



Universität Stuttgart

Modulhandbuch
Studiengang Künstlerisches Lehramt (GymPO I) Informatik
Prüfungsordnung: 2010

Universität Stuttgart
Keplerstr. 7
70174 Stuttgart

Inhaltsverzeichnis

Präambel	3
200 Pflichtmodule	4
10280 Programmierung und Software-Entwicklung	5
10940 Theoretische Grundlagen der Informatik	7
3000 Zwischenprüfung	9
12060 Datenstrukturen und Algorithmen	10
10280 Programmierung und Software-Entwicklung	12
10940 Theoretische Grundlagen der Informatik	14



Präambel

nicht verfügbar

200 Pflichtmodule

Zugeordnete Module: 10280 Programmierung und Software-Entwicklung
 10940 Theoretische Grundlagen der Informatik

Modul: 10280 Programmierung und Software-Entwicklung

2. Modulkürzel:	051520005	5. Moduldauer:	1 Semester
3. Leistungspunkte:	9.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	6.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Bernhard Mitschang		
9. Dozenten:	Bernhard Mitschang		
10. Zuordnung zum Curriculum:	B.Sc. Informatik, 1. Semester → Basismodule BA (Komb) Informatik, 1. Semester → Module im Nebenfach		
11. Voraussetzungen:	Keine. Teilnahme an einem Mathematik Vorkurs wird empfohlen.		
12. Lernziele:	Die Teilnehmer haben die wichtigsten Konzepte einer höheren Programmiersprache und ihrer Verwendung verstanden und sind in der Lage, kleine Programme (bis zu einigen hundert Zeilen) zu analysieren und selbst zu konzipieren und zu implementieren. Sie kennen die Möglichkeiten, Daten- und Ablaufstrukturen zu entwerfen, zu beschreiben und zu codieren. Sie haben die Abstraktionskonzepte moderner Programmiersprachen verstanden. Sie kennen die Techniken und Notationen zur Definition kontextfreier Programmiersprachen und können damit arbeiten.		
13. Inhalt:	• Algorithmen und funktionale Programmierung Kap. 1 verwendet nur die funktionale Teilmenge der Programmiersprache Ada, keine Variablen, keine Prozeduren. Grammatik, Formale Sprachen und BNF werden eingeführt. • Imperative Programmierung Kap. 2 erweitert die verwendete Sprache durch die prozeduralen Konzepte, also Variablen und Prozeduren. Zu den Sprachkonstrukten werden Vor- und Nachbedingungen, mit den Schleifen die Invarianten eingeführt. Datentypen werden schrittweise ausgebaut. In Zusammenhang mit den Zeigern werden die Konzepte für Keller und Halde vermittelt. Die Entwicklung einfacher Programme wird gezeigt und geübt. • Aufbau und Organisation komplexer Programme. Die Modularisierung, die bei größeren Programmen notwendig ist, führt zur Kapselung und zu den abstrakten Datentypen. Damit entsteht die Möglichkeit, neue Datenstrukturen und Datentypen sicher zu definieren. Die Konzepte der Kompilation und der Interpretation werden erläutert. Wichtige Beispiele komplexer Datentypen werden entwickelt. Die Konzepte der Generalisierung (generische Einheiten) werden vermittelt. • Ausnahmebehandlung Möglichkeiten und Probleme der Ausnahmebehandlung sind Gegenstände dieses kurzen Kapitels. • Objektorientierte Programmierung Am Ende des Semesters steht ein Ausblick in die objektorientierte Programmierung, d.h. die Umsetzung der bereits bekannten Konzepte (ADTs) in die objektorientierte Sichtweise und die Vererbung. Dieser Teil bereitet die Programmierung in einer objektorientierten Sprache (3. Semester) vor.		
14. Literatur:	Manuskripte: V.Claus (WS 08/09 bis SS 2009)		

Appelrath, Hans-Jürgen und Ludewig, Jochen, "Skriptum Informatik - eine konventionelle Einführung", Verlag der Fachvereine Zürich und B.G. Teubner Stuttgart, 4. Auflage 1999

Nagl., M., "Softwaretechnik mit Ada 95. Entwicklung großer Systeme.", Vieweg-Verlag, Wiesbaden 1999

Barnes, J.G.P., "Programming in Ada 95", 2. Auflage, Addison-Wesley 1998

15. Lehrveranstaltungen und -formen: • 102801 Vorlesung Programmierung und Softwareentwicklung
• 102802 Übung Programmierung und Softwareentwicklung

16. Abschaetzung Arbeitsaufwand: Präsenzzeit: 63 Stunden
Vor-/Nachbearbeitungszeit: 187 Stunden

Prüfungsvorbereitung: 20 Stunden

17a. Studienleistung: Studienleistung: Übungsschein, Vor. 3 mal vortragen in den

Übungen und mindestens 50% der Übungspunkte erwerben,

Teilnahme an den Zwischenklausuren.

17b. Prüfungsleistungen: Modulprüfung: Schriftliche Prüfung von 120 Minuten Dauer

18. Grundlage für ... :

19. Medienform:

20. Prüfungsnummer/n und -name: 10281 Programmierung und Software-Entwicklung

21. Angeboten von:

22. Zuordnung zu weiteren Curricula:

- B.Sc. Softwaretechnik, 1. Semester
→ Basismodule
- B.Sc. Wirtschaftsinformatik, 1. Semester
→ Informatik (B 1)
- B.Sc. Mechatronik, 1. Semester
→ Kernmodule
- B.Sc. Maschinelle Sprachverarbeitung, 1. Semester
→ Basismodule
- B.Sc. Technikpädagogik, 1. Semester
→ Hauptfach Informatik
→ Basismodule Informatik
- B.Sc. Technikpädagogik, 1. Semester
→ Wahlpflichtfach
→ Informatik
- M.Sc. Technikpädagogik, 1. Semester
→ Wahlpflichtfach B
→ Affines Wahlpflichtfach Informatik
→ Affines Wahlpflichtfach Informatik Basismodule
- B.Sc. Simulation Technology, 1. Semester
→ Grundstudium
- ohne Absch Lehramt-Pool, 1. Semester
→ Pflichtmodule

Modul: 10940 Theoretische Grundlagen der Informatik

2. Modulkürzel:	050420005	5. Moduldauer:	2 Semester
3. Leistungspunkte:	12.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	8.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Volker Diekert		
9. Dozenten:	• Ulrich Hertrampf • Volker Diekert		
10. Zuordnung zum Curriculum:	B.Sc. Informatik, 1. Semester → Basismodule		
11. Voraussetzungen:	Keine		
12. Lernziele:	• Logik und Diskrete Strukturen: Die Studierenden haben die grundsätzlichen Kenntnisse in Logik und Diskreter Mathematik erworben, wie sie in den weiteren Grundvorlesungen der Informatik in verschiedenen Bereichen benötigt werden. • Automaten und Formale Sprachen: Die Studierenden beherrschen wichtige theoretische Grundlagen der Informatik, insbesondere die Theorie und Algorithmik endlicher Automaten. Hierzu gehört das Kennenlernen, Einordnung und Trennung der Chomskyschen Sprachklassen.		
13. Inhalt:	• Logik und Diskrete Strukturen: Einführung in die Aussagenlogik; formale Sprache; Semantik (Wahrheitswerte); Syntax (Axiome und Schlussregeln); Normalformen; Hornformeln; aussagenlogische Resolution; Korrektheit und Vollständigkeit für die Aussagenlogik; Einführung in die Prädikatenlogik 1. Stufe; formale Sprache; Semantik und Syntax; Normalformen; Herbrand-Theorie; prädikatenlogische Resolution; Kombinatorik, Graphen, elementare Zahlentheorie: Rechnen mit Restklassen, endliche Körper, RSA-Verfahren. • Automaten und Formale Sprachen: Deterministische- bzw. nichtdeterministische endliche Automaten, reguläre Ausdrücke, Minimierung endlicher Automaten, Iterationslemmata für reguläre und kontextfreie Sprachen, Normalformen, Kellerautomaten, Lösen des Wortproblems kontextfreier Sprachen mit dem CYK-Algorithmus, linear beschränkte Automaten, kontextsensitive Grammatiken, Typ 0-Grammatiken und Turingmaschinen.		
14. Literatur:	• John Hopcroft, Jeffrey Ullman, Einführung in die Automatentheorie, formale Sprachen und Komplexitätstheorie, 1988 • Uwe Schöningh, Theoretische Informatik - kurzgefasst, 1999		
15. Lehrveranstaltungen und -formen:	• 109401 Vorlesung Logik und Diskrete Strukturen • 109402 Übung Logik und Diskrete Strukturen • 109403 Vorlesung Automaten und Formale Sprachen • 109404 Übung Automaten und Formale Sprachen		
16. Abschaetzung Arbeitsaufwand:	Präsenzzeit: 84 Stunden		

	Nachbearbeitungszeit: 276 Stunden
17a. Studienleistung:	Prüfungsvorleistung: Übungsschein.
17b. Prüfungsleistungen:	Schriftliche Prüfung von 120 Minuten Dauer.
18. Grundlage für ... :	
19. Medienform:	
20. Prüfungsnummer/n und -name:	10941 Theoretische Grundlagen der Informatik
21. Angeboten von:	
22. Zuordnung zu weiteren Curricula:	B.Sc. Softwaretechnik, 1. Semester → Basismodule B.Sc. Maschinelle Sprachverarbeitung, 1. Semester → Basismodule B.Sc. Technikpädagogik, 1. Semester → Hauptfach Informatik → Kernmodule Informatik B.Sc. Technikpädagogik, 1. Semester → Wahlpflichtfach → Informatik M.Sc. Technikpädagogik, 1. Semester → Wahlpflichtfach B → Affines Wahlpflichtfach Informatik → Affines Wahlpflichtfach Informatik Kernmodule ohne Absch Lehramt-Pool, 1. Semester → Pflichtmodule

3000 Zwischenprüfung

Zugeordnete Module: 12060 Datenstrukturen und Algorithmen
 10280 Programmierung und Software-Entwicklung
 10940 Theoretische Grundlagen der Informatik

Modul: 12060 Datenstrukturen und Algorithmen

2. Modulkürzel:	051510005	5. Moduldauer:	1 Semester
3. Leistungspunkte:	9.0 LP	6. Turnus:	jedes 2. Semester, SoSe
4. SWS:	6.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Erhard Plödereder		
9. Dozenten:	Stefan Funke		
10. Zuordnung zum Curriculum:	B.Sc. Informatik, 2. Semester → Basismodule BA (Komb) Informatik, 2. Semester → Module im Nebenfach		
11. Voraussetzungen:	• Modul 051520005 Programmierung und Software-Entwicklung		
12. Lernziele:	Die Studierenden kennen nach engagierter Mitarbeit in dieser Veranstaltung diverse zentrale Algorithmen auf geeigneten Datenstrukturen, die für eine effiziente Nutzung von Computern unverzichtbar sind. Sie können am Ende zu gängigen Problemen geeignete programmiersprachliche Lösungen angeben und diese in einer konkreten Programmiersprache formulieren. Konkret: • Kenntnis der Eigenschaften elementarer und häufig benötigter Algorithmen • Verständnis für die Auswirkungen theoretischer und tatsächlicher Komplexität • Erweiterung der Kompetenz im Entwurf und Verstehen von Algorithmen und der zugehörigen Datenstrukturen • Erste Begegnung mit nebenläufigen Algorithmen; sowohl „originär“ parallel, als auch parallelisierte Versionen bereits vorgestellter sequentieller Algorithmen		
13. Inhalt:	• Vorgehensweise bei der Entwicklung und Implementierung von Algorithmen • Komplexität und Effizienz von Algorithmen, O-Notation • Wahl der Datenstrukturen; Listen, Bäume, Graphen; deren Definitionen, deren Datenstrukturen • diverse interne und externe Such- und Sortierverfahren (z.B. Linear-, Binär-, Interpolationssuche, AVL-, B-Bäume, internes und externes Hashing, mehrere langsame Sortierungen, Heap-, Quick-, Bucket-, Mergesort) • diverse Graphenalgorithmen (DFS, BFS, Besuchssequenzen, topol. Traversierung, Zusammenhangskomponenten, minimale Spannbäume, Dijkstra-, Floyd- kürzeste Wege) • Algorithmen auf Mengen und Relationen (transitive Hüllen, Warshall) • Korrektheitsbegriff und -formalismen; Spezifikation und Implementierung • Einige parallele und parallelisierte Algorithmen • einfache Elemente paralleler Programmierung, soweit für obiges notwendig		
14. Literatur:	• Appelrath H.J., Ludewig. J., Skriptum Informatik, 1999 • Sedgewick, R., Algorithms in C, 1998		
15. Lehrveranstaltungen und -formen:	• 120601 Vorlesung Datenstrukturen und Algorithmen		

• 120602 Übung Datenstrukturen und Algorithmen

16. Abschaetzung Arbeitsaufwand:	Präsenzzeit: 63 Stunden Nachbearbeitungszeit: 207 Stunden
17a. Studienleistung:	Prüfungsvorleistung: Übungsschein.
17b. Prüfungsleistungen:	Schriftliche Prüfung von 120 Minuten Dauer. Die genauen Details der Übungsleistungen und Ihrer Anrechnung werden zu Beginn der Lehrveranstaltung bekannt gegeben.
18. Grundlage für ... :	
19. Medienform:	
20. Prüfungsnummer/n und -name:	12061 Datenstrukturen und Algorithmen
21. Angeboten von:	
22. Zuordnung zu weiteren Curricula:	B.Sc. Mathematik, 2. Semester → Nebenfach → Nebenfach Informatik B.Sc. Softwaretechnik, 2. Semester → Basismodule B.Sc. Wirtschaftsinformatik, 2. Semester → Informatik (B 1) B.Sc. Mechatronik, 2. Semester → Kernmodule B.Sc. Maschinelle Sprachverarbeitung, 2. Semester → Basismodule B.Sc. Technikpädagogik, 2. Semester → Hauptfach Informatik → Basismodule Informatik B.Sc. Technikpädagogik, 2. Semester → Wahlpflichtfach → Informatik M.Sc. Technikpädagogik, 2. Semester → Wahlpflichtfach B → Affines Wahlpflichtfach Informatik → Affines Wahlpflichtfach Informatik Basismodule

Modul: 10280 Programmierung und Software-Entwicklung

2. Modulkürzel:	051520005	5. Moduldauer:	1 Semester
3. Leistungspunkte:	9.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	6.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Bernhard Mitschang		
9. Dozenten:	Bernhard Mitschang		
10. Zuordnung zum Curriculum:	B.Sc. Informatik, 1. Semester → Basismodule BA (Komb) Informatik, 1. Semester → Module im Nebenfach		
11. Voraussetzungen:	Keine. Teilnahme an einem Mathematik Vorkurs wird empfohlen.		
12. Lernziele:	Die Teilnehmer haben die wichtigsten Konzepte einer höheren Programmiersprache und ihrer Verwendung verstanden und sind in der Lage, kleine Programme (bis zu einigen hundert Zeilen) zu analysieren und selbst zu konzipieren und zu implementieren. Sie kennen die Möglichkeiten, Daten- und Ablaufstrukturen zu entwerfen, zu beschreiben und zu codieren. Sie haben die Abstraktionskonzepte moderner Programmiersprachen verstanden. Sie kennen die Techniken und Notationen zur Definition kontextfreier Programmiersprachen und können damit arbeiten.		
13. Inhalt:	• Algorithmen und funktionale Programmierung Kap. 1 verwendet nur die funktionale Teilmenge der Programmiersprache Ada, keine Variablen, keine Prozeduren. Grammatik, Formale Sprachen und BNF werden eingeführt. • Imperative Programmierung Kap. 2 erweitert die verwendete Sprache durch die prozeduralen Konzepte, also Variablen und Prozeduren. Zu den Sprachkonstrukten werden Vor- und Nachbedingungen, mit den Schleifen die Invarianten eingeführt. Datentypen werden schrittweise ausgebaut. In Zusammenhang mit den Zeigern werden die Konzepte für Keller und Halde vermittelt. Die Entwicklung einfacher Programme wird gezeigt und geübt. • Aufbau und Organisation komplexer Programme. Die Modularisierung, die bei größeren Programmen notwendig ist, führt zur Kapselung und zu den abstrakten Datentypen. Damit entsteht die Möglichkeit, neue Datenstrukturen und Datentypen sicher zu definieren. Die Konzepte der Kompilation und der Interpretation werden erläutert. Wichtige Beispiele komplexer Datentypen werden entwickelt. Die Konzepte der Generalisierung (generische Einheiten) werden vermittelt. • Ausnahmebehandlung Möglichkeiten und Probleme der Ausnahmebehandlung sind Gegenstände dieses kurzen Kapitels. • Objektorientierte Programmierung Am Ende des Semesters steht ein Ausblick in die objektorientierte Programmierung, d.h. die Umsetzung der bereits bekannten Konzepte (ADTs) in die objektorientierte Sichtweise und die Vererbung. Dieser Teil bereitet die Programmierung in einer objektorientierten Sprache (3. Semester) vor.		
14. Literatur:	Manuskripte: V.Claus (WS 08/09 bis SS 2009)		

Appelrath, Hans-Jürgen und Ludewig, Jochen, "Skriptum Informatik - eine konventionelle Einführung", Verlag der Fachvereine Zürich und B.G. Teubner Stuttgart, 4. Auflage 1999

Nagl., M., "Softwaretechnik mit Ada 95. Entwicklung großer Systeme.", Vieweg-Verlag, Wiesbaden 1999

Barnes, J.G.P., "Programming in Ada 95", 2. Auflage, Addison-Wesley 1998

15. Lehrveranstaltungen und -formen:	• 102801 Vorlesung Programmierung und Softwareentwicklung • 102802 Übung Programmierung und Softwareentwicklung
16. Abschaetzung Arbeitsaufwand:	Präsenzzeit: 63 Stunden Vor-/Nachbearbeitungszeit: 187 Stunden Prüfungsvorbereitung: 20 Stunden
17a. Studienleistung:	Studienleistung: Übungsschein, Vor. 3 mal vortragen in den Übungen und mindestens 50% der Übungspunkte erwerben, Teilnahme an den Zwischenklausuren.
17b. Prüfungsleistungen:	Modulprüfung: Schriftliche Prüfung von 120 Minuten Dauer
18. Grundlage für ... :	
19. Medienform:	
20. Prüfungsnummer/n und -name:	10281 Programmierung und Software-Entwicklung
21. Angeboten von:	
22. Zuordnung zu weiteren Curricula:	B.Sc. Softwaretechnik, 1. Semester → Basismodule B.Sc. Wirtschaftsinformatik, 1. Semester → Informatik (B 1) B.Sc. Mechatronik, 1. Semester → Kernmodule B.Sc. Maschinelle Sprachverarbeitung, 1. Semester → Basismodule B.Sc. Technikpädagogik, 1. Semester → Hauptfach Informatik → Basismodule Informatik B.Sc. Technikpädagogik, 1. Semester → Wahlpflichtfach → Informatik M.Sc. Technikpädagogik, 1. Semester → Wahlpflichtfach B → Affines Wahlpflichtfach Informatik → Affines Wahlpflichtfach Informatik Basismodule B.Sc. Simulation Technology, 1. Semester → Grundstudium ohne Absch Lehramt-Pool, 1. Semester → Pflichtmodule

Modul: 10940 Theoretische Grundlagen der Informatik

2. Modulkürzel:	050420005	5. Moduldauer:	2 Semester
3. Leistungspunkte:	12.0 LP	6. Turnus:	jedes 2. Semester, WiSe
4. SWS:	8.0	7. Sprache:	Deutsch
8. Modulverantwortlicher:	Volker Diekert		
9. Dozenten:	• Ulrich Hertrampf • Volker Diekert		
10. Zuordnung zum Curriculum:	B.Sc. Informatik, 1. Semester → Basismodule		
11. Voraussetzungen:	Keine		
12. Lernziele:	• Logik und Diskrete Strukturen: Die Studierenden haben die grundsätzlichen Kenntnisse in Logik und Diskreter Mathematik erworben, wie sie in den weiteren Grundvorlesungen der Informatik in verschiedenen Bereichen benötigt werden. • Automaten und Formale Sprachen: Die Studierenden beherrschen wichtige theoretische Grundlagen der Informatik, insbesondere die Theorie und Algorithmik endlicher Automaten. Hierzu gehört das Kennenlernen, Einordnung und Trennung der Chomskyschen Sprachklassen.		
13. Inhalt:	• Logik und Diskrete Strukturen: Einführung in die Aussagenlogik; formale Sprache; Semantik (Wahrheitswerte); Syntax (Axiome und Schlussregeln); Normalformen; Hornformeln; aussagenlogische Resolution; Korrektheit und Vollständigkeit für die Aussagenlogik; Einführung in die Prädikatenlogik 1. Stufe; formale Sprache; Semantik und Syntax; Normalformen; Herbrand-Theorie; prädikatenlogische Resolution; Kombinatorik, Graphen, elementare Zahlentheorie: Rechnen mit Restklassen, endliche Körper, RSA-Verfahren. • Automaten und Formale Sprachen: Deterministische- bzw. nichtdeterministische endliche Automaten, reguläre Ausdrücke, Minimierung endlicher Automaten, Iterationslemmata für reguläre und kontextfreie Sprachen, Normalformen, Kellerautomaten, Lösen des Wortproblems kontextfreier Sprachen mit dem CYK-Algorithmus, linear beschränkte Automaten, kontextsensitive Grammatiken, Typ 0-Grammatiken und Turingmaschinen.		
14. Literatur:	• John Hopcroft, Jeffrey Ullman, Einführung in die Automatentheorie, formale Sprachen und Komplexitätstheorie, 1988 • Uwe Schöningh, Theoretische Informatik - kurzgefasst, 1999		
15. Lehrveranstaltungen und -formen:	• 109401 Vorlesung Logik und Diskrete Strukturen • 109402 Übung Logik und Diskrete Strukturen • 109403 Vorlesung Automaten und Formale Sprachen • 109404 Übung Automaten und Formale Sprachen		
16. Abschaetzung Arbeitsaufwand:	Präsenzzeit: 84 Stunden		

	Nachbearbeitungszeit: 276 Stunden
17a. Studienleistung:	Prüfungsvorleistung: Übungsschein.
17b. Prüfungsleistungen:	Schriftliche Prüfung von 120 Minuten Dauer.
18. Grundlage für ... :	
19. Medienform:	
20. Prüfungsnummer/n und -name:	10941 Theoretische Grundlagen der Informatik
21. Angeboten von:	
22. Zuordnung zu weiteren Curricula:	- B.Sc. Softwaretechnik, 1. Semester → Basismodule - B.Sc. Maschinelle Sprachverarbeitung, 1. Semester → Basismodule - B.Sc. Technikpädagogik, 1. Semester → Hauptfach Informatik → Kernmodule Informatik - B.Sc. Technikpädagogik, 1. Semester → Wahlpflichtfach → Informatik - M.Sc. Technikpädagogik, 1. Semester → Wahlpflichtfach B → Affines Wahlpflichtfach Informatik → Affines Wahlpflichtfach Informatik Kernmodule - ohne Absch Lehramt-Pool, 1. Semester → Pflichtmodule
