

## 4.2.5 Informatik 2

<b>Informatik 2</b>
<b>Modulübersicht</b>
EDV-Bezeichnung: QUCB250
Modulverantwortliche(r): Prof. Dr. NN
Modulumfang (ECTS): 6 CP
Einordnung (Semester): 2. Semester
Inhaltliche Voraussetzungen: Kenntnisse aus den dem Modul Informatik 2
Voraussetzungen nach SPO: Nach SPO sind keine formellen Voraussetzungen erforderlich.
<p>Lernergebnisse und Kompetenzen: Die Studierenden vertiefen ihre Kenntnisse der Programmierung und erwerben ein grundlegendes Verständnis für Algorithmen und Datenstrukturen. Sie sind in der Lage, algorithmische Probleme zu analysieren, geeignete Lösungsverfahren auszuwählen und effizient zu implementieren.</p> <p><b>Fachliche Kompetenzen</b> Die Studierenden</p> <ul style="list-style-type: none"> <li>• verstehen grundlegende Algorithmen und elementare Datenstrukturen (z. B. Listen, Bäume, Hashtabellen, Graphen),</li> <li>• formulieren Algorithmen in Pseudocode und schätzen deren Zeit- und Speicheraufwand ab,</li> <li>• wählen für gegebene Problemstellungen geeignete Algorithmen und Datenstrukturen aus,</li> <li>• analysieren und begründen die Korrektheit einfacher Algorithmen,</li> <li>• kennen theoretische Grenzen der Berechenbarkeit (z. B. Turingmaschinen, NP-Vollständigkeit),</li> <li>• setzen ausgewählte Algorithmen praktisch in einer Programmiersprache um und bewerten deren Effizienz.</li> </ul> <p><b>Methodische Kompetenzen</b> Die Studierenden</p> <ul style="list-style-type: none"> <li>• wenden systematische Verfahren zur Analyse, Entwurf und Optimierung von Algorithmen an,</li> <li>• testen, vergleichen und dokumentieren Implementierungen,</li> <li>• nutzen Werkzeuge zur Entwicklungs- und Laufzeitanalyse,</li> <li>• strukturieren komplexe Problemstellungen in Teilprobleme und abstrahieren diese algorithmisch.</li> </ul> <p><b>Sozial- und Selbstkompetenzen</b> Die Studierenden</p> <ul style="list-style-type: none"> <li>• arbeiten eigenständig und im Team an Programmieraufgaben,</li> </ul>

<ul style="list-style-type: none"> <li>• planen und reflektieren den Entwicklungsprozess,</li> <li>• dokumentieren und präsentieren Ergebnisse adressatengerecht.</li> </ul>
<p>Prüfungsleistungen:                  Informatik 2: Klausur, 120 Minuten                  Übungen Informatik 2 (Studienleistung): Die Übungen gelten als bestanden, wenn die Übungsblätter/ -projekte erfolgreich bearbeitet und testiert wurden.</p>

<b>Lehrveranstaltung: Informatik 2</b>
EDV-Bezeichnung: QUCB251
Dozierende(r): Prof. Dr. NN
Umfang (SWS): 4
Turnus: Sommersemester
Art, Modus: Vorlesung, Pflichtfach
Lehrsprache: Deutsch
<p>Studieninhalte:</p> <ul style="list-style-type: none"> <li>• Einführung in das algorithmische Problemlösen und die Analyse von Algorithmen</li> <li>• Formulierung von Algorithmen in Pseudocode, Ablaufdiagrammen und Programmiersprachen</li> <li>• Grundlegende Datenstrukturen: Arrays, Listen, Stacks, Queues, Bäume, Hashtabellen, Graphen</li> <li>• Zentrale Algorithmen: Suchverfahren (linear, binär, Hash-basierte Suche), Sortierverfahren (Insertion-, Merge-, Quick-Sort), Traversierungen und Graphalgorithmen (DFS, BFS, kürzeste Wege)</li> <li>• Rekursive und iterative Lösungsansätze</li> <li>• Analyse von Laufzeit- und Speicherkomplexität (Big-O-Notation)</li> <li>• Nachweis der Korrektheit einfacher Algorithmen</li> <li>• Einführung in theoretische Konzepte: Turingmaschinen, Entscheidbarkeit, NP-Vollständigkeit</li> <li>• Vergleich von Implementierungsstrategien und Effizienz</li> <li>• Diskussion praktischer Anwendungen (z. B. Suchen, Planen, Sortieren, Netzwerke)</li> </ul>
<p>Empfohlene Literatur:</p> <ul style="list-style-type: none"> <li>• Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein: Introduction to Algorithms, 3. Aufl., MIT Press, 2009.</li> <li>• Robert Sedgwick: Algorithms in C++, Parts 1-4: Fundamentals, Data Structure, Sorting, Searching, 3. Aufl., Pearson, 1998.</li> <li>• Robert Sedgwick, Algorithms in C++, Part 5: Graph Algorithms, 3. Aufl., Pearson, 2001.</li> <li>• D. W. Hoffmann: Theoretische Informatik, 5. Aufl., Hanser, 2022.</li> <li>• Michael Sipser: Introduction to the Theory of Computation, 3. Aufl., Cengage Learning, 2012.</li> </ul>

<b>Lehrveranstaltung: Übungen Informatik 2</b>
EDV-Bezeichnung: QUCB252

Dozierende(r): Prof. Dr. NN
Umfang (SWS): 2
Turnus: Sommersemester
Art, Modus: Übungen, Pflichtfach
Lehrsprache: Deutsch
Studieninhalte: <ul style="list-style-type: none"><li>• Implementierung und Test zentraler Algorithmen (Such-, Sortier- und Graphverfahren)</li><li>• Anwendung und Vergleich verschiedener Datenstrukturen</li><li>• Analyse von Laufzeiten durch Messung und Profiling</li><li>• Praktische Belege zur Korrektheit und Effizienz von Implementierungen</li><li>• Bearbeitung komplexerer Aufgabenstellungen im Team</li><li>• Abschlussprojekt: Entwicklung einer kleinen Anwendung unter Nutzung geeigneter Datenstrukturen</li></ul>
Empfohlene Literatur: <ul style="list-style-type: none"><li>• Siehe Vorlesung Informatik 2</li></ul>