

# OOP - Objektorientierte Programmierung

## OOP - Object oriented programming

---

<b>Allgemeine Informationen</b>	
<b>Modulkürzel oder Nummer</b>	OOP
<b>Eindeutige Bezeichnung</b>	OOProg-01-BA-M
<b>Modulverantwortlich(e)</b>	Prof. Dr. Manzke, Robert (robert.manzke@haw-kiel.de)
<b>Lehrperson(en)</b>	Bode, Ralf (ralf.bode@haw-kiel.de) Lee, Michael (michael.lee@haw-kiel.de)
<b>Wird angeboten zum</b>	Sommersemester 2018
<b>Moduldauer</b>	1 Fachsemester
<b>Angebotsfrequenz</b>	Regelmäßig
<b>Angebotsturnus</b>	In der Regel im Sommersemester
<b>Lehrsprache</b>	Deutsch
<b>Empfohlen für internationale Studierende</b>	Ja
<b>Ist als Wahlmodul auch für andere Studiengänge freigegeben (ggf. Interdisziplinäres Modulangebot - IDL)</b>	Ja

<b>Studiengänge und Art des Moduls (gemäß Prüfungsordnung)</b>
Studiengang: B.Eng. - E - Elektrotechnik (PO 2017, V3) Vertiefungsrichtung: Technische Informatik Modulart: Pflichtmodul Fachsemester: 4
Studiengang: B.Eng. - Wing - Wirtschaftsingenieurwesen - Elektrotechnik (PO 2017, V1) Vertiefungsrichtung: Digitale Wirtschaft Modulart: Pflichtmodul Fachsemester: 4
Studiengang: B.Sc. - INI - Informationstechnologie (PO 2017, V1) Modulart: Pflichtmodul Fachsemester: 2
Studiengang: B.Sc. - WINF - Wirtschaftsinformatik (6 Sem.) Modulart: Pflichtmodul Fachsemester: 4

<b>Kompetenzen / Lernergebnisse</b>
<i>Kompetenzbereiche: Wissen und Verstehen; Einsatz, Anwendung und Erzeugung von Wissen; Kommunikation und Kooperation; Wissenschaftliches Selbstverständnis/Professionalität.</i>
Die Studierenden lernen die Grundelemente objektorientierter Softwareentwicklung kennen. Am Beispiel der Programmiersprache Java werden die objektorientierte Analyse und der objektorientierte Entwurf eingeführt mit dem Ziel, fortgeschrittene Programmiersprachkonzepte zu beherrschen und anwenden zu können. In Laborübungen mit kleinen Gruppen wird aufgaben- und problemorientiertes Denken gestärkt, um die Probleme der modernen Softwareentwicklung beurteilen und verstehen zu können.

Das in der Vorlesung erlernte Wissen wird in Übungen und ggf. einer Semesterarbeit an vorlesungsbezogenen Beispielen angewendet und vertieft.

In den Übungen wird die Programmiersprache Java unter den gängigen Entwicklungsumgebungen eingesetzt, um die Grundkonzepte der objektorientierten Programmierung nachzuvollziehen.

Die Vorlesung und Übung sind nicht als Step-by-Step für das Erlernen einer Programmiersprache ausgelegt. Eigenverantwortliches Einarbeiten in die Lerninhalte sind daher für jeden Studenten für den Lernerfolg notwendig.

### Angaben zum Inhalt

<b>Lehrinhalte</b>	<ul style="list-style-type: none"> <li>- Objektorientierter Entwurf: Definition und Begriffe,</li> <li>- Objektmodellierung, Klassen und Objekte, Vererbung, Kapselung, Methoden (überladen, überschreiben), Polymorphismus, Delegation</li> <li>- Objektorientierte Programmiersprachen: Java, C++; ABAP</li> <li>- Speicherverwaltung unter Java vs. C++</li> <li>- Polymorphismus durch VMT</li> <li>- Designpattern: Singleton, Kompositum, Visitor, ...</li> <li>- Java-Erweiterungen: Lambda, Interfaces</li> <li>- Delegation über Behaviour injection</li> </ul>
<b>Literatur</b>	<p>Christian Ullenboom: Java ist auch eine Insel  Gamma et al: Designpatterns  Grady, Booch: Objektorientierte Analyse und Design  Aho, Ullman, Sethi: Compilerbau</p>

### Lehrformen der Lehrveranstaltungen

Lehrform	SWS
Labor	1
Lehrvortrag	3

### Arbeitsaufwand

<b>Anzahl der SWS</b>	4 SWS
<b>Leistungspunkte</b>	5,00 Leistungspunkte
<b>Präsenzzeit</b>	48 Stunden
<b>Selbststudium</b>	102 Stunden

### Modulprüfungsleistung

<b>Voraussetzung für die Teilnahme an der Prüfung gemäß PO</b>	Keine
<b>OOP - Übung</b>	Prüfungsform: Übung Gewichtung: 0% wird angerechnet gem. § 11 Absatz 2 PVO: Ja Benotet: Nein
<b>OOP - Klausur</b>	Prüfungsform: Klausur Dauer: 120 Minuten Gewichtung: 100% wird angerechnet gem. § 11 Absatz 2 PVO: Ja Benotet: Ja

### Sonstiges

<b>Empfohlene Voraussetzungen</b>	PRG Programmieren
-----------------------------------	-------------------