

## Modul

# PI2 - Praktische Informatik 2

Bachelor Technische Informatik 2020

---

Version: 3 | Letzte Änderung: 25.09.2019 11:21 | Entwurf: 0 | Status: vom Modulverantwortlichen freigegeben | Verantwortlich: Yuan

### ^ Allgemeine Informationen

<b>Anerkannte Lehrveranstaltungen</b>	<u>PI2_Yuan</u>
<b>Fachsemester</b>	2
<b>Dauer</b>	1 Semester
<b>ECTS</b>	5
<b>Zeugnistext (de)</b>	Praktische Informatik 2
<b>Zeugnistext (en)</b>	Practical Informatics 2
<b>Unterrichtssprache</b>	deutsch oder englisch
<b>abschließende Modulprüfung</b>	Ja

### Modulprüfung

<b>Benotet</b>	Ja
<b>Frequenz</b>	Jedes Semester

### Prüfungskonzept

Schriftliche Klausur

Die Prüfung stellt sicher, dass Studierende auch individuell die Ziele des L.O. durch Aufgaben der folgenden Typen erreicht haben:

\*Fragen zum Grundwissen (K.3, K.8)

\*Objektorientierte Programmierung (K.2, K.5, K.6)

\*Finden von Fehlern in vorgegebenen Programmstücken (K.4, K.7)

\*Arbeiten mit dynamischen Datenstrukturen, insbesondere mit Bäumen (K.3, K.4, K.7)

\*Arbeiten mit rekursiven Methoden (K.4, K.7, K.6)

## ^ Allgemeine Informationen

### Inhaltliche Voraussetzungen

PI1

Grundkenntnisse in PI1

-Praktische Informatik 1

Programmiererfahrung mit Entwicklungsumgebungen wie Eclipse

### Kompetenzen

Kompetenz	Ausprägung
fachliche Probleme abstrahieren und formalisieren	diese Kompetenz wird vermittelt
Konzepte und Methoden der Informatik, Mathematik und Technik kennen und anwenden	diese Kompetenz wird vermittelt
Systeme entwerfen	diese Kompetenz wird vermittelt
Systeme realisieren	diese Kompetenz wird vermittelt
Systeme prüfen	diese Kompetenz wird vermittelt
Typische Werkzeuge, Standards und Best Practices der industriellen Praxis kennen und einsetzen	diese Kompetenz wird vermittelt
In vorhandene Systeme einarbeiten und vorhandene Komponenten sinnvoll nutzen	diese Kompetenz wird vermittelt

## ^ Vorlesung / Übungen

### Exemplarische inhaltliche Operationalisierung

Aufbauend auf PI1 führt die Lehrveranstaltung in grundlegende Konzepte der Programmierung ein und illustriert sie am Beispiel einer höheren Programmiersprache. Die Studierenden sollen lernen, diese Konzepte selbstständig zur Lösung von Anwendungsproblemen einzusetzen.

### Separate Prüfung

## ^ Praktikum

### Exemplarische inhaltliche Operationalisierung

Die Studierenden schließen sich zu Kleingruppen zusammen.

Jede Kleingruppe absolviert mehrere Praktikumsrunden mit zugewiesenen Laborterminen.

In jeder Runde werden Programmieraufgaben algorithmischer und objektorientierter Art gelöst.

Zur Vorbereitung eines Labortermins muss ein Vorbereitungsblatt praktisch gelöst werden.

Im Laborterminen werden zusätzliche Aufgaben unter Aufsicht (und ggf. mit Hilfestellung) bearbeitet.

### Separate Prüfung

<b>Benotet</b>	Nein
<b>Frequenz</b>	Einmal im Jahr
<b>Voraussetzung für Teilnahme an Modulprüfung</b>	Ja

### Prüfungskonzept

Verschiedene praxisnahe Aufgabenstellungen in kleinen Teams bearbeiten, d.h. die Aufgabenstellung analysieren und ein Programm zu ihrer Lösung entwerfen, programmieren, testen, fristgerecht die Lösung abzuliefern, im Abnahmegespräch mit dem Lehrenden begründen und kleine Erweiterungen vornehmen zu können.