

# Lehrveranstaltung

## VMA - Programmierung verteilter und mobiler Anwendungen

---

Version: 1 | Letzte Änderung: 02.08.2019 08:53 | Entwurf: 0 | Status: vom verantwortlichen Dozent freigegeben

### ^ Allgemeine Informationen

<b>Langname</b>	Programmierung verteilter und mobiler Anwendungen
<b>Anerkennende LModule</b>	<a href="#">VMA_BaET</a> , <a href="#">VMA_BaTIN</a>
<b>Verantwortlich</b>	Prof. Dr. Cartsten Vogt Professor Fakultät IME
<b>Niveau</b>	Bachelor
<b>Semester im Jahr</b>	Sommersemester
<b>Dauer</b>	Semester
<b>Stunden im Selbststudium</b>	60
<b>ECTS</b>	5
<b>Dozenten</b>	Prof. Dr. Cartsten Vogt Professor Fakultät IME
<b>Voraussetzungen</b>	Objektorientierte Programmierung (idealerweise Java) Struktur und Funktionalität von Betriebssystemen Programmierung mit Nebenläufigkeit / Threading und Sockets Kommunikationsprotokolle für Datennetze Relationale Datenbanken
<b>Unterrichtssprache</b>	deutsch, englisch bei Bedarf
<b>separate Abschlussprüfung</b>	Ja

## Abschlussprüfung

### Details

Die Studierenden sollen in einer schriftlichen Klausur folgende Kompetenzen nachweisen: 1.) Sicherer Umgang mit grundlegenden Begrifflichkeiten, 2.) Anwendung programmiersprachlicher Konstrukte zur Lösung von Anwendungsproblemen im Bereich der Mobilgeräteprogrammierung, 3.) Prüfung von Aussagen und programmiersprachlicher Konstrukte auf Korrektheit. Typische Aufgabenformen zu 1.) sind Multiple-Choice-Fragen, Lückentexte,

Bewertung von Aussagen hinsichtlich ihrer Korrektheit, zu 2.) Lösung kleinerer umgangssprachlich formulierter Probleme durch Programmstücke und zu 3.) das Finden von Fehlern in Aussagen und in vorgegebenen Programmstücken.

### Mindeststandard

Mindestens 50% der möglichen Gesamtpunktzahl.

### Prüfungstyp

Die Studierenden sollen in einer schriftlichen Klausur folgende Kompetenzen nachweisen: 1.) Sicherer Umgang mit grundlegenden Begrifflichkeiten, 2.) Anwendung programmiersprachlicher Konstrukte zur Lösung von Anwendungsproblemen im Bereich der Mobilgeräteprogrammierung, 3.) Prüfung von Aussagen und programmiersprachlicher Konstrukte auf Korrektheit. Typische Aufgabenformen zu 1.) sind Multiple-Choice-Fragen, Lückentexte, Bewertung von Aussagen hinsichtlich ihrer Korrektheit, zu 2.) Lösung kleinerer umgangssprachlich formulierter Probleme durch Programmstücke und zu 3.) das Finden von Fehlern in Aussagen und in vorgegebenen Programmstücken.

## ^ Vorlesung / Übungen

### Lernziele

---

#### Kenntnisse

grundlegende Begriffe und Techniken

Eigenschaften von Mobilgeräten im Vergleich zu traditionellen Computern

Übersicht über aktuelle Mobilgeräte-Betriebssysteme und -Programmierplattformen

grundlegende Vorgehensweise bei der Programmierung eines Mobilgeräts (Programmerstellung, -emulation und -installation)

---

Programmierung von Mobilgeräten am Beispiel eines oder mehrerer aktueller Systeme (Randbemerkung: Dieser (Haupt-)Teil der Lehrveranstaltung wird jeweils den aktuellen technischen Gegebenheiten und der aktuellen Marktsituation angepasst. An dieser Stelle werden daher die voraussichtlich zu behandelnden Themengebiete nur grob genannt.)

Komponenten einer Mobilgeräte-Applikation

graphische Oberflächen

Datenhaltung

Nebenläufigkeit

Datenkommunikation, insbes. Internetzugriff

ortsabhängige Dienste

Sicherheit

---

Bewertung der Risiken bei der Programmierung und beim Gebrauch von Mobilgeräten

---

#### Fertigkeiten

Umgang mit Softwareentwicklungsumgebungen für Mobilgeräte

---

Programmetechnische Realisierung von Mobilgeräte-Applikationen mittlerer Komplexität

### Aufwand Präsenzlehre

Typ	Präsenzzeit (h/Wo.)
Vorlesung	2
Übungen (ganzer Kurs)	1
Übungen (geteilter Kurs)	1
Tutorium (freiwillig)	0

## Separate Prüfung

keine

## ^ Praktikum

### Lernziele

---

#### Kenntnisse

Programmierung von Smartphones am Beispiel eines aktuellen Systems - Details wie unter "Vorlesung/Übung" angegeben

---

#### Fertigkeiten

Umgang mit Softwareentwicklungsumgebungen für Smartphones

---

Realisierung von Smartphone-Applikationen mittlerer Komplexität in kleinen Teams

### Aufwand Präsenzlehre

Typ	Präsenzzeit (h/Wo.)
Praktikum	1
Tutorium (freiwillig)	0

## Separate Prüfung

### Prüfungstyp

praxisnahes Szenario bearbeiten (z.B. im Praktikum)

## Details

Die Studierenden schließen sich zu Kleingruppen zusammen. Jede Kleingruppe absolviert mehrere "Praktikumsrunden" mit zugewiesenen Laborterminen. In jeder Runde werden Programmieraufgaben gelöst.

Zur Vorbereitung eines Labortermins muss ein "Vorbereitungsblatt" praktisch gelöst werden. Die dabei erworbenen Kenntnisse werden zu Beginn des Termins geprüft (kurzer schriftlicher Eingangstest, persönliches Gespräch mit dem Betreuer). Wird diese Prüfung nicht bestanden, so muss ein Folgetermin wahrgenommen werden; im Wiederholungsfall führt dies zum Nichtbestehen des Praktikums. Im Erfolgsfall wird ein "Laborarbeitsblatt" mit weiteren Aufgaben unter Aufsicht (und ggf. mit Hilfestellung) bearbeitet.

## Mindeststandard

Erfolgreiche Teilnahme an allen Laborterminen, d.h. insbesondere selbstständige (ggf. mit Hilfestellung) Lösung der Programmieraufgaben.