

## Programmierung

<b>Modul:</b> Programmierung	
<b>Studiengang:</b> Bibliotheksinformatik	<b>Abschluss:</b> Master of Science
<b>Modulverantwortliche/r:</b> Janett Mohnke & Frank Seeliger	

<b>Semester:</b> 1	<b>Dauer:</b> 2	
<b>Präsenzstunden:</b> 50.0	<b>davon V/Ü/L/P:</b> 25.0/0.0/25.0/0.0	<b>CP nach ECTS:</b> 8.0
<b>Art der Lehrveranstaltung:</b> Pflicht	<b>Sprache:</b> Deutsch	<b>Stand vom:</b> 2018-07-04
<b>Empfohlene Voraussetzungen:</b>		
<b>Pauschale Anrechnung von:</b>		
<b>Besondere Regelungen:</b>		

<b>Aufschlüsselung des Workload</b>	<b>Stunden:</b>
Präsenz:	50.0
Vor- und Nachbereitung:	189.0
Projektarbeit:	0.0
Prüfung:	1.0
<b>Gesamt:</b>	<b>240</b>

## Programmierung

Lernziele	Anteil
Fachkompetenzen	
<p>Kenntnisse/Wissen</p> <ul style="list-style-type: none"> <li>• Die Studierenden kennen die Merkmale und Unterschiede von Programmiersprachen und können dieses Wissen praktisch anwenden.</li> <li>• Sie kennen die wichtigen Elemente einer Programmiersprache, insbesondere der Programmiersprache Java.</li> <li>• Sie kennen die Konzepte der objektorientierten Programmierung und können diese an Beispielen erklären.</li> <li>• Die Studierenden haben ein grundlegendes Verständnis für das Vorgehen beim Entwurf und der Analyse von Algorithmen.</li> <li>• Sie kennen eine Auswahl wichtiger Datenstrukturen und sind in der Lage, diese für typische Problemstellungen einzusetzen.</li> </ul>	50%
<p>Fertigkeiten</p> <ul style="list-style-type: none"> <li>• Die Studierenden können die Methoden und Konzepte der imperativen und der objektorientierten Programmierung praktisch zur Lösung von Problemen anwenden.</li> <li>• Sie beherrschen grundlegende Entwicklungswerkzeuge zum Entwurf und zum Testen von Softwaresystemen in Java.</li> <li>• Insbesondere können sie gegebene Aufgabenstellungen analysieren, mit Hilfe der Programmiersprache Java implementieren und testen.</li> </ul>	45%
Personale Kompetenzen	
<p>Soziale Kompetenz</p> <ul style="list-style-type: none"> <li>• Die Studierenden erwerben bzw. vertiefen die Fähigkeit, sich in kleineren Lerngruppen zu organisieren, um gemeinsam Aufgaben zu bearbeiten.</li> <li>• Sie lernen, sich dabei gegenseitig zu helfen, den Lehrstoff zu verstehen und dieses Wissen auch gemeinsam praktisch zur Lösung fachspezifischer Aufgaben anzuwenden.</li> </ul>	5%
<p>Selbstständigkeit</p> <ul style="list-style-type: none"> <li>• Die Studierenden sind in der Lage, eine gegebene Aufgabenstellung eigenständig zu analysieren, eine Lösung zu konzipieren, diese zu implementieren und zu testen.</li> </ul>	

## Programmierung

### Inhalt:

1. Merkmale von Programmiersprachen, Compiler, Interpreter und virtuelle Maschinen
2. Elemente von Programmiersprachen (Datentypen und Variablen, Kontrollstrukturen, Methoden (Prozeduren und Funktionen)) und Aufbau eines Java-Programmes
3. Recheninterne Darstellung von Zahlen und Zeichen
4. Boolesche Operatoren und deren Logik
5. Klassen und Objekte
6. Grundpfeiler der objektorientierten Programmierung, objektorientierte Analyse und Design mit UML, Entwurfsmuster (Singleton, Iterator)
7. Einführung in den Entwurf von Algorithmen am Beispiel von Sortierverfahren (inkl. Rekursion, Aufwand und Komplexität)
8. Ausgewählte Java-Standardklassen (z.B. ArrayList, String)
9. Ausnahmebehandlung
10. Dokumentation, Erzeugung von ausführbaren jar-Dateien
11. Lineare Datenstrukturen und spezielle Zugriffsformen (Arrays, Listen, FIFO, LIFO)
12. Hashtabellen

### Prüfungsform:

Mündliche Prüfung (100%)

Zusätzliche Regelungen:

Die erfolgreiche Bearbeitung aller Übungsaufgaben ist Zulassungsvoraussetzung für die mündliche Prüfung.

## Programmierung

### Pflichtliteratur:

**Gumm, H. & Sommer, M.** (2013). *Einführung in die Informatik*. München: Oldenbourg.

**Ratz, D.** (2011). *Grundkurs Programmieren in Java*. München [u.a.]: Hanser.

**Mössenböck, H.** (2014). *Sprechen Sie Java?: Eine Einführung in das systematische Programmieren*. dpunkt.verlag GmbH.

### Empfohlene Literatur:

**Hunt, A. & Thomas, D.** (2004). *Unit-Tests mit JUnit*. München: Hanser.

**Cormen, T.** (2007). *Algorithmen*. München [u.a.]: Oldenbourg.

**Rupp, C. & Queins, S.** (2012). *UML 2 glasklar*. München: Hanser.