

# Lehrveranstaltung

## INF2 - Informatik 2

---

Version: 5 | Letzte Änderung: 29.09.2019 16:57 | Entwurf: 0 | Status: vom verantwortlichen Dozent freigegeben

### ^ Allgemeine Informationen

<b>Langname</b>	Informatik 2
<b>Anerkennende LModule</b>	<u>INF2_BaMT</u>
<b>Verantwortlich</b>	Prof. Dr.-Ing. Arnulph Fuhrmann Professor Fakultät IME
<b>Niveau</b>	Bachelor
<b>Semester im Jahr</b>	Sommersemester
<b>Dauer</b>	Semester
<b>Stunden im Selbststudium</b>	90
<b>ECTS</b>	6
<b>Dozenten</b>	Prof. Dr.-Ing. Arnulph Fuhrmann Professor Fakultät IME
<b>Voraussetzungen</b>	Informatik 1
<b>Unterrichtssprache</b>	deutsch
<b>separate Abschlussprüfung</b>	Ja

## Abschlussprüfung

### Details

Die Studierenden weisen in einer schriftlichen Klausur folgende Kompetenzen nach:

- Entwicklung kurzer Programme zur Lösung von umgangssprachlich beschriebenen Problemen (K.4, K.9)
- Entwicklung kurzer Programme zur Lösung von abstrakt beschriebenen Problemen (K.2)
- Anwendung programmiersprachlicher und abstrakterer Konstrukte zur Lösung von Anwendungsproblemen (K.8)
- Lesen, verstehen und ggf. Korrektur von vorgegebenen Programmfragmenten (K.4, K.10)
- Bewertung von Aussagen hinsichtlich ihrer Korrektheit (K.11)

### Mindeststandard

Mindestens 50% der möglichen Gesamtpunktzahl.

## Prüfungstyp

Die Studierenden weisen in einer schriftlichen Klausur folgende Kompetenzen nach:

- Entwicklung kurzer Programme zur Lösung von umgangssprachlich beschriebenen Problemen (K.4, K.9)
- Entwicklung kurzer Programme zur Lösung von abstrakt beschriebenen Problemen (K.2)
- Anwendung programmiersprachlicher und abstrakterer Konstrukte zur Lösung von Anwendungsproblemen (K.8)
- Lesen, verstehen und ggf. Korrektur von vorgegebenen Programmfragmenten (K.4, K.10)
- Bewertung von Aussagen hinsichtlich ihrer Korrektheit (K.11)

## ^ Vorlesung / Übungen

### Lernziele

---

#### Kenntnisse

Fortgeschrittene Methoden der Objektorientierung  
Polymorphismus  
Abstrakte Klassen  
Interfaces  
Modellierung  
Generische Programmierung  
Dynamische Datenstrukturen  
verkettete Listen  
Stacks  
Queues  
Hash-Tabellen  
Bäume  
Algorithmen  
Komplexität  
O-Notation  
Zeitaufwand  
Speicheraufwand  
Messung der Performance  
Allgemeine Strategien zum Entwurf von Algorithmen  
Brute-force  
greedy  
divide-and-conquer  
backtracking  
Sortierverfahren  
Selection Sort  
Insertion Sort  
Merge Sort  
Suchverfahren  
Lineare Suche  
Binäre Suche

---

#### Fertigkeiten

Erstellen von objektorientierten Programmen in Java  
Entwerfen von objekt-orientierten Modellen zu einer gegebenen Problemstellung  
Verwenden von Klassendiagrammen  
Umsetzen in Software  
dynamische Datenstrukturen  
Verwenden von dynamischen Datenstrukturen in Java  
Entwerfen von dynamischen Datenstrukturen  
Implementieren von dynamischen Datenstrukturen in Java  
Bestimmen der Komplexität von Algorithmen  
Lösen einer Problemstellung mittels geeigneter Algorithmen  
Auswählen von Algorithmen  
Entwerfen von Algorithmen  
Implementieren von Algorithmen in Java

## Aufwand Präsenzlehre

Typ	Präsenzzeit (h/Wo.)
Vorlesung	3
Übungen (ganzer Kurs)	0
Übungen (geteilter Kurs)	2
Tutorium (freiwillig)	2

## Separate Prüfung

### Prüfungstyp

Übungsaufgabe mit fachlich / methodisch eingeschränktem Fokus lösen

### Details

Selbständiges Lösen von Selbstlernaufgaben zu den Themen der Vorlesung in Form der Entwicklung komplexerer Programme zur Lösung von umgangssprachlich oder abstrakt beschriebenen Problemen (K.4, K.5, K.9, K.2).

### Mindeststandard

Mehr als 80% aller Übungsaufgaben abgegeben. Eine Aufgabe, gilt als abgegeben, wenn diese zum überwiegenden Teil und selbstständig gelöst wurde.