

# Modulhandbuch

für den Studiengang

1. Staatsprüfung für das Lehramt  
an Mittelschulen Informatik  
(Prüfungsordnungsversion: 20242)

für das Sommersemester 2025

# Inhaltsverzeichnis

Grundlagen der Programmierung (93104).....	3
Didaktik der Informatik 1 (93211).....	5
Sichere Systeme (93105).....	8
Didaktik der Informatik 2 (93224).....	10
Didaktik der Informatik 3 (93225).....	12
Einführung in das Software Engineering (93097).....	14
Einführung in die Algorithmik (93106).....	16
Einführung in Datenbanken (93008).....	18
Parallele und Funktionale Programmierung (93040).....	21
Praktikum Informatik (93089).....	22
Theoretische Informatik für Wirtschaftsinformatik und Lehramt (93201).....	23
Praktikum Maschinenprogrammierung (93085).....	25

1	<b>Modulbezeichnung</b> 93104	<b>Grundlagen der Programmierung</b> Foundations of programming	<b>5 ECTS</b>
2	Lehrveranstaltungen	Im aktuellen Semester werden keine Lehrveranstaltungen zu dem Modul angeboten. Für weitere Auskünfte zum Lehrveranstaltungsangebot kontaktieren Sie bitte die Modul-Verantwortlichen.	
3	Lehrende	-	

4	<b>Modulverantwortliche/r</b>	Dr.-Ing. Vanessa Klein Prof. Dr. Tim Weyrich	
5	<b>Inhalt</b>	<ul style="list-style-type: none"> <li>• Grundlegende Begriffe: Problem, Algorithmus, Programm, Syntax, Semantik, von Neumann Architektur</li> <li>• Imperative Programmkonstrukte: Variablen, Zahlen, Strings, Arrays, Kontrollstrukturen, Methoden</li> <li>• Grundlagen asymptotische Aufwandsanalyse: Einführung O-Notation und einfache Abschätzungen</li> <li>• Robustes Programmieren: Exceptions, Assert, Testen, Verifikation, Debugging</li> <li>• Objektorientierte Programmierung: Klassen, Objekte, Vererbung, Polymorphie, Module</li> <li>• Datenstrukturen: Parametrisierte Typen, abstrakte Datentypen, Listen, dynamische Arrays, binäre Suche, Suchbäume, Hashtabellen</li> </ul>	
6	<b>Lernziele und Kompetenzen</b>	<p><i>Wissen:</i> Die Studierenden ...</p> <ul style="list-style-type: none"> <li>• erlernen die Grundlagen und das Vokabular der Programmierung anhand der Programmiersprache Java</li> </ul> <p><i>Verstehen:</i> Die Studierenden ...</p> <ul style="list-style-type: none"> <li>• können algorithmische Beschreibungen in natürlicher Sprache verstehen</li> <li>• können einfache Algorithmen im Code verstehen und analysieren</li> <li>• verstehen die grundlegende Behälterdatentypen und deren Eigenschaften (insbesondere Laufzeit- und Speicherplatzbedarf ihrer Operationen)</li> </ul> <p><i>Anwenden:</i> Die Studierenden ...</p> <ul style="list-style-type: none"> <li>• implementieren einfache Algorithmen in Java unter Verwendung verschiedener Kontrollstrukturen</li> <li>• strukturieren Java-Code in Paketen, Klassen und Methoden und entwickeln wiederverwendbare Funktionen</li> <li>• können einfache Komplexitätsanalysen erstellen (O-Kalkül)</li> <li>• benutzen verschiedene Möglichkeiten zur Absicherung gegen Fehlersituationen und zur Fehlerrückmeldung (Rückgabewert, Ausnahmebehandlung)</li> <li>• wenden geeignete Testverfahren an</li> <li>• kennen die Konzepte der objektorientierten Programmierung und können diese einsetzen</li> <li>• setzen Verfahren und Werkzeuge zur systematischen Lokalisierung und Behebung von Programmfehlern an</li> </ul>	

		(Debugging) und verbessern ihre Lösungen auf diese Weise iterativ <ul style="list-style-type: none"> <li>• verwenden generische Behälterdatentypen sachgerecht in eigenen Programmen</li> <li>• setzen Lambda-Ausdrücke effektiv ein</li> </ul>
7	<b>Voraussetzungen für die Teilnahme</b>	Keine
8	<b>Einpassung in Studienverlaufsplan</b>	Semester: 1
9	<b>Verwendbarkeit des Moduls</b>	Module Fachwissenschaft Informatik 1. Staatsprüfung für das Lehramt an Mittelschulen Informatik 20242
10	<b>Studien- und Prüfungsleistungen</b>	Klausur (90 Minuten)
11	<b>Berechnung der Modulnote</b>	Klausur (100%)
12	<b>Turnus des Angebots</b>	nur im Wintersemester
13	<b>Arbeitsaufwand in Zeitstunden</b>	Präsenzzeit: 60 h Eigenstudium: 90 h
14	<b>Dauer des Moduls</b>	1 Semester
15	<b>Unterrichts- und Prüfungssprache</b>	Deutsch
16	<b>Literaturhinweise</b>	

1	<b>Modulbezeichnung</b> 93211	<b>Didaktik der Informatik 1</b> Teaching computer science I	<b>5 ECTS</b>
2	Lehrveranstaltungen	Vorlesung mit Übung: Curriculare Themen der Fachdidaktik Informatik (LA RS/LA MS/MEd BPT/LA GYM) (2 SWS, SoSe 2025)	2,5 ECTS
3	Lehrende	Prof. Dr. Marc-Pascal Berges	

4	<b>Modulverantwortliche/r</b>	Prof. Dr. Marc-Pascal Berges	
5	<b>Inhalt</b>	<ul style="list-style-type: none"> <li>• Grundlagen der informatikbezogenen Unterrichtsplanung und -gestaltung</li> <li>• Informatik und Informatikdidaktik im Wissenschaftskontext</li> <li>• Informatische Modellbildung</li> <li>• Programmieren im Informatikunterricht</li> <li>• Werkzeuge für den Informatikunterricht</li> <li>• Unterrichtsmethoden und -techniken</li> <li>• Aufgaben und Aufgabenkultur für einen kompetenzorientierten Informatikunterricht</li> </ul>	
6	<b>Lernziele und Kompetenzen</b>	<p>Nach erfolgreichem Abschluss des Moduls sind die Studierenden in der Lage, Informatikunterricht begründet zu planen, durchzuführen und zu reflektieren. Sie sind in der Lage, Unterrichtsinhalte motivierend, schülernah, verständlich und zielführend zu vermitteln. Sie können Elemente der Informatik in Alltagssituationen zur Motivation und als Modellierungsgrundlage heranziehen, Realsituationen informatisch modellieren, den Prozess des Modellierens schülerbezogen gestalten und Schülerinnen und Schüler beim Modellieren unterstützen.</p> <p>Sie</p> <ul style="list-style-type: none"> <li>• analysieren informatische Unterrichtsgegenstände fachdidaktisch und geben Unterrichtsziele outcomeorientiert an</li> <li>• charakterisieren die Wissenschaft Informatik und ihre Rolle im Bildungskontext (Computer Literacy, Great Principles of Computing, Computational Thinking) und geben eine eigene Definition für Informatik an</li> <li>• geben Ziele des Informatikunterrichts (gemäß Lehrplan Bayern) an und beschreiben beispielhaft Möglichkeiten zur Umsetzung dieser Ziele</li> <li>• geben zu Inhalten des Lehrplans konkrete durch die SuS zu erwerbende Kompetenzen an und gestalten entsprechenden Unterricht</li> <li>• beschreiben die "roten Fäden" in den Lehrplänen für Informatik in Bayern und berücksichtigen diese in der Gestaltung von Unterricht</li> <li>• erläutern den Informationszentrierten Ansatz und seinen Einfluss auf den bayerischen Lehrplan</li> <li>• ordnen Inhalte des Lehrplans dem Gesamtkonzept des Lehrplans zu</li> <li>• beschreiben Informatische Modellbildung, geben Beispiele und Darstellungsformen für Modellierungstechniken an und</li> </ul>	

		<p>begründen die Relevanz informatischen Modellierens für die Schulinformatik</p> <ul style="list-style-type: none"> <li>• erläutern und illustrieren den Modellbegriff und Modellbildungsprozess aus Sicht der Informatik an selbst gewählten Beispielen</li> <li>• wenden Theorie und Begriffe informatischer Modellbildung in der Gestaltung und Bewertung von Unterrichtsszenarien an</li> <li>• ordnen Beispiele und Werkzeuge des Informatikunterrichts den Klassen von Modellen zu (EIS)</li> <li>• diskutieren Stellenwert, Rolle und Ziele des Programmierens in der informatischen Bildung und im informationszentrierten Ansatz</li> <li>• diskutieren den Stellenwert von Modellierung und Programmierung im Informatikunterricht ihrer Schulform</li> <li>• grenzen die Begriffe Modellieren, Programmieren und Codieren voneinander ab</li> <li>• begründen aus historischer und aktueller Perspektive den Einsatz von Methoden und Werkzeugen für die Vermittlung von Programmierkompetenz</li> <li>• diskutieren den Einsatz visueller und textueller Programmiersprachen</li> <li>• wenden Werkzeuge für den Informatikunterricht begründet in der Gestaltung von Unterricht an.</li> <li>• nennen Kriterien für Werkzeuge und wählen Werkzeuge für den Informatikunterricht begründet aus</li> <li>• begründen den Einsatz der Projektmethode im Informatikunterricht und erläutern deren Ziele</li> <li>• ordnen die Projektmethode in Kategorien der Sozial- und Lehr-/Lernformen ein</li> <li>• erstellen ein Szenario für ein Informatikunterrichtsprojekt</li> <li>• vergleichen Wasserfallmodell und Agile Methoden als Grundlage für die Durchführung eines Informatikprojekts</li> <li>• beschreiben agile Techniken und wenden diese in der methodischen Unterrichtsgestaltung an</li> <li>• strukturieren und bewerten Unterrichtsmethoden für den Informatikunterricht</li> <li>• wählen für gegebene Inhalte und Kompetenzen adäquate Unterrichtsmethoden begründet aus</li> <li>• erläutern verschiedene Unterrichtstechniken und -prinzipien anhand von adressierten Problemen, Zielen und Beispielen</li> <li>• nennen Qualitätskriterien für Aufgaben und Leitfragen zur Aufgabenentwicklung und wenden diese in der Analyse und Entwicklung von Aufgaben an</li> <li>• entwickeln Aufgaben hinsichtlich eines kompetenzorientierten Informatikunterrichts unter verschiedenen Gesichtspunkten (z.B. Öffnen von Aufgaben, Kontextorientierung, Kreativität) (weiter) und ordnen diese den GI-Bildungsstandards zu</li> </ul>
7	<b>Voraussetzungen für die Teilnahme</b>	Keine

8	<b>Einpassung in Studienverlaufsplan</b>	Semester: 3
9	<b>Verwendbarkeit des Moduls</b>	Module Fachdidaktik Informatik 1. Staatsprüfung für das Lehramt an Mittelschulen Informatik 20242
10	<b>Studien- und Prüfungsleistungen</b>	Portfolio Im Rahmen des Moduls erstellen die Studierenden ein Portfolio im Umfang von 40-100 Seiten. Sie wenden dabei die Methode des reflexiven Schreibens an.
11	<b>Berechnung der Modulnote</b>	Portfolio (100%)
12	<b>Turnus des Angebots</b>	nur im Wintersemester
13	<b>Arbeitsaufwand in Zeitstunden</b>	Präsenzzeit: 60 h Eigenstudium: 90 h
14	<b>Dauer des Moduls</b>	2 Semester
15	<b>Unterrichts- und Prüfungssprache</b>	Deutsch
16	<b>Literaturhinweise</b>	<ul style="list-style-type: none"> <li>• Hubwieser, Peter. Didaktik der Informatik. Springer-Verlag, 2007.</li> <li>• Schubert, Sigrid, and Andreas Schwill. Didaktik der Informatik. Spektrum Akademischer Verlag, 2011.</li> <li>• Werner Hartmann, Michael Näf, and Raimond Reichert. Informatikunterricht planen und durchführen. Springer, 2007.</li> <li>• Meyer, Hilbert. Leitfaden Unterrichtsvorbereitung. Cornelsen Scriptor, 2007.</li> </ul>

1	<b>Modulbezeichnung</b> 93105	<b>Sichere Systeme</b> Secure Systems	<b>5 ECTS</b>
2	Lehrveranstaltungen	Im aktuellen Semester werden keine Lehrveranstaltungen zu dem Modul angeboten. Für weitere Auskünfte zum Lehrveranstaltungsangebot kontaktieren Sie bitte die Modul-Verantwortlichen.	
3	Lehrende	-	

4	<b>Modulverantwortliche/r</b>	Prof. Dr.-Ing. Felix Freiling	
5	<b>Inhalt</b>	<p>Die Vorlesung gibt einen einführenden Überblick über Konzepte und Methoden der IT-Sicherheit. Themen (unter anderem):</p> <ul style="list-style-type: none"> <li>• Angreifer und Schutzziele</li> <li>• Cyberkriminalität und Strafbarkeit</li> <li>• Ethik und Privatsphäre</li> <li>• grundlegende Muster von Unsicherheit in technischen Systemen</li> <li>• grundlegende Sicherheitsmechanismen</li> <li>• Techniken der Sicherheitsanalyse</li> <li>• ausgewählte Beispiele aus dem Bereich der Kryptographie und Internetsicherheit (Web-Security)</li> </ul> <p>In der Übung werden die Themen der Veranstaltung beispielhaft eingeübt. Themen (unter anderem):</p> <ul style="list-style-type: none"> <li>• Kryptanalyse und Angreifbarkeit kryptographischer Protokolle</li> <li>• Schutzziele und Strafbarkeit</li> <li>• Zertifikate und Public-Key-Infrastrukturen</li> <li>• Web-Security</li> <li>• anonyme Kommunikation</li> <li>• formale Sicherheitsanalyse</li> <li>• Sicherheitstesten</li> </ul>	
6	<b>Lernziele und Kompetenzen</b>	Die Teilnehmenden erwerben einen Überblick über Konzepte und Methoden aus dem Bereich der IT-Sicherheit und können diese im Kontext der Informatik und der Lebenswirklichkeit anhand von Beispielen einordnen und erläutern. Die Studierenden können die Schwächen in Internetprotokollen erkennen und benennen. Sie können außerdem erläutern, wie man diese Schwachstellen ausnutzt und welche technischen und organisatorischen Maßnahmen geeignet sind, diese Schwachstellen zu vermeiden. Die Studierenden lernen, die Wirksamkeit von IT-Sicherheitsmechanismen im gesellschaftlichen Kontext und in Kenntnis professioneller Strukturen der Cyberkriminalität aus technischen, ethischen und rechtlichen Perspektiven zu bewerten.	
7	<b>Voraussetzungen für die Teilnahme</b>	Keine	
8	<b>Einpassung in Studienverlaufsplan</b>	Semester: 1	
9	<b>Verwendbarkeit des Moduls</b>	Module Fachwissenschaft Informatik 1. Staatsprüfung für das Lehramt an Mittelschulen Informatik 20242	

10	<b>Studien- und Prüfungsleistungen</b>	Klausur (90 Minuten)
11	<b>Berechnung der Modulnote</b>	Klausur (100%)
12	<b>Turnus des Angebots</b>	nur im Wintersemester
13	<b>Arbeitsaufwand in Zeitstunden</b>	Präsenzzeit: 60 h Eigenstudium: 90 h
14	<b>Dauer des Moduls</b>	1 Semester
15	<b>Unterrichts- und Prüfungssprache</b>	Deutsch
16	<b>Literaturhinweise</b>	<ul style="list-style-type: none"> <li>• Dieter Gollmann: Computer Security. 3. Auflage, Wiley, 2010.</li> <li>• Joachim Biskup: Security in Computing Systems. Springer, 2008.</li> </ul> <p>Weitere Literatur wird in der Vorlesung bekanntgegeben.</p>

1	<b>Modulbezeichnung</b> 93224	<b>Didaktik der Informatik 2</b> Teaching computer science II	<b>5 ECTS</b>
2	Lehrveranstaltungen	Im aktuellen Semester werden keine Lehrveranstaltungen zu dem Modul angeboten. Für weitere Auskünfte zum Lehrveranstaltungsangebot kontaktieren Sie bitte die Modul-Verantwortlichen.	
3	Lehrende	-	

4	<b>Modulverantwortliche/r</b>	Prof. Dr. Marc-Pascal Berges	
5	<b>Inhalt</b>	<ul style="list-style-type: none"> <li>• Theoretische Fundierung der Didaktik der Informatik</li> <li>• Voraussetzungen und Rahmenbedingungen für Informatikunterricht</li> <li>• Lern- und Kompetenzziele des Informatikunterrichts</li> </ul>	
6	<b>Lernziele und Kompetenzen</b>	<p>Nach erfolgreichem Abschluss des Moduls sind die Studierenden in der Lage, Informatikunterricht zu planen, durchzuführen, zu reflektieren und auf wissenschaftlicher Grundlage weiterzuentwickeln. Sie haben vertiefte fachbezogene Reflexionskompetenzen, informatikdidaktische Basis- und diagnostische Kompetenzen sowie informatikunterrichtsbezogene Handlungskompetenzen erworben. Sie sind in der Lage, Unterrichtsinhalte motivierend, schülernah, verständlich und zielführend zu vermitteln. Hierzu können sie entscheiden, welche Inhalte der Informatik für die Schule relevant sind und diese lerngruppenadäquat aufbereiten. Sie können Elemente der Informatik in Alltagssituationen zur Motivation und als Modellierungsgrundlage heranziehen und den Beitrag des Faches zur Allgemeinbildung beschreiben und transportieren. Sie können Realsituationen informatisch modellieren, den Prozess des Modellierens schülerbezogen gestalten und Schülerinnen und Schüler beim Modellieren unterstützen.</p>	
7	<b>Voraussetzungen für die Teilnahme</b>	Es wird eine vorherige erfolgreiche Teilnahme am Modul "Didaktik der Informatik 1" (93211) wird empfohlen	
8	<b>Einpassung in Studienverlaufsplan</b>	Semester: 5	
9	<b>Verwendbarkeit des Moduls</b>	Module Fachdidaktik Informatik 1. Staatsprüfung für das Lehramt an Mittelschulen Informatik 20242	
10	<b>Studien- und Prüfungsleistungen</b>	Portfolio Im Rahmen des Moduls erstellen die Studierenden ein Portfolio im Umfang von 40-100 Seiten. Sie wenden dabei die Methode des reflexiven Schreibens an.	
11	<b>Berechnung der Modulnote</b>	Portfolio (100%)	
12	<b>Turnus des Angebots</b>	nur im Wintersemester	
13	<b>Arbeitsaufwand in Zeitstunden</b>	Präsenzzeit: 90 h Eigenstudium: 60 h	
14	<b>Dauer des Moduls</b>	1 Semester	

15	<b>Unterrichts- und Prüfungssprache</b>	Deutsch
16	<b>Literaturhinweise</b>	<ul style="list-style-type: none"><li>• Hubwieser, Peter. Didaktik der Informatik. Springer-Verlag, 2007.</li><li>• Schubert, Sigrid, and Andreas Schwill. Didaktik der Informatik. Spektrum Akademischer Verlag, 2011.</li></ul>

1	<b>Modulbezeichnung</b> 93225	<b>Didaktik der Informatik 3</b> Teaching computer science 3	<b>5 ECTS</b>
2	Lehrveranstaltungen	Vorlesung mit Übung: Extracurriculare Themen der Didaktik der Informatik (2 SWS, SoSe 2025) Praktikum: Praktikum Informatik in der Bildung (4 SWS, SoSe 2025)	2,5 ECTS 5 ECTS
3	Lehrende	Prof. Dr. Marc-Pascal Berges	

4	<b>Modulverantwortliche/r</b>	Prof. Dr. Marc-Pascal Berges	
5	<b>Inhalt</b>	<ul style="list-style-type: none"> <li>• Didaktik der Informatik als Wissenschaft</li> <li>• Gamebased Learning im Informatikunterricht</li> <li>• Informatikwettbewerbe</li> <li>• Digitalisierung</li> <li>• Lehr-/Lernlabore</li> <li>• Physical Computing</li> </ul>	
6	<b>Lernziele und Kompetenzen</b>	<p>Nach Abschluss des Moduls sind die Studierenden in der Lage, fachdidaktische Problemstellungen auch außerhalb der Schulcurricula zu reflektieren. Sie sind in der Lage, extracurriculare Inhalte zu planen und durchzuführen und geeignete Anknüpfungspunkte an die curriculare Lehre zu identifizieren.</p> <p>Sie</p> <ul style="list-style-type: none"> <li>• erkennen die Fachdidaktik Informatik als eigenständige Wissenschaft und können aktuelle Fragestellungen nachvollziehen</li> <li>• erstellen Unterrichtsentwürfe nach der Theorie des gamebased learning</li> <li>• konzipieren und erzeugen Unterrichtsmaterialien (z.B. auf Basis von physical computing)</li> <li>• kennen die gängigsten Wettbewerbe für Informatik</li> <li>• beschreiben die Rolle der Informatikdidaktik im Rahmen der Digitalisierung</li> <li>• wenden die Methode des explorativen Lernens auf Unterrichtsmaterialien und im Rahmen von Lehr-/Lernlaboren an.</li> </ul>	
7	<b>Voraussetzungen für die Teilnahme</b>	Es wird dringend empfohlen, die Module "Didaktik der Informatik 1" (93211) und "Didaktik der Informatik 2" (93224) vorab erfolgreich abgeschlossen zu haben.	
8	<b>Einpassung in Studienverlaufsplan</b>	Semester: 8	
9	<b>Verwendbarkeit des Moduls</b>	Module Fachdidaktik Informatik 1. Staatsprüfung für das Lehramt an Mittelschulen Informatik 20242	
10	<b>Studien- und Prüfungsleistungen</b>	Portfolio Im Rahmen des Moduls erstellen die Studierenden ein Portfolio im Umfang von 40-100 Seiten. Sie wenden dabei die Methode des reflexiven Schreibens an.	

11	<b>Berechnung der Modulnote</b>	Portfolio (100%)
12	<b>Turnus des Angebots</b>	nur im Sommersemester
13	<b>Arbeitsaufwand in Zeitstunden</b>	Präsenzzeit: 90 h Eigenstudium: 60 h
14	<b>Dauer des Moduls</b>	2 Semester
15	<b>Unterrichts- und Prüfungssprache</b>	Deutsch
16	<b>Literaturhinweise</b>	<ul style="list-style-type: none"> <li>• Hubwieser, Peter. Didaktik der Informatik. Springer-Verlag, 2007.</li> <li>• Schubert, Sigrid, and Andreas Schwill. Didaktik der Informatik. Spektrum Akademischer Verlag, 2011.</li> </ul>

1	<b>Modulbezeichnung</b> 93097	<b>Einführung in das Software Engineering</b> Introduction to software engineering	<b>5 ECTS</b>
2	Lehrveranstaltungen	Im aktuellen Semester werden keine Lehrveranstaltungen zu dem Modul angeboten. Für weitere Auskünfte zum Lehrveranstaltungsangebot kontaktieren Sie bitte die Modul-Verantwortlichen.	
3	Lehrende	-	

4	<b>Modulverantwortliche/r</b>	Prof. Dr.-Ing. Andreas Maier	
5	<b>Inhalt</b>	<ul style="list-style-type: none"> <li>• Einführung in die einzelnen Phasen der Softwareentwicklung: Anforderungsanalyse, Spezifikation, Entwurf, Implementierung, Test, Wartung</li> <li>• Prozessmodelle</li> <li>• Agile Softwareentwicklung</li> <li>• Anforderungsanalyse und -verwaltung</li> <li>• Modellierung von Systemen (u.a. mit UML)</li> <li>• Software-Architekturen und Designmuster</li> <li>• Teststrategien</li> <li>• Umgang mit Software-Alterung</li> <li>• Projektmanagement</li> <li>• Software-Engineering im Bereich Machine Learning</li> <li>• Refactoring zur Unterstützung der Wartungsphase</li> </ul>	
6	<b>Lernziele und Kompetenzen</b>	<p>Die Studierenden</p> <ul style="list-style-type: none"> <li>• Beschreiben Prozessmodelle und unterscheiden plangesteuerte (wie das Wasserfall- und V-Modell) und agile Prozessmodelle (wie XP, Scrum, RUP und Kanban)</li> <li>• Erläutern verschiedene Techniken der Anforderungsanalyse und –Ermittlung (wie Endliche Zustandsautomaten, Petri-Netze, Use Cases, User Stories) und wenden diese für plangesteuerte und agile Prozesse an</li> <li>• Stellen die Unterschiede zwischen agilem und plangesteuertem Requirements-Engineering dar</li> <li>• Verstehen und erläutern UML-Diagramme (wie Use Case-, Klassen-, Sequenz- und Kommunikationsdiagramme) und wenden diese auf praktische Beispiele der Objektorientierung an</li> <li>• Reproduzieren allgemeine Entwurfslösungen wiederkehrender Probleme des Software-Engineerings und wenden diese an</li> <li>• Wenden funktionale und strukturelle Testansätze an</li> <li>• Erklären Methoden zur Änderung und Weiterentwicklung von Software</li> <li>• Beschreiben Ansätze für das Projekt-Management von Softwareprojekten</li> <li>• Erläutern wie Methoden des Maschinellen Lernens für Software-Engineering eingesetzt werden können</li> </ul>	
7	<b>Voraussetzungen für die Teilnahme</b>	Keine	

8	<b>Einpassung in Studienverlaufsplan</b>	Semester: 3
9	<b>Verwendbarkeit des Moduls</b>	Module Fachwissenschaft Informatik 1. Staatsprüfung für das Lehramt an Mittelschulen Informatik 20242
10	<b>Studien- und Prüfungsleistungen</b>	Klausur (90 Minuten)
11	<b>Berechnung der Modulnote</b>	Klausur (100%)
12	<b>Turnus des Angebots</b>	in jedem Semester
13	<b>Arbeitsaufwand in Zeitstunden</b>	Präsenzzeit: 60 h Eigenstudium: 90 h
14	<b>Dauer des Moduls</b>	1 Semester
15	<b>Unterrichts- und Prüfungssprache</b>	Deutsch
16	<b>Literaturhinweise</b>	<ul style="list-style-type: none"> <li>• Software Engineering, Ian Sommerville, 10. Auflage, 2016</li> <li>• Software-Engineering Kompakt, Anja Metzner, 2020</li> <li>• Handbook of Software Engineering, Sungdeok Cha, Richard N. Taylor, Kyochul Kang (Hrsg.), 2019</li> </ul>

1	<b>Modulbezeichnung</b> 93106	<b>Einführung in die Algorithmik</b> Introduction to algorithms	<b>7,5 ECTS</b>
2	Lehrveranstaltungen	Vorlesung: Einführung in die Algorithmik - Vorlesung (4 SWS) Übung: Einführung in die Algorithmik - Übung (2 SWS)	- -
3	Lehrende	Christian Riess Julian Thomas	

4	<b>Modulverantwortliche/r</b>	Carina Harrius Christian Riess	
5	<b>Inhalt</b>	<p>Die Vorlesung "Einführung in die Algorithmik" gibt eine fundierte Einführung in die Gebiete der Algorithmen und Datenstrukturen. Diese Einführung umfasst grundlegende Designkonzepte von Algorithmen und deren formale Analyse. Folgende Themen werden behandelt:</p> <ul style="list-style-type: none"> <li>• Grundlagen</li> <li>• Design und Analyse von Algorithmen Korrektheit von Algorithmen</li> <li>• Wachstumsfunktionen</li> <li>• Rekurrenz</li> <li>• Probabilistische Algorithmen und deren Analyse</li> <li>• Grundlegende Algorithmen und Datenstrukturen und deren formale Analyse</li> <li>• Datenstrukturen Sortierverfahren Graphalgorithmen</li> <li>• Ausgewählte Themen</li> <li>• Algorithmen in der Zahlentheorie String matching</li> <li>• Matrix Operationen</li> </ul>	
6	<b>Lernziele und Kompetenzen</b>	<p>Die Studierenden erwerben eine grundlegende Einführung in die Konzepte und Methoden aus dem Bereich der Algorithmen und Datenstrukturen. Die Teilnehmer kennen grundlegende Techniken und Prinzipien zum Design von Algorithmen und Datenstrukturen. Die Studierenden kennen grundlegende Algorithmen im Bereich der Sortierung, der Graphentheorie und der Zahlentheorie. Des Weiteren kennen die Studierenden die notwendigen Datenstrukturen und verstehen deren Vor- und Nachteile in Bezug auf deren Effizienz und Komplexität. Die Studierenden können die unterschiedlichen Designparadigmen von Datenstrukturen und Algorithmen auf neue Probleme anwenden und deren Korrektheit formal analysieren. Aus der Analyse können die Studierenden Algorithmen bewerten und vergleichen.</p>	
7	<b>Voraussetzungen für die Teilnahme</b>	Keine	
8	<b>Einpassung in Studienverlaufsplan</b>	Semester: 2	
9	<b>Verwendbarkeit des Moduls</b>	Module Fachwissenschaft Informatik 1. Staatsprüfung für das Lehramt an Mittelschulen Informatik 20242	

10	<b>Studien- und Prüfungsleistungen</b>	<p>Übungsleistung Klausur (90 Minuten)</p> <p>Die Vorlesung wird durch Übungen begleitet. Die Aufgaben von fünf Übungsblättern werden bewertet, diese können in Zweiergruppen bearbeitet werden. Für die erforderliche unbenotete Prüfungsleistung sind insgesamt 50% der Punkte dieser Übungsblätter zu erreichen.</p>
11	<b>Berechnung der Modulnote</b>	<p>Übungsleistung (bestanden/nicht bestanden) Klausur (100%)</p> <p>Die Modulnote wird durch die Abschlussklausur bestimmt.</p>
12	<b>Turnus des Angebots</b>	nur im Sommersemester
13	<b>Arbeitsaufwand in Zeitstunden</b>	<p>Präsenzzeit: 90 h Eigenstudium: 135 h</p>
14	<b>Dauer des Moduls</b>	1 Semester
15	<b>Unterrichts- und Prüfungssprache</b>	Deutsch
16	<b>Literaturhinweise</b>	<ul style="list-style-type: none"> <li>• Introduction to Algorithms, Thomas H. Cormen , Charles E. Leiserson, Ronald L. Rivest, Clifford Stein</li> <li>• Weitere Literatur wird ggfs. nach Bedarf in der Vorlesung bekannt gegeben.</li> </ul>

1	<b>Modulbezeichnung</b> 93008	<b>Einführung in Datenbanken</b> Introduction to databases	<b>7,5 ECTS</b>
2	Lehrveranstaltungen	Übung: Übung zu Einführung in Datenbanken - PG1: Fr 16 (3 SWS)	2,5 ECTS
		Übung: Übung zu Einführung in Datenbanken - PG2: Do 18 (3 SWS)	2,5 ECTS
		Übung: Übung zu Einführung in Datenbanken - PG3: Di 08 (3 SWS)	2,5 ECTS
		Übung: Übung zu Einführung in Datenbanken - PG4: Do 16 (3 SWS)	2,5 ECTS
		Übung: Übung zu Einführung in Datenbanken - PG5: Mo 18 (3 SWS)	2,5 ECTS
		Übung: Übung zu Einführung in Datenbanken - PG6: Do 08 (3 SWS)	2,5 ECTS
		Übung: Übung zu Einführung in Datenbanken - PG7: Di 18 (3 SWS)	2,5 ECTS
		Übung: Übung zu Einführung in Datenbanken - PG8: Mi 14 (3 SWS)	2,5 ECTS
		Vorlesung: Einführung in Datenbanken (3 SWS)	5 ECTS
3	Lehrende	Felix Hanika David Haller Tobias Bittner Alexander Seifert Fabian Nitschke Joshua Orendt Prof. Dr.-Ing. Richard Lenz	

4	<b>Modulverantwortliche/r</b>	Prof. Dr.-Ing. Richard Lenz	
5	<b>Inhalt</b>	<p>Ziel des Moduls ist die Vermittlung von Kenntnissen zur systematischen und bedarfsorientierten Erstellung konzeptioneller Datenbankschemata sowie die relationale Datenbanksprache SQL. Darüber hinaus werden Grundkenntnisse zur Funktionsweise und zur Implementierung von Datenbankmanagementsystemen vermittelt, im Einzelnen:</p> <ul style="list-style-type: none"> <li>• Grundbegriffe von Datenbanken</li> <li>• Entity-Relationship Modell und erweitertes E/R-Modell</li> <li>• UML Klassendiagramme</li> <li>• Das Relationale Datenmodell</li> <li>• Systematische Abbildung von ER-Diagrammen auf Relationale Datenbankschemata</li> <li>• Normalisierung</li> <li>• Relationale Algebra</li> <li>• SQL</li> <li>• Multidimensionale Modellierung und Data Warehousing</li> <li>• Schichtenmodell zur Implementierung von Datenbanksystemen</li> </ul>	

		<ul style="list-style-type: none"> <li>• Pufferverwaltung</li> <li>• Indexstrukturen (B-Bäume, B+-Bäume)</li> <li>• Anfrageverarbeitung</li> <li>• Transaktionen</li> <li>• Synchronisation</li> <li>• Recovery</li> <li>• Andere Datenmodelle, No-SQL Systeme</li> <li>• Ontologien, Semantic Web, RDF, SPARQL</li> </ul>
6	<b>Lernziele und Kompetenzen</b>	<p>Die Studierenden</p> <ul style="list-style-type: none"> <li>• Können die zentralen Begriffe aus der Datenbankfachliteratur definieren</li> <li>• Erstellen ER-Diagramme und erweiterte ER Diagramme</li> <li>• Können ER-Diagramme systematisch in geeignete relationale Datenbankschemata überführen</li> <li>• Definieren die Normalformen 1NF, 2NF, 3NF, BCNF und 4NF</li> <li>• Können ein nicht normalisiertes Relationenschema in 3NF überführen</li> <li>• Erstellen Anfragen auf der Basis der Relationalen Algebra</li> <li>• Erstellen Datenbankschemata mit Hilfe der SQL DDL</li> <li>• Erstellen Datenbankanfragen mit SQL</li> <li>• Erstellen multidimensionale ER-Diagramme und bilden diese auf Star- oder Snowflake-Schemata ab</li> <li>• Erklären die Funktionsweise von Datenbankpuffern</li> <li>• Erklären die Funktionsweise von Indexstrukturen</li> <li>• Erklären die Grundlagen der Anfrageoptimierung</li> <li>• Erläutern und bewerten die Funktionsweise verschiedener Join-Algorithmen</li> <li>• Erklären die ACID Eigenschaften von Transaktionen</li> <li>• Erklären die Funktionsweise des Zwei-Phasen-Freigabe-Protokolls</li> <li>• Erläutern die Funktionsweise des Zwei-Phasen-Sperr-Protokolls</li> <li>• Vergleichen die verschiedenen Klassen von Wiederherstellungs-Algorithmen</li> <li>• Erläutern die grundlegende Funktionsweise der Protokoll-basierten Wiederherstellung</li> <li>• Beschreiben und vergleichen verschiedene Datenmodelle</li> </ul>
7	<b>Voraussetzungen für die Teilnahme</b>	Keine
8	<b>Einpassung in Studienverlaufsplan</b>	Semester: 4
9	<b>Verwendbarkeit des Moduls</b>	<p>Informatik 1. Staatsprüfung für das Lehramt an Mittelschulen Informatik 20242</p> <p>Module Fachwissenschaft Informatik 1. Staatsprüfung für das Lehramt an Mittelschulen Informatik 20242</p>
10	<b>Studien- und Prüfungsleistungen</b>	Klausur mit MultipleChoice (90 Minuten)

11	<b>Berechnung der Modulnote</b>	Klausur mit MultipleChoice (100%)
12	<b>Turnus des Angebots</b>	nur im Sommersemester
13	<b>Arbeitsaufwand in Zeitstunden</b>	Präsenzzeit: 90 h Eigenstudium: 135 h
14	<b>Dauer des Moduls</b>	1 Semester
15	<b>Unterrichts- und Prüfungssprache</b>	Deutsch
16	<b>Literaturhinweise</b>	

1	<b>Modulbezeichnung</b> 93040	<b>Parallele und Funktionale Programmierung</b> Parallel and functional programming	<b>5 ECTS</b>
2	Lehrveranstaltungen	Übung: Intensivübungen zu Parallele und Funktionale Programmierung (2 SWS)	0 ECTS
3	Lehrende	David Schwarzbeck Dr.-Ing. Norbert Oster Prof. Dr. Michael Philippsen	

4	<b>Modulverantwortliche/r</b>	Dr.-Ing. Norbert Oster Prof. Dr. Michael Philippsen	
5	<b>Inhalt</b>	<ul style="list-style-type: none"> <li>• Grundlagen der funktionale Programmierung</li> <li>• Grundlagen der parallelen Programmierung</li> <li>• Datenstrukturen</li> <li>• Objektorientierung</li> <li>• Scala-Kenntnisse</li> <li>• Erweiterte JAVA-Kenntnisse</li> <li>• Aufwandsabschätzungen</li> <li>• Grundlegende Algorithmen</li> </ul>	
6	<b>Lernziele und Kompetenzen</b>	Die Studierenden <ul style="list-style-type: none"> <li>• erlernen die Grundlagen der funktionalen Programmierung anhand der Programmiersprache Scala</li> <li>• verstehen paralleles Programmieren mit Java</li> <li>• kennen fundamentale Datenstrukturen und Algorithmen</li> <li>• können funktionale und parallele Algorithmen entwickeln und analysieren</li> </ul>	
7	<b>Voraussetzungen für die Teilnahme</b>	Keine	
8	<b>Einpassung in Studienverlaufsplan</b>	Semester: 3	
9	<b>Verwendbarkeit des Moduls</b>	Module Fachwissenschaft Informatik 1. Staatsprüfung für das Lehramt an Mittelschulen Informatik 20242	
10	<b>Studien- und Prüfungsleistungen</b>	Klausur (60 Minuten)	
11	<b>Berechnung der Modulnote</b>	Klausur (100%)	
12	<b>Turnus des Angebots</b>	nur im Wintersemester	
13	<b>Arbeitsaufwand in Zeitstunden</b>	Präsenzzeit: 60 h Eigenstudium: 90 h	
14	<b>Dauer des Moduls</b>	1 Semester	
15	<b>Unterrichts- und Prüfungssprache</b>	Deutsch	
16	<b>Literaturhinweise</b>		

1	<b>Modulbezeichnung</b> 93089	<b>Praktikum Informatik</b> Secure Systems	<b>6 ECTS</b>
2	Lehrveranstaltungen	Im aktuellen Semester werden keine Lehrveranstaltungen zu dem Modul angeboten. Für weitere Auskünfte zum Lehrveranstaltungsangebot kontaktieren Sie bitte die Modul-Verantwortlichen.	
3	Lehrende	-	

4	<b>Modulverantwortliche/r</b>	Prof. Dr. Marc-Pascal Berges	
5	<b>Inhalt</b>	<ul style="list-style-type: none"> <li>• Agile Entwicklung eines Projektes/Programms für den Einsatz im Mittelschul-Unterricht</li> <li>• Organisation eines mittleren Projektes</li> <li>• Zeitmanagement</li> <li>• Kommunikation in der Gruppe</li> </ul>	
6	<b>Lernziele und Kompetenzen</b>	Die Studierenden <ul style="list-style-type: none"> <li>• entwickeln in Gruppen ein lauffähiges Programm für den Einsatz im Mittelschul-Unterricht</li> <li>• verwalten ihren Projektfortschritt mit Hilfe eines Projektboards</li> <li>• lernen agile Entwicklung und deren Einsatzmöglichkeit im Unterricht kennen</li> </ul>	
7	<b>Voraussetzungen für die Teilnahme</b>	Es wird dringen empfohlen, die Module "Einführung in die Programmierung", "Einführung in das Software-Engineering" erfolgreich abgeschlossen zu haben.	
8	<b>Einpassung in Studienverlaufsplan</b>	Semester: 5	
9	<b>Verwendbarkeit des Moduls</b>	Module Fachwissenschaft Informatik 1. Staatsprüfung für das Lehramt an Mittelschulen Informatik 20242	
10	<b>Studien- und Prüfungsleistungen</b>	Praktikumsleistung Prüfungsleistung: Entwicklung und Vorstellung eines Softwareprojektes in Gruppen. Die Vorstellung erfolgt in einem 20-30minütigem Vortrag. Die Entwicklung wird durch einen 30-60 seitigen Praktikumsbericht dokumentiert.	
11	<b>Berechnung der Modulnote</b>	Praktikumsleistung (100%)	
12	<b>Turnus des Angebots</b>	nur im Wintersemester	
13	<b>Arbeitsaufwand in Zeitstunden</b>	Präsenzzeit: 90 h Eigenstudium: 90 h	
14	<b>Dauer des Moduls</b>	1 Semester	
15	<b>Unterrichts- und Prüfungssprache</b>	Deutsch	
16	<b>Literaturhinweise</b>	<ul style="list-style-type: none"> <li>• Best Practice Software-Engineering, A.Schatten, Spektrum Verlag Software-Qualität, Dirk W. Hoffmann, Springer Verlag</li> </ul>	

1	<b>Modulbezeichnung</b> 93201	<b>Theoretische Informatik für Wirtschaftsinformatik und Lehramt</b> Theoretical computer science for information systems and teaching degree students	<b>5 ECTS</b>
2	Lehrveranstaltungen	Übung: ÜThInfWI-A (2 SWS) Vorlesung: Theoretische Informatik für Wirtschaftsinformatik und Lehramt (2 SWS) Übung: Intensivübung zu Theoretische Informatik für Wirtschaftsinformatik und Lehramt (optional) (2 SWS)	- - -
3	Lehrende	Simon Prucker Max Ole Elliger apl. Prof. Dr. Stefan Milius	

4	<b>Modulverantwortliche/r</b>	apl. Prof. Dr. Stefan Milius	
5	<b>Inhalt</b>	<p>Grundlegende Begriffe und Kernergebnisse der Automatentheorie, Berechenbarkeitstheorie und Komplexitätstheorie werden überblickhaft behandelt:</p> <ul style="list-style-type: none"> <li>• endliche Automaten und reguläre Grammatiken und Sprachen</li> <li>• Kellerautomaten, kontextfreie Grammatiken und Sprachen</li> <li>• Turingmaschinen und berechenbare Funktionen</li> <li>• Primitiv rekursive und mü-rekursive Funktionen</li> <li>• LOOP- und WHILE-Berechenbarkeit</li> <li>• Entscheidbare Sprachen und Unentscheidbarkeit</li> <li>• Chomsky-Hierarchie</li> <li>• Komplexitätsklassen P und NP</li> <li>• NP-Vollständigkeit</li> </ul>	
6	<b>Lernziele und Kompetenzen</b>	<p><b>Fachkompetenz</b> Wissen Die Studierenden geben elementare Definitionen und Fakten zu formalen Sprachen und entsprechenden Maschinenmodellen und Grammatiken wieder.</p> <p><b>Verstehen</b> Die Studierenden</p> <ul style="list-style-type: none"> <li>• erklären grundlegende Konzepte der Begriffe der Automaten- und Berechenbarkeits- und Komplexitätstheorie.</li> <li>• beschreiben Beispiele dieser Konzepte.</li> <li>• erläutern grundlegende Konstruktionen, Algorithmen und wesentliche Resultate und entsprechende Beweise (z.B. Unentscheidbarkeit des Halteproblems).</li> </ul> <p><b>Anwenden</b> Die Studierenden</p> <ul style="list-style-type: none"> <li>• führen Konstruktionen auf vorgelegten Maschinen und Grammatiken und Sprachen durch (z.B. Automatenminimierung, Potenzmengen-Konstruktion, Chomsky-Normierung, CYK-Algorithmus).</li> <li>• wenden grundlegende Beweisverfahren der theoretischen Informatik an (z.B. Induktionsbeweise, Pumping-Lemma, Reduktionen).</li> </ul>	

		<p>Analysieren</p> <p>Die Studierenden</p> <ul style="list-style-type: none"> <li>• analysieren formale Sprachen und ermitteln ihre Zugehörigkeit zu den Klassen der Chomsky-Hierarchie.</li> <li>• untersuchen die Entscheidbarkeit von vorgelegten formalen Sprachen.</li> <li>• analysieren die Komplexität eines Entscheidungsproblems und klassifizieren es als Problem in P, NP bzw. NP-vollständig.</li> </ul> <p>Lern- bzw. Methodenkompetenz</p> <p>Die Studierenden</p> <ul style="list-style-type: none"> <li>• beherrschen das grundsätzliche Konzept des Beweises als hauptsächliche Methode des Erkenntnisgewinns in der theoretischen Informatik. Sie überblicken abstrakte Begriffsarchitekturen.</li> <li>• vollziehen mathematische Argumentationen nach, erklären diese, führen diese selbst und legen sie schriftlich nieder.</li> </ul> <p>Sozialkompetenz</p> <p>Die Studierenden lösen Probleme in kollaborativer Gruppenarbeit und präsentieren erarbeitete Lösungen.</p>
7	<b>Voraussetzungen für die Teilnahme</b>	Keine
8	<b>Einpassung in Studienverlaufsplan</b>	Semester: 6
9	<b>Verwendbarkeit des Moduls</b>	Module Fachwissenschaft Informatik 1. Staatsprüfung für das Lehramt an Mittelschulen Informatik 20242
10	<b>Studien- und Prüfungsleistungen</b>	Klausur (90 Minuten)
11	<b>Berechnung der Modulnote</b>	Klausur (100%)
12	<b>Turnus des Angebots</b>	nur im Sommersemester
13	<b>Arbeitsaufwand in Zeitstunden</b>	Präsenzzeit: 56 h Eigenstudium: 94 h
14	<b>Dauer des Moduls</b>	1 Semester
15	<b>Unterrichts- und Prüfungssprache</b>	Deutsch
16	<b>Literaturhinweise</b>	<ul style="list-style-type: none"> <li>• U. Schöning: Theoretische Informatik - kurz gefasst, 5. Aufl., Spektrum 2008.</li> <li>• J.E. Hopcroft, R. Motwani und J.D. Ullman: Introduction to Automata Theory, Languages, and Computation, 2. Aufl., Addison Wesley, 2001.</li> </ul>

1	<b>Modulbezeichnung</b> 93085	<b>Praktikum Maschinenprogrammierung</b> Machine code lab	<b>5 ECTS</b>
2	Lehrveranstaltungen	Im aktuellen Semester werden keine Lehrveranstaltungen zu dem Modul angeboten. Für weitere Auskünfte zum Lehrveranstaltungsangebot kontaktieren Sie bitte die Modul-Verantwortlichen.	
3	Lehrende	-	

4	<b>Modulverantwortliche/r</b>	Prof. Dr. Marc-Pascal Berges	
5	<b>Inhalt</b>	<p>Aufbau und Prinzip von Rechnern, Daten und ihre Codierung, Boolesche Algebra und Schaltalgebra, Schaltnetze (Symbole, Darstellung), Optimierung von Schaltnetzen (Minimierung Boolescher Funktionen), Realisierungsformen von Schaltnetzen (ROM, PLA, FPGA), Automaten und Schaltwerke (Moore/Mealy, Zustandskodierung und -minimierung), Flipflops, Register, Zähler, Speicher (RAM, ROM), Taktung und Synchronisation, Realisierungsformen von Schaltwerken, Realisierung der Grundrechenarten Addition/Subtraktion, Multiplikation und Division, Gleitkommazahlen (Darstellung, Fehler, Rundung, Standards, Einheiten), Steuerwerksentwurf, Spezialeinheiten und Co-Prozessoren, Mikrocontroller; vorlesungsbegleitende Einführung und Beschreibung der Schaltungen mit VHDL.</p>	
6	<b>Lernziele und Kompetenzen</b>	<ul style="list-style-type: none"> <li>• Die Studierenden veranschaulichen fundierte theoretische und praxisorientierte Grundlagen der Informationstheorie, Rechnerarithmetik, Digitaltechnik und des Schaltungsentwurfs.</li> <li>• Die Studierenden führen den Entwurf, die Synthese und das Testen von digitalen Schaltungen auf programmierbarer Hardware (FPGAs) durch.</li> <li>• Die Studierenden verstehen, dass Hardware heutzutage mit Software am Rechner entwickelt und simuliert wird.</li> <li>• Die Studierenden verstehen den Schaltungsentwurf mittels einer Beschreibungssprache (VHDL).</li> <li>• Die Studierenden erarbeiten und diskutieren verschiedene Lösungswege für die Datencodierung sowie den Entwurf und die Optimierung von digitalen Hardwareschaltungen.</li> <li>• Die Studierenden erlernen die Fähigkeit, digitale Schaltungen und Systeme eigenständig zu konzipieren und zu implementieren.</li> <li>• Die Studierenden reflektieren den Umgang mit schulgeeigneten Mikrocontrollern</li> </ul>	
7	<b>Voraussetzungen für die Teilnahme</b>	Keine	
8	<b>Einpassung in Studienverlaufsplan</b>	Semester: 7	
9	<b>Verwendbarkeit des Moduls</b>	Module Fachwissenschaft Informatik 1. Staatsprüfung für das Lehramt an Mittelschulen Informatik 20242	

10	<b>Studien- und Prüfungsleistungen</b>	schriftlich/mündlich Die Prüfungsleistung besteht aus 8-10 praktischen Aufgaben sowie 3-4 schriftlichen Testaten
11	<b>Berechnung der Modulnote</b>	schriftlich/mündlich (100%)
12	<b>Turnus des Angebots</b>	nur im Wintersemester
13	<b>Arbeitsaufwand in Zeitstunden</b>	Präsenzzeit: 60 h Eigenstudium: 90 h
14	<b>Dauer des Moduls</b>	1 Semester
15	<b>Unterrichts- und Prüfungssprache</b>	Deutsch
16	<b>Literaturhinweise</b>	