

Prep-For-Study

Studienvorbereitungsprogramm

Modulbeschreibung

Informatik

Inhaltsverzeichnis

| | |
|---|---|
| Modulbeschreibung | 3 |
| Lerninhalte für das Studienvorbereitungsprogramm im Fach Informatik | 4 |
| 1. Theoretische Informatik | 5 |
| 2. Technische Informatik | 6 |
| 3. Praktische Informatik | 7 |

Modulbeschreibung

| | |
|---|--|
| Modulbezeichnung | Informatik |
| Studiensemester: | 1 und 2 |
| Modulverantwortliche(r): | Prof. Dr. Markus Esch, Dr. Eric Wagner |
| Unterrichtssprache: | Deutsch |
| Lehrform/SWS: | 4-5 Vorlesungen in Präsenz / im Durchschnitt 4-5 SWS / maximal 30 Teilnehmer*innen |
| Arbeitsaufwand: | Die Präsenzzeit dieses Moduls umfasst bei 4-5 SWS 144 Veranstaltungsstunden (= 108 Zeitstunden). Der Gesamtumfang des Moduls / Arbeitslast beträgt 189 h. Daher stehen für das Eigenstudium einschließlich Prüfungsvorbereitung, jeweils in Zeitstunden und summiert 81 h zur Verfügung. |
| Kreditpunkte: | Keine |
| Voraussetzungen nach Prüfungsordnung | Keine |
| Empfohlene Voraussetzungen: | Sprachkenntnisse B1 |
| Literatur | |
| Studien-/ Prüfungsleistungen/ Prüfungsformen: | Schriftliche Feststellungsprüfung |

Lerninhalte für das Studienvorbereitungsprogramm im Fach Informatik

Die Lerninhalte sind unter der Annahme gewählt, dass keine Vorkenntnisse im Bereich der Informatik vorhanden sind. Alle Themen müssen daher fachsprachlich sensibilisiert werden. Neben der Fachsprache sollen auch das strukturierte Denken sowie die Fähigkeit zur Abstraktion von Problemstellungen und entsprechenden Lösungswegen gesichert werden.

Die Inhalte sind in vier Module zu den Themengebieten Theoretische Informatik, Praktische Informatik und Technische Informatik aufgeteilt. In allen Modulen werden grundlegende Kenntnisse vermittelt; informatikspezifisches Vorwissen wird nicht vorausgesetzt. Durch die Module werden die wichtigsten Teilgebiete der Informatik abgedeckt. Somit erhalten die Teilnehmenden einen breiten Überblick über Informatik-spezifische Fragestellungen.

1. Theoretische Informatik

Lernziele

Die Studierenden können zentrale Fachbegriffe der Informatik und der Boolesche Algebra erklären. Sie sind in der Lage, die grundlegenden Rechenregeln der Booleschen Algebra anzuwenden. Sie können die grundlegenden Modelle der Theoretischen Informatik nach Shannon sowie der Automatentheorie beschreiben und anwenden. Sie können die Eigenschaften einer Turing-Maschine benennen und einfache Programme für eine Turing-Maschine entwickeln. Basierend darauf können Sie das Halteproblem und die Church-Turing-These erklären. Sie können die Bedeutung der Laufzeit von Algorithmen erklären und einfache Laufzeitanalysen unter Verwendung der O-Notation durchführen.

Inhalte

- Fachbegriffe der Informatik
- Mathematische Grundlagen
 - Zahlensysteme
 - Boolesche Algebra
- Informationstheorie nach Shannon
- Turing-Maschine
- O-Notation

2. Technische Informatik

Lernziele

Im Modul Technische Informatik erlernen die Studierenden die technischen Aspekte moderner Computersysteme. Sie lernen die Historie von Rechenmaschinen bis hin zu modernen Computersystemen. Die Studierenden verstehen die Von-Neumann-Architektur als Grundlage modernen Computersysteme und können deren wichtigsten Merkmale benennen. Zum Austausch von Informationen zwischen Computersystemen lernen die Studierenden verschiedene Arten der Datenspeicherung, unterschiedliche Verfahren der Datenübertragung und darauf aufbauend unterschiedliche Netzwerktopologien sowie das OSI-Referenzmodells kennen. Sie können einzelne Verfahren Anwendungen, Vor- und Nachteile der der einzelnen Verfahren und Netzwerktopologien erklären und verstehen die Funktionen der einzelnen Schichten des OSI-Referenzmodells.

Inhalte

- Historie von Computersystemen
- Architektur des Von-Neumann-Rechners
- Speichermedien
- Aufzeichnungsverfahren/Verfahren der Datenübertragung
- Netzwerktechnologie

3. Praktische Informatik

Lernziele

Die Studierenden lernen ein Unix-basiertes Betriebssystem kennen und können grundlegende Befehle über das Command Line Interface ausführen. Die Studierenden lernen unterschiedliche Paradigmen der Programmierung kennen, können deren Unterschiede erklären und beispielhafte Programmiersprachen für die jeweiligen Paradigmen nennen. Die Studierenden können die Konzepte der prozeduralen Programmierung und der Datenabstraktion erklären und diese in einer gängigen Programmiersprache umsetzen. Die Studierenden sind sie in der Lage, gut strukturierte Programme zu erstellen. Die Studierenden können grundlegende Konzepte objektorientierter Programmiersprachen erklären. Sie sind in der Lage, einfache objektorientierte Programme in einer gängigen Programmiersprache zu konzipieren, zu implementieren und auszuführen.

Inhalte

- Unix-Betriebssysteme und Command Line Interface
- Prozedurale Programmierung
 - Fundamentale Datentypen
 - Operatoren
 - Kontrollstrukturen
- Objektorientierte Programmierung
 - Entwurf und Implementierung von Klassen
 - Erzeugung von Objekten
 - Kapselung von Daten