos
- Die Studierenden erstellen wissenschaftlich-orientierte Texte, Präsentationen, Poster, Videos auf der Basis wissenschaftlicher Standards
- Die Studierenden erlernen relevante Zielsetzungen, Fähigkeiten und Best Practices zur Planung, Durchführung und Nachbereitung verschiedener Befragungstechniken (wie Interviews, Umfragen)
- Die Studierenden wenden Methoden zur effektiven Kommunikation in Teams an
- Die Studierenden wenden Methoden der Gesprächsführung zur Ermittlung von Anforderungen an

Literatur
Wird in Vorlesung bekanntgegeben

fiw.thws
Fakultät Informatik und Wirtschaftsinformatik
Technische Hochschule Würzburg-Schweinfurt

Modul: 1511300
Projekt 2

Modulprofil

Prüfungsnummer
1511300

Dauer
1 Semester

Häufigkeit des Angebots
Jedes Wintersemester

SWS
1

ECTS-Credits (CP)
5.0

Workload
Angeleitete Studienzeit:
Synchrone Teilnahme: 1 Std.
Asynchrone Teilnahme: 0 Std.

Selbststudienzeit: 135 Std.

Gesamt: 150 Std.

Lehrveranstaltungsart(en)
Projekt

Lehrsprache
Deutsch

Organisation

Modulverantwortung
Prof. Dr. Peter Braun

Dozierende
Prof. Dr. Peter Braun,
Prof. Dr. Isabel John,
Michael Rott,
Prof. Dr. Rolf Schillinger,
Prof. Dr.-Ing. Tobias Fertig,
Prof. Dr. Frank-Michael Schleif,
Prof. Dr.-Ing. Sebastian Biedermann,
Prof. Dr. Tristan Wimmer,
Prof. Dr.-Ing. Anne Heß

Verwendbarkeit	BSED
Studiensemester	3. Semester
Art des Moduls	Pflichtmodul

Verpflichtende Voraussetzungen gemäß SPO
Keine

Empfohlene Voraussetzungen
Programmieren 1, Projekt 1, Programmieren 2, Introduction to Software Engineering, Analyse und Design, Algorithmen und Datenstrukturen

Inhalte
In Gruppen von vier bis fünf Personen entwickeln die Studierenden eine Anwendung mit dem Fokus auf das Backend und die Datenbank. Sie wenden Elemente des agilen Projektmanagement an. Die Studierenden erhalten ein Thema und ermitteln mit der betreuenden Personen die Anforderungen. Eine grafische Nutzeroberfläche ist nicht notwendig bzw. sollte sehr einfach gehalten werden. Die Software muss eine erkennbare Software Architektur (z.B. Schichten, Pipeline, Hexagonal) aufweisen. Die Qualitätssicherung erfolgt durch Unit-Tests. Die Studierenden arbeiten mit:

- Datenbank-Anbindung mit Schema-Entwurf (SQL)
- Erweiterten Software-Entwurfsmethoden (z. B. UML, Architektur-Patterns)
- Versionskontrolle mit Git (Branching, Pull Requests, Code Reviews)
- Sprint-basierten Entwicklungsprozessen (z. B. Scrum, Kanban)
- Die Dokumentation umfasst Architekturdiagramme, User Stories und eine automatische API-Dokumentation.
- Maßnahmen zur Qualitätssicherung durch automatisierte Unit-Tests

Die Studierenden werden während der Projektbearbeitung kontinuierlich von einer Dozentin oder einem Dozenten betreut. Die Betreuung erfolgt durch regelmäßige Treffen, in denen der Arbeitsfortschritt besprochen wird, sowie durch begleitende Zwischenpräsentationen, in denen die Entwicklung des Projekts reflektiert und weiterentwickelt wird.

Prüfung

Verpflichtende Voraussetzung gemäß SPO für die Teilnahme an der Prüfung
Keine

Art der Prüfung
Sonstige Prüfung (soP) gemäß §§ 26, 27 APO

Dauer/Form der Prüfung
Dokumentation, Präsentation

Die konkrete Festlegung der abzuleistenden Prüfung erfolgt im Studienplan

Prüfungssprache
Deutsch

Voraussetzung für die Vergabe von Leistungspunkten
Keine

Lernergebnisse
Nach erfolgreichem Abschluss des Moduls besitzen die Studierenden die folgenden Kompetenzen:

- Die Studierenden verstehen grundlegende Architekturprinzipien wie das Schichtenmodell und die Modularisierung.
- Die Studierenden wenden das Schichtenmodell und Modularisierungskonzepte bei der Entwicklung einer verteilten Anwendung mit Datenbankanbindung an.
- Die Studierenden wenden agile Methoden an, um Aufgaben in Sprints zu planen und umzusetzen.
- Die Studierenden analysieren Code-Änderungen im Rahmen von Pull Requests und führen Code Reviews im Team durch.
- Die Studierenden erstellen eine strukturierte Software-Dokumentation für ihre Anwendung.
- Die Studierenden erstellen eine technische Projektpräsentation zur Vorstellung ihrer Ergebnisse.

Literatur
Wird in Abhängigkeit von den konkreten Projekten und verwendeten Technologie in der Veranstaltung bekannt gegeben.

fiw.thws
Fakultät Informatik und Wirtschaftsinformatik
Technische Hochschule Würzburg-Schweinfurt

Modul: 1511500
Software Qualität

Modulprofil

Prüfungsnummer
1511500

Dauer
1 Semester

Häufigkeit des Angebots
Jedes Wintersemester

SWS
4

ECTS-Credits (CP)
5.0

Workload
Angeleitete Studienzeit:
Synchrone Teilnahme: 1 Std.
Asynchrone Teilnahme: 3 Std.

Selbststudienzeit: 90 Std.

Gesamt: 150 Std.

Lehrveranstaltungsart(en)
Seminaristischer Unterricht, Übung

Lehrsprache
Deutsch

Organisation

Modulverantwortung
Prof. Dr.-Ing. Anne Heß

Dozierende
Prof. Dr. Isabel John,
Prof. Dr.-Ing. Tobias Fertig,
Prof. Dr.-Ing. Anne Heß

Verwendbarkeit	BSED
Studiensemester	3. Semester
Art des Moduls	Pflichtmodul

Verpflichtende Voraussetzungen gemäß SPO
Keine

Empfohlene Voraussetzungen
Introduction to Software Engineering
Analyse und Design

Inhalte
Dieses Modul leistet eine umfassende Einführung in den Themenkomplex Software Qualität. Dabei werden sowohl technische Grundlagen als auch Prozess- , Management- und Usability Themen näher gebracht.

Die Inhalte umfassen die folgenden Themen:
- Softwarequalität & Software-Qualitätsmanagement
- Analytische Qualitätssicherung (Testen von Software)
- Testprozess, -aktivitäten und -artifakte
- Strategien zur Testfallermittlung (Black-Box Testing, White-Box Testing)
- Testarten (Funktionales Testen, Regressionstests, Performance Tests)
- Test-Driven Development
- Testing Tools
- Bug Tracking und Reporting
- Statisches Testen (Reviews & Statische Analyse)
- Konstruktive Qualitätssicherung am Beispiel Usability / User Experience
- Human-Centered Design und Usability Testing

Prüfung

Verpflichtende Voraussetzung gemäß SPO für die Teilnahme an der Prüfung
Keine

Art der Prüfung
Sonstige Prüfung (soP) gemäß §§ 26, 27 APO

Dauer/Form der Prüfung
Portfolio

Die konkrete Festlegung der abzuleistenden Prüfung erfolgt im Studienplan

Prüfungssprache
Deutsch

Voraussetzung für die Vergabe von Leistungspunkten
Keine

Lernergebnisse

- Die Studierenden verstehen Begrifflichkeiten der Software Qualität anhand eines Qualitätsmodells
- Die Studierenden können Qualitätsmodelle zur systematischen Ermittlung von Qualitätsanforderungen anwenden
- Die Studierenden verstehen relevante Aktivitäten und Artefakte im Testprozess
- Die Studierenden können im Kontext analytischer Qualitätssicherung Teststrategien entwickeln, Testfälle definieren und Testüberdeckungen ermitteln
- Die Studierenden können Testmethoden für funktionale und nicht-funktionale Anforderungen auswählen und anzuwenden.
- Die Studierenden können ein geeignetes Testmanagement etablieren und individuell auf die Anforderungen von Projekten abstimmen
- Die Studierenden verstehen die Bedeutung von Usability und User Experience
- Die Studierenden verstehen Zielsetzungen und Techniken zur konstruktiven Qualitätssicherung basierend auf dem Human-Centered Design Prozess
- Die Studierenden können die Techniken im Human-Centered Design zur Sicherstellung einer guten Usability / User Experience anwenden

Literatur

- Liggesmeyer, P. (2009). Software-Qualität. Springer-Verlag.
- Spillner, A., & Linz, T. (2019). Basiswissen Softwaretest: Aus- und Weiterbildung zum Certified Tester – Foundation Level nach ISTQB®-Standard. dpunkt.verlag.
- Schneider, K. (2012). Abenteuer Softwarequalität: Grundlagen und Verfahren für Qualitätssicherung und Qualitätsmanagement (2. Aufl.). dpunkt.verlag.

fiw.thws
Fakultät Informatik und Wirtschaftsinformatik
Technische Hochschule Würzburg-Schweinfurt

Modul: 1511400 System-oriented Programming

Modulprofil

Prüfungsnummer
1511400

Dauer
1 Semester

Häufigkeit des Angebots
Jedes Wintersemester

SWS
4

ECTS-Credits (CP)
5.0

Workload
Angeleitete Studienzeit:
Synchrone Teilnahme: 1 Std.
Asynchrone Teilnahme: 3 Std.

Selbststudienzeit: 90 Std.

Gesamt: 150 Std.

Lehrveranstaltungsart(en)
Seminaristischer Unterricht, Übung

Lehrsprache
Englisch

Organisation

Modulverantwortung
Prof. Dr. Peter Braun

Dozierende
Prof. Dr. Peter Braun

Verwendbarkeit	BSED
Studiensemester	3. Semester
Art des Moduls	Pflichtmodul

Verpflichtende Voraussetzungen gemäß SPO
None

Empfohlene Voraussetzungen
Programmieren 1 and Programmieren 2

Inhalte
- Definition and meaning of system-oriented programming
- Using the command line of an operating system
- Shell programming using the example of Bash
- Working on remote computers with ssh
- Data processing on the command line with sed, awk, sort, jq
- Using AI assistance systems (e.g. GitHub Copilot, ChatGPT)
- Editing text documents with vim
- Using the version control system Git
- Introduction to the C programming language (syntax, data types, pointers, memory management)
- Build systems make, cmake, bazel
- System programming under Linux (system calls, handling files, processes)
- Debugging, profiling and performance optimisation (gdb, strace, ltrace, gprof)
- Security aspects of system-related programming and protection mechanisms
- Structure of the Linux operating system
- Processes, process management, scheduling
- Inter-process communication, race conditions, deadlocks, semaphores, Petri nets and deadlock detection, philosopher problem, producer-consumer problem
- Memory management, memory abstraction, partitioning, fragmentation, free memory management, virtual memory, page exchange algorithms
- Input and output, direct memory access, interrupts, hard disks, file systems for hard disks
- Backup methods, automation, data integrity
- Network communication and implementation of network protocols
- Hypervisor technologies, Docker containers, resource management

Prüfung

Verpflichtende Voraussetzung gemäß SPO für die Teilnahme an der Prüfung
Keine

Art der Prüfung
Sonstige Prüfung (soP) gemäß §§ 26, 27 APO

Dauer/Form der Prüfung
Portfolio

Die konkrete Festlegung der abzuleistenden Prüfung erfolgt im Studienplan

Prüfungssprache
Englisch

Voraussetzung für die Vergabe von Leistungspunkten
Keine

Lernergebnisse

- The students understand the definition and principles of system-oriented programming, including its role in software development.
- The students demonstrate proficiency in using the command line of an operating system, applying advanced tools like sed, awk, sort, and jq for data processing tasks.
- The students develop shell scripts in Bash to automate system tasks and streamline operations effectively.
- The students utilize AI assistance systems, such as GitHub Copilot and ChatGPT, to improve coding efficiency and solve programming challenges.
- The students manage text documents using the vim text editor, employing advanced editing and configuration techniques.
- The students implement version control practices with Git to support collaborative software development workflows.
- The students program in C, focusing on syntax, data types, pointers, memory management, and use debugging and profiling tools like gdb, strace, ltrace, and gprof to analyze code performance.
- The students evaluate security aspects of system-related programming and apply protection mechanisms to ensure code security.

Literatur

- D. J. Barrett, Efficient Linux at the command line: boost your command-line skills, First edition. Sebastopol, CA: O'Reilly, 2022.
- A. S. Tanenbaum und H. Bos, Modern operating systems, 4th ed. Boston: Prentice Hall, 2015.
- K. Hitchcock, Linux System Administration for the 2020s: The Modern Sysadmin Leaving Behind the Culture of Build and Maintain. Berkeley, CA: Apress, 2022.
- M. Kalin, Modern C Up and Running: A Programmer's Guide to Finding Fluency and Bypassing the Quirks. Berkeley, CA: Apress, 2022.
- K. Hitchcock, The Enterprise Linux Administrator: Journey to a New Linux Career. Berkeley, CA: Apress, 2023.
- J. Varma, Pro Bash: Learn to Script and Program the GNU/Linux Shell. Berkeley, CA: Apress, 2023.
- S. M. Palakollu, Practical System Programming with C: Pragmatic Example Applications in Linux and Unix-Based Operating Systems. Berkeley, CA: Apress, 2021.

4. Semester

fiw.thws
Fakultät Informatik und Wirtschaftsinformatik
Technische Hochschule Würzburg-Schweinfurt

Modul: 9999999
Allgemeinwissenschaftliches Wahlpflichtmodul

Modulprofil

Prüfungsnummer
9999999

Dauer
1 Semester

Häufigkeit des Angebots
Jedes Semester

SWS
4

ECTS-Credits (CP)
5.0

Workload
Angeleitete Studienzeit:
Synchrone Teilnahme: 1 Std.
Asynchrone Teilnahme: 3 Std.

Selbststudienzeit: 90 Std.

Gesamt: 150 Std.

Lehrveranstaltungsart(en)
Seminaristischer Unterricht

Lehrsprache
Deutsch/Englisch

Organisation

Modulverantwortung
Prof. Dr. Jochen Seufert

Dozierende
Beate Wassermann

Verwendbarkeit	BSED
Studiensemester	4. Semester
Art des Moduls	AWPM

Verpflichtende Voraussetzungen gemäß SPO
i. d. R. keine; Ausnahmen werden durch die Fakultät Angewandte Natur- und Geisteswissenschaften festgelegt und bekanntgegeben.

Empfohlene Voraussetzungen
keine

Inhalte
Auswahl von zwei Allgemeinwissenschaftlichen Wahlpflichtfächern (AWPF) (2 x 2,5 ECTS) bzw. einem AWPF (1 x 5 ECTS) aus dem Fächerangebot der Fakultät Angewandte Natur- und Geisteswissenschaften (FANG).

Fächerangebot der FANG aus den Bereichen
- Sprachen
- Kulturwissenschaften
- Naturwissenschaften und Technik
- Politik, Recht und Wirtschaft
- Pädagogik, Psychologie und Sozialwissenschaften
- Soft Skills
- Kreativität und Kunst.

Ausgeschlossen aus dem Angebotskatalog der FANG sind Veranstaltungen, deren Inhalte bereits Bestandteile oder unmittelbar fachlich verwandt mit Teilen anderer Module des Studiengangs sind. Die entsprechenden Veranstaltungen sind im Fächerkatalog der FANG mit einem Sperrvermerk versehen. Die Inhalte der einzelnen AWPFs sind auf der fakultätseigenen Homepage der FANG veröffentlicht.

Prüfung

Verpflichtende Voraussetzung gemäß SPO für die Teilnahme an der Prüfung
Keine

Art der Prüfung
Schriftliche Prüfung (sP) gemäß § 23 APO

Dauer/Form der Prüfung
90 Minuten

Die konkrete Festlegung der abzuleistenden Prüfung erfolgt im Studienplan

Prüfungssprache
Deutsch/Englisch

Voraussetzung für die Vergabe von Leistungspunkten
Keine

Lernergebnisse
Die fachspezifischen Lernziele sind abhängig von den jeweils ausgewählten AWPF. Die Studierenden

- erwerben zudem Wissen und Kompetenzen, die nicht fachspezifisch sind, aber für das angestrebte Berufsziel bedeutsam sein können wie beispielsweise spezielle Kenntnisse bei Fremdsprachen, in naturwissenschaftlichen oder auch in sozialwissenschaftlichen Gebieten
- analysieren unterschiedlichste Fragestellungen
- ordnen das fachspezifische Wissen in einen interdisziplinären Zusammenhang ein
- übertragen das Gelernte auf die aktuelle Ausbildung
- haben ihre Schlüsselkompetenzen und ggf. Fremdsprachenkompetenzen erweitert, wodurch die Persönlichkeitsbildung unterstützt wird, auch in interkultureller Hinsicht
- sind sich ihrer Verantwortung in persönlicher, gesellschaftlicher und ethischer Hinsicht bewusst.

Literatur
je nach gewählten AWPFs

fiw.thws
Fakultät Informatik und Wirtschaftsinformatik
Technische Hochschule Würzburg-Schweinfurt

Modul: 1512200
IT-Sicherheit

Modulprofil

Prüfungsnummer
1512200

Dauer
1 Semester

Häufigkeit des Angebots
Jedes Sommersemester

SWS
4

ECTS-Credits (CP)
5.0

Workload
Angeleitete Studienzeit:
Synchrone Teilnahme: 1 Std.
Asynchrone Teilnahme: 3 Std.

Selbststudienzeit: 90 Std.

Gesamt: 150 Std.

Lehrveranstaltungsart(en)
Seminaristischer Unterricht, Übung

Lehrsprache
Deutsch

Organisation

Modulverantwortung
Prof. Dr.-Ing. Sebastian Biedermann

Dozierende
Prof. Dr.-Ing. Sebastian Biedermann

Verwendbarkeit	BSED
Studiensemester	4. Semester
Art des Moduls	Pflichtmodul

Verpflichtende Voraussetzungen gemäß SPO
Keine

Empfohlene Voraussetzungen
Keine

Inhalte
Das Modul "IT-Sicherheit" vermittelt grundlegende Konzepte und praktische Ansätze zur Sicherung von IT-Systemen und Softwareanwendungen. Studierende lernen zentrale Begriffe, Bedrohungsszenarien und Angriffsvektoren kennen und verstehen die Sicherheitsziele Vertraulichkeit, Integrität und Verfügbarkeit. Ein Schwerpunkt liegt auf der Kryptografie, einschließlich symmetrischer und asymmetrischer Verschlüsselungsverfahren, digitaler Signaturen, Zertifikaten sowie Public Key Infrastrukturen (PKI) und Alternativen wie Transport-Layer-Security (TLS). Darüber hinaus werden Schwachstellen in Webanwendungen, wie SQL-Injections, sowie in klassischen Anwendungen, wie Buffer-Overflows, analysiert und Gegenmaßnahmen entwickelt. Das Modul behandelt auch Authentifizierungs- und Autorisierungskonzepte, einschließlich Multi-Faktor-Authentifizierung, Rollen- und Rechtekonzepte sowie Single Sign-On (SSO). Ein weiterer wichtiger Bestandteil ist die Bedrohungsmodellierung und Risikoanalyse, um Schwachstellen systematisch zu identifizieren und Sicherheitsmaßnahmen zu priorisieren. Praktische Übungen und Fallstudien vertiefen die theoretischen Inhalte und ermöglichen die Anwendung auf reale Szenarien. Das Modul zielt darauf ab, Studierende mit den Fähigkeiten auszustatten, IT-Systeme sicher zu gestalten und Sicherheitsstrategien effektiv in Softwareentwicklungsprozesse zu integrieren.

Prüfung

Verpflichtende Voraussetzung gemäß SPO für die Teilnahme an der Prüfung
Keine

Art der Prüfung
Schriftliche Prüfung (sP) gemäß § 23 APO

Dauer/Form der Prüfung
90 Minuten

Die konkrete Festlegung der abzuleistenden Prüfung erfolgt im Studienplan

Prüfungssprache
Deutsch

Voraussetzung für die Vergabe von Leistungspunkten
Keine

Lernergebnisse
Die Studierenden

- erklären grundlegende Begriffe, Bedrohungsszenarien und Angriffsvektoren der IT-Sicherheit,
- wenden diese Kenntnisse auf unterschiedliche Anwendungskontexte an,
- erläutern die Sicherheitsziele Vertraulichkeit, Integrität und Verfügbarkeit,
- bewerten deren Relevanz für IT-Systeme und Softwareprojekte,
- beschreiben die Funktionsweise symmetrischer und asymmetrischer Verschlüsselungsverfahren sowie digitaler Signaturen und Zertifikate,
- setzen kryptografische Verfahren in geeigneten Szenarien zielgerichtet ein,
- analysieren Schwachstellen in Webanwendungen (z. B. SQL-Injection) und klassischen Anwendungen (z. B. Buffer Overflows),
- entwickeln geeignete Gegenmaßnahmen zur Behebung dieser Schwachstellen und begründen deren Auswahl,
- erläutern moderne Authentifizierungs- und Autorisierungskonzepte wie Multi-Faktor-Authentifizierung, Rollen- und Rechtekonzepte sowie Single Sign-On (SSO),
- und setzen diese Konzepte praktisch in sicherheitsrelevanten IT-Szenarien um.

Literatur

- Wolfgang Ertel: Angewandte Kryptographie, Hanser, 2019.
- Dafydd Stuttard, Marcus Pinto: The Web Application Hacker's Handbook: Finding and Exploiting Security Flaws, Wiley, 2011.
- Tobias Klein: Buffer Overflows und Format-String-Schwachstellen: Funktionsweisen, Exploits und Gegenmaßnahmen, dpunkt.verlag, 2005.

fiw.thws
Fakultät Informatik und Wirtschaftsinformatik
Technische Hochschule Würzburg-Schweinfurt

Modul: 1512300
Machine Learning

Modulprofil

Prüfungsnummer
1512300

Dauer
1 Semester

Häufigkeit des Angebots
Jedes Sommersemester

SWS
4

ECTS-Credits (CP)
5.0

Workload
Angeleitete Studienzeit:
Synchrone Teilnahme: 1 Std.
Asynchrone Teilnahme: 3 Std.

Selbststudienzeit: 90 Std.

Gesamt: 150 Std.

Lehrveranstaltungsart(en)
Seminaristischer Unterricht, Übung

Lehrsprache
Englisch

Organisation

Modulverantwortung
Prof. Dr. Frank-Michael Schleif

Dozierende
Prof. Dr. Frank-Michael Schleif

Verwendbarkeit	BSED
Studiensemester	4. Semester
Art des Moduls	Pflichtmodul

Verpflichtende Voraussetzungen gemäß SPO
Data Science, Math modules, programming courses

Empfohlene Voraussetzungen
Math skills are crucial and are only briefly repeated on demand and in an initial refresher.
The students should have a reasonable knowledge in programming with python

Inhalte
Machine learning (ML) is a core technology in software engineering, driving data analysis, automation, and AI. This module builds on prior data science knowledge to introduce essential ML concepts, models, and algorithms.

Students learn traditional AI approaches, core learning paradigms (supervised and unsupervised), and ethical considerations. A hands-on component emphasizes practical skills in Python using libraries such as NumPy, Pandas, and Scikit-learn. By the end, students can build and evaluate ML models.

Module contents:

- Introduction to AI and ML (math refresher, AI history, symbolic vs. sub-symbolic methods, expert systems, classical models)
- Core ML Concepts (learning paradigms, prediction vs. discovery, ethics)
- Learning Foundations (loss functions, model evaluation, bias-variance trade-off)
- ML Algorithms (linear models, regularization, clustering, decision trees)
- Practical Skills (Python, libraries, Jupyter, model tuning)
- Ethical & Societal Aspects (bias, fairness, privacy, AI impact)

Prüfung

Verpflichtende Voraussetzung gemäß SPO für die Teilnahme an der Prüfung
Keine

Art der Prüfung
Sonstige Prüfung (soP) gemäß §§ 26, 27 APO

Dauer/Form der Prüfung
Portfolio

Die konkrete Festlegung der abzuleistenden Prüfung erfolgt im Studienplan

Prüfungssprache
Englisch

Voraussetzung für die Vergabe von Leistungspunkten
Keine

Lernergebnisse
Upon successful completion of this module, students will be able to:
Knowledge and Understanding:

- Explain AI/ML history (symbolic & sub-symbolic)
- Distinguish classical AI from modern ML
- Describe ML paradigms and applications
- Understand learning formalism: loss, risk, complexity
- Explain strengths, weaknesses, and efficiency of ML algorithms

Skills and Competences:

- Apply core ML algorithms (regression, classification, clustering)
- Use Python + libraries (Scikit-learn, Pandas, NumPy) for modeling
- Evaluate and compare models with metrics and validation
- Interpret results (accuracy, efficiency, robustness)
- Present findings via visualizations, reports, notebooks
- Assess ethical issues and bias in ML/AI

Literatur

- Christopher M. Bishop: Pattern Recognition and Machine Learning, Springer, 2006.
- Kevin P. Murphy: Machine Learning: A Probabilistic Perspective, MIT Press, 2012.
- Trevor Hastie, Robert Tibshirani, Jerome Friedman: The Elements of Statistical Learning, Springer, 2001.
- Stuart Russell, Peter Norvig: Artificial Intelligence: A Modern Approach, Pearson, 2022.
- Ian Goodfellow, Yoshua Bengio, Aaron Courville: Deep Learning, MIT Press, 2016.
- European Commission: Ethical AI and Machine Learning Guidelines, 2023.

fiw.thws
Fakultät Informatik und Wirtschaftsinformatik
Technische Hochschule Würzburg-Schweinfurt

Modul: 1512000
Mobile Systems

Modulprofil

Prüfungsnummer
1512000

Dauer
1 Semester

Häufigkeit des Angebots
Jedes Sommersemester

SWS
4

ECTS-Credits (CP)
5.0

Workload
Angeleitete Studienzeit:
Synchrone Teilnahme: 1 Std.
Asynchrone Teilnahme: 3 Std.

Selbststudienzeit: 90 Std.

Gesamt: 150 Std.

Lehrveranstaltungsart(en)
Seminaristischer Unterricht, Übung

Lehrsprache
Englisch

Organisation

Modulverantwortung
Prof. Dr. Peter Braun

Dozierende
Prof. Dr. Peter Braun

Verwendbarkeit BSED

Studiensemester 4. Semester

Art des Moduls Pflichtmodul

Verpflichtende Voraussetzungen gemäß SPO
Keine

Empfohlene Voraussetzungen
Keine

Inhalte
This module introduces students to the principles and practical aspects of mobile application development. It focuses on modern cross-platform technologies and best practices for building efficient, scalable, and user-friendly mobile applications.

- Introduction to Mobile User Interface Development with Flutter
- Overview of Flutter and the Dart programming language
- Designing responsive layouts, handling user input, and implementing navigation
- State Management in Mobile Applications
- Understanding stateful and stateless widgets in Flutter
- Exploring state management techniques such as Provider and Riverpod

- Introduction to React Native: Understanding the fundamentals of React Native and its component-based architecture
- Building cross-platform mobile apps using JavaScript and React principles
- Utilizing React Native components and navigation techniques
- Integrating APIs and Asynchronous Data Handling in Mobile Apps

- Connecting mobile applications with backend services using REST APIs and GraphQL
- Handling asynchronous operations with Dart's Future & Streams and JavaScript's Promises & Async/Await
- Managing data fetching, caching, and error handling
- Testing and Debugging Mobile Applications

In the traditional degree programme, the lecturer provides or agrees with the topics of the practical examples for the examination. In the dual study programme, the lecturer consults with the company on a task, ensuring practical relevance and feedback from the company.

Prüfung

Verpflichtende Voraussetzung gemäß SPO für die Teilnahme an der Prüfung
Keine

Art der Prüfung
Sonstige Prüfung (soP) gemäß §§ 26, 27 APO

Dauer/Form der Prüfung
Portfolio

Die konkrete Festlegung der abzuleistenden Prüfung erfolgt im Studienplan

Prüfungssprache
Englisch

Voraussetzung für die Vergabe von Leistungspunkten
Keine

Lernergebnisse
After completing this module, students will be able to:

- Explain the fundamental concepts of cross-platform mobile development using Flutter and React Native, including their architectures, strengths, and limitations
- Compare and evaluate the architectural patterns of Flutter and React Native to justify technology choices for specific development scenarios
- Design responsive and user-friendly mobile interfaces using Flutter's widget system and React Native's component model
- Implement mobile applications with effective state management by using Provider, Riverpod, or React Context API
- Integrate backend services into mobile applications by consuming REST APIs and GraphQL, and by managing asynchronous data, caching, and error handling
- Develop and execute unit, widget, and integration tests using tools like Flutter's testing suite, Jest, and React Testing Library to ensure application correctness and stability

Literatur
- Thomas Bailey, Alessandro Biessek: Flutter for Beginners: Cross-platform mobile development from Hello, World! to app release with Flutter 3.10+ and Dart 3.x. Packt, 2023.
- Sakhniuk, Mikhail, und Adam Boduch. React and React Native: Build Cross-platform JavaScript and TypeScript Apps for the Web, Desktop, and Mobile. Fifth edition. Birmingham, UK: Packt, 2024.

fiw.thws
Fakultät Informatik und Wirtschaftsinformatik
Technische Hochschule Würzburg-Schweinfurt

Modul: 1511900
Projekt 3

Modulprofil

Prüfungsnummer
1511900

Dauer
1 Semester

Häufigkeit des Angebots
Jedes Sommersemester

SWS
1

ECTS-Credits (CP)
5.0

Workload
Angeleitete Studienzeit:
Synchrone Teilnahme: 1 Std.
Asynchrone Teilnahme: 0 Std.

Selbststudienzeit: 135 Std.

Gesamt: 150 Std.

Lehrveranstaltungsart(en)
Projekt

Lehrsprache
Deutsch

Organisation

Modulverantwortung
Prof. Dr. Peter Braun

Dozierende
Prof. Dr. Peter Braun,
Prof. Dr. Isabel John,
Michael Rott,
Prof. Dr. Rolf Schillinger,
Prof. Dr.-Ing. Tobias Fertig,
Prof. Dr. Frank-Michael Schleif,
Prof. Dr.-Ing. Sebastian Biedermann,
Prof. Dr. Tristan Wimmer,
Prof. Dr.-Ing. Anne Heß

Verwendbarkeit	BSED
Studiensemester	4. Semester
Art des Moduls	Pflichtmodul

Verpflichtende Voraussetzungen gemäß SPO
Keine

Empfohlene Voraussetzungen
Projekt1, Projekt 2, Programmieren 1, Programmieren 2, Introduction to Software Engineering, Analyse und Design, Software Qualität, Datenbanken

Inhalte
Die Studierenden arbeiten in größeren Teams (5–6 Personen) an einem komplexen Softwareprojekt mit echten funktionalen und nicht-funktionalen Anforderungen, die von einem Kunden durch Interviews zu ermitteln sind. Die Studierenden entwickeln sie ein verteiltes Softwaresystem, das moderne Software Engineering Techniken integriert. Das Team organisiert seine Aufgaben durch eine agile Projektmanagement-Methodik und verteilt die Rollen im Team entsprechend. Die Anwendung könnte enthalten:

- Microservices oder eine verteilte Architektur
- Datenbankanbindung (SQL oder NoSQL)
- Eine grafische Benutzeroberfläche, zum Beispiel als Web- oder Mobile Anwendung
- Sicherheitsaspekte (z. B. Authentifizierung, Berechtigungen, Verschlüsselung)
- Performanz- und Skalierbarkeitsoptimierung
- Automatisierte Unit- und Integrations-Tests zur Qualitätssicherung
- Die Dokumentation folgt Industrie-Standards mit vollständiger Architektur-, API- und Deployment-Dokumentation.

Die Studierenden werden während der Projektbearbeitung kontinuierlich von einer Dozentin oder einem Dozenten betreut. Die Betreuung erfolgt durch regelmäßige Treffen, in denen der Arbeitsfortschritt besprochen wird, sowie durch begleitende Zwischenpräsentationen, in denen die Entwicklung des Projekts reflektiert und weiterentwickelt wird.

Prüfung

Verpflichtende Voraussetzung gemäß SPO für die Teilnahme an der Prüfung
Keine

Art der Prüfung
Sonstige Prüfung (soP) gemäß §§ 26, 27 APO

Dauer/Form der Prüfung
Dokumentation, Präsentation

Die konkrete Festlegung der abzuleistenden Prüfung erfolgt im Studienplan

Prüfungssprache
Deutsch

Voraussetzung für die Vergabe von Leistungspunkten
Keine

Lernergebnisse

- Ein verteiltes Software-System mit modernen Technologien zu entwickeln
- IT-Sicherheitsaspekte in die Software-Architektur zu integrieren
- Komplexe Softwarelösungen in Teams zu konzipieren und umzusetzen
- Verschiedene Rollen in einem agilen Team zu übernehmen
- Fortgeschrittene Methoden des agilen Projektmanagement anzuwenden, insbesondere Reviews und Retrospektiven durchzuführen
- Eine vollständige Software-Dokumentation zu erstellen
- Ihr Projekt in einer öffentlichen Abschlusspräsentation überzeugend darzustellen

Literatur
Wird in Abhängigkeit von den konkreten Projekten und verwendeten Technologie in der Veranstaltung bekannt gegeben.

fiw.thws
Fakultät Informatik und Wirtschaftsinformatik
Technische Hochschule Würzburg-Schweinfurt

Modul: 1512100
Web Systems

Modulprofil

Prüfungsnummer
1512100

Dauer
1 Semester

Häufigkeit des Angebots
Jedes Sommersemester

SWS
4

ECTS-Credits (CP)
5.0

Workload
Angeleitete Studienzeit:
Synchrone Teilnahme: 1 Std.
Asynchrone Teilnahme: 3 Std.

Selbststudienzeit: 90 Std.

Gesamt: 150 Std.

Lehrveranstaltungsart(en)
Seminaristischer Unterricht, Übung

Lehrsprache
Deutsch

Organisation

Modulverantwortung
Prof. Dr. Rolf Schillinger

Dozierende
Prof. Dr. Rolf Schillinger

Verwendbarkeit	BSED
Studiensemester	4. Semester
Art des Moduls	Pflichtmodul

Verpflichtende Voraussetzungen gemäß SPO
Keine

Empfohlene Voraussetzungen
Keine

Inhalte
Web Systeme haben in den letzten Jahrzehnten viele klassische GUI Anwendungen ersetzt, die vormals unter Nutzung von GUI Frameworks wie MFC/.NET/WPF, Qt, Swing und ähnlichem programmiert wurden. Vielen Vorteilen wie zum Beispiel der großen Platformunabhängigkeit, der zentralen Wartung und auch der einfache Verteilung an beliebig viele Clients standen immer wieder auch erhebliche Nachteile wie eingeschränktem Offlinefähigkeit oder mangelnder Systemintegration gegenüber. In der Praxis überwiegen in der Mehrzahl der Anwendungsfälle allerdings mittlerweile die Vorteile. In diese Modul lernen Studierende die verschiedenen Ansätze und Technologien kennen und nutzen, auf denen moderne Web Systeme basieren. Dazu gehören insbesondere die folgenden Themen:

- Grundlagen: HTML DOM und Rendering Prozesse im Browser, klassische HTML Templating Systeme wie Blade in PHP oder Django in Python
- Reaktivität und Frameworks, Reaktivität in Webapplikationen, dabei insbesondere reaktive JavaScript Frameworks wie React, Komponentenbasierte Entwicklung, Single Page Web Apps, Progressive Web Apps
- Performance und Optimierung des Page Renderings, Critical Rendering Path, Server-side Rendering
- Authentifizierung und Autorisierung: Grundlagen von JWT und OAuth2, Einführung in FIDO2 (Passkeys)
- Fortgeschrittene Themen in reaktiven JavaScript Frameworks, State-Management (z.B. via Redux), Tooling (Vite und Parcel), Frontend Testing
- Zukunftstechnologien und hybride Ansätze, WebAssembly, Electron & Desktop-Apps auf Webbasis

Prüfung

Verpflichtende Voraussetzung gemäß SPO für die Teilnahme an der Prüfung
Keine

Art der Prüfung
Schriftliche Prüfung (sP) gemäß § 23 APO

Dauer/Form der Prüfung
90 Minuten

Die konkrete Festlegung der abzuleistenden Prüfung erfolgt im Studienplan

Prüfungssprache
Deutsch

Voraussetzung für die Vergabe von Leistungspunkten
Keine

Lernergebnisse
Nach erfolgreichem Abschluß dieses Moduls sind die Studierenden in der Lage

- die wesentlichen Komponenten eines modernen Websystems zu benennen und deren Zusammenspiel zu skizzieren
- verschiedene HTML Templating Systeme zu vergliechen und deren wichtigste Eigenschaften voneinander abzugrenzen um deren Eignung für unterschiedliche Szenarien zu bwerten
- vorgegebene HTML Layouts mittels Templating Systemen strukturiert umzusetzen
- eine reaktive Webapplikation zu designen und als SPA in React umzusetzen
- die entstandene Webapplikation auf deren Rendering Performanz hin zu analyisieren und gezielt zu optimieren
- die JWT und OAuth2 Spezifikationen im Detail zu erklären und auf dieser Basis zu entscheiden, welche Lösung für vorgegebene Anforderungen geeignet ist
- sowohl JWT- als auch OAuth2-basierte Authentifizierungsmechanismen praktisch in jeweils einer React-Beispielanwendung zu integrieren

Literatur

- Banks, A., & Porcello, E. (2020). Learning React: Modern Patterns for Developing React Apps (2nd ed.). O'Reilly Media.
- Google Developers (https://web.dev)
- Grigorik, I. (2013). High Performance Browser Networking: What every web developer should know about networking and web performance. O'Reilly Media.
- Mozilla Developer Network (https://developer.mozilla.org/de/)
- Müller, P. (2022). Einstieg in HTML und CSS: Webseiten programmieren und gestalten (2. Aufl.). Rheinwerk Verlag.
- Springer, S. (2022). React: Das umfassende Handbuch für moderne Frontend-Entwicklung. Mit Praxisbeispielen zu Redux, TypeScript, SSR und PWA (2. Aufl.). Rheinwerk Verlag.

fiw.thws
Fakultät Informatik und Wirtschaftsinformatik
Technische Hochschule Würzburg-Schweinfurt

5. Semester

fiw.thws
Fakultät Informatik und Wirtschaftsinformatik
Technische Hochschule Würzburg-Schweinfurt

Modul: 1512500
Praxismodul

Modulprofil

Prüfungsnummer
1512500

Dauer
1 Semester

Häufigkeit des Angebots
Jedes Semester

SWS
1

ECTS-Credits (CP)
30.0

Workload
Angeleitete Studienzeit:
Synchrone Teilnahme: 1 Std.
Asynchrone Teilnahme: 0 Std.

Selbststudienzeit: 885 Std.

Gesamt: 900 Std.

Lehrveranstaltungsart(en)
Praktikum

Lehrsprache
Deutsch/Englisch

Organisation

Modulverantwortung
Prof. Dr. Peter Braun

Dozierende
Prof. Dr. Peter Braun

Verwendbarkeit	BSED
Studiensemester	5. Semester
Art des Moduls	Pflichtmodul

Verpflichtende Voraussetzungen gemäß SPO
Mehr als 90 ECTS Punkte, davon 55 ECTS aus Modulen aus dem ersten Studienjahr sowie das Modul Professional Skills ist bestanden.

Empfohlene Voraussetzungen
Zusätzlich zu den genannten Voraussetzungen insbesondere auch: Software Qualität, Backend Systems, Mobile Systems, Web Systems

Inhalte
Im Rahmen des Praxismoduls erhalten die Studierenden die Möglichkeit, ihre im Studium erworbenen fachlichen und methodischen Kompetenzen in einem realen Unternehmensumfeld anzuwenden und zu vertiefen. Die Praxisphase umfasst ein größeres IT-Projekt, in dem die Studierenden möglichst viele Phasen des Softwareentwicklungsprozesses eigenverantwortlich durchlaufen, darunter:

- Anforderungsanalyse und Systemspezifikation
- Softwarearchitektur und -design
- Implementierung und Testen
- Systemintegration und -einführung
- Dokumentation und Qualitätssicherung
- Das Projekt soll einen zeitlichen Umfang von mindestens 12 Wochen haben und praxisnahe Problemstellungen aus dem Bereich Software Engineering adressieren.

Zusätzlich wird empfohlen, dass die Studierenden vor dem Projekt verschiedene Abteilungen und Unternehmensbereiche kennenlernen, um ein ganzheitliches Verständnis für betriebliche Prozesse und die Zusammenarbeit in interdisziplinären Teams zu entwickeln. Die Praxisphase wird durch die Hochschule begleitet. Ansprechpartner für die begleitete Praxisphase ist der oder die Praktikumsbeauftragte.

Prüfung

Verpflichtende Voraussetzung gemäß SPO für die Teilnahme an der Prüfung
Keine

Art der Prüfung
Sonstige Prüfung (soP) gemäß §§ 26, 27 APO

Dauer/Form der Prüfung
Präsentation, Dokumentation

Die konkrete Festlegung der abzuleistenden Prüfung erfolgt im Studienplan

Prüfungssprache
Deutsch/Englisch

Voraussetzung für die Vergabe von Leistungspunkten
Keine

Lernergebnisse
Die Studierenden sollen während des Praxissemester folgende Lernziele erreichen:

- Praxisorientierte Kenntnisse betrieblicher Abläufe und Prozesse zu erwerben und deren Relevanz für die Softwareentwicklung zu verstehen.
- Selbstständig und eigenverantwortlich in IT-Projekten zu arbeiten, unter Anleitung und mit zunehmender Eigeninitiative.
- Die im Studium erworbenen theoretischen und praktischen Kompetenzen in realen Projekten anzuwenden und mit den Erfahrungen der Praxis zu verknüpfen.
- Anforderungen aus der Praxis (z. B. Kundenwünsche, technische Anforderungen) zu analysieren und deren Bedeutung für die Entwicklung und Umsetzung von Softwarelösungen zu verstehen.
- Problemlösungen für betriebliche Prozesse oder IT-Projekte zu entwerfen, zu implementieren und deren Wirksamkeit zu bewerten.
- Effektiv im Team zu arbeiten, Rollenverteilungen zu verstehen und kollaborative Entwicklungsprozesse aktiv mitzugestalten.
- Die Einbettung von Softwareprojekten in die organisatorischen Strukturen eines Unternehmens zu erkennen und aktiv in betriebliche Abläufe zu integrieren.
- Das Berufsfeld eines Software Engineers in einem professionellen Umfeld kennenzulernen und Erfahrungen für die eigene Karriereplanung zu sammeln.
- Bei Problemen eigenständig die richtigen Ansprechpartner im Unternehmen zu identifizieren und mit ihnen zielführend zu kommunizieren.
- Best Practices für professionelle Softwareentwicklung zu erleben, den hohen Qualitätsanspruch in Unternehmen nachzuvollziehen und Exzellenz als Zielsetzung zu verinnerlichen.
- Die Dynamik und Motivation in professionellen Entwicklungsteams zu erleben und die Bedeutung von Kommunikation, Verantwortung und Teamkultur für den Projekterfolg zu erkennen.
- Den Sinn und die Wirkung der eigenen Tätigkeit im Unternehmen zu reflektieren und den Beitrag zur Wertschöpfung und zum Gesamterfolg der Organisation zu erkennen.

Literatur

6. Semester

fiw.thws
Fakultät Informatik und Wirtschaftsinformatik
Technische Hochschule Würzburg-Schweinfurt

Modul: 1513000
Advanced Software Testing

Modulprofil

Prüfungsnummer
1513000

Dauer
1 Semester

Häufigkeit des Angebots
Jedes Sommersemester

SWS
4

ECTS-Credits (CP)
5.0

Workload
Angeleitete Studienzeit:
Synchrone Teilnahme: 1 Std.
Asynchrone Teilnahme: 3 Std.

Selbststudienzeit: 90 Std.

Gesamt: 150 Std.

Lehrveranstaltungsart(en)
Seminaristischer Unterricht, Übung

Lehrsprache
Englisch

Organisation

Modulverantwortung
Prof. Dr. Steffen Heinzl

Dozierende
Prof. Dr. Steffen Heinzl,
Prof. Dr. Tristan Wimmer

Verwendbarkeit	BSED
Studiensemester	6. Semester
Art des Moduls	Pflichtmodul

Verpflichtende Voraussetzungen gemäß SPO
Keine

Empfohlene Voraussetzungen
Programmieren 1 und 2, Introduction to Software Engineering, Analyse und Design, Software Qualität

Inhalte
Grundlagen des Testens

Motivation und Ziele von Softwaretests
- Testarten: Black-, White- & Grey-Box-Testing
- Funktionale & nicht-funktionale Tests
- Testabdeckung, Testpfade & Testpyramide
- Testautomatisierung

Einführung in automatisierte Tests und Erfolgsfaktoren
- Testframeworks in Abhängigkeit der verwendeten Sprachen
- Testarten: Record Replay, Scripted & Keyword-driven Testing
- Testarchitektur & Design Patterns

SOLID-Prinzipien für Testarchitekturen
- 4-Schichten-Testkonzept: Modellierung, Definition, Execution, Adaptation
- Wichtige Design Patterns für Testing
- Automatisiertes UI- & API-Testing

Einführung in Selenium: Driver, PageObject Pattern, Identifikatoren
- Mocking mit Wiremock für API-Tests
- Behaviour Driven Development (BDD)

Einführung in Cucumber & Gherkin
- Feature-Files, Step-Files & Szenario Outlines
- Exploratives Testen & DevOps-Integration

Explorative Testmethoden und Techniken
- Continuous Testing, z.B. mit Jenkins & DevOps Pipelines

Prüfung

Verpflichtende Voraussetzung gemäß SPO für die Teilnahme an der Prüfung
Keine

Art der Prüfung
Sonstige Prüfung (soP) gemäß §§ 26, 27 APO

Dauer/Form der Prüfung
Portfolio, Schriftliche Prüfung

Die konkrete Festlegung der abzuleistenden Prüfung erfolgt im Studienplan

Prüfungssprache
Englisch

Voraussetzung für die Vergabe von Leistungspunkten
Keine

Lernergebnisse

- Die Studierenden verstehen die grundlegenden Konzepte und Prinzipien des Softwaretestens, einschließlich Testarten, Testabdeckung und Testpyramide.
- Die Studierenden analysieren verschiedene Testarchitekturen und bewerten deren Eignung für unterschiedliche Softwareprojekte.
- Die Studierenden wenden Testautomatisierungsstrategien an und implementieren automatisierte Tests mit JUnit, Selenium und Wiremock.
- Die Studierenden entwickeln eine strukturierte Testarchitektur unter Berücksichtigung von SOLID-Prinzipien und Design Patterns.
- Die Studierenden erstellen Behaviour Driven Development (BDD)-Tests mit Cucumber & Gherkin und binden sie in den Entwicklungsprozess ein.
- Die Studierenden integrieren automatisierte Tests in einen DevOps-Prozess und setzen Continuous Testing mit Jenkins um.
- Die Studierenden bewerten die Qualität von Tests anhand von Metriken wie Testabdeckung, Robustheit und Wartbarkeit.
- Die Studierenden experimentieren mit explorativen Testmethoden und wenden geeignete Techniken zur Fehlererkennung an.

Literatur

- Essentials of Software Testing von Ralf Bierig, Stephen Brown, Edgar Galván, Joe Timoney, 2021, Cambridge University Press

fiw.thws
Fakultät Informatik und Wirtschaftsinformatik
Technische Hochschule Würzburg-Schweinfurt

Modul: 1512900
Clean Code und Design Pattern

Modulprofil

Prüfungsnummer
1512900

Dauer
1 Semester

Häufigkeit des Angebots
Jedes Sommersemester

SWS
4

ECTS-Credits (CP)
5.0

Workload
Angeleitete Studienzeit:
Synchrone Teilnahme: 1 Std.
Asynchrone Teilnahme: 3 Std.

Selbststudienzeit: 90 Std.

Gesamt: 150 Std.

Lehrveranstaltungsart(en)
Seminaristischer Unterricht, Übung

Lehrsprache
Deutsch

Organisation

Modulverantwortung
Prof. Dr.-Ing. Tobias Fertig

Dozierende
Prof. Dr. Steffen Heinzl,
Prof. Dr.-Ing. Tobias Fertig

Verwendbarkeit	BSED
Studiensemester	6. Semester
Art des Moduls	Pflichtmodul

Verpflichtende Voraussetzungen gemäß SPO
Keine

Empfohlene Voraussetzungen
Die Studierenden sollen ein eigenes Projekt aus bspw. dem Programmierprojekt, Softwareentwicklungsprojekt, Praktikum, o.ä. mitbringen.

Inhalte
Dieses Modul vermittelt weit-verbreitete Design Patterns und Clean Code Techniken, die essentiell für die Code-Qualität und Wartbarkeit sind. Es werden u.a. folgende Inhalte behandelt:

- Naming, Functions, Comments, Formatting
- Objects and Data Structures
- Law of Demeter
- DTOs
- Exceptions, Don't return null
- Code Tangling and Scattering, Logging, AOP
- Fluent Interfaces, Internal DSLs
- Coding Dojos
- Dependency Injection
- Traditionelle Design Patterns
- Moderne Web und Mobile Design Patterns
- Refactoring Strategien

Prüfung

Verpflichtende Voraussetzung gemäß SPO für die Teilnahme an der Prüfung
Keine

Art der Prüfung
Sonstige Prüfung (soP) gemäß §§ 26, 27 APO

Dauer/Form der Prüfung
Portfolio, Schriftliche Prüfung

Die konkrete Festlegung der abzuleistenden Prüfung erfolgt im Studienplan

Prüfungssprache
Deutsch

Voraussetzung für die Vergabe von Leistungspunkten
Keine

Lernergebnisse

- Studierende können verständlicheren Code entwickeln
- Studierende können besser wartbaren Code entwickeln
- Studierende können weniger fehleranfälligen Code entwickeln
- Studierende beschreiben an geeigneten Code-Stellen, WAS erreicht werden soll und nicht WIE
- Studierende verstehen von Crosscutting Concerns
- Studierende führen Refactorings selbstständig durch
- Studierende verstehen Design Patterns und wenden sie an

Literatur

- Robert C. Martin (2008). Clean Code: A Handbook of Agile Software Craftsmanship. Prentice Hall.
- Gamma, E., Helm, R., Johnson, R., Vlissides, J. (1994). Design Patterns: Elements of Reusable Object-Oriented Software. Pearson Education.
- Fowler, M., Beck, K. (2018). Refactoring: Improving the Design of Existing Code. Addison-Wesley Professional.

fiw.thws
Fakultät Informatik und Wirtschaftsinformatik
Technische Hochschule Würzburg-Schweinfurt

Modul: 1512800
Cloud Computing

Modulprofil

Prüfungsnummer
1512800

Dauer
1 Semester

Häufigkeit des Angebots
Jedes Sommersemester

SWS
4

ECTS-Credits (CP)
5.0

Workload
Angeleitete Studienzeit:
Synchrone Teilnahme: 1 Std.
Asynchrone Teilnahme: 3 Std.

Selbststudienzeit: 90 Std.

Gesamt: 150 Std.

Lehrveranstaltungsart(en)
Seminaristischer Unterricht, Übung

Lehrsprache
Deutsch

Organisation

Modulverantwortung
Prof. Dr. Rolf Schillinger

Dozierende
Prof. Dr. Rolf Schillinger

Verwendbarkeit	BSED
Studiensemester	6. Semester
Art des Moduls	Pflichtmodul

Verpflichtende Voraussetzungen gemäß SPO
Keine

Empfohlene Voraussetzungen
Keine

Inhalte
Seit den ersten Anfängen der Virtualisierung in IBMs Mainframe Betriebssystemen der 80er Jahre haben sich Technologien der Virtualisierung zum heutigen Cloud Computing Komplex hin entwickelt, der für einige bahnbrechende Neuerungen im Betrieb von IT-Services gesorgt hat. Viele der heute von modernen Rechenzentren geforderten Eigenschaften wie größte Flexibilität, Agilität und Skalierbarkeit waren vor der großflächigen Verfügbarkeit von Cloud Technologien nicht oder nur sehr schwer umsetzbar.
Dieser Kurs beschäftigt sich mit den wichtigsten Grundlagen, Funktionalitäten und Eigenschaften des Cloud Computings, insbesondere mit den folgenden Teilgebieten:

- Grundlagen und Abgrenzung
- Virtualisierung als Grundlage des Cloud Computings
- Cloud Betreibermodelle und deren Charakteristika
- HyperScaler (AWS, Azure, Google Cloud Platform)
- Virtuelle Maschinen
- Überblick über gängige Virtualisierungstechnologien (KVM / QEMU, Proxmox, Hyper-V)
- VM Introspection / Sicherheit von VMs
- Deployment in Cloud Umgebungen und DevOps
- Cloud im Kontext DevOps
- Automatisierung des Deployments (Pipelines, CI/CD)
- Infrastructure as code
- Skalierungsmöglichkeiten
- Containerisierung
- BSD Jails, LXC
- Docker Container
- Orchestrierung von Containern (Kubernetes)
- Überblick über die PaaS Angebote einiger GigaScaler, z.B. AWS Elastic Beanstalk

Prüfung

Verpflichtende Voraussetzung gemäß SPO für die Teilnahme an der Prüfung
Keine

Art der Prüfung
Sonstige Prüfung (soP) gemäß §§ 26, 27 APO

Dauer/Form der Prüfung
Portfolio

Die konkrete Festlegung der abzuleistenden Prüfung erfolgt im Studienplan

Prüfungssprache
Deutsch

Voraussetzung für die Vergabe von Leistungspunkten
Keine

Lernergebnisse
Nach erfolgreichem Abschluß des Moduls sind die Studiernden in der Lage

- Unterschiede zwischen traditionellen IT-Infrastrukturen und Cloud Deployments beschreiben
- die verschiedenen Cloud Betreibermodelle zu beschreiben und voneinander abzugrenzen
- für gegebene Applikationen geeignete Cloud Technologien auszuwählen und deren Nutzung zu planen
- eine virtuelle Maschine auf einer bereitgestellten Proxmox-Instanz zu erstellen, zu konfigurieren und eine Backend-Applikation (bestehend aus Apache-Webserver und Datenbank) darauf bereitzustellen
- diese Applikation containerisiert mit LXC bereitzustellen
- die Applikation in Docker Container zu verpacken und manuell zu deployen
- diese Container mittels Kubernetes skalierbar zu orchestrieren
- Methoden des load testings anzuwenden, um das Verhalten des Deployments unter verschiedenen Lastprofilen abschätzen zu können

Literatur

- Humble, J., & Farley, D. (2010). Continuous delivery: Reliable software releases through build, test, and deployment automation. Addison-Wesley.
- Liebel, O. (2023). Skalierbare Container-Infrastrukturen: Das Handbuch für Planung und Administration. Rheinwerk Verlag.
- Merkel, D. (2023). Docker: Das umfassende Handbuch (aktuelle Aufl.). Rheinwerk Verlag. ISBN 978-3836294722
- Stender, D. (2020). Cloud-Infrastrukturen: Das Handbuch für DevOps-Teams und Administratoren. Rheinwerk Verlag.
- Tanenbaum, A. S., & Bos, H. (2014). Modern operating systems (4th ed.). Pearson International.
- Turnbull, J. (2021). The Docker book: Containerization is the new virtualization (latest ed.). Self-published.

fiw.thws
Fakultät Informatik und Wirtschaftsinformatik
Technische Hochschule Würzburg-Schweinfurt

Modul: 1512700
Datenschutz und Ethik

Modulprofil

Prüfungsnummer
1512700

Dauer
1 Semester

Häufigkeit des Angebots
Jedes Sommersemester

SWS
4

ECTS-Credits (CP)
5.0

Workload
Angeleitete Studienzeit:
Synchrone Teilnahme: 1 Std.
Asynchrone Teilnahme: 3 Std.

Selbststudienzeit: 90 Std.

Gesamt: 150 Std.

Lehrveranstaltungsart(en)
Seminaristischer Unterricht, Übung

Lehrsprache
Deutsch

Organisation

Modulverantwortung
Prof. Dr. Markus Oermann

Dozierende
Prof. Dr. Markus Oermann

Verwendbarkeit	BSED
Studiensemester	6. Semester
Art des Moduls	Pflichtmodul

Verpflichtende Voraussetzungen gemäß SPO

Empfohlene Voraussetzungen
Introduction to Software Engineering
Analyse und Design
Software Qualität

Inhalte
Das Modul vermittelt Wissen und ein Verständnis für Grundpositionen der digitalen Ethik und von Grundstrukturen des Datenschutzrechts. Die Studierenden werden befähigt, das Erlernte auf praktische Fälle von Softwareprojekten anzuwenden und grundlegende ethische und datenschutzrechtliche Fragestellungen während der Softwareentwicklung zu bewerten.

Im Abschnitt zu Ethik werden essenzielle begriffliche Grundlagen der Moralphilosophie erläutert. Auf der Grundlage etablierter Schulen der Ethik wird die normative Begründung und anhand von problematischen Praxisfällen der tatsächliche Bedarf für eine ethische fundierte Softwareentwicklung und einen entsprechenden Ethos der Verantwortlichen herausgearbeitet. Die Betrachtung von Modellen für die Integration ethischer Überlegungen in Entwicklungs- und Systemdesignprozesse verdeutlicht, wie ethischen Grundsätze in der Praxis in die Softwareentwicklung einfließen können.

In der Praxis sind zudem Fragen der Compliance mit dem geltenden Datenschutzrecht von besonderer Relevanz. Nach einem Überblick über dessen Grundstrukturen liegt der Schwerpunkt auf den Anforderungen an den technischen und organisatorischen Datenschutz sowie der Durchsetzung und den Folgen von Rechtsverstößen. Hier fließen auch relevante Aspekte des Informationssicherheitsrechts ein. Abschließend wird ein Überblick über den europäischen Rechtsrahmen für künstliche Intelligenz vermittelt.

Prüfung

Verpflichtende Voraussetzung gemäß SPO für die Teilnahme an der Prüfung
Keine

Art der Prüfung
Schriftliche Prüfung (sP) gemäß § 23 APO

Dauer/Form der Prüfung
90 Minuten

Die konkrete Festlegung der abzuleistenden Prüfung erfolgt im Studienplan

Prüfungssprache
Deutsch

Voraussetzung für die Vergabe von Leistungspunkten
Keine

Lernergebnisse

- Die Studierenden verfügen über einen Überblick über ethische Anforderungen an die Softwareentwicklung und können diese in Arbeitsprozessen abbilden.
- Die Studierenden können Grundpositionen der digitalen Ethik in praktischen Fällen anwenden und ethische Grundfragen bei Softwareentwicklungen beantworten.
- Die Studierenden haben Kenntnisse der Grundstrukturen des Datenschutzrechts gewonnen und können Grundfragen der Datenschutzcompliance von Softwareprojekten bewerten.
- Die Studierenden haben Kenntnisse der Grundstrukturen des Rechtsrahmens für Künstliche Intelligenz erworben.
- Die Studierenden sind kommunikations- und dialogfähig mit den entsprechenden Expertinnen und Experten für ethische und rechtliche Compliance in ihrem späteren Arbeitsumfeld.

Literatur

- Schweppenhäuser, Gerhard (2021): Grundbegriffe der Ethik. Ditzingen, Reclam: Kap. 2.3 - 3.
- Spieker gen. Döhmann, Indra (2025): Datenschutzrecht. München, Nomos.
- Windholz, Natascha et al. (2024): Praxishandbuch KI-VO. 1 Auflage. München, Hanser.

fiw.thws
Fakultät Informatik und Wirtschaftsinformatik
Technische Hochschule Würzburg-Schweinfurt

Modul: 1512600
Projekt 4

Modulprofil

Prüfungsnummer
1512600

Dauer
1 Semester

Häufigkeit des Angebots
Jedes Semester

SWS
1

ECTS-Credits (CP)
10.0

Workload
Angeleitete Studienzeit:
Synchrone Teilnahme: 1 Std.
Asynchrone Teilnahme: 0 Std.

Selbststudienzeit: 285 Std.

Gesamt: 300 Std.

Lehrveranstaltungsart(en)
Projekt

Lehrsprache
Deutsch/Englisch

Organisation

Modulverantwortung
Prof. Dr. Peter Braun

Dozierende
Prof. Dr. Peter Braun,
Prof. Dr. Isabel John,
Michael Rott,
Prof. Dr. Rolf Schillinger,
Prof. Dr.-Ing. Tobias Fertig,
Prof. Dr. Frank-Michael Schleif,
Prof. Dr.-Ing. Sebastian Biedermann,
Prof. Dr. Tristan Wimmer,
Prof. Dr.-Ing. Anne Heß

Verwendbarkeit	BSED
Studiensemester	6. Semester
Art des Moduls	Pflichtmodul

Verpflichtende Voraussetzungen gemäß SPO
Projekt 1, Projekt 2, Projekt 3. Es müssen min. 100 ECTS erworben worden sein.

Empfohlene Voraussetzungen
Keine

Inhalte
Die Studierenden arbeiten in größeren Teams (5–6 Personen) an einem wissenschaftlich-motivierten und/oder praxisnahen Softwareprojekt mit komplexen funktionalen und nicht-funktionalen Anforderungen. Diese Anforderungen werden entweder durch Interviews mit einem externen Stakeholder (z. B. Unternehmen oder Forschungspartner) oder durch wissenschaftliche Fragestellungen aus aktuellen Forschungsthemen ermittelt. Das Projekt beinhaltet sowohl die technische Umsetzung als auch eine wissenschaftliche Evaluierung der getroffenen Entscheidungen. Die Anwendung könnte folgende Elemente enthalten:

- Moderne Softwarearchitektur, z. B. Microservices oder verteilte Systeme mit Datenbankanbindung (SQL oder NoSQL) mit optimierter Abfrageperformance
- Eine grafische Benutzeroberfläche (Web oder Mobile) mit Usability-Tests
- Sicherheitsaspekte (z. B. Authentifizierung, Verschlüsselung, Security Audits)
- Automatisierte Tests und Code-Qualitätsanalysen zur Sicherstellung langfristiger Wartbarkeit

Die Studierenden werden während der Projektbearbeitung kontinuierlich von einer Dozentin oder einem Dozenten betreut. Die Betreuung erfolgt durch regelmäßige Treffen, in denen der Arbeitsfortschritt besprochen wird, sowie durch begleitende Zwischenpräsentationen, in denen die Entwicklung des Projekts reflektiert und weiterentwickelt wird. Im traditionellen Studiengang geben die Dozierenden die Themen der Praxisbeispiele für die Prüfung vor oder stimmen sie mit den Studierenden ab. Im dualen Studiengang stimmen die Dozierenden eine Aufgabe mit dem Unternehmen ab, so dass der Praxisbezug und das Feedback des Unternehmens gewährleistet sind.

Prüfung

Verpflichtende Voraussetzung gemäß SPO für die Teilnahme an der Prüfung
Keine

Art der Prüfung
Sonstige Prüfung (soP) gemäß §§ 26, 27 APO

Dauer/Form der Prüfung
error

Die konkrete Festlegung der abzuleistenden Prüfung erfolgt im Studienplan

Prüfungssprache
Deutsch/Englisch

Voraussetzung für die Vergabe von Leistungspunkten
Keine

Lernergebnisse
Nach Abschluss des Moduls sind die Studierenden in der Lage:

- Ein komplexes Software-System mit modernen Technologien zu entwickeln
- Eine wissenschaftliche Fragestellung aus dem Bereich Software Engineering zu bearbeiten und methodisch zu evaluieren
- IT-Sicherheitsaspekte in die Software-Architektur zu integrieren
- Mit externen Stakeholdern aus Industrie oder Forschung zu arbeiten und deren Anforderungen zu erheben
- Methoden des agilen Projektmanagement anzuwenden, inkl. Zeitschätzung und Zeiterfassung
- Eine vollständige wissenschaftliche Software-Dokumentation nach akademischen Standards zu erstellen
- Ergebnisse auf einer Konferenz oder in einem wissenschaftlichen Kolloquium zu präsentieren

Literatur
Wird in Abhängigkeit von den konkreten Projekten und verwendeten Technologie in der Veranstaltung bekannt gegeben.

7. Semester

fiw.thws
Fakultät Informatik und Wirtschaftsinformatik
Technische Hochschule Würzburg-Schweinfurt

Modul: 1513400
Bachelorarbeitsmodul

Modulprofil

Prüfungsnummer
1513400

Dauer
1 Semester

Häufigkeit des Angebots
Jedes Semester

SWS
1

ECTS-Credits (CP)
15.0

Workload
Angeleitete Studienzeit:
Synchrone Teilnahme: 1 Std.
Asynchrone Teilnahme: 0 Std.

Selbststudienzeit: 435 Std.

Gesamt: 450 Std.

Lehrveranstaltungsart(en)
Seminar

Lehrsprache
Deutsch/Englisch

Organisation

Modulverantwortung
Prof. Dr. Peter Braun

Dozierende
Prof. Dr. Peter Braun,
Prof. Dr. Isabel John,
Prof. Dr. Eva Wedlich,
Prof. Dr. Rolf Schillinger,
Prof. Dr.-Ing. Tobias Fertig,
Prof. Dr. Frank-Michael Schleif,
Prof. Dr.-Ing. Sebastian Biedermann,
Prof. Dr. Tristan Wimmer,
Prof. Dr.-Ing. Anne Heß

Verwendbarkeit	BSED
Studiensemester	7. Semester
Art des Moduls	Pflichtmodul

Verpflichtende Voraussetzungen gemäß SPO
120 ECTS-Punkte aus den ersten vier Semestern, Praxismodul, Projektarbeit

Empfohlene Voraussetzungen
Grundlagen des wissenschaftlichen Arbeitens im Modul Professional Skills

Inhalte
Das Bachelorarbeitsmodul setzt sich zusammen aus der Bachelorarbeit (12 CP) sowie dem Bachelorseminar (3 CP). Die Bachelorarbeit umfasst eigene Studien und Recherchen über den Stand der Technik und der Wissenschaft des jeweiligen Themengebiets. Die Arbeit muss von Randbedingungen abstrahieren, die ihrer Natur nach nicht technisch begründet sind, sondern aus den spezifischen Gegebenheiten des Unternehmens resultieren. Soweit softwaretechnische Lösungen als Teil der Aufgabe gefordert sind, heißt das in der Regel, dass Prototypen implementiert werden, nicht aber die Sicherstellung von Produkteigenschaften (inkl. begleitender Handbücher etc.) eingeschlossen ist. Im Bachelorseminar werden die Methoden des wissenschaftlichen Arbeitens weiter vertieft und eingeübt.

Prüfung

Verpflichtende Voraussetzung gemäß SPO für die Teilnahme an der Prüfung
Keine

Art der Prüfung
Sonstige Prüfung (soP) gemäß §§ 26, 27 APO

Dauer/Form der Prüfung
Thesis, Präsentation

Die konkrete Festlegung der abzuleistenden Prüfung erfolgt im Studienplan

Prüfungssprache
Deutsch/Englisch

Voraussetzung für die Vergabe von Leistungspunkten
Keine

Lernergebnisse
Mit der Bachelorarbeit / dem Bachelorseminar erbringen die Studierenden den Nachweis, dass sie fähig sind, selbständig eine anspruchsvolle Aufgabenstellung aus dem Gebiet Software Engineering (ggf. fächerübergreifend) zu lösen und dass sie dabei die methodischen und wissenschaftlichen Grundlagen des Faches beherrschen sowie das Ergebnis adäquat darstellen können.

Literatur
in Abhängigkeit des gestellten Themas; wissenschaftliche Literatur ist entsprechend des Themas intensiv zu sichten, zu verwenden und zu zitieren

fiw.thws
Fakultät Informatik und Wirtschaftsinformatik
Technische Hochschule Würzburg-Schweinfurt

Modul: 1513900
FWPM 1

Modulprofil

Prüfungsnummer
1513900

Dauer
1 Semester

Häufigkeit des Angebots
Jedes Wintersemester

SWS
4

ECTS-Credits (CP)
5.0

Workload
Angeleitete Studienzeit:
Synchrone Teilnahme: 1 Std.
Asynchrone Teilnahme: 3 Std.

Selbststudienzeit: 90 Std.

Gesamt: 150 Std.

Lehrveranstaltungsart(en)
Seminar

Lehrsprache
Deutsch/Englisch

Organisation

Modulverantwortung
Prof. Dr. Peter Braun

Dozierende
Prof. Dr. Peter Braun

Verwendbarkeit	BSED
Studiensemester	7. Semester
Art des Moduls	FWPM

Verpflichtende Voraussetzungen gemäß SPO
Keine

Empfohlene Voraussetzungen
Keine

Inhalte
Die Wahlveranstaltungen ermöglichen den Studierenden eine fachliche Vertiefung in spezifischen Bereichen des Software Engineering. Dabei stehen aktuelle Technologien, Methoden und Konzepte im Fokus, die praxisnah vermittelt und in Projekten angewendet werden. Die inhaltliche Ausgestaltung der Wahlveranstaltungen variiert je nach angebotenen Themen, kann jedoch folgende Schwerpunkte umfassen:

- Moderne Softwarearchitekturen: Vertiefung in Microservices, Event-Driven Architecture, Domain-Driven Design (DDD) oder Cloud-native Entwicklung
- Fortgeschrittene Programmierkonzepte: Funktionale Programmierung, Metaprogrammierung, Compilerbau oder domänenspezifische Sprachen (DSLs)
- Softwarequalität und Teststrategien: Testautomatisierung, Continuous Integration/Continuous Deployment (CI/CD), statische Codeanalyse und Performance-Optimierung
- Software Security: Sichere Softwareentwicklung, Penetration Testing, Sicherheitskonzepte in verteilten Systemen
- Mobile und Web-Technologien: Entwicklung von Progressive Web Apps (PWA), native vs. hybride App-Entwicklung, moderne Frameworks und UI/UX-Optimierung
- KI und maschinelles Lernen im Software Engineering: Grundlagen des maschinellen Lernens, Einsatz von KI für Softwaretests oder Codegenerierung
- Datenbank- und Persistenztechnologien: NoSQL-Datenbanken, verteilte Datenhaltung, Optimierung von Datenbankabfragen

Die Studierenden setzen sich in den Wahlveranstaltungen intensiv mit aktuellen Herausforderungen und Lösungen in der Softwareentwicklung auseinander. Praktische Übungen, Projektarbeiten und Diskussionen ergänzen die theoretischen Inhalte und fördern eine anwendungsorientierte Auseinandersetzung mit den jeweiligen Themenbereichen.

Prüfung

Verpflichtende Voraussetzung gemäß SPO für die Teilnahme an der Prüfung
Keine

Art der Prüfung
Sonstige Prüfung (soP) gemäß §§ 26, 27 APO

Dauer/Form der Prüfung
Portfolio

Die konkrete Festlegung der abzuleistenden Prüfung erfolgt im Studienplan

Prüfungssprache
Deutsch/Englisch

Voraussetzung für die Vergabe von Leistungspunkten
Keine

Lernergebnisse
Nach erfolgreichem Abschluss der Wahlveranstaltung sind die Studierenden in der Lage:

- Spezialisierte Konzepte und Technologien in einem ausgewählten Bereich des Software Engineering zu verstehen, anzuwenden und kritisch zu reflektieren, um komplexe Problemstellungen gezielt zu lösen.
- Eigenständig moderne Softwarelösungen zu entwerfen und zu implementieren, unter Berücksichtigung von Best Practices, aktuellen Entwicklungsmethoden und Qualitätsstandards.
- Innovative Ansätze und aktuelle Forschungsergebnisse aus dem gewählten Themenbereich zu analysieren und deren praktische Relevanz für die Softwareentwicklung zu bewerten.

Literatur
Abhängig vom gewählten Modul.

fiw.thws
Fakultät Informatik und Wirtschaftsinformatik
Technische Hochschule Würzburg-Schweinfurt

Modul: 1513901
FWPM 2

Modulprofil

Prüfungsnummer
1513901

Dauer
1 Semester

Häufigkeit des Angebots
Jedes Wintersemester

SWS
4

ECTS-Credits (CP)
5.0

Workload
Angeleitete Studienzeit:
Synchrone Teilnahme: 1 Std.
Asynchrone Teilnahme: 3 Std.

Selbststudienzeit: 90 Std.

Gesamt: 150 Std.

Lehrveranstaltungsart(en)
Seminar

Lehrsprache
Deutsch/Englisch

Organisation

Modulverantwortung
Prof. Dr. Peter Braun

Dozierende
Prof. Dr. Peter Braun

Verwendbarkeit	BSED
Studiensemester	7. Semester
Art des Moduls	FWPM

Verpflichtende Voraussetzungen gemäß SPO
Keine

Empfohlene Voraussetzungen
Keine

Inhalte
Die Wahlveranstaltungen ermöglichen den Studierenden eine fachliche Vertiefung in spezifischen Bereichen des Software Engineering. Dabei stehen aktuelle Technologien, Methoden und Konzepte im Fokus, die praxisnah vermittelt und in Projekten angewendet werden. Die inhaltliche Ausgestaltung der Wahlveranstaltungen variiert je nach angebotenen Themen, kann jedoch folgende Schwerpunkte umfassen:

- Moderne Softwarearchitekturen: Vertiefung in Microservices, Event-Driven Architecture, Domain-Driven Design (DDD) oder Cloud-native Entwicklung
- Fortgeschrittene Programmierkonzepte: Funktionale Programmierung, Metaprogrammierung, Compilerbau oder domänenspezifische Sprachen (DSLs)
- Softwarequalität und Teststrategien: Testautomatisierung, Continuous Integration/Continuous Deployment (CI/CD), statische Codeanalyse und Performance-Optimierung
- Software Security: Sichere Softwareentwicklung, Penetration Testing, Sicherheitskonzepte in verteilten Systemen
- Mobile und Web-Technologien: Entwicklung von Progressive Web Apps (PWA), native vs. hybride App-Entwicklung, moderne Frameworks und UI/UX-Optimierung
- KI und maschinelles Lernen im Software Engineering: Grundlagen des maschinellen Lernens, Einsatz von KI für Softwaretests oder Codegenerierung
- Datenbank- und Persistenztechnologien: NoSQL-Datenbanken, verteilte Datenhaltung, Optimierung von Datenbankabfragen

Die Studierenden setzen sich in den Wahlveranstaltungen intensiv mit aktuellen Herausforderungen und Lösungen in der Softwareentwicklung auseinander. Praktische Übungen, Projektarbeiten und Diskussionen ergänzen die theoretischen Inhalte und fördern eine anwendungsorientierte Auseinandersetzung mit den jeweiligen Themenbereichen.

Prüfung

Verpflichtende Voraussetzung gemäß SPO für die Teilnahme an der Prüfung
Keine

Art der Prüfung
Sonstige Prüfung (soP) gemäß §§ 26, 27 APO

Dauer/Form der Prüfung
Portfolio

Die konkrete Festlegung der abzuleistenden Prüfung erfolgt im Studienplan

Prüfungssprache
Deutsch/Englisch

Voraussetzung für die Vergabe von Leistungspunkten
Keine

Lernergebnisse
Nach erfolgreichem Abschluss der Wahlveranstaltung sind die Studierenden in der Lage:

- Spezialisierte Konzepte und Technologien in einem ausgewählten Bereich des Software Engineering zu verstehen, anzuwenden und kritisch zu reflektieren, um komplexe Problemstellungen gezielt zu lösen.
- Eigenständig moderne Softwarelösungen zu entwerfen und zu implementieren, unter Berücksichtigung von Best Practices, aktuellen Entwicklungsmethoden und Qualitätsstandards.
- Innovative Ansätze und aktuelle Forschungsergebnisse aus dem gewählten Themenbereich zu analysieren und deren praktische Relevanz für die Softwareentwicklung zu bewerten.

Literatur
Abhängig vom gewählten Modul.

fiw.thws
Fakultät Informatik und Wirtschaftsinformatik
Technische Hochschule Würzburg-Schweinfurt

Modul: 1513300
Green IT

Modulprofil

Prüfungsnummer
1513300

Dauer
1 Semester

Häufigkeit des Angebots
Jedes Wintersemester

SWS
4

ECTS-Credits (CP)
5.0

Workload
Angeleitete Studienzeit:
Synchrone Teilnahme: 1 Std.
Asynchrone Teilnahme: 3 Std.

Selbststudienzeit: 90 Std.

Gesamt: 150 Std.

Lehrveranstaltungsart(en)
Seminaristischer Unterricht, Übung

Lehrsprache
Deutsch

Organisation

Modulverantwortung
Prof. Dr. Frank-Michael Schleif

Dozierende
Prof. Dr. Frank-Michael Schleif

Verwendbarkeit	BSED
Studiensemester	7. Semester
Art des Moduls	Pflichtmodul

Verpflichtende Voraussetzungen gemäß SPO
Given the advanced stage in the curriculum, students are expected to bring prior knowledge in software engineering, cloud computing, and machine learning, allowing for a deeper examination of sustainability challenges and solutions in IT.
Programming experience (e.g., Python) and familiarity with AI/ML frameworks (e.g., TensorFlow, PyTorch) are recommended but not mandatory.

Empfohlene Voraussetzungen
- Software Engineering
- Machine Learning
- Data Science

Inhalte
Green AI & IT for Sustainable Digital Systems:
This module addresses the environmental impact of digital technologies and explores sustainable approaches to IT and AI. Students learn core principles of Green IT—such as energy-efficient hardware, data centers, and software—and emerging Green AI methods aimed at reducing the carbon footprint of machine learning. Topics include sustainable software engineering, AI model efficiency, and energy-aware computing. The module also covers process optimization for sustainability, regulatory frameworks (e.g., EU AI Act, GDPR), and ethical considerations. Theory is complemented by practical strategies and expert-led enrichment lectures.

Core Topics:
- Green IT foundations: energy-efficient computing, sustainable infrastructure
- Green software engineering: efficient coding, lifecycle sustainability
- Green AI: low-energy model design, federated learning, inference optimization
- Sustainable process modeling and green BPM
- Legal, regulatory, and ethical frameworks for sustainable IT/AI
- Emerging trends in sustainable computing and AI for global impact

Prüfung

Verpflichtende Voraussetzung gemäß SPO für die Teilnahme an der Prüfung
Keine

Art der Prüfung
Sonstige Prüfung (soP) gemäß §§ 26, 27 APO

Dauer/Form der Prüfung
Portfolio, Schriftliche Prüfung

Die konkrete Festlegung der abzuleistenden Prüfung erfolgt im Studienplan

Prüfungssprache
Deutsch

Voraussetzung für die Vergabe von Leistungspunkten
Keine

Lernergebnisse
Upon successful completion of this module, students will be able to:

1. Understand and explain the core principles of Green IT and Green AI, including their environmental and economic impacts.
2. Describe and evaluate sustainable IT infrastructures and energy-aware AI systems, including data centers, cloud solutions, and efficient model architectures
3. Interpret and apply relevant legal regulations, standards, and policies related to sustainability in IT and AI (e.g., EU AI Act, ISO 14001)
4. Assess the carbon footprint and resource consumption of IT systems and AI models using established tools; optimize their efficiency through software and AI techniques
5. Model and analyze IT processes with a focus on energy/resource efficiency and sustainability goals
6. Develop comprehensive strategies for implementing sustainable IT and AI practices in organizational settings, integrating technical, regulatory, and business considerations

Literatur

- Hilty, L. M., & Aebischer, B., ICT Innovations for Sustainability. Springer.
- Strubell, E., Ganesh, A., & McCallum, A. (2019), Energy and Policy Considerations for Deep Learning in NLP, https://arxiv.org/abs/1906.02243
- Bolón-Canedo, V., et al. (2023), Green Machine Learning: Advances in Energy-Efficient AI Models, Proceedings of ESANN 2023.
- European Parliament, AI Act: Environmental Regulations for Artificial Intelligence in the EU

fiw.thws
Fakultät Informatik und Wirtschaftsinformatik
Technische Hochschule Würzburg-Schweinfurt

Modul: 1513100
Transferkolloquium

Modulprofil

Prüfungsnummer
1513100

Dauer
4 Semester

Häufigkeit des Angebots
Jedes Semester

SWS
4

ECTS-Credits (CP)
5.0

Workload
Angeleitete Studienzeit:
Synchrone Teilnahme: 1 Std.
Asynchrone Teilnahme: 3 Std.

Selbststudienzeit: 90 Std.

Gesamt: 150 Std.

Lehrveranstaltungsart(en)
Seminar

Lehrsprache
Deutsch

Organisation

Modulverantwortung
Prof. Dr.-Ing. Sebastian Biedermann

Dozierende
Prof. Dr. Peter Braun,
Prof. Dr.-Ing. Sebastian Biedermann

Verwendbarkeit	BSED
Studiensemester	7. Semester
Art des Moduls	Pflichtmodul

Verpflichtende Voraussetzungen gemäß SPO
Das Modul richtet sich nur an Studierende in der dualen Studienvariante.

Empfohlene Voraussetzungen
Keine

Inhalte
Das Transferkolloquium dient dem Austausch der Studierenden in der dualen Studienvariante und ersetzt eines der FWPM. Studierende in der normalen Variante des Studiengangs nehmen nicht an dieser Veranstaltung teil. Die Veranstaltung findet verteilt über das dritte bis siebte Semester mit Ausnahme des Praxissemesters statt. Unter Leitung eines Dozierenden tauschen sich die dualen Studierenden über folgende Themen aus:

- Bedeutung des Austauschs Hochschule–Unternehmen
- Reflexion von Praxis- und Studienerfahrungen im beruflichen Kontext
- Austausch zu Herausforderungen und Lösungen im dualen Studium
- Best-Practice-Beispiele aus dem Unternehmensalltag
- Strategien für Zeitmanagement und Problemlösung
- Anwendung theoretischer Inhalte im Betrieb
- Grundlagen und Umsetzung von IT-Sicherheit in der Praxis
- Dokumentation und Präsentation von Projektarbeiten

Prüfung

Verpflichtende Voraussetzung gemäß SPO für die Teilnahme an der Prüfung
Keine

Art der Prüfung
Sonstige Prüfung (soP) gemäß §§ 26, 27 APO

Dauer/Form der Prüfung
Portfolio

Die konkrete Festlegung der abzuleistenden Prüfung erfolgt im Studienplan

Prüfungssprache
Deutsch

Voraussetzung für die Vergabe von Leistungspunkten
Keine

Lernergebnisse

- Die Studierenden sind in der Lage, ihre Praxiserfahrungen systematisch zu reflektieren und daraus Schlüsse für ihre berufliche Entwicklung zu ziehen.
- Die Studierenden entwickeln Strategien zur Lösung von Problemen, die bei der Abstimmung zwischen Studium und Praxis auftreten.
- Die Studierenden können theoretisch erworbenes Wissen praxisnah im Unternehmen anwenden und somit den Wissenstransfer zwischen Hochschule und Praxis optimieren.
- Die Studierenden sind in der Lage, ihre Praxiserfahrungen klar und strukturiert zu präsentieren und effektiv zu kommunizieren.
- Die Studierenden verstehen die Bedeutung von IT-Sicherheit im Unternehmenskontext und können entsprechende Maßnahmen in ihrer täglichen Arbeit umsetzen.
- Die Studierenden sind in der Lage, Projektdokumentationen professionell zu erstellen und zu präsentieren, um ihre Arbeit transparent und nachvollziehbar zu machen.
- Die Studierenden lernen, Feedback konstruktiv zu nutzen, um ihre Arbeitsweise und die Zusammenarbeit zwischen Hochschule und Unternehmen kontinuierlich zu verbessern.
- Die Studierenden entwickeln ein Verständnis für ihre beruflichen Entwicklungsmöglichkeiten und können fundierte Entscheidungen für ihre Karriereplanung treffen.

Literatur
Wird im Seminar bekannt gegeben.