

Qualitative Inhaltsanalyse (nach Mayring) von 129 Modulbeschreibungen 35 deutscher Universitäten und (Fach-)Hochschulen

Stand: 07.04.2020

Inhaltsverzeichnis

1. Übersichten	5
1.1 Dokumentenliste	5
1.2 Überblick nicht-kognitiver Kompetenzen	6
1.3 Überblick kognitiver Kompetenzen.....	6
2. nicht operationalisiert	9
3. nicht-kognitive Kompetenzen	40
3.1. Sozial-kommunikative Fähigkeiten	40
3.1.1. Kooperieren und kollaborieren.....	40
3.1.2. Kommunizieren.....	42
3.1.3. Organisation von Teamarbeit und Projekten (Projektmanagement)	44
3.2. Programmiererfahrung sammeln.....	44
3.3. Sensibilisierung für IT-Sicherheit (affektiv)	46
4. Wissensdimensionen	47
4.1. Faktenwissen.....	47
4.2. Konzeptionelles Wissen	47
4.3. Prozedurales Wissen	51
4.4. Meta-kognitives Wissen.....	131
5. kognitive Prozessdimensionen	135
5.1. Erinnern.....	135
5.1.1. Literale zu Datentypen zuordnen.....	135
5.1.2. Begriffe & Kategorien zu Komplexität & Effizienz von Algorithmen kennen.....	135
5.1.3. Merkmale von Algorithmen kennen.....	136
5.1.4. Funktionsbibliotheken für Algorithmen kennen.....	136
5.1.5. Elemente von GUIs kennen.....	136
5.1.6. Grundprinzipien und Methoden von Programmiersprachen und –paradigmen kennen.....	137
5.1.7. Kenntnis grundlegender Algorithmen & Datenstrukturen	143
5.1.8. Methoden der Softwareentwicklung kennen	147
5.1.9. Konzepte zur Modellierung von Algorithmen und Prozessen kennen	149
5.1.10. Entwurfsmöglichkeiten von Algorithmen & Datenstrukturen kennen.....	150
5.1.11. Aufbau und Funktionsweise des Computers kennen	151
5.1.12. Implementierungsmöglichkeiten von Datentypen & -strukturen kennen	152
5.1.13. Prozesskenntnis der Laufzeitanalyse	152
5.1.14. Werkzeuge zur Softwareentwicklung, deren Funktionen und Anwendungszwecke kennen (IDEs, Debugger, Profiler).....	153
5.1.15. Qualitätskriterien für Quellcode nennen.....	153
5.1.16. Mathematische Grundlagen von Algorithmen kennen	154
5.1.17. Konzepte zur Datenverwaltung kennen	154

5.1.18. Begriff der Kontrollstrukturen kennen.....	154
5.1.19. Aufbau und Funktionsweise von Netzen kennen	155
5.1.20. Funktionsweise des Compilers kennen.....	155
5.1.21. Verteilte Systeme & paralleles Programmieren kennen.....	155
5.1.22. Techniken zur formalen Definition von Programmiersprachen kennen	155
5.2. Verstehen	155
5.2.1. Konzepte von Programmierparadigmen (und –sprachen) beschreiben.....	155
5.2.2. Algorithmische Probleme und Aufgabenstellungen beschreiben	157
5.2.3. Begriffe formaler Verifikationstechniken erläutern.....	157
5.2.4. Algorithmen, Datenstrukturen und Datentypen charakterisieren	157
5.2.5. Softwarearchitektur & deren Entwicklung beschreiben.....	159
5.2.6. Notwendigkeit von Werkzeugen zur Softwareentwicklung begründen	159
5.3. Anwenden	159
5.3.1. Werkzeuge zur Software-Entwicklung nutzen	159
5.3.2. Mathematische Berechnungen und Codierungen durchführen.....	163
5.3.3. Bestehende Bibliotheken einsetzen.....	164
5.3.4. Programme professionell dokumentieren.....	164
5.3.5. Qualitätskriterien und Programmierkonventionen auf Quellcode anwenden.....	165
5.3.6. Computer bedienen	165
5.3.7. LaTeX benutzen	165
5.3.8. UNIX-Systeme bedienen mit Hilfe der Kommandozeile	166
5.4. Analysieren.....	166
5.4.1. Programmiersprachen & -paradigmen differenzieren	166
5.4.2. Komplexität von Algorithmen analysieren.....	167
5.4.3. Problemstellung zerlegen	169
5.4.4. Algorithmen auf Eigenschaften und Eignung hin analysieren	172
5.4.5. (Fremde) Programme beschreiben können.....	173
5.4.6. Datenstrukturen auf Eigenschaften und Eignung hin analysieren.....	174
5.4.7. Formale Sprachen durch Chomsky-Hierarchie differenzieren.....	175
5.4.8. Grenzen der Berechnungskraft von Rechnermodellen aufzeigen	175
5.5. Bewerten	175
5.5.1. Angemessenheit von Algorithmen, Datenstrukturen und Datentypen beurteilen	175
5.5.2. Testen von Algorithmen und Programmen auf Fehler	180
5.5.3. Komplexität von Algorithmen beurteilen	182
5.5.4. Eigenschaften von Algorithmen und Programmen beweisen	184
5.5.5. Angemessenheit programmiersprachlicher Lösungen beurteilen.....	185
5.5.6. Beurteilen von Programmierwerkzeugen & Templates	185
5.5.7. Angemessenheit des Selbstkonzepts bewerten	186
5.6. Erzeugen.....	186
5.6.1. Programmiersprachliche, lauffähige Lösungen für Probleme schreiben.....	186

5.6.2. Algorithmen mit Hilfe von Entwurfsmustern problemadäquat entwerfen	196
5.6.3. Algorithmen implementieren	201
5.6.4. Modellieren von Problemen und Programmen	205
5.6.5. kleinere Programme strukturiert entwerfen	208
5.6.6. Standard-Algorithmen anwenden	211
5.6.7. Standard-Datenstrukturen anwenden	212
5.6.8. Datenstrukturen entwerfen	214
5.6.9. Datenstrukturen anpassen	216
5.6.10. Algorithmen anpassen	217
5.6.11. (Abstrakte) Datentypen konzipieren & einsetzen	218
5.6.12. Gegebene Programm(-komponenten) anpassen & verändern	218
5.6.13. Formale Syntaxbeschreibungen entwerfen	219
5.6.14. Programm-Spezifikation entwerfen	220
5.6.15. größere Anwendungen/Applikationen entwickeln	220
5.6.16. Formale Werkzeuge & Testfälle entwickeln	220
5.6.17. GUIs programmieren	221
5.6.18. nebenläufige und parallele Prozesse entwerfen	221
5.6.19. Bibliotheken entwickeln	221
5.6.20. Methoden aufrufen	221
5.6.21. Objekte erzeugen	222
5.6.22. Schnittstellen entwerfen	222
5.6.23. Transfer von Wissen und Erfahrung auf neue Aufgaben und Programmiersprachen	222
5.6.24. Selbstständige Organisation des Lernprozesses (unter Anwendung externer Ressourcen) ..	224
5.6.25. Abstraktion von Problemen	225

1. Übersichten

1.1 Dokumentenliste¹

Nr.	Dokument
1	Modulbeschreibungen\MH01
2	Modulbeschreibungen\MH02
3	Modulbeschreibungen\MH03
4	Modulbeschreibungen\MH04
5	Modulbeschreibungen\MH05
6	Modulbeschreibungen\MH06
7	Modulbeschreibungen\MH07
8	Modulbeschreibungen\MH08
9	Modulbeschreibungen\MH09
10	Modulbeschreibungen\MH10
11	Modulbeschreibungen\MH11
12	Modulbeschreibungen\MH12
13	Modulbeschreibungen\MH13
14	Modulbeschreibungen\MH14
15	Modulbeschreibungen\MH15
16	Modulbeschreibungen\MH16
17	Modulbeschreibungen\MH17
18	Modulbeschreibungen\MH18
19	Modulbeschreibungen\MH19
20	Modulbeschreibungen\MH20
21	Modulbeschreibungen\MH21
22	Modulbeschreibungen\MH22
23	Modulbeschreibungen\MH23
24	Modulbeschreibungen\MH24
25	Modulbeschreibungen\MH25

¹ Liste der aus Studienordnungen und Modullisten extrahierten Modulbeschreibungen zum definierten Inhaltsbereiche (Grundlagenausbildung Programmierung, 1.-3. (bzw. 4.) Fachsemester, B. Sc. Informatik)

26	Modulbeschreibungen\MH26
27	Modulbeschreibungen\MH27
28	Modulbeschreibungen\MH28
29	Modulbeschreibungen\MH29
30	Modulbeschreibungen\MH30
31	Modulbeschreibungen\MH31
32	Modulbeschreibungen\MH32
33	Modulbeschreibungen\MH33
34	Modulbeschreibungen\MH34
35	Modulbeschreibungen\MH35

1.2 Überblick nicht-kognitiver Kompetenzen

Sozial-kommunikative Fähigkeiten (insgesamt 37)²

- Kooperieren und kollaborieren (je 22)
- Kommunizieren (je 13)
- Organisation von Teamarbeit und Projekten (Projektmanagement) (je 2)

Programmiererfahrung sammeln (10)

Sensibilisierung für IT-Sicherheit (affektiv) (1)

1.3 Überblick kognitiver Kompetenzen

<i>Kognitive Prozess- dimen- sion</i>	<i>Wissens-Dimensionen</i>			
	<i>Fakten- wissen</i>	<i>Konzeptionelles Wissen</i>	<i>Prozedurales Wissen</i>	<i>Meta- kognitives Wissen</i>
<i>Stufe 1 Erinnern</i>	Literale zu Daten- typen zuordnen (1)	Begriffe & Kategorien zu Komplexität & Effizienz von Algorithmen kennen (7) Merkmale von Algorithmen kennen (3) Funktionsbibliothe- ken für	Grundprinzipien und Methoden von Programmiersprachen und -paradigmen kennen (50) Kenntnis grundlegender Algorithmen & Datenstrukturen (39) Methoden der Softwareentwicklung kennen (12) Konzepte zur Modellierung von Algorithmen und Prozessen kennen (9) Entwurfsmöglichkeiten von Algorithmen & Datenstrukturen kennen (8)	

² In den Klammern werden die Häufigkeiten im Material angezeigt.

		<p>Algorithmen kennen (3)</p> <p>Elemente von GUIs kennen (1)</p>	<p>Aufbau und Funktionsweise des Computers kennen (5)</p> <p>Implementierungsmöglichkeiten von Datentypen &-strukturen kennen (4)</p> <p>Prozesskenntnis der Laufzeitanalyse (4)</p> <p>Werkzeuge zur Softwareentwicklung, deren Funktionen und Anwendungszwecke kennen (IDEs, Debugger, Profiler) (3)</p> <p>Qualitätskriterien für Quellcode kennen (3)</p> <p>Mathematische Grundlagen von Algorithmen kennen (2)</p> <p>Konzepte zur Datenverwaltung kennen (2)</p> <p>Begriff der Kontrollstrukturen kennen (2)</p> <p>Aufbau und Funktionsweise von Netzen kennen (1)</p> <p>Funktionsweise des Compilers kennen (1)</p> <p>Verteilte Systeme & paralleles Programmieren kennen (1)</p> <p>Techniken zur formalen Definition von Programmiersprachen kennen (1)</p>	
<i>Stufe 2 Verstehen</i>		<p>Konzepte von Programmierparadigmen (und -sprachen) beschreiben (11)</p> <p>Algorithmische Probleme und Aufgabenstellungen beschreiben (3)</p> <p>Begriffe formaler Verifikationstechniken erläutern (2)</p>	<p>Algorithmen, Datenstrukturen und Datentypen charakterisieren (9)</p> <p>Softwarearchitektur & deren Entwicklung beschreiben (2)</p> <p>Notwendigkeit von Werkzeugen zur Softwareentwicklung begründen (2)</p>	
<i>Stufe 3 Anwenden</i>			<p>Werkzeuge zur Software-Entwicklung nutzen (30)</p> <p>Mathematische Berechnungen und Codierungen durchführen (5)</p> <p>Bestehende Bibliotheken einsetzen (5)</p> <p>Programme professionell dokumentieren (4)</p> <p>Qualitätskriterien und Programmierkonventionen auf Quellcode anwenden (4)</p> <p>Computer bedienen (1)</p> <p>LaTeX benutzen (1)</p> <p>UNIX-Systeme bedienen mit Hilfe der Kommandozeile (1)</p>	
<i>Stufe 4 Analysieren</i>		<p>Programmiersprachen & -paradigmen differenzieren (6)</p>	<p>Komplexität von Algorithmen analysieren (24)</p> <p>Problemstellung zerlegen (20)</p> <p>Algorithmen auf Eigenschaften und Eignung hin analysieren (9)</p> <p>(Fremde) Programme beschreiben können (8)</p> <p>Datenstrukturen auf Eigenschaften und Eignung hin analysieren (6)</p>	

			Formale Sprachen durch Chomsky-Hierarchie differenzieren (1) Grenzen der Berechnungskraft von Rechnermodellen aufzeigen (1)	
<i>Stufe 5 Bewerten</i>			Angemessenheit von Algorithmen, Datenstrukturen und Datentypen beurteilen (34) Testen von Algorithmen und Programmen auf Fehler (19) Komplexität von Algorithmen beurteilen (14) Eigenschaften von Algorithmen und Programmen beweisen (10) Angemessenheit programmiersprachlicher Lösungen beurteilen (5) Beurteilen von Programmierwerkzeugen & Templates (2)	Angemessenheit des Selbstkonzepts bewerten (1)
<i>Stufe 6 Erzeugen</i>			Programmiersprachliche, lauffähige Lösungen für Probleme schreiben (80) Algorithmen mit Hilfe von Entwurfsmustern problemadäquat entwerfen (41) Algorithmen implementieren (33) Modellieren von Problemen und Programmen (26) Kleinere Programme strukturiert entwerfen (17) Standard-Algorithmen anwenden (16) Standard-Datenstrukturen anwenden (14) Datenstrukturen entwerfen (12) Datenstrukturen anpassen (8) Algorithmen anpassen (7) (Abstrakte) Datentypen konzipieren & einsetzen (5) Gegebene Programm(-komponenten) anpassen & verändern (5) Formale Syntaxbeschreibungen entwerfen (4) Programm-Spezifikation entwerfen (3) Größere Anwendungen/Applikationen entwickeln (2) Formale Werkzeuge & Testfälle entwickeln (2) GUIs programmieren (2) Nebenläufige und parallele Prozesse entwerfen (2) Bibliotheken entwickeln (1) Methoden aufrufen (1) Objekte erzeugen (1) Schnittstellen entwerfen (1)	Transfer von Wissen und Erfahrung auf neue Aufgaben und Programmiersprachen (13) Selbstständige Organisation des Lernprozesses (unter Anwendung externer Ressourcen) (7) Abstraktion von Problemen (2)

2. nicht operationalisiert³

1.

"Auf der Basis mathematisch exakter Beweise erfassen [die Studierenden] hierfür wichtige Gebiete der Theoretischen Informatik."

[Modulbeschreibungen\MH01; Position: 6 - 6]

2.

"Weiterhin erfassen die Studierenden die Berechnungskraft gängiger Rechnermodelle durch endliche Automaten."

[Modulbeschreibungen\MH01; Position: 9 - 9]

3.

"Die Lehrveranstaltungen dieses Moduls vermitteln fachlichen Grundlagen der Softwareentwicklung und der Informatik."

[Modulbeschreibungen\MH01; Position: 13 - 13]

4.

"[Die Studierenden lernen, kleine Probleme] mit Hilfe von Programmen zu lösen."

[Modulbeschreibungen\MH01; Position: 15 - 15]

5.

"Darüber hinaus lernen [die Studierenden] existierende Algorithmen zu bewerten."

[Modulbeschreibungen\MH01; Position: 17 - 17]

6.

"Die Studierenden erkennen, dass das erlernte Wissen erforderlich ist, um einfache mobile Apps zu schreiben."

[Modulbeschreibungen\MH01; Position: 25 - 25]

7.

"Die Studierenden bekommen einen Einblick in die Programmiersprachen C/C++."

[Modulbeschreibungen\MH01; Position: 28 - 28]

8.

"Die Teilnehmer haben einen Überblick über das Gebiet der Informatik."

³ In diesem Abschnitt werden alle als „nicht operationalisiert“ kodierten Lehr- und Lernziele des Samples aufgeführt. Ursache ist die fehlende Sichtbarkeit bzw. Messbarkeit der angestrebten Lernergebnisse.

[Modulbeschreibungen\MH02; Position: 5 - 5]

9.

"[Die Teilnehmer] haben die wichtigsten Konzepte einer höheren Programmiersprache und ihrer Verwendung verstanden."

[Modulbeschreibungen\MH02; Position: 6 - 6]

10.

"[Die Teilnehmer] haben die Abstraktionskonzepte moderner Programmiersprachen verstanden."

[Modulbeschreibungen\MH02; Position: 13 - 13]

11.

"Verständnis für die Auswirkungen theoretischer und tatsächlicher Komplexität."

[Modulbeschreibungen\MH02; Position: 21 - 21]

12.

"[Erweiterung der Kompetenz] im Verstehen von Algorithmen und der zugehörigen Datenstrukturen."

[Modulbeschreibungen\MH02; Position: 24 - 24]

13.

"Erste Begegnung mit nebenläufigen Algorithmen."

[Modulbeschreibungen\MH02; Position: 25 - 25]

14.

"Beherrschung von grundlegenden Konzepten der (allgemeinen sowie deklarativen) Programmierung."

[Modulbeschreibungen\MH03; Position: 10 - 10]

15.

"Vorbereitung auf die zukünftige Entwicklung von Programmiersprachen."

[Modulbeschreibungen\MH03; Position: 13 - 13]

16.

"Es soll der Umgang mit einer systemnahen Programmiersprache am Beispiel von C erlernt werden."

[Modulbeschreibungen\MH03; Position: 30 - 30]

17.

"Die Themen und Aufgaben umfassen u.a. die Spezifikation von Schnittstellen, den modularen Aufbau von Systemsoftware, formatierte Ein-/Ausgabe, Parser-Generierung, CPU Zuteilungsstrategien (Scheduling), Prozesse und deren Verwaltung, Verfahren der Prozesskommunikation und -synchronisation, Signale, Pipes und Sockets, Kommunikationsprotokolle, die TCP/IP-Protokollfamilie sowie das Client/Server-Modell."

[Modulbeschreibungen\MH03; Position: 31 - 31]

18.

"Die Studierenden erwerben ein allgemeines Verständnis der wichtigsten Aspekte moderner Softwareentwicklung."

[Modulbeschreibungen\MH03; Position: 33 - 33]

19.

"Die Studierenden lernen die angemessene Anwendung wichtiger Techniken zur Beherrschung komplexer Systeme."

[Modulbeschreibungen\MH04; Position: 7 - 7]

20.

"Die Studierenden erwerben vertiefte Kenntnis in den Bereichen Rekursion und Abstraktion."

[Modulbeschreibungen\MH04; Position: 8 - 8]

21.

"Die Studierenden schulen ihre analytischen Denkweisen."

[Modulbeschreibungen\MH04; Position: 9 - 9]

22.

"[Für die Umsetzung lernen Studenten am Beispiel der Programmiersprache Java, wie man mit analytischem und konzeptionellem Denken die Problemstellungen] löst."

[Modulbeschreibungen\MH04; Position: 17 - 17]

23.

"[Die Studierenden lernen aus dem Kanon der Grundlagen der Informatik die objektorientierte Programmierung] anwenden."

[Modulbeschreibungen\MH04; Position: 20 - 20]

24.

"[Darüber hinaus] entwickeln [die Studierenden] ein Verständnis für die Realisierung großer Softwaresysteme."

[Modulbeschreibungen\MH04; Position: 25 - 25]

25.

"[Darüber hinaus] üben [die Studierenden] den Umgang mit ausgewählten Klassenbibliotheken."

[Modulbeschreibungen\MH04; Position: 26 - 26]

26.

"Die Studierenden erhalten ein tiefgehendes Verständnis für professionellen Code, automatisierte Tests und testgetriebene Softwareentwicklung."

[Modulbeschreibungen\MH04; Position: 54 - 54]

27.

"Die Teilnehmenden begreifen die Tool Chain der C/C++ Entwicklung und deren Bedeutung."

[Modulbeschreibungen\MH04; Position: 56 - 56]

28.

"Die Studierenden erlernen die Grundlagen der Programmiersprache C++."

[Modulbeschreibungen\MH04; Position: 57 - 57]

29.

"[Die Studierenden] beherrschen die Grundlagen der C++ Standardbibliothek."

[Modulbeschreibungen\MH04; Position: 60 - 60]

30.

"Die Studentinnen und Studenten beherrschen am Ende des Moduls imperative und objektorientierte Programmierkonzepte."

[Modulbeschreibungen\MH05; Position: 5 - 5]

31.

"[Die Studentinnen und Studenten beherrschen am Ende des Moduls] [...] die Anwendbarkeit [imperativer und objektorientierter Programmierkonzepte]."

[Modulbeschreibungen\MH05; Position: 6 - 6]

32.

"[Die Studentinnen und Studenten] [können am Ende des Moduls Beschreibungen und Quelltexte elementarer Algorithmen] verstehen."

[Modulbeschreibungen\MH05; Position: 19 - 19]

33.

"[Die Studentinnen und Studenten] [können am Ende des Moduls] Beschreibungen von Programmkomponenten verstehen."

[Modulbeschreibungen\MH05; Position: 22 - 22]

34.

"[Die Studentinnen und Studenten] verstehen die Begriffe Laufzeit und Korrektheit."

[Modulbeschreibungen\MH05; Position: 27 - 27]

35.

"[Die Studentinnen und Studenten] haben ein grundlegendes Verständnis der Berechenbarkeit."

[Modulbeschreibungen\MH05; Position: 29 - 29]

36.

"[Studentinnen und Studenten] verstehen den Unterschied zwischen Spezifikation und Implementierung."

[Modulbeschreibungen\MH05; Position: 33 - 33]

37.

"[Studentinnen und Studenten] verstehen die praktische Bedeutung von NP-Vollständigkeit für die effiziente Lösbarkeit von Problemen."

[Modulbeschreibungen\MH05; Position: 41 - 41]

38.

"Die Studierenden beherrschen Grundkenntnisse einer Programmiersprache."

[Modulbeschreibungen\MH06; Position: 7 - 7]

39.

"[Die Studierenden] verstehen die Notwendigkeit, korrekte und robuste Programme zu entwickeln."

[Modulbeschreibungen\MH06; Position: 9 - 9]

40.

"[Die Studierenden] erwerben Fachkompetenz zu Analyse und Design von Algorithmen."

[Modulbeschreibungen\MH06; Position: 16 - 16]

41.

"Die Studierenden erlangen grundlegende Kenntnisse und Fähigkeiten im Testen modularer bzw. objektorientierter Software."

[Modulbeschreibungen\MH06; Position: 22 - 22]

42.

"[Die Studierenden] erfahren anhand von konkreten Anwendungen die Bedeutung der Mathematik für die Informatik."

[Modulbeschreibungen\MH07; Position: 12 - 12]

43.

"[Die Studierenden] verstehen die Konzepte der Objektorientierung und objektorientierten Programmierung am Beispiel der Programmiersprache Java."

[Modulbeschreibungen\MH07; Position: 17 - 17]

44.

"[Die Studierenden] verstehen die Entwurfsvorgaben in Klassendiagrammen."

[Modulbeschreibungen\MH07; Position: 18 - 18]

45.

"[Die Studierenden] verstehen die Konzepte der Prozeduralen und objektorientierten Programmierung, wie sie in den Programmiersprache C und C++ umgesetzt sind."

[Modulbeschreibungen\MH07; Position: 24 - 24]

46.

"Die Studierenden beherrschen die Syntax [...] [der] Programmiersprachen [C und C++]."

[Modulbeschreibungen\MH07; Position: 25 - 25]

47.

"Beherrschung der Konzepte von Programmiersprachen (z.B. Python)."

[Modulbeschreibungen\MH08; Position: 16 - 16]

48.

"Beherrschung effizienter Standardalgorithmen zum Multiplizieren und Matrixmultiplizieren, auf Folgen, Bäumen, Graphen und Punktmengen, u.a. zum Suchen und Sortieren auf Folgen, zum Durchlaufen, zum Suchen, Einfügen, Löschen auf allgemeinen und ausgeglichenen Suchbäumen."

[Modulbeschreibungen\MH08; Position: 21 - 21]

49.

"Die Studierenden haben ein breites und integriertes Wissen der wissenschaftlichen Grundlagen auf dem Gebiet der Deklarativen Programmierung nachgewiesen."

[Modulbeschreibungen\MH08; Position: 26 - 26]

50.

"Die Studierenden haben ein breites und integriertes [Verstehen der wissenschaftlichen Grundlagen auf dem Gebiet der Deklarativen Programmierung nachgewiesen]."

[Modulbeschreibungen\MH08; Position: 27 - 27]

51.

"Die Studierenden verfügen über ein kritisches Verständnis der wichtigsten Theorien, Prinzipien und Methoden auf dem Gebiet der Deklarativen Programmierung."

[Modulbeschreibungen\MH08; Position: 28 - 28]

52.

"Das Wissen und Verstehen der Studierenden entspricht dem Stand der Fachliteratur und schließt einige vertiefte Wissensbestände auf dem aktuellen Stand der Forschung auf dem Gebiet ein."

[Modulbeschreibungen\MH08; Position: 30 - 30]

53.

"[Die Studierenden sind in der Lage] Problemlösungen und Argumente auf dem Gebiet der Deklarativen Programmierung zu erarbeiten."

[Modulbeschreibungen\MH08; Position: 32 - 32]

54.

"[Die Studierenden sind in der Lage] [Problemlösungen und Argumente auf dem Gebiet der Deklarativen Programmierung] weiterzuentwickeln."

[Modulbeschreibungen\MH08; Position: 33 - 33]

55.

"Die Studentinnen erlangen einen Überblick über die grundlegenden Begriffe und Konzepte der Informatik, wobei der Begriff des Algorithmus eine zentrale Rolle einnimmt."

[Modulbeschreibungen\MH09; Position: 13 - 13]

56.

"[Die Studentinnen] erfahren, wie auf Basis von theoretischen Modellen und Verfahren ein Computer die ihm übergebenen Algorithmen verarbeitet."

[Modulbeschreibungen\MH09; Position: 14 - 14]

57.

"Theoretische Konzepte werden mit Hilfe formaler Beschreibungsverfahren, speziell UML, erarbeitet."

[Modulbeschreibungen\MH09; Position: 19 - 19]

58.

"Grundlegende Informatikkonzepte wiedergeben können."

[Modulbeschreibungen\MH10; Position: 5 - 5]

59.

"[Grundlegende Informatikkonzepte] erklären können."

[Modulbeschreibungen\MH10; Position: 6 - 6]

60.

"[Konzepte einer imperativen Programmiersprache] verstehen."

[Modulbeschreibungen\MH10; Position: 8 - 8]

61.

"[Konzepte einer imperativen Programmiersprache] anwenden können."

[Modulbeschreibungen\MH10; Position: 9 - 9]

62.

"Formale Syntaxbeschreibungen verstehen."

[Modulbeschreibungen\MH10; Position: 17 - 17]

63.

"Operationelle Semantik einfacher While-Sprachen verstehen."

[Modulbeschreibungen\MH10; Position: 19 - 19]

64.

"Die Vorlesungen Praktische Informatik 1 und 2 vermitteln essenzielles Grundwissen und Basisfähigkeiten, deren Beherrschung für nahezu jede vertiefte Beschäftigung mit Informatik – sowohl in der industriellen Anwendung, als auch in der Forschung – Voraussetzung ist."

[Modulbeschreibungen\MH10; Position: 26 - 26]

65.

"Die Vorlesungen Praktische Informatik 1 und 2 vermitteln essenzielles Grundwissen und Basisfähigkeiten, deren Beherrschung für nahezu jede vertiefte Beschäftigung mit Informatik – sowohl in der industriellen Anwendung, als auch in der Forschung – Voraussetzung ist."

[Modulbeschreibungen\MH10; Position: 39 - 39]

66.

"[Konzepte und typische Merkmale des funktionalen Programmierens] verstehen."

[Modulbeschreibungen\MH10; Position: 42 - 42]

67.

"Die Vorlesung Praktische Informatik 3 vermittelt essenzielles Grundwissen und Basisfähigkeiten, deren Beherrschung für nahezu jede vertiefte Beschäftigung mit Informatik Voraussetzung ist."

[Modulbeschreibungen\MH10; Position: 48 - 48]

68.

"[Die Studierenden können die Basiskonzepte einer modernen Programmiersprache] anwenden."

[Modulbeschreibungen\MH11; Position: 6 - 6]

69.

"Die Studierenden beherrschen fortgeschrittene Programmiertechniken."

[Modulbeschreibungen\MH11; Position: 18 - 18]

70.

"[Die Studierenden] beherrschen die grundlegenden Konzepte, Formalismen und Notationen sowie die wichtigsten Algorithmen."

[Modulbeschreibungen\MH11; Position: 23 - 23]

71.

"[Die Studierenden] [besitzen die Fähigkeit zur] einfachen Analyse."

[Modulbeschreibungen\MH11; Position: 27 - 27]

72.

"[Die Teilnehmer] beherrschen das grundlegende Handwerkszeug der Programmierung im Kleinen."

[Modulbeschreibungen\MH12; Position: 14 - 14]

73.

"[Die Teilnehmer] verstehen die Konzepte der Programmierung über eine konkrete Programmiersprache hinaus."

[Modulbeschreibungen\MH12; Position: 18 - 18]

74.

"[Die Teilnehmer] haben einen ersten Eindruck vom Komplexitätsbegriff."

[Modulbeschreibungen\MH12; Position: 20 - 20]

75.

"Weiterhin sind [die Studierenden] vertraut mit fortgeschrittenen Programmiersprachkonzepten, sowie mit Konzepten von Entwurfsmustern und Refactorings."

[Modulbeschreibungen\MH12; Position: 25 - 25]

76.

"Die Studierenden verfügen über ein grundlegendes Verständnis einfacher formaler Konzepte und mathematischer Methoden der Informatik."

[Modulbeschreibungen\MH12; Position: 28 - 28]

77.

"[Die Studierenden] sind in der Lage, [...] [geeignete Abstraktionen, Modellbildungen und Verfahren zur Beschreibung und Analyse von Algorithmen und Prozessen] auf einem sauberen, theoretischen Fundament anzuwenden."

[Modulbeschreibungen\MH12; Position: 31 - 31]

78.

"Die Studierenden verstehen den grundsätzlichen Ansatz und die Vorgehensweise der objektorientierten Programmierung."

[Modulbeschreibungen\MH13; Position: 5 - 5]

79.

"[Die Studierenden] begreifen den Aufbau und die Wechselwirkung von Objekten."

[Modulbeschreibungen\MH13; Position: 6 - 6]

80.

"[Die Studierenden] beherrschen die grundlegenden imperativen Programmiertechniken in Java."

[Modulbeschreibungen\MH13; Position: 7 - 7]

81.

"Die Studierenden erlangen ein vertieftes Verständnis objektorientierter Programmentwicklung."

[Modulbeschreibungen\MH13; Position: 16 - 16]

82.

"[Die Studierenden] verstehen das Konzept der Klassenhierarchien in seiner Tragweite."

[Modulbeschreibungen\MH13; Position: 17 - 17]

83.

"[Die Studierenden] beherrschen [die] Nutzung [des Konzepts der Klassenhierarchien]."

[Modulbeschreibungen\MH13; Position: 18 - 18]

84.

"[Die Studierenden] [können die Effizienz von Algorithmen mit Hilfe mathematischer Methoden] anwenden."

[Modulbeschreibungen\MH13; Position: 28 - 28]

85.

"Insbesondere beherrschen [die Studierenden] den Umgang mit Zeigern und zeigerbasierten Datenstrukturen."

[Modulbeschreibungen\MH13; Position: 33 - 33]

86.

"[Insbesondere beherrschen] [die Studierenden] die Implementierung von iterativen und rekursiven Algorithmen."

[Modulbeschreibungen\MH13; Position: 34 - 34]

87.

"Verständnis für Abläufe im Rechner bei Programmausführung."

[Modulbeschreibungen\MH14; Position: 6 - 6]

88.

"Verstehen grundlegender Programmierkonzepte."

[Modulbeschreibungen\MH14; Position: 7 - 7]

89.

"Gute Fertigkeiten im Programmieren im Kleinen."

[Modulbeschreibungen\MH14; Position: 8 - 8]

90.

"Fertigkeiten in objektorientierter Programmierung."

[Modulbeschreibungen\MH14; Position: 9 - 9]

91.

"Fertigkeiten im Erfassen gegebener Algorithmen und Datenstrukturen."

[Modulbeschreibungen\MH14; Position: 12 - 12]

92.

"[Fertigkeiten im] Entwickeln eigener Algorithmen und Datenstrukturen."

[Modulbeschreibungen\MH14; Position: 13 - 13]

93.

"Fertigkeiten in Effizienz- und Korrektheitsanalyse."

[Modulbeschreibungen\MH14; Position: 14 - 14]

94.

"Vertiefung Programmierfertigkeiten."

[Modulbeschreibungen\MH14; Position: 15 - 15]

95.

"Analyse mit Hilfe von Szenarien, Objekt- und Klassendiagrammen."

[Modulbeschreibungen\MH14; Position: 17 - 17]

96.

"Fachunabhängige Kompetenzen (Kommunikation und Kooperation): Fachunabhängige Kompetenzen werden integriert erworben."

[Modulbeschreibungen\MH15; Position: 8 - 8]

97.

"Die LV trägt zu den Lernergebnissen des Moduls mit der theoriebezogenen Erarbeitung der angegebenen Themen/Inhalte bei.

Themen/Inhalte der LV

Iteratoren, Streams mit Faltungen und Maps

Baum-, Ereignis- oder Strombasierte Verarbeitung Hierarchischer Strukturen

Speicherbelegung und Freigabe, explizit und implizit

Nebenläufigkeit

Kommunikation

Funktionen als Typen für Argumente

Reflection und dynamisches Laden von Klassen"

[Modulbeschreibungen\MH15; Position: 9 - 17]

98.

"Die LV trägt zu den Lernergebnissen des Moduls mit der anwendungsbezogenen Erarbeitung der angegebenen Themen/Inhalte bei.

Themen/Inhalte der LV

siehe zugehörige Vorlesung"

[Modulbeschreibungen\MH15; Position: 19 - 21]

99.

"Die erworbenen Fähigkeiten tragen in besonderem Maße zur Ausprägung von algorithmischen Kompetenzen bei, erweitern die Methodenkompetenzen und die Analyse-, Design- und Realisierungskompetenzen."

[Modulbeschreibungen\MH15; Position: 34 - 34]

100.

"Fachunabhängige Kompetenzen (Kommunikation und Kooperation): Fachunabhängige Kompetenzen werden integriert erworben."

[Modulbeschreibungen\MH15; Position: 35 - 35]

101.

"Die LV trägt zu den Lernergebnissen des Moduls mit der theoriebezogenen Erarbeitung der angegebenen Themen/Inhalte bei.

Themen/Inhalte der LV:

Eigenschaften von Algorithmen, Probleme versus Algorithmen

Suchen, einfache Sortierverfahren, effiziente Sortierverfahren

Laufzeit und Komplexität, O-Notation, Analyse von Algorithmen, Lösen von Rekurrenzen

Algorithmenentwurf und Algorithmenmuster

Abstrakte Datentypen und deren Implementierung (Listen, Mengen)

Einfache dynamische Datenstrukturen (verkettete Listen, Keller, Warteschlangen)

Bäume, Durchlaufen, Binärbäume, Suchbäume, Ausgeglichene Bäume

Hashing, Hash-Funktionen, Kollisionsbehandlung

Graphen"

[Modulbeschreibungen\MH15; Position: 36 - 46]

102.

"Die LV trägt zu den Lernergebnissen des Moduls mit der anwendungsbezogenen Erarbeitung der angegebenen Themen/Inhalte bei."

[Modulbeschreibungen\MH15; Position: 48 - 48]

103.

"[Wissen und Verstehen:] [Die Studierenden sollen grundlegende Datenstrukturen mit deren Eigenschaften und Leistungsparametern] [...] in asymptotischer Notation verstehen"

[Modulbeschreibungen\MH16; Position: 6 - 6]

104.

"Die Studierenden sollen Grundbegriffe der Informatik aus Programmiersicht kennen."

[Modulbeschreibungen\MH16; Position: 13 - 13]

105.

"Die Studierenden sollen strukturiertes und objektorientiertes Programmieren am Beispiel einer imperativen Programmiersprache erlernen."

[Modulbeschreibungen\MH16; Position: 20 - 20]

106.

"[Die Studierenden] [sollen Wissen über funktionale Sprachen] auf einfache Probleme anwenden können."

[Modulbeschreibungen\MH16; Position: 27 - 27]

107.

"Nach erfolgreichem Abschluss der Veranstaltung sind Studierende mit den Grundlagen von funktionalen und objektorientierten Programmiersprachen vertraut."

[Modulbeschreibungen\MH17; Position: 5 - 5]

108.

"[Die Studierenden können] [...] die Komplexitätsklassen von Algorithmen und Datenstrukturen verstehen."

[Modulbeschreibungen\MH17; Position: 8 - 8]

109.

"[Die Studierenden] erwerben die Fähigkeiten die Grundprinzipien der Algorithmik anzuwenden."

[Modulbeschreibungen\MH17; Position: 14 - 14]

110.

"Außerdem verstehen [die Studierenden] bedeutende algorithmische Strategien."

[Modulbeschreibungen\MH17; Position: 17 - 17]

111.

"Nach dem erfolgreichen Besuch der Veranstaltung verstehen Studierende die Grundlagen paralleler Systeme und ihrer effizienten Programmierung."

[Modulbeschreibungen\MH17; Position: 20 - 20]

112.

"Beherrschen von Grundlagen der prozeduralen und der objektorientierten Programmierung am Beispiel der Sprachen C und C++."

[Modulbeschreibungen\MH18; Position: 5 - 5]

113.

"Wichtige Algorithmen verstehen."

[Modulbeschreibungen\MH18; Position: 11 - 11]

114.

"Wichtige Datenstrukturen verstehen (z.B. Arrays, Stapel, Bäume)."

[Modulbeschreibungen\MH18; Position: 13 - 13]

115.

"Prinzipien der Generischen Programmierung verstehen (Java, C#, C++)."

[Modulbeschreibungen\MH18; Position: 22 - 22]

116.

"Probleme und Lösungsansätze der nebenläufigen Programmierung verstehen (Java)."

[Modulbeschreibungen\MH18; Position: 24 - 24]

117.

"Fachkompetenz: Beherrschung einer imperativen Programmiersprache."

[Modulbeschreibungen\MH19; Position: 5 - 5]

118.

"[Fachkompetenz:] Beherrschung einer deklarativen Programmiersprache."

[Modulbeschreibungen\MH19; Position: 6 - 6]

119.

"Methodenkompetenz: erfolgreiches Bearbeiten einfacher programmiertechnischer Probleme."

[Modulbeschreibungen\MH19; Position: 7 - 7]

120.

"[Selbst- und Sozialkompetenz:] Algorithmisches Denken."

[Modulbeschreibungen\MH19; Position: 10 - 10]

121.

"[Methodenkompetenz:] Umsetzung formal spezifizierter Anforderungen in korrekte und effiziente Lösungen."

[Modulbeschreibungen\MH19; Position: 16 - 16]

122.

"Fachkompetenz: Prinzipien und Techniken des Software Engineering, Modellierung, Softwarearchitektur, Muster und Bibliotheken."

[Modulbeschreibungen\MH19; Position: 24 - 24]

123.

"[Fachkompetenz:] Prinzipien der Aufwandsabschätzung und Projektplanung."

[Modulbeschreibungen\MH19; Position: 25 - 25]

124.

"[Fachkompetenz:] Standards."

[Modulbeschreibungen\MH19; Position: 26 - 26]

125.

"Selbst- und Sozialkompetenz: Berücksichtigung nichttechnischer Rahmenbedingungen bei der Bearbeitung einer komplexen Aufgabe."

[Modulbeschreibungen\MH19; Position: 30 - 30]

126.

"[Selbst- und Sozialkompetenz:] Kernkompetenzen für Berufsqualifizierung im nichtakademischen Bereich."

[Modulbeschreibungen\MH19; Position: 32 - 32]

127.

"Realisierungs-Kompetenz: Beherrschung des imperativen Programmierparadigmas unter Nutzung von Objektbibliotheken."

[Modulbeschreibungen\MH20; Position: 9 - 9]

128.

"[Informatik-Kompetenz:] verstehen des Aufbaus und der Funktionsweise von Computern."

[Modulbeschreibungen\MH20; Position: 14 - 14]

129.

"Verstehen des Aufbaus und der Funktionsweise von Netzen wie dem Internet."

[Modulbeschreibungen\MH20; Position: 18 - 18]

130.

"Realisierungs-Kompetenz: Beherrschung des objektorientierten Programmierparadigmas."

[Modulbeschreibungen\MH20; Position: 25 - 25]

131.

"[Technologische Kompetenzen:] Compiler-Fehler verstehen."

[Modulbeschreibungen\MH20; Position: 45 - 45]

132.

"Design-, Realisierungs- und Methodenkompetenzen: Studierende verstehen das Speichermodell von C."

[Modulbeschreibungen\MH20; Position: 46 - 46]

133.

"[Design-, Realisierungs- und Methodenkompetenzen:] Studierende beherrschen die Kernkonzepte der Objektorientierung in C++."

[Modulbeschreibungen\MH20; Position: 48 - 48]

134.

"Design- und Realisierungskompetenzen: erworbenes technisches Wissen zur Lösung einer Problemstellung einsetzen."

[Modulbeschreibungen\MH20; Position: 51 - 51]

135.

"Das Programmieren ist eine der Basistätigkeiten von Informatikern und Voraussetzung für viele andere Veranstaltungen des Informatikstudiums."

[Modulbeschreibungen\MH21; Position: 5 - 5]

136.

"Ziel des Moduls „Programmierung, Datenstrukturen und Algorithmen“ ist das Erlernen grundlegender Konzepte der imperativen und prozeduralen Programmierung anhand der Programmiersprache Java."

[Modulbeschreibungen\MH21; Position: 6 - 6]

137.

"[Ziel des Moduls „Programmierung, Datenstrukturen und Algorithmen“ ist] die Vorstellung bekannter, effizienter Algorithmen und Datenstrukturen für verschiedene, häufig vorkommende Problemstellungen."

[Modulbeschreibungen\MH21; Position: 7 - 7]

138.

"Methodenkompetenzen: Die Studierenden lösen gegebene Probleme unter den Gesichtspunkt der imperativen Programmierung."

[Modulbeschreibungen\MH21; Position: 25 - 25]

139.

"[Selbstkompetenzen:] [Die Studierenden] beziehen die Konzepte des allgemeinen Programmentwurfs in ihr Handeln ein."

[Modulbeschreibungen\MH21; Position: 30 - 30]

140.

"Die Objektorientierung stellt heutzutage den Stand der Technik in der Softwareentwicklung dar."

[Modulbeschreibungen\MH21; Position: 32 - 32]

141.

"Fähigkeiten: Programmiertechniken imperativer und objektorientierter Programmiersprachen."

[Modulbeschreibungen\MH22; Position: 11 - 11]

142.

"[Kenntnisse:] Wichtigste Technologien und Konzepte, die für den Entwurf und die Analyse rechnergestützter Systeme benötigt werden."

[Modulbeschreibungen\MH22; Position: 28 - 28]

143.

"Fähigkeiten: Umgang mit der Programmiersprache C und deren Zusammenspiel mit der Hardware."

[Modulbeschreibungen\MH22; Position: 29 - 29]

144.

"[Fähigkeiten:] Nutzung grundlegender Funktionalitäten unter Berücksichtigung des Zusammenspiels der Basiskomponenten eines Betriebssystems."

[Modulbeschreibungen\MH22; Position: 30 - 30]

145.

"[Fähigkeiten:] Effiziente Ressourcenverwaltung."

[Modulbeschreibungen\MH22; Position: 31 - 31]

146.

"Dieses Modul vermittelt grundlegende Programmierkenntnisse in einer objektorientierten Programmiersprache.

Darüber hinaus werden einführend Aspekte von Algorithmen und Datenstrukturen behandelt.

Es wird keine Programmiererfahrung vorausgesetzt.

Grundlegende Begriffe der Informatik .

Primitive Datentypen und Variablen.

Kontrollstrukturen.

Eigene Datentypen (Klassen) und Arrays.

Programmstrukturen im Speicher (Heap, Stack).

Konzepte der Objektorientierung (Polymorphie, Schnittstellen).

Rekursion.

Fehlerbehandlung.

Dynamische Datenstrukturen (Listen, Binärbäume, Hashing).

Suchen und Sortieren (ausgewählte Algorithmen, u.a. binäre Suche, BubbleSort, QuickSort).

Datenströme (Standard-Eingabe und -Ausgabe, einfache 2D-Grafik, Dateien)."

[Modulbeschreibungen\MH23; Position: 5 - 18]

147.

"Studierende sollen nach Absolvierung der Lehrveranstaltungen in der Lage sein, Begriffe der Informatik und der Programmierung zu nennen."

[Modulbeschreibungen\MH23; Position: 19 - 19]

148.

"[Studierende sollen nach Absolvierung der Lehrveranstaltungen in der Lage sein, Begriffe der Informatik und der Programmierung] zu erläutern."

[Modulbeschreibungen\MH23; Position: 20 - 20]

149.

"[Studierende sollen nach Absolvierung der Lehrveranstaltungen in der Lage sein,] einfache Algorithmen (iterativ und rekursiv) zu verstehen."

[Modulbeschreibungen\MH23; Position: 21 - 21]

150.

"Dieses Modul vermittelt grundlegende Kenntnisse aus folgenden Bereichen: Algorithmen und ihre formalen Grundlagen.

Rechenmodelle, Effizienzmaße.

Sortierverfahren (Quicksort, Heapsort, Mergesort, ...).

Aufwandsabschätzung im Mittel.

Suchstrategien (Binärsuche, Interpolationsuche, Textsuche, ...).

Dictionaries (offene Hashverfahren, dynamische Hashverfahren).

Suchbäume (AVL-Bäume, B-Bäume, Splay-Trees, ...).

Vorrangwarteschlangen (Heaps, Binominal Queues, Fibonacci-Heaps, ...).

Amortisierte Laufzeitanalysen.

Einführung in Graphenalgorithmien (Tiefensuche, Breitensuche, Zusammenhangsprobleme, ...)."

[Modulbeschreibungen\MH23; Position: 29 - 38]

151.

"Studierende, die das Modul erfolgreich absolviert haben, besitzen anschließend ein Basisverständnis der wichtigsten Grundlagen über Algorithmen."

[Modulbeschreibungen\MH23; Position: 39 - 39]

152.

"[Studierende, die das Modul erfolgreich absolviert haben,] haben die Fähigkeit zur algorithmischen Problembearbeitung erworben."

[Modulbeschreibungen\MH23; Position: 41 - 41]

153.

"[Nach erfolgreichem Abschluss des Moduls sollen die Studierenden in der Lage sein,] eine gestellte komplexere Aufgabe verstehen."

[Modulbeschreibungen\MH23; Position: 51 - 51]

154.

"Die Studierenden sollen einen Überblick über Geschichte und Entwicklung des Fachgebietes Informatik erhalten."

[Modulbeschreibungen\MH24; Position: 5 - 5]

155.

"[Die Studierenden sollen] alle wichtigen Elemente der Programmiersprache Java anwenden können."

[Modulbeschreibungen\MH24; Position: 10 - 10]

156.

"[Die Studierenden sollen] die Grundkonzepte der objektorientierten Programmierung verstehen können."

[Modulbeschreibungen\MH24; Position: 13 - 13]

157.

"[Die Studierenden sollen] [die] Wechselwirkungen [grundlegender Datenstrukturen und Algorithmen], insbesondere unter Laufzeit- und Speicherplatzbetrachtungen, verstehen."

[Modulbeschreibungen\MH24; Position: 18 - 18]

158.

"[Die Studierenden sollen] die wesentlichen Such- und Sortieralgorithmen verstehen"

[Modulbeschreibungen\MH24; Position: 20 - 20]

159.

"[Die Studierenden sollen] die wesentlichen Testverfahren verstehen."

[Modulbeschreibungen\MH24; Position: 33 - 33]

160.

"[Studierende sollten nach der Veranstaltung in der Lage sein, Probleme] zu lösen."

[Modulbeschreibungen\MH24; Position: 38 - 38]

161.

"[Studierende sollten nach der Veranstaltung in der Lage sein,] funktionale Konzepte in anderen Programmiersprachen und Frameworks, z. B. Java Streams, zu verstehen."

[Modulbeschreibungen\MH24; Position: 43 - 43]

162.

"Die Studierenden sollen algorithmische Aufgabenstellungen verstehen können."

[Modulbeschreibungen\MH24; Position: 46 - 46]

163.

"Im Rahmen von Vorlesung und Übungen erlernen die Studierenden die Beherrschung einer Programmiersprache sowie Grundfertigkeiten zum Algorithmen- und Software-Entwurf."

[Modulbeschreibungen\MH25; Position: 5 - 5]

164.

"Die Veranstaltung führt in die Grundlagen der Entwicklung von Software ein und erprobt diese am praktischen Beispiel."

[Modulbeschreibungen\MH25; Position: 6 - 6]

165.

"Im Fokus stehen dabei imperative Programmierkonstrukte, die selbst im Rahmen objektorientierter Softwareentwicklung meist die Grundlagen zur Implementierung von Datenstrukturen und Algorithmen liefern."

[Modulbeschreibungen\MH25; Position: 7 - 7]

166.

"Dazu gehören insbesondere: Das systematische Testen von Softwareartefakten, Techniken zum Debugging, Leitlinien zur Benennung von Bezeichnern, zur Kommentierung, und zur Modularisierung (einfache und explizite Schnittstellen, strukturierter Kontrollfluss)."

[Modulbeschreibungen\MH25; Position: 9 - 9]

167.

"In dieser Veranstaltung erlernen die Studierenden die Grundlagen der objektorientierten Programmierung zur Umsetzung großer Softwaresysteme anhand einer geeigneten Programmiersprache (z.B. Java oder C#)."

[Modulbeschreibungen\MH25; Position: 11 - 11]

168.

"Neben objektorientierten Konzepten (Vererbung, Schnittstellen-Implementierung, Geheimnisprinzip, ...) wird die Modellierung anhand von UML eingeführt."

[Modulbeschreibungen\MH25; Position: 12 - 12]

169.

"Der Begriff des Design Patterns wird eingeführt und einige relevante Beispiele erarbeitet."

[Modulbeschreibungen\MH25; Position: 13 - 13]

170.

"[Die Studierenden] haben vertiefte Kenntnisse in mindestens einer objektorientierten Programmiersprache."

[Modulbeschreibungen\MH25; Position: 16 - 16]

171.

"[Die Studierenden] können [...] [grundlegende Datenstrukturen und Algorithmen] umsetzen."

[Modulbeschreibungen\MH25; Position: 18 - 18]

172.

"Die Studierenden verfügen über ein Verständnis für verschiedene Programmierparadigmen."

[Modulbeschreibungen\MH25; Position: 22 - 22]

173.

"Kennenlernen von Grundbegriffen der Informatik."

[Modulbeschreibungen\MH26; Position: 5 - 5]

174.

"Verstehen des Algorithmusbegriffes."

[Modulbeschreibungen\MH26; Position: 6 - 6]

175.

"[Laufzeitanalysen] verstehen."

[Modulbeschreibungen\MH26; Position: 8 - 8]

176.

"Bereitstellen von Kenntnissen für die Digitaltechnik-Vorlesung"

[Modulbeschreibungen\MH26; Position: 11 - 11]

177.

"Die Veranstaltung führt in die Grundlagen der objektorientierten Programmierung mit der Programmiersprache Java ein."

[Modulbeschreibungen\MH26; Position: 13 - 13]

178.

"[Die Studierenden] sollen dabei die Prinzipien der objektorientierten Programmierung und die wichtigsten Sprachelemente von Java beherrschen."

[Modulbeschreibungen\MH26; Position: 15 - 15]

179.

"Aufbauend auf dem Modul Programmierung 1 vertiefen die Studierenden ihre Kenntnisse und ihr Verständnis objektorientierter Programmierung."

[Modulbeschreibungen\MH26; Position: 17 - 17]

180.

"[Die Studierenden sollen die wichtigsten Vorgehensmodelle der Softwareentwicklung] verstehen lernen."

[Modulbeschreibungen\MH26; Position: 32 - 32]

181.

"Höherstufige, getypte funktionale Programmierung anwenden können."

[Modulbeschreibungen\MH27; Position: 5 - 5]

182.

"Verständnis rekursiver Datenstrukturen und Algorithmen, Zusammenhänge mit Mengenlehre."

[Modulbeschreibungen\MH27; Position: 6 - 6]

183.

"Typabstraktion und Modularisierung verstehen."

[Modulbeschreibungen\MH27; Position: 9 - 9]

184.

"Struktur von Programmiersprachen verstehen."

[Modulbeschreibungen\MH27; Position: 10 - 10]

185.

"Anwendungsnahe Rechenmodelle mit maschinennahen Rechenmodellen realisieren können."

[Modulbeschreibungen\MH27; Position: 13 - 13]

186.

"[Bei dem Kennenlernen der Grundprinzipien der imperativen/objektorientierten Programmierung] [...] wird primär Java als Programmiersprache verwendet."

[Modulbeschreibungen\MH27; Position: 18 - 18]

187.

"[In dieser Vorlesung lernen sie] die Grundlagen imperativer und objektorientierten Sprachen."

[Modulbeschreibungen\MH27; Position: 20 - 20]

188.

"Jeder Teilnehmende beherrscht wesentliche Bestandteile der Syntax und Semantik der Programmiersprache JAVA."

[Modulbeschreibungen\MH28; Position: 8 - 8]

189.

"Die Studenten besitzen die notwendigen theoretischen Grundkenntnisse und praktischen Fähigkeiten und Fertigkeiten für das systematische Programmieren im Kleinen als Voraussetzung für alle weiteren Informatik-Module."

[Modulbeschreibungen\MH28; Position: 12 - 12]

190.

"Darüber hinaus wird im Rahmen des Moduls eine Harmonisierung der informatikbezogenen Kenntnisse und Fertigkeiten der Studierenden bedingt durch weiter auseinandergehende Ausgangsniveaus angestrebt."

[Modulbeschreibungen\MH28; Position: 13 - 13]

191.

"In diesem Modul steht die Vertiefung der Fach- und Methodenkompetenzen im Bereich der Programmierung im Vordergrund."

[Modulbeschreibungen\MH28; Position: 15 - 15]

192.

"Die Studenten besitzen vertiefte Kenntnisse im Bereich der programmiertechnischen Umsetzung von praxisrelevanten Problemstellungen."

[Modulbeschreibungen\MH28; Position: 22 - 22]

193.

"Das Modul fördert die Schlüsselkompetenz des komplexen Denkens."

[Modulbeschreibungen\MH28; Position: 33 - 33]

194.

"Dadurch wird zusätzlich die Sozialkompetenz gefördert."

[Modulbeschreibungen\MH28; Position: 35 - 35]

195.

"Unabhängig von der speziellen Aufgabenstellung treten beim Entwurf von Lösungen immer wieder ähnliche kerninformatische Probleme auf, für die es gut etablierte Standardalgorithmen und Datenstrukturen gibt."

[Modulbeschreibungen\MH28; Position: 37 - 37]

196.

"Gleichzeitig wird das Abstraktionsvermögen gefördert."

[Modulbeschreibungen\MH28; Position: 40 - 40]

197.

"Oft bieten sich zur Lösung eines Problems mehrere alternative Verfahren an."

[Modulbeschreibungen\MH28; Position: 41 - 41]

198.

"Die Programmiersprache C spielt in der Systemprogrammierung und der Programmierung eingebetteter Rechnersysteme eine wichtige Rolle."

[Modulbeschreibungen\MH28; Position: 48 - 48]

199.

"Die Absolventen des Moduls erlangen sach- und fachbezogene Methodenkompetenzen auf dem Gebiet der hardwarenahen Programmierung."

[Modulbeschreibungen\MH28; Position: 49 - 49]

200.

"Die Absolventen vertiefen ihr Verständnis über das Zusammenwirken von Hard- und Software (Darstellung von Informationen, little / big endian Bytereihenfolge, Speichermodell, Parameterübergabe)."

[Modulbeschreibungen\MH28; Position: 50 - 50]

201.

"Die Absolventen verfügen über umfassender Fähigkeiten und Fertigkeiten bei der sicheren Nutzung dieser Programmiersprache für hardware- und systemnahe Anwendungen."

[Modulbeschreibungen\MH28; Position: 52 - 52]

202.

"Die Absolventen verfügen über Fähigkeiten und Fertigkeiten im Umgang mit Werkzeugen zur Programmentwicklung (Compiler, Debugger, Code-Analysator splint, Versionsverwaltung, Projektverwaltung)."

[Modulbeschreibungen\MH28; Position: 56 - 56]

203.

"Die Studierenden besitzen Verständnis innerhalb der Zahlensysteme und Rechenoperationen."

[Modulbeschreibungen\MH30; Position: 5 - 5]

204.

"In einfache Codierungen können sich die Studierenden hineindenken."

[Modulbeschreibungen\MH30; Position: 7 - 7]

205.

"Die Studierenden beherrschen einfache und strukturierte Datentypen sowie Kontrollstrukturen und das Prozedurkonzept von Java."

[Modulbeschreibungen\MH30; Position: 11 - 11]

206.

"[Die Studierenden] sind in der Lage, typische Fragestellungen, Probleme und Aufgaben diesbezüglich zu bearbeiten."

[Modulbeschreibungen\MH30; Position: 12 - 12]

207.

"Auch das Arbeiten mit einer Programmierumgebung ist [den Studierenden] geläufig."

[Modulbeschreibungen\MH30; Position: 15 - 15]

208.

"[Die Studenten] haben einen Überblick über den Aufbau von Betriebssystemen erhalten."

[Modulbeschreibungen\MH30; Position: 18 - 18]

209.

"[Die Studierenden] beherrschen die wichtigsten Unix-Befehle."

[Modulbeschreibungen\MH30; Position: 19 - 19]

210.

"[Die Studierenden] können [...] [die wichtigsten Konzepte der objektorientierten Programmierung] anwenden."

[Modulbeschreibungen\MH30; Position: 23 - 23]

211.

"Die Studierenden sind in der Lage, sich in die objektorientierte Programmierung in C++ hineinzudenken."

[Modulbeschreibungen\MH30; Position: 32 - 32]

212.

"Des Weiteren beherrschen [die Studierenden] weiterführende Techniken der objektorientierten Programmierung."

[Modulbeschreibungen\MH30; Position: 34 - 34]

213.

"[Die Studierenden] verfügen über Kenntnisse von objektorientierten Werkzeugen."

[Modulbeschreibungen\MH30; Position: 35 - 35]

214.

"[Die Studierenden] [...] [können objektorientierte Werkzeuge] anwenden."

[Modulbeschreibungen\MH30; Position: 36 - 36]

215.

"Erwerb von Grundkenntnissen über die Konzepte der Informatik."

[Modulbeschreibungen\MH31; Position: 5 - 5]

216.

"Vertrautheit mit der informatischen Denkweise beim Problemlösen."

[Modulbeschreibungen\MH31; Position: 8 - 8]

217.

"Erwerb von Grundkenntnissen über die Konzepte der Informatik."

[Modulbeschreibungen\MH31; Position: 10 - 10]

218.

"Vertrautheit mit der informatischen Denkweise beim Problemlösen."

[Modulbeschreibungen\MH31; Position: 13 - 13]

219.

"Schaffung der methodischen Grundlagen zur Umsetzung realweltlicher Problemstellungen in komplexe Softwaresysteme."

[Modulbeschreibungen\MH31; Position: 15 - 15]

220.

"Schaffung eines Grundverständnisses für die Modellierung."

[Modulbeschreibungen\MH31; Position: 16 - 16]

221.

"Grundverständnis für Programmierparadigmen."

[Modulbeschreibungen\MH31; Position: 21 - 21]

222.

"[Studierende sollen am Bsp. der Programmiersprache JAVA die folgenden Aspekte einer imperativen/prozeduralen Programmiersprache beherrschen:] Verständnis des Unterschieds zwischen Werte- und Referenzsemantik."

[Modulbeschreibungen\MH32; Position: 12 - 12]

223.

"[Studierende sollen am Bsp. der Programmiersprache JAVA die folgenden Aspekte einer imperativen/prozeduralen Programmiersprache beherrschen:] Entwicklung eines praktisch verinnerlichten Algorithmusbegriffs am Beispiel einfacher Sortieralgorithmen (Bubble Sort, BinSort)."

[Modulbeschreibungen\MH32; Position: 19 - 19]

224.

"Studierende sollen am Bsp. der Programmiersprache JAVA die folgenden Aspekte einer objektorientierten Programmiersprache beherrschen: Datenkapselung."

[Modulbeschreibungen\MH32; Position: 20 - 20]

225.

"[Studierende sollen am Bsp. der Programmiersprache JAVA die folgenden Aspekte einer objektorientierten Programmiersprache beherrschen:] Polymorphie."

[Modulbeschreibungen\MH32; Position: 21 - 21]

226.

"[Studierende sollen am Bsp. der Programmiersprache JAVA die folgenden Aspekte einer objektorientierten Programmiersprache beherrschen:] Vererbung."

[Modulbeschreibungen\MH32; Position: 22 - 22]

227.

"[Studierende sollen am Bsp. der Programmiersprache JAVA die folgenden Aspekte einer objektorientierten Programmiersprache beherrschen:] Objektkommunikation."

[Modulbeschreibungen\MH32; Position: 23 - 23]

228.

"Dieses erste Informatik- Modul soll den Studierenden die für die Informatik-relevanten Teilgebiete der diskreten Mathematik nahebringen."

[Modulbeschreibungen\MH32; Position: 27 - 27]

229.

"[Die Studierenden] beherrschen die grundlegenden Konzepte und Methoden der Kombinatorik und der Wahrscheinlichkeitsrechnung."

[Modulbeschreibungen\MH32; Position: 31 - 31]

230.

"[Die Studierenden] verstehen den Begriff der Komplexität von Algorithmen und beherrschen die Methoden zu deren Bestimmung."

[Modulbeschreibungen\MH32; Position: 33 - 33]

231.

"[Studierende sollen am Bsp. der Programmiersprache JAVA die folgenden Aspekte einer objektorientierten Programmiersprache beherrschen:] Verständnis was Unit Testing und Code Coverage ist."

[Modulbeschreibungen\MH32; Position: 40 - 40]

232.

"Sie beherrschen die notwendigen theoretischen Grundlagen: Abstrakte Datentypen, Berechenbarkeit sowie Algorithmenparadigmen."

[Modulbeschreibungen\MH32; Position: 46 - 46]

233.

"Die Studierenden sind in der Lage, Algorithmen und Datenstrukturen zu verstehen."

[Modulbeschreibungen\MH32; Position: 47 - 47]

234.

"Tiefgehendes Verständnis des Algorithmusbegriffs."

[Modulbeschreibungen\MH33; Position: 5 - 5]

235.

"Tiefgehendes Verständnis der Grundlagen imperativer und objektorientierter Programmierung."

[Modulbeschreibungen\MH33; Position: 7 - 7]

236.

"Tiefgehendes Verständnis von Syntax und Semantik von Programmierfragen."

[Modulbeschreibungen\MH33; Position: 8 - 8]

237.

"Verständnis grundlegender Algorithmen."

[Modulbeschreibungen\MH33; Position: 17 - 17]

238.

"Verständnis über elementare Datenstrukturen."

[Modulbeschreibungen\MH33; Position: 19 - 19]

239.

"Beherrschen grundlegender Prinzipien und Methoden für Entwurf von Algorithmen."

[Modulbeschreibungen\MH33; Position: 21 - 21]

240.

"[Die Studierenden] können syntaktische und semantische Beschreibungen von Programmiersprachen verstehen."

[Modulbeschreibungen\MH33; Position: 27 - 27]

241.

"[Die Studierenden können] Konzepte objektorientierter Programmierung verstehen."

[Modulbeschreibungen\MH34; Position: 34 - 34]

242.

"[Fachkompetenz:] [Die Studierenden] sind mit grundlegenden Datenstrukturen der Informatik vertraut."

[Modulbeschreibungen\MH35; Position: 6 - 6]

243.

"Systemkompetenz: Die Studierenden verstehen die Wirkungsweise von Standardalgorithmen und –datenstrukturen."

[Modulbeschreibungen\MH35; Position: 12 - 12]

244.

"Die Studierenden erwerben Basiswissen über Programmiersprachparadigmen, einschließlich der zugrundeliegenden Denk- und Verarbeitungsmodelle."

[Modulbeschreibungen\MH35; Position: 19 - 19]

245.

"Die Studierenden verstehen verschiedene Programmiersprachkonzepte im Kontext einer Programmiersprache (Systemkompetenz)."

[Modulbeschreibungen\MH35; Position: 24 - 24]

3. nicht-kognitive Kompetenzen

3.1. Sozial-kommunikative Fähigkeiten

3.1.1. Kooperieren und kollaborieren

1.

"Die Studierenden können eine größere Software in einem kleinen Team entwickeln."

[Modulbeschreibungen\MH02; Position: 27 - 27]

2.

"Es soll die Fähigkeit entwickelt werden, in einem kleinen Team ein größeres Softwareprojekt erfolgreich durchzuführen."

[Modulbeschreibungen\MH03; Position: 27 - 27]

3.

"In Gruppen Probleme analysieren."

[Modulbeschreibungen\MH10; Position: 24 - 24]

4.

"[In Gruppen] gemeinsam Lösungsstrategien entwickeln und präsentieren können."

[Modulbeschreibungen\MH10; Position: 25 - 25]

5.

"In Gruppen Probleme analysieren."

[Modulbeschreibungen\MH10; Position: 37 - 37]

6.

"[In Gruppen] gemeinsam Lösungsstrategien entwickeln und präsentieren können."

[Modulbeschreibungen\MH10; Position: 38 - 38]

7.

"In Gruppen Probleme analysieren."

[Modulbeschreibungen\MH10; Position: 46 - 46]

8.

"[In Gruppen] gemeinsam Lösungsstrategien entwickeln und präsentieren können."

[Modulbeschreibungen\MH10; Position: 47 - 47]

9.

"Ein weiteres wesentliches Ziel ist das Erlernen der Teamkompetenz, um später größere Implementierungsaufgaben in der Gruppe lösen zu können (systemische und kommunikative Kompetenz)."

[Modulbeschreibungen\MH16; Position: 23 - 23]

10.

"Komplexe Problemlösungen sollen im Team erarbeitet und implementiert werden (systemische und kommunikative Kompetenz)."

[Modulbeschreibungen\MH16; Position: 34 - 34]

11.

"[Zur komplexen Problemlösung im Team] gehören die Strukturierung, die Schnittstellendefinition, die Implementierung sowie ihre Verifikation unter Benutzung von Entwicklungsumgebungen und die Erstellung einer angemessenen Dokumentation (instrumentale Kompetenz)."

[Modulbeschreibungen\MH16; Position: 35 - 35]

12.

"Selbst- und Sozialkompetenz: Gemeinsame Bearbeitung programmiertechnischer Aufgaben in Kleingruppen."

[Modulbeschreibungen\MH19; Position: 9 - 9]

13.

"[Design- und Realisierungskompetenzen:] Erstellen eines größeren Software-Programms im Team."

[Modulbeschreibungen\MH20; Position: 52 - 52]

14.

"Soziale Kompetenzen: Teamarbeit."

[Modulbeschreibungen\MH20; Position: 55 - 55]

15.

"[Kompetenzen:] Das eigenverantwortliche Lösen von Aufgaben und die Präsentation der Ergebnisse steigert in diesem Zusammenhang die soziale Kompetenz der Studierenden."

[Modulbeschreibungen\MH22; Position: 33 - 33]

16.

"Im Rahmen der praktischen Übungen erlernen die Studierenden die Zusammenarbeit in Kleingruppen."

[Modulbeschreibungen\MH26; Position: 29 - 29]

17.

"[Die Lerneinheit ist darauf ausgerichtet, den Studierenden ein breites Spektrum informatischer Werkzeuge zu eröffnen, die die Teilnehmer] in Gruppenarbeit einsetzen (Befähigung Teamkompetenz)"

[Modulbeschreibungen\MH28; Position: 46 - 46]

18.

"Sozialkompetenz: Die Studierenden erarbeiten Lösungen zu einfachen Programmieraufgaben [...] in der Gruppe."

[Modulbeschreibungen\MH35; Position: 15 - 15]

19.

"[Sozialkompetenz:] [Die Studierenden] können [...] [erarbeitete Lösungen zu einfachen Programmieraufgaben] in der Gruppe analysieren."

[Modulbeschreibungen\MH35; Position: 16 - 16]

20.

"[Sozialkompetenz:] [Die Studierenden] [können] [erarbeitete Lösungen zu einfachen Programmieraufgaben] [in der Gruppe] bewerten."

[Modulbeschreibungen\MH35; Position: 17 - 17]

21.

"Die Studierenden können erarbeitete Lösungen einfacher Programmieraufgaben in der Gruppe analysieren (Sozialkompetenz)."

[Modulbeschreibungen\MH35; Position: 25 - 25]

22.

"[Die Studierenden können erarbeitete Lösungen einfacher Programmieraufgaben in der Gruppe] bewerten (Sozialkompetenz)."

[Modulbeschreibungen\MH35; Position: 26 - 26]

3.1.2. Kommunizieren

1.

"[Die Teilnehmer] sind in der Lage, Lösungen zu rechtfertigen."

[Modulbeschreibungen\MH12; Position: 15 - 15]

2.

"Kommunikative Kompetenzen werden durch Arbeiten in Gruppenübungen und die dortige Vorstellung und Diskussion von Übungsaufgaben erworben."

[Modulbeschreibungen\MH16; Position: 11 - 11]

3.

"In den Übungsgruppen werden Teilnehmerinnen und Teilnehmer Lösungen präsentieren, bzw. im Dialog erarbeiten (kommunikative Kompetenz)."

[Modulbeschreibungen\MH16; Position: 18 - 18]

4.

"Lösungen zu Übungsaufgaben werden in Kleingruppen präsentiert bzw. im Dialog erarbeitet (kommunikative Kompetenz)."

[Modulbeschreibungen\MH16; Position: 31 - 31]

5.

"[Selbst- und Sozialkompetenz:] Fähigkeit zur Argumentation über die Qualität einer programmiertechnischen Lösung."

[Modulbeschreibungen\MH19; Position: 21 - 21]

6.

"[Soziale Kompetenzen:] Kommunikation."

[Modulbeschreibungen\MH20; Position: 56 - 56]

7.

"Sozialkompetenzen: Die Studierenden vermitteln die Struktur und Wirkungsweise selbst entwickelter Programme an andere."

[Modulbeschreibungen\MH21; Position: 27 - 27]

8.

"[Sozialkompetenzen:] [Die Studierenden] präsentieren Lösungen zu kleinen Aufgaben vor Gruppen."

[Modulbeschreibungen\MH21; Position: 28 - 28]

9.

"Sozialkompetenzen: Die Studierenden vermitteln die Struktur und Wirkungsweise selbst entwickelter Modelle und Programme an andere."

[Modulbeschreibungen\MH21; Position: 50 - 50]

10.

"[Sozialkompetenzen:] [Die Studierenden] präsentieren selbstständig entwickelte Lösungen vor Gruppen."

[Modulbeschreibungen\MH21; Position: 51 - 51]

11.

"Kompetenzen: Ziel des Praktikums ist neben dem grundlegenden Erwerb von Fachwissen auch die Befähigung der Studierenden, im beruflichen Umfeld kompetent kommunizieren zu können."

[Modulbeschreibungen\MH22; Position: 32 - 32]

12.

"Die von den Studenten bearbeiteten Themenstellung und Lösungswege werden von diesen in dem begleitenden Seminar vorgestellt und in der Gruppe diskutiert."

[Modulbeschreibungen\MH28; Position: 34 - 34]

13.

"Beim Studium der empfohlenen Fachliteratur verbessern die Absolventen ihre Englischkenntnisse."

[Modulbeschreibungen\MH28; Position: 55 - 55]

3.1.3. Organisation von Teamarbeit und Projekten (Projektmanagement)

1.

"[Selbst- und Sozialkompetenz:] Organisation von Teamarbeit."

[Modulbeschreibungen\MH19; Position: 31 - 31]

2.

"Projektmanagementkompetenz: Fähigkeit zur Organisation und Steuerung von Projekten."

[Modulbeschreibungen\MH20; Position: 54 - 54]

3.2. Programmiererfahrung sammeln

1.

"In der Übung wenden [die Studierenden] Ihre erlangten Kenntnisse anhand verschiedener Aufgaben an."

[Modulbeschreibungen\MH01; Position: 26 - 26]

2.

"Anhand einer Reihe von Übungsaufgaben wird das erlernte Wissen praktisch erprobt."

[Modulbeschreibungen\MH01; Position: 31 - 31]

3.

"Das Softwareentwicklungspraktikum vermittelt praktische Erfahrung in der teamorientierten Entwicklung eines größeren und komplexen Software-Systems unter Benutzung marktüblicher Werkzeuge und Methoden."

[Modulbeschreibungen\MH03; Position: 26 - 26]

4.

"Praktische Erfahrungen beim Einsatz von Algorithmen mit Java sammeln."

[Modulbeschreibungen\MH04; Position: 41 - 41]

5.

"Die Studierenden entwickeln durch die praktischen Übungen erste Anwendungs-, Analyse-, Problemlöse- und Methodenkompetenzen in der objektorientierten Programmierung."

[Modulbeschreibungen\MH07; Position: 21 - 21]

6.

"Praktische Programmiererfahrung."

[Modulbeschreibungen\MH27; Position: 14 - 14]

7.

"Vermittlung praktischer Erfahrungen in der modellgetriebenen Systementwicklung."

[Modulbeschreibungen\MH31; Position: 19 - 19]

8.

"[Nach Durcharbeiten des Moduls sollen die Lernenden in der Lage sein, für die meisten typischerweise in der Informatik auftretenden Probleme und Fragestellungen] schnell zu den entsprechenden Lösungen zu gelangen."

[Modulbeschreibungen\MH32; Position: 36 - 36]

9.

"Anwendungserfahrung grundlegender Algorithmen."

[Modulbeschreibungen\MH33; Position: 18 - 18]

10.

"Anwendungserfahrung über elementare Datenstrukturen."

[Modulbeschreibungen\MH33; Position: 20 - 20]

3.3. Sensibilisierung für IT-Sicherheit (affektiv)

1.

"[Die Studierenden] sollen weiterhin für Sicherheitsprobleme sensibilisiert sein."

[Modulbeschreibungen\MH16; Position: 16 - 16]

4. Wissensdimensionen⁴

4.1. Faktenwissen

1.

"[Die Studierenden] ordnen Literale der Sprache C sicher den entsprechenden Datentypen zu."

[Modulbeschreibungen\MH34; Position: 7 - 7]

4.2. Konzeptionelles Wissen

1.

"[Die Studierenden] klassifizieren formale Sprachen mit Hilfe der sog. Chomsky-Hierarchie."

[Modulbeschreibungen\MH01; Position: 7 - 7]

2.

"[Weiterhin] können [die Studierenden] mit exakten logischen Argumenten [...] Grenzen [der Berechnungskraft gängiger Rechnermodelle durch endliche Automaten] aufzeigen."

[Modulbeschreibungen\MH01; Position: 10 - 10]

3.

"[Bei erfolgreichem Abschluss des Moduls sollten die Teilnehmer(innen) Kenntnisse über] wesentliche Komplexitätsklassen für das Laufzeitverhalten und den Speicherplatzbedarf von Algorithmen [erlangt haben]."

[Modulbeschreibungen\MH03; Position: 17 - 17]

4.

"Kenntnis des Umfangs typischer Funktionsbibliotheken für Algorithmen in Programmiersprachen."

[Modulbeschreibungen\MH04; Position: 39 - 39]

5.

"Die Studentinnen und Studenten kennen die Grundbegriffe zu Algorithmen."

[Modulbeschreibungen\MH05; Position: 16 - 16]

⁴ Unter den nachfolgenden Überschriften werden alle Segmente der gleichnamigen Kodierung aufgelistet. Da kognitive Kompetenzen doppelt kodiert wurden (einmal pro Wissensdimension und einmal pro kognitiver Prozessdimension) finden sich alle kodierten Segmente genau zwei Mal im Sample. (Eine Ausnahme stellt die Kodierung der als „nicht operationalisierten“ Lehr- und Lernziele dar. Diese sind nur einmalig kodiert und demnach keiner zweiten Kategorie hinzugefügt.)

6.

"[Die Studentinnen und Studenten] können [...] [die Begriffe Laufzeit und Korrektheit] einsetzen."

[Modulbeschreibungen\MH05; Position: 28 - 28]

7.

"Studentinnen und Studenten können die Grundbegriffe der Algorithmik definieren."

[Modulbeschreibungen\MH05; Position: 31 - 31]

8.

"[Studentinnen und Studenten] kennen die Definition [...] von NP-Vollständigkeit für die effiziente Lösbarkeit von Problemen."

[Modulbeschreibungen\MH05; Position: 40 - 40]

9.

"Kenntnis des Algorithmusbegriffs, von Merkmalen von Algorithmen und Grenzen der Algorithmisierung."

[Modulbeschreibungen\MH08; Position: 5 - 5]

10.

"Churchsche These kennen."

[Modulbeschreibungen\MH08; Position: 7 - 7]

11.

"Kenntnis der Effizienzmaße auf Parallelrechner-systemen und von effizienten parallelen Algorithmen."

[Modulbeschreibungen\MH08; Position: 24 - 24]

12.

"Grundbegriffe der formalen Verifikation erläutern können."

[Modulbeschreibungen\MH10; Position: 35 - 35]

13.

"[Die Studierenden] kennen einige grundlegende Klassen der Java-Bibliothek."

[Modulbeschreibungen\MH13; Position: 11 - 11]

14.

"[Die Studierenden] kennen grafische Benutzerschnittstellen."

[Modulbeschreibungen\MH13; Position: 19 - 19]

15.

"[In dieser Veranstaltung lernen Studierende] die Komplexitätsklassen P, NP und NPC kennen."

[Modulbeschreibungen\MH17; Position: 13 - 13]

16.

"Fachkompetenzen: Die Studierenden beschreiben grundlegende Konzepte der imperativen Programmierung mit Java."

[Modulbeschreibungen\MH21; Position: 10 - 10]

17.

"[Fachkompetenzen:] [Die Studierenden] verwenden die [...] Begriffe [der imperativen Programmierung] präzise bei Diskussionen."

[Modulbeschreibungen\MH21; Position: 12 - 12]

18.

"[Fachkompetenzen:] [Die Studierenden] benennen Probleme der Effizienz von algorithmischen Lösungen konkreter Fragestellungen."

[Modulbeschreibungen\MH21; Position: 20 - 20]

19.

"[Fachkompetenzen:] [Die Studierenden] verwenden die entsprechenden Begriffe [der objektorientierten Modellierung und Programmierung] präzise bei Diskussionen."

[Modulbeschreibungen\MH21; Position: 41 - 41]

20.

"[Fähigkeiten:] Verwendung grundlegender Beschreibungsformen für Programmiersprachen."

[Modulbeschreibungen\MH22; Position: 14 - 14]

21.

"[Beim erfolgreichen Abschluss des Moduls sollten die Studierenden Kenntnisse über] [...] wesentliche Komplexitätskategorien für Laufzeit und Speicherbedarf von Algorithmen [haben]."

[Modulbeschreibungen\MH22; Position: 19 - 19]

22.

"[Studierende sollten nach der Veranstaltung in der Lage sein, Probleme] zu beschreiben."

[Modulbeschreibungen\MH24; Position: 37 - 37]

23.

"[Die Studierenden sollen algorithmische Aufgabenstellungen] formulieren können."

[Modulbeschreibungen\MH24; Position: 47 - 47]

24.

"Nach erfolgreicher Absolvierung dieses Moduls können die Studierenden fortgeschrittene Konzepte objektorientierter Programmierung in Java erklären."

[Modulbeschreibungen\MH26; Position: 18 - 18]

25.

"[Die Studierenden] können Unterschiede sowie Vor- und Nachteile zu objektorientierten und typsicheren Sprachen erläutern."

[Modulbeschreibungen\MH26; Position: 27 - 27]

26.

"Einfache Programmiersprachen formal beschreiben können."

[Modulbeschreibungen\MH27; Position: 11 - 11]

27.

"Jeder Studierende kann präzise die von ihm bearbeitete Aufgabe formulieren."

[Modulbeschreibungen\MH28; Position: 26 - 26]

28.

"Die Absolventen kennen die C-Standardbibliothek und weitere ausgewählte Bibliotheken."

[Modulbeschreibungen\MH28; Position: 53 - 53]

29.

"Die Studierenden können wichtige Programmiersprachen charakterisieren."

[Modulbeschreibungen\MH33; Position: 25 - 25]

30.

"[Die Studierenden] können [...] Anwendungsgebiete [wichtiger Programmiersprachen] gegenüberstellen."

[Modulbeschreibungen\MH33; Position: 26 - 26]

31.

"[Die Studierenden] können den Aufbau und die Prinzipien von Programmiersprachen analysieren."

[Modulbeschreibungen\MH33; Position: 30 - 30]

32.

"[Die Studierenden] [können neue Sprachkonstrukte selbstständig] einordnen."

[Modulbeschreibungen\MH33; Position: 32 - 32]

33.

"[Die Studierenden können die grundlegenden Prinzipien der OOP] mit eigenen Worten am Beispiel beschreiben."

[Modulbeschreibungen\MH34; Position: 20 - 20]

34.

"[Die Studierenden können] Anwendungsmöglichkeiten [von Konzepten wie Referenzen, abstrakte Klassen, virtuelle Methoden, Klassenkonstanten, -attribute und -methoden] aufzeigen."

[Modulbeschreibungen\MH34; Position: 22 - 22]

35.

"[Die Studierenden können] [Gemeinsamkeiten und Unterschiede hinsichtlich der Sprachphilosophie und Konzepte von Java und C/C++] erklären."

[Modulbeschreibungen\MH34; Position: 40 - 40]

36.

"[Die Studierenden] können Programmiersprachen und deren Konzepte nach wesentlichen Paradigmen klassifizieren (Fachkompetenz)."

[Modulbeschreibungen\MH35; Position: 20 - 20]

4.3. Prozedurales Wissen

1.

"Die Studierenden erlernen die prinzipiellen Beschränkungen heutiger Computer bei der Lösung von wichtigen Problemen."

[Modulbeschreibungen\MH01; Position: 5 - 5]

2.

"[Die Studierenden] erkennen [...] [durch die Klassifikation formaler Sprachen] ihre algorithmische Komplexität."

[Modulbeschreibungen\MH01; Position: 8 - 8]

3.

"Diverse Probleme erkennen [die Studierenden] von vorneherein als durch Computer unlösbare Aufgabenstellungen."

[Modulbeschreibungen\MH01; Position: 11 - 11]

4.

"Die vorgestellten Ergebnisse können die Studierenden durch den sicheren Umgang mit verschiedenen Beweistechniken belegen."

[Modulbeschreibungen\MH01; Position: 12 - 12]

5.

"Die Studierenden lernen, kleine Probleme zu analysieren."

[Modulbeschreibungen\MH01; Position: 14 - 14]

6.

"[Die Studierenden] wenden bestehende Lösungskonzepte an, um komplexere Probleme in kleinere aufzuteilen."

[Modulbeschreibungen\MH01; Position: 16 - 16]

7.

"Darüber hinaus lernen [die Studierenden] existierende Algorithmen anzuwenden."

[Modulbeschreibungen\MH01; Position: 18 - 18]

8.

"Die Studierenden lernen objekt-orientierte Techniken kennen, um Software-Entwicklungs-Projekte eigenständig zu implementieren."

[Modulbeschreibungen\MH01; Position: 20 - 20]

9.

"[Die Studierenden] nutzen weitergehende Analyse-, Design- und Realisierungskompetenzen, um das Zusammenspiel und die architektonische Organisation vieler Klassen in einem Projekt anhand der Programmiersprache Java sowie der UML zu formulieren."

[Modulbeschreibungen\MH01; Position: 21 - 21]

10.

"Weiterhin können [die Studierenden] abschätzen, in welcher Situation bestimmte komplexe Datentypen eingesetzt werden."

[Modulbeschreibungen\MH01; Position: 22 - 22]

11.

"Weiterhin können [die Studierenden] abschätzen, wie [...] [komplexe Datentypen] funktionieren."

[Modulbeschreibungen\MH01; Position: 23 - 23]

12.

"Weiterhin können [die Studierenden] abschätzen, welchen Laufzeitaufwand [...] [komplexe Datentypen] besitzen."

[Modulbeschreibungen\MH01; Position: 24 - 24]

13.

"Die Studierenden sind in der Lage die verschiedenen Sprachmerkmale (imperative, objekt-orientiert) einzuordnen."

[Modulbeschreibungen\MH01; Position: 29 - 29]

14.

"[Die Studierenden sind in der Lage die verschiedenen Sprachmerkmale (imperative, objekt-orientiert)] je nach Anwendungsfall geeignet einzusetzen."

[Modulbeschreibungen\MH01; Position: 30 - 30]

15.

"[Die Teilnehmer] sind in der Lage, kleine Programme (bis zu einigen hundert Zeilen) zu analysieren."

[Modulbeschreibungen\MH02; Position: 7 - 7]

16.

"[Die Teilnehmer] [sind in der Lage, kleine Programme (bis zu einigen hundert Zeilen)] selbst zu konzipieren."

[Modulbeschreibungen\MH02; Position: 8 - 8]

17.

"[Die Teilnehmer] [sind in der Lage, kleine Programme (bis zu einigen hundert Zeilen)] zu implementieren."

[Modulbeschreibungen\MH02; Position: 9 - 9]

18.

"[Die Teilnehmer] kennen die Möglichkeiten, Daten- und Ablaufstrukturen zu entwerfen."

[Modulbeschreibungen\MH02; Position: 10 - 10]

19.

"[Die Teilnehmer] [kennen die Möglichkeiten, Daten- und Ablaufstrukturen] zu beschreiben."

[Modulbeschreibungen\MH02; Position: 11 - 11]

20.

"[Die Teilnehmer] [kennen die Möglichkeiten, Daten- und Ablaufstrukturen] zu codieren."

[Modulbeschreibungen\MH02; Position: 12 - 12]

21.

"[Die Teilnehmer] kennen die Techniken und Notationen zur Definition kontextfreier Programmiersprachen."

[Modulbeschreibungen\MH02; Position: 14 - 14]

22.

"[Die Teilnehmer] können mit den Techniken und Notationen zur Definition kontextfreier Programmiersprachen [...] arbeiten."

[Modulbeschreibungen\MH02; Position: 15 - 15]

23.

"Die Studierenden kennen nach engagierter Mitarbeit in dieser Veranstaltung diverse zentrale Algorithmen auf geeigneten Datenstrukturen, die für eine effiziente Nutzung von Computern unverzichtbar sind."

[Modulbeschreibungen\MH02; Position: 17 - 17]

24.

"[Die Studierenden] können am Ende zu gängigen Problemen geeignete programmiersprachliche Lösungen angeben."

[Modulbeschreibungen\MH02; Position: 18 - 18]

25.

"[Die Studierenden] [können am Ende zu gängigen Problemen geeignete programmiersprachliche Lösungen] [...] in einer konkreten Programmiersprache formulieren."

[Modulbeschreibungen\MH02; Position: 19 - 19]

26.

"Kenntnis der Eigenschaften elementarer und häufig benötigter Algorithmen."

[Modulbeschreibungen\MH02; Position: 20 - 20]

27.

"Erweiterung der Kompetenz im Entwurf von Algorithmen."

[Modulbeschreibungen\MH02; Position: 22 - 22]

28.

"Erweiterung der Kompetenz im Entwurf von [...] zu Algorithmen zugehörigen Datenstrukturen."

[Modulbeschreibungen\MH02; Position: 23 - 23]

29.

"Die Studierenden werden in die Lage versetzt, Lösungen für kleinere und überschaubare Probleme algorithmisch umzusetzen."

[Modulbeschreibungen\MH03; Position: 5 - 5]

30.

"[Die Studierenden werden in die Lage versetzt, Lösungen für kleinere und überschaubare Probleme] mit einer höheren Programmiersprache als ausführbare Programme zu realisieren."

[Modulbeschreibungen\MH03; Position: 6 - 6]

31.

"Die Benutzung einer Entwicklungsumgebung wie Eclipse fördert die Professionalisierung."

[Modulbeschreibungen\MH03; Position: 7 - 7]

32.

"Fähigkeit, kleine Algorithmen funktional zu programmieren."

[Modulbeschreibungen\MH03; Position: 11 - 11]

33.

"[Fähigkeit, kleine Algorithmen] [...] im Vergleich mit imperativen Lösungen zu bewerten."

[Modulbeschreibungen\MH03; Position: 12 - 12]

34.

"Bei erfolgreichem Abschluss des Moduls sollten die Teilnehmer(innen) Kenntnisse über grundlegende Eigenschaften und Entwurfmethoden für Algorithmen [erlangt haben]."

[Modulbeschreibungen\MH03; Position: 15 - 15]

35.

"[Bei erfolgreichem Abschluss des Moduls sollten die Teilnehmer(innen) Kenntnisse über] effiziente Algorithmen und Datenstrukturen für grundlegende Probleme [erlangt haben]."

[Modulbeschreibungen\MH03; Position: 16 - 16]

36.

"Teilnehmer(innen) des Moduls erlernen die Fähigkeit zur Analyse des Laufzeitverhaltens und des Speicherplatzbedarfs für gegebene Algorithmen."

[Modulbeschreibungen\MH03; Position: 18 - 18]

37.

"[Teilnehmer(innen) des Moduls erlernen die Fähigkeit zur] formalen Modellierung von Algorithmen Problemstellungen."

[Modulbeschreibungen\MH03; Position: 19 - 19]

38.

"[Teilnehmer(innen) des Moduls erlernen die Fähigkeit zur] Adaption bekannter Datenstrukturen an modifizierte Problemstellungen."

[Modulbeschreibungen\MH03; Position: 20 - 20]

39.

"[Teilnehmer(innen) des Moduls erlernen die Fähigkeit zur] Adaption bekannter Algorithmen an modifizierte Problemstellungen."

[Modulbeschreibungen\MH03; Position: 21 - 21]

40.

"Unter Anwendung der erlangten Fähigkeiten und Kenntnisse sollen die Teilnehmer in die Lage versetzt werden, Programme unter Verwendung der erlernten algorithmischen Techniken eigenständig zu konzipieren."

[Modulbeschreibungen\MH03; Position: 22 - 22]

41.

"[Unter Anwendung der erlangten Fähigkeiten und Kenntnisse sollen die Teilnehmer in die Lage versetzt werden, Programme unter Verwendung der erlernten algorithmischen Techniken] in einer Programmiersprache zu implementieren."

[Modulbeschreibungen\MH03; Position: 23 - 23]

42.

"[Unter Anwendung der erlangten Fähigkeiten und Kenntnisse sollen die Teilnehmer in die Lage versetzt werden,] für ein gegebenes Problem die Verwendung verschiedener Lösungsmöglichkeiten nach formalen Kriterien zu beurteilen."

[Modulbeschreibungen\MH03; Position: 24 - 24]

43.

"[Die Studierenden] [wenden] Notationen und Werkzeugen gemäß des Stands der Forschung und der industriellen Praxis [an]."

[Modulbeschreibungen\MH03; Position: 34 - 34]

44.

"[Die Studierenden] sind in der Lage, strukturelle und dynamische Eigenschaften komplexer Softwaresysteme zu modellieren."

[Modulbeschreibungen\MH03; Position: 35 - 35]

45.

"[Die Studierenden] sind in der Lage, die Modelle [struktureller und dynamischer Eigenschaften komplexer Softwaresysteme] in Software zu überführen."

[Modulbeschreibungen\MH03; Position: 36 - 36]

46.

"Die Studierenden entwickeln zu Beginn ihrer Ausbildung ein Verständnis für Stilistik und Ästhetik der Programmierung."

[Modulbeschreibungen\MH04; Position: 5 - 5]

47.

"Die Studierenden erwerben Kenntnisse der grundlegenden Techniken zur algorithmischen Problemlösung."

[Modulbeschreibungen\MH04; Position: 6 - 6]

48.

"Die Studierenden lernen aus dem Kanon der Grundlagen der Informatik die prozedurale Programmierung kennen sowie einführend auch Grundzüge der Objektorientierung."

[Modulbeschreibungen\MH04; Position: 11 - 11]

49.

"Die Studierenden sind in der Lage eigenständig eine Lösungsstrategie zum Schreiben kleiner prozeduraler und objektorientierter Java-Programme nach einer vorgegebenen Entwurfsidee zu entwickeln."

[Modulbeschreibungen\MH04; Position: 12 - 12]

50.

"[Die Studierenden sind in der Lage eigenständig eine Lösungsstrategie zum Schreiben kleiner prozeduraler und objektorientierter Java-Programme nach einer vorgegebenen Entwurfsidee] umzusetzen."

[Modulbeschreibungen\MH04; Position: 13 - 13]

51.

"Um diese Lösungsstrategien umzusetzen, lernen die Studenten zunächst, wie man an einfache mathematische und technische Problemstellungen herangeht (Analyse)."

[Modulbeschreibungen\MH04; Position: 14 - 14]

52.

"[Um diese Lösungsstrategien umzusetzen, lernen die Studenten zunächst,] wie man einfache Probleme strukturiert."

[Modulbeschreibungen\MH04; Position: 15 - 15]

53.

"Für die Umsetzung lernen Studenten am Beispiel der Programmiersprache Java, wie man mit analytischem und konzeptionellem Denken die Problemstellungen zerlegt."

[Modulbeschreibungen\MH04; Position: 16 - 16]

54.

"Die Studierenden lernen aus dem Kanon der Grundlagen der Informatik die objektorientierte Programmierung kennen."

[Modulbeschreibungen\MH04; Position: 19 - 19]

55.

"Sie erwerben die Fertigkeit zur Analyse technischer Problemstellungen an größeren geeigneten Beispielen."

[Modulbeschreibungen\MH04; Position: 21 - 21]

56.

"[Sie erwerben die Fertigkeit] zur Strukturierung technischer Problemstellungen an größeren geeigneten Beispielen."

[Modulbeschreibungen\MH04; Position: 22 - 22]

57.

"Darüber hinaus erlernen die Studierenden das eigenständige Schreiben objektorientierter Java-Programme mit einer geeigneten Erstellungsumgebung (z.B. Eclipse)."

[Modulbeschreibungen\MH04; Position: 23 - 23]

58.

"[Darüber hinaus erlernen die Studierenden das eigenständige] Testen objektorientierter Java-Programme mit einer geeigneten Erstellungsumgebung (z.B. Eclipse)."

[Modulbeschreibungen\MH04; Position: 24 - 24]

59.

"Datenstrukturen benennen."

[Modulbeschreibungen\MH04; Position: 28 - 28]

60.

"[Datenstrukturen] bzgl. ihrer Leistungsmerkmale charakterisieren können."

[Modulbeschreibungen\MH04; Position: 29 - 29]

61.

"Such- und Sortierv Verfahren benennen."

[Modulbeschreibungen\MH04; Position: 30 - 30]

62.

"[Such- und Sortierv Verfahren] einsetzen."

[Modulbeschreibungen\MH04; Position: 31 - 31]

63.

"[Such- und Sortierv Verfahren] bzgl. ihrer Leistung und Anwendbarkeit charakterisieren können."

[Modulbeschreibungen\MH04; Position: 32 - 32]

64.

"Für vorgegebene Anwendungsfälle geeignete Datenstrukturen und Algorithmen finden."

[Modulbeschreibungen\MH04; Position: 33 - 33]

65.

"[Für vorgegebene Anwendungsfälle geeignete Datenstrukturen] analysieren."

[Modulbeschreibungen\MH04; Position: 34 - 34]

66.

"[Für vorgegebene Anwendungsfälle geeignete Algorithmen] analysieren."

[Modulbeschreibungen\MH04; Position: 35 - 35]

67.

"[Für vorgegebene Anwendungsfälle geeignete Datenstrukturen und Algorithmen] bewerten können."

[Modulbeschreibungen\MH04; Position: 36 - 36]

68.

"Algorithmen entwickeln können."

[Modulbeschreibungen\MH04; Position: 37 - 37]

69.

"[Algorithmen] implementieren können."

[Modulbeschreibungen\MH04; Position: 38 - 38]

70.

"Befähigung zum Einsatz typischer Funktionsbibliotheken für Algorithmen in Programmiersprachen."

[Modulbeschreibungen\MH04; Position: 40 - 40]

71.

"Grundlagen Informatik: Die Studierenden lernen grundlegende Algorithmen und Datenstrukturen kennen."

[Modulbeschreibungen\MH04; Position: 42 - 42]

72.

"[Fertigkeit zur Analyse und Strukturierung technischer Problemstellungen: Die Studierenden lernen, wie sie für vorgegebene Anwendungsfälle geeignete Datenstrukturen und Algorithmen] bzgl. ihrer Leistung analysieren."

[Modulbeschreibungen\MH04; Position: 44 - 44]

73.

"Fertigkeit zur Entwicklung und zum Umsetzen von Lösungsstrategien: Die Studierenden lernen, für praktische Problemstellungen algorithmische Lösungen zu entwickeln."

[Modulbeschreibungen\MH04; Position: 45 - 45]

74.

"[Fertigkeit zur Entwicklung und zum Umsetzen von Lösungsstrategien: Die Studierenden lernen, für praktische Problemstellungen] vorhandene Algorithmen einzusetzen."

[Modulbeschreibungen\MH04; Position: 46 - 46]

75.

"Kenntnisse von praxisrelevanten Aufgabenstellungen: Anhand praktischer Beispiele werden die Einsatzszenarien für verschiedene Algorithmen erarbeitet."

[Modulbeschreibungen\MH04; Position: 48 - 48]

76.

"Die Studierenden kennen die Grundlagen der professionellen Softwareentwicklung und des Konfigurationsmanagements (ALM)."

[Modulbeschreibungen\MH04; Position: 50 - 50]

77.

"[Die Studierenden] können Werkzeuge zur Build-Automatisierung in größeren Softwareprojekten anwenden."

[Modulbeschreibungen\MH04; Position: 51 - 51]

78.

"Die Teilnehmenden lernen den gekonnten Einsatz einer IDE (Integrated Development Environment)."

[Modulbeschreibungen\MH04; Position: 52 - 52]

79.

"Editor, Refactoring, Build, Versionsverwaltungssystem und Test der Software in einer Applikation zeigen Ihnen den aktuellen Stand moderner Entwicklungsumgebungen."

[Modulbeschreibungen\MH04; Position: 53 - 53]

80.

"[Die Studierenden] kennen die [...] nötigen Methoden und Werkzeuge für [professionellen Code, automatisierte Tests und testgetriebene Softwareentwicklung]."

[Modulbeschreibungen\MH04; Position: 55 - 55]

81.

"[Die Studierenden] kennen insbesondere die Unterschiede bei der objektorientierten Programmierung zwischen C++ und Java."

[Modulbeschreibungen\MH04; Position: 58 - 58]

82.

"[Die Studierenden] können die in C++ typischen Sprachkonstrukte erfolgreich anwenden."

[Modulbeschreibungen\MH04; Position: 59 - 59]

83.

"[Die Studentinnen und Studenten] können objektorientierte Modelle erstellen."

[Modulbeschreibungen\MH05; Position: 7 - 7]

84.

"[Die Studentinnen und Studenten] kennen grundlegende Datenstrukturen und Algorithmen."

[Modulbeschreibungen\MH05; Position: 8 - 8]

85.

"[Die Studentinnen und Studenten] sind in der Lage, abstrakte Datentypen zu spezifizieren."

[Modulbeschreibungen\MH05; Position: 9 - 9]

86.

"[Die Studentinnen und Studenten] [sind in der Lage, abstrakte Datentypen] zu implementieren."

[Modulbeschreibungen\MH05; Position: 10 - 10]

87.

"[Die Studentinnen und Studenten] kennen grundlegende Datenstrukturen und Algorithmen."

[Modulbeschreibungen\MH05; Position: 11 - 11]

88.

"[Die Studentinnen und Studenten] können die Eigenschaften von kleinen Programmen formal und informell beweisen."

[Modulbeschreibungen\MH05; Position: 12 - 12]

89.

"[Die Studentinnen und Studenten] können objektorientierte Programme implementieren."

[Modulbeschreibungen\MH05; Position: 13 - 13]

90.

"[Die Studentinnen und Studenten] [...] können Entwurfsmuster [objektorientierter Programme] problemadäquat einsetzen."

[Modulbeschreibungen\MH05; Position: 14 - 14]

91.

"[Die Studentinnen und Studenten kennen die Grundbegriffe zu] der funktionalen Programmierung."

[Modulbeschreibungen\MH05; Position: 17 - 17]

92.

"[Die Studentinnen und Studenten] können am Ende des Moduls Beschreibungen und Quelltexte elementarer Algorithmen lesen."

[Modulbeschreibungen\MH05; Position: 18 - 18]

93.

"[Die Studentinnen und Studenten] [können am Ende des Moduls] elementare Algorithmen funktional entwerfen."

[Modulbeschreibungen\MH05; Position: 20 - 20]

94.

"[Die Studentinnen und Studenten] [können am Ende des Moduls] Anforderungen an funktionale Programme spezifizieren."

[Modulbeschreibungen\MH05; Position: 21 - 21]

95.

"[Die Studentinnen und Studenten] [können am Ende des Moduls] [Beschreibungen von Programmkomponenten] [...] in eigenen Programmen benutzen."

[Modulbeschreibungen\MH05; Position: 23 - 23]

96.

"[Die Studentinnen und Studenten] [können am Ende des Moduls] einfache Programme an geänderte Anforderungen anpassen."

[Modulbeschreibungen\MH05; Position: 24 - 24]

97.

"[Die Studentinnen und Studenten] [können am Ende des Moduls] strukturierte Programme entwickeln."

[Modulbeschreibungen\MH05; Position: 25 - 25]

98.

"[Die Studentinnen und Studenten] [können am Ende des Moduls] Eigenschaften von funktionalen Programmen formal beweisen."

[Modulbeschreibungen\MH05; Position: 26 - 26]

99.

"[Studentinnen und Studenten] wissen, was ein abstrakter Datentyp ist."

[Modulbeschreibungen\MH05; Position: 32 - 32]

100.

"[Studentinnen und Studenten] kennen die wichtigsten abstrakten Datentypen und die Datenstrukturen zu deren Implementierung."

[Modulbeschreibungen\MH05; Position: 34 - 34]

101.

"[Studentinnen und Studenten] können [...] [die wichtigsten abstrakten Datentypen und die Datenstrukturen zu deren Implementierung] in Bezug auf ihre Eigenschaften beurteilen."

[Modulbeschreibungen\MH05; Position: 35 - 35]

102.

"[Studentinnen und Studenten] können [...] [die wichtigsten abstrakten Datentypen und die Datenstrukturen zu deren Implementierung] [in Bezug auf ihre Eigenschaften] geeignet auswählen."

[Modulbeschreibungen\MH05; Position: 36 - 36]

103.

"[Studentinnen und Studenten] können [...] [die wichtigsten abstrakten Datentypen und die Datenstrukturen zu deren Implementierung] [in Bezug auf ihre Eigenschaften] einsetzen."

[Modulbeschreibungen\MH05; Position: 37 - 37]

104.

"[Studentinnen und Studenten] können die Korrektheit von Algorithmen nachweisen."

[Modulbeschreibungen\MH05; Position: 38 - 38]

105.

"[Studentinnen und Studenten] können die asymptotische Laufzeit von Algorithmen bestimmen."

[Modulbeschreibungen\MH05; Position: 39 - 39]

106.

"Die Studierenden können einfache Probleme in Form von Algorithmen lösen."

[Modulbeschreibungen\MH06; Position: 5 - 5]

107.

"[Die Studierenden können] [...] [die Lösungen zu einfachen Problemen in Form von Algorithmen] als strukturierte Programme implementieren."

[Modulbeschreibungen\MH06; Position: 6 - 6]

108.

"[Die Studierenden] haben gelernt, ihre Problemlösungen [...] [in einer Programmiersprache] auszudrücken."

[Modulbeschreibungen\MH06; Position: 8 - 8]

109.

"Die Studierenden kennen wichtige Algorithmen und die zugehörigen Datenstrukturen."

[Modulbeschreibungen\MH06; Position: 11 - 11]

110.

"[Die Studierenden] können für grundlegende Aufgabenstellungen passende Algorithmen und Datenstrukturen auswählen."

[Modulbeschreibungen\MH06; Position: 12 - 12]

111.

"Die Studierenden können die Laufzeit von Algorithmen bewerten."

[Modulbeschreibungen\MH06; Position: 13 - 13]

112.

"[Die Studierenden können die Laufzeit von Algorithmen] vergleichen."

[Modulbeschreibungen\MH06; Position: 14 - 14]

113.

"[Die Studierenden können die Laufzeit von Algorithmen] in die zugehörigen Komplexitätsklassen einordnen."

[Modulbeschreibungen\MH06; Position: 15 - 15]

114.

"Die Studierenden können Probleme in Form von Klassendiagrammen formulieren."

[Modulbeschreibungen\MH06; Position: 18 - 18]

115.

"[Die Studierenden können Probleme in Form von Klassendiagrammen] [...] objektorientiert implementieren."

[Modulbeschreibungen\MH06; Position: 19 - 19]

116.

"Die Studierenden erwerben Grundkenntnisse einer objektorientierten Programmiersprache."

[Modulbeschreibungen\MH06; Position: 20 - 20]

117.

"[Die Studierenden] lernen, ihre Problemlösungen [...] [in einer objektorientierten Programmiersprache] auszudrücken."

[Modulbeschreibungen\MH06; Position: 21 - 21]

118.

"Die Studierenden können anspruchsvolle Probleme in Form von Klassendiagrammen formulieren."

[Modulbeschreibungen\MH06; Position: 24 - 24]

119.

"[Die Studierenden können anspruchsvolle Probleme in Form von Klassendiagrammen] [...] objektorientiert implementieren."

[Modulbeschreibungen\MH06; Position: 25 - 25]

120.

"Die Studierenden haben ihre Kenntnisse einer objektorientierten Programmiersprache erweitert."

[Modulbeschreibungen\MH06; Position: 26 - 26]

121.

"[Die Studierenden haben] gelernt, Problemlösungen [...] [in einer objektorientierten Programmiersprache] auszudrücken."

[Modulbeschreibungen\MH06; Position: 27 - 27]

122.

"Fähigkeit zur Implementierung von Programmen mit wahlweise textueller bzw. grafischer Benutzeroberfläche."

[Modulbeschreibungen\MH06; Position: 28 - 28]

123.

"Die Studierenden kennen Standardalgorithmen für typische Problemstellungen aus den Bereichen Suchen, Sortieren, Mustererkennung, Rekursion, Bäume und Graphen."

[Modulbeschreibungen\MH07; Position: 5 - 5]

124.

"[Die Studierenden] erwerben die Fähigkeit, Algorithmen anzuwenden."

[Modulbeschreibungen\MH07; Position: 6 - 6]

125.

"[Die Studierenden] [erwerben die Fähigkeit, Algorithmen] zu konstruieren."

[Modulbeschreibungen\MH07; Position: 7 - 7]

126.

"[Die Studierenden] [erwerben die Fähigkeit, Algorithmen] zu implementieren."

[Modulbeschreibungen\MH07; Position: 8 - 8]

127.

"[Die Studierenden] können die Leistungsfähigkeit von Algorithmen abschätzen."

[Modulbeschreibungen\MH07; Position: 9 - 9]

128.

"[Die Studierenden] [können die Leistungsfähigkeit von Algorithmen] beurteilen."

[Modulbeschreibungen\MH07; Position: 10 - 10]

129.

"[Die Studierenden] kennen die Datenstrukturen Liste, Array, verkettete Liste, Stapel, Schlange, Baum, Graph."

[Modulbeschreibungen\MH07; Position: 11 - 11]

130.

"[Die Studierenden] kennen in konkreten Problemstellungen der Informatik das nötige mathematische Handwerkszeug."

[Modulbeschreibungen\MH07; Position: 13 - 13]

131.

"[Die Studierenden] können [...] [das nötige mathematische Handwerkszeug in konkreten Problemstellungen der Informatik] anwenden."

[Modulbeschreibungen\MH07; Position: 14 - 14]

132.

"Die Studierenden kennen die Konzepte der Objektorientierung und objektorientierten Programmierung am Beispiel der Programmiersprache Java."

[Modulbeschreibungen\MH07; Position: 16 - 16]

133.

"[Die Studierenden] können [...] [die Entwurfsvorgaben in Klassendiagrammen] in Programme umsetzen."

[Modulbeschreibungen\MH07; Position: 19 - 19]

134.

"[Die Studierenden] sind in der Lage, Programme in einem guten Programmierstil zu programmieren."

[Modulbeschreibungen\MH07; Position: 20 - 20]

135.

"Die Studierenden kennen die Konzepte der Prozeduralen und objektorientierten Programmierung, wie sie in den Programmiersprache C und C++ umgesetzt sind."

[Modulbeschreibungen\MH07; Position: 23 - 23]

136.

"[Die Studierenden] sind in der Lage, selbständig Programme in C und C++ zu schreiben."

[Modulbeschreibungen\MH07; Position: 26 - 26]

137.

"[Die Studierenden] sind in der Lage fremde Programme zu lesen."

[Modulbeschreibungen\MH07; Position: 27 - 27]

138.

"[Die Studierenden] [sind in der Lage fremde Programme] zu modifizieren."

[Modulbeschreibungen\MH07; Position: 28 - 28]

139.

"Die Studierenden beherrschen die Techniken: Projektorganisation, Editieren, Debuggen und Fehlersuche."

[Modulbeschreibungen\MH07; Position: 29 - 29]

140.

"[Die Studierenden] können [...] [die Techniken: Projektorganisation, Editieren, Debuggen und Fehlersuche] beispielhaft in der integrierten Entwicklungsumgebung Visual Studio anwenden."

[Modulbeschreibungen\MH07; Position: 30 - 30]

141.

"Einfache Algorithmen in einer halbformalen Notation erstellen können."

[Modulbeschreibungen\MH08; Position: 6 - 6]

142.

"Einfache Algorithmen in Programme funktionaler und imperativer Notation (z.B. Python) umsetzen können."

[Modulbeschreibungen\MH08; Position: 8 - 8]

143.

"Funktionale Spezifikationen zu einfachen Problemen angeben können."

[Modulbeschreibungen\MH08; Position: 9 - 9]

144.

"Elementare Datentypen und Datentypkonstruktoren mit ihren mathematischen Konzepten beschreiben."

[Modulbeschreibungen\MH08; Position: 10 - 10]

145.

"Wichtige Datenstrukturen (z.B. Sequenz, Baum, File) in Programmiersprachen (z.B. Python) definieren können."

[Modulbeschreibungen\MH08; Position: 11 - 11]

146.

"Grundprinzipien funktionaler Programmierung kennen."

[Modulbeschreibungen\MH08; Position: 12 - 12]

147.

"Kleinere funktionale Programme schreiben können."

[Modulbeschreibungen\MH08; Position: 13 - 13]

148.

"Programmierparadigmen und -sprachen, Syntax und Semantik bei Programmiersprachen definieren können."

[Modulbeschreibungen\MH08; Position: 14 - 14]

149.

"Entwurf einfacher Algorithmen."

[Modulbeschreibungen\MH08; Position: 17 - 17]

150.

"Bewertung [einfacher Algorithmen] hinsichtlich Zeit- und Platzkomplexität."

[Modulbeschreibungen\MH08; Position: 18 - 18]

151.

"Implementierung einfacher Algorithmen"

[Modulbeschreibungen\MH08; Position: 19 - 19]

152.

"Bewertung [einfacher Algorithmen] hinsichtlich Zeit- und Platzkomplexität."

[Modulbeschreibungen\MH08; Position: 20 - 20]

153.

"Suchen kürzester Wege und minimaler Spannbäume auf Graphen."

[Modulbeschreibungen\MH08; Position: 22 - 22]

154.

"Suchen kürzester Abstände und Bilden konvexer Hüllen auf Punktmengen."

[Modulbeschreibungen\MH08; Position: 23 - 23]

155.

"Die Studierenden erwerben grundlegende theoretische und praktische Fähigkeiten zur Entwicklung von Software mit Hilfe höherer Programmiersprachen."

[Modulbeschreibungen\MH09; Position: 5 - 5]

156.

"Anhand kleinerer Problemstellungen lernen [die Studierenden], problemorientiert Algorithmen zu entwerfen."

[Modulbeschreibungen\MH09; Position: 6 - 6]

157.

"[Anhand kleinerer Problemstellungen lernen] [die Studierenden], programmiersprachlich zu formulieren."

[Modulbeschreibungen\MH09; Position: 7 - 7]

158.

"[Anhand kleinerer Problemstellungen lernen] [die Studierenden], [programmiersprachlich] zu implementieren."

[Modulbeschreibungen\MH09; Position: 8 - 8]

159.

"[Anhand kleinerer Problemstellungen lernen] [die Studierenden], [programmiersprachlich] zu testen."

[Modulbeschreibungen\MH09; Position: 9 - 9]

160.

"[Anhand kleinerer Problemstellungen lernen] [die Studierenden], professionell zu dokumentieren."

[Modulbeschreibungen\MH09; Position: 10 - 10]

161.

"Neben der Fähigkeit zur selbstständigen Entwicklung programm-technischer Lösungen wird insbesondere der Umgang mit modernen Programmierumgebungen erlernt und geübt."

[Modulbeschreibungen\MH09; Position: 11 - 11]

162.

"[Die Studentinnen] formulieren für kleinere Problemstellungen selbständig Algorithmen."

[Modulbeschreibungen\MH09; Position: 15 - 15]

163.

"[Die Studentinnen] stellen [...] [selbstständig formulierte Algorithmen für kleinere Problemstellungen] geeignet dar."

[Modulbeschreibungen\MH09; Position: 16 - 16]

164.

"Die Studierenden erwerben grundlegende theoretische und praktische Fähigkeiten zur Entwicklung von Software mit Hilfe objektorientierter Programmiersprachen."

[Modulbeschreibungen\MH09; Position: 18 - 18]

165.

"Schwerpunkt der praktischen Arbeit ist der Entwurf von Applikationen unter Nutzung von Standard- APIs."

[Modulbeschreibungen\MH09; Position: 20 - 20]

166.

"[Schwerpunkt der praktischen Arbeit] ist die selbständige Realisierung [von Applikationen unter Nutzung von Standard- APIs]."

[Modulbeschreibungen\MH09; Position: 21 - 21]

167.

"Neben der Fähigkeit zur selbstständigen Entwicklung komplexer programmtechnischer Lösungen wird insbesondere der Umgang mit modernen Spezifikationstools und Programmierumgebungen erlernt und geübt."

[Modulbeschreibungen\MH09; Position: 22 - 22]

168.

"Die Studentinnen sollen die in der Veranstaltung vermittelten Begriffe und Konzepte zu Algorithmen und Datenstrukturen derart auf beispielhafte Problemstellungen anwenden können, dass sie geeignete Standard-Datenstrukturen auswählen."

[Modulbeschreibungen\MH09; Position: 24 - 24]

169.

"[Die Studentinnen sollen die in der Veranstaltung vermittelten Begriffe und Konzepte zu Algorithmen und Datenstrukturen derart auf beispielhafte Problemstellungen anwenden können, dass sie] eigene Datenstrukturen passend definieren können."

[Modulbeschreibungen\MH09; Position: 25 - 25]

170.

"[Die Studentinnen sollen die in der Veranstaltung vermittelten Begriffe und Konzepte zu Algorithmen und Datenstrukturen derart auf beispielhafte Problemstellungen anwenden können, dass sie] Algorithmen zur Bearbeitung entwickeln."

[Modulbeschreibungen\MH09; Position: 26 - 26]

171.

"[Die Studentinnen sollen die in der Veranstaltung vermittelten Begriffe und Konzepte zu Algorithmen und Datenstrukturen derart auf beispielhafte Problemstellungen anwenden können, dass sie] diese auf Korrektheit beurteilen können."

[Modulbeschreibungen\MH09; Position: 27 - 27]

172.

"[Die Studentinnen sollen die in der Veranstaltung vermittelten Begriffe und Konzepte zu Algorithmen und Datenstrukturen derart auf beispielhafte Problemstellungen anwenden können, dass sie] diese auf Komplexität beurteilen können."

[Modulbeschreibungen\MH09; Position: 28 - 28]

173.

"[Die Studentinnen sollen die in der Veranstaltung vermittelten Begriffe und Konzepte zu Algorithmen und Datenstrukturen derart auf beispielhafte Problemstellungen anwenden können, dass sie] diese auf Eleganz beurteilen können."

[Modulbeschreibungen\MH09; Position: 29 - 29]

174.

"Konzepte einer imperativen Programmiersprache kennen."

[Modulbeschreibungen\MH10; Position: 7 - 7]

175.

"Anschauliche Sachverhalte im Modell der Objektorientierung ausdrücken können."

[Modulbeschreibungen\MH10; Position: 10 - 10]

176.

"Einfache Algorithmen entwickeln."

[Modulbeschreibungen\MH10; Position: 11 - 11]

177.

"[Einfache Algorithmen] in Java umsetzen können."

[Modulbeschreibungen\MH10; Position: 12 - 12]

178.

"Einfache in Java realisierte Algorithmen systematisch testen können."

[Modulbeschreibungen\MH10; Position: 13 - 13]

179.

"Probleme in Teilprobleme zerlegen."

[Modulbeschreibungen\MH10; Position: 14 - 14]

180.

"[Die Strukturierung von Problemen in Teilprobleme] mit Mitteln von Java umsetzen"

[Modulbeschreibungen\MH10; Position: 15 - 15]

181.

"[Die Strukturierung von Problemen in Teilprobleme] [mit Mitteln von Java] aussagekräftig dokumentieren können."

[Modulbeschreibungen\MH10; Position: 16 - 16]

182.

"[Formale Syntaxbeschreibungen] für einfache Sprachen entwickeln können."

[Modulbeschreibungen\MH10; Position: 18 - 18]

183.

"[Operationelle Semantik einfacher While-Sprachen] zum Nachweis einfacher Programmeigenschaften anwenden können."

[Modulbeschreibungen\MH10; Position: 20 - 20]

184.

"Eine Entwicklungsumgebung nutzen können."

[Modulbeschreibungen\MH10; Position: 21 - 21]

185.

"LaTeX zur Erstellung einfacher Dokumente nutzen können."

[Modulbeschreibungen\MH10; Position: 22 - 22]

186.

"Versionsverwaltungssysteme einsetzen können."

[Modulbeschreibungen\MH10; Position: 23 - 23]

187.

"Typische Datenstrukturen identifizieren."

[Modulbeschreibungen\MH10; Position: 28 - 28]

188.

"[Typische Datenstrukturen] problemadäquat einsetzen können."

[Modulbeschreibungen\MH10; Position: 29 - 29]

189.

"Datenstrukturen und Algorithmen in Java umsetzen können."

[Modulbeschreibungen\MH10; Position: 30 - 30]

190.

"Wesentliche Algorithmen der Informatik erklären können."

[Modulbeschreibungen\MH10; Position: 31 - 31]

191.

"[Wesentliche Algorithmen der Informatik] anwenden können."

[Modulbeschreibungen\MH10; Position: 32 - 32]

192.

"[Wesentliche Algorithmen der Informatik] modifizieren können."

[Modulbeschreibungen\MH10; Position: 33 - 33]

193.

"Algorithmische Alternativen bezüglich der Eignung für ein Problem beurteilen können."

[Modulbeschreibungen\MH10; Position: 34 - 34]

194.

"Die Komplexität von einfachen Algorithmen analysieren können."

[Modulbeschreibungen\MH10; Position: 36 - 36]

195.

"Konzepte und typische Merkmale des funktionalen Programmierens kennen."

[Modulbeschreibungen\MH10; Position: 41 - 41]

196.

"[Konzepte und typische Merkmale des funktionalen Programmierens] anwenden können."

[Modulbeschreibungen\MH10; Position: 43 - 43]

197.

"Datenstrukturen und Algorithmen in einer funktionalen Programmiersprache umsetzen können."

[Modulbeschreibungen\MH10; Position: 44 - 44]

198.

"[Datenstrukturen und Algorithmen in einer funktionalen Programmiersprache] auf einfachere praktische Probleme anwenden können."

[Modulbeschreibungen\MH10; Position: 45 - 45]

199.

"Die Studierenden können die Basiskonzepte einer modernen Programmiersprache benennen."

[Modulbeschreibungen\MH11; Position: 5 - 5]

200.

"[Die Studierenden] können eine abstrakte Problembeschreibung in einen programmierbaren Algorithmus übertragen."

[Modulbeschreibungen\MH11; Position: 7 - 7]

201.

"[Die Studierenden] können objektorientierte Problemlösungen modellieren."

[Modulbeschreibungen\MH11; Position: 8 - 8]

202.

"Die Studierenden können eine moderne Entwicklungsumgebung bedienen (Editor, Debugger)."

[Modulbeschreibungen\MH11; Position: 10 - 10]

203.

"[Die Studierenden] können einfache Programme in einer modernen Programmiersprache entwickeln."

[Modulbeschreibungen\MH11; Position: 11 - 11]

204.

"[Die Studierenden] können fremden Quellcode analysieren."

[Modulbeschreibungen\MH11; Position: 12 - 12]

205.

"[Die Studierenden] [können fremden Quellcode] in eigene Programme integrieren."

[Modulbeschreibungen\MH11; Position: 13 - 13]

206.

"[Die Studierenden] können Qualitätskriterien für lesbaren, wartbaren, wiederverwendbaren Quellcode nennen."

[Modulbeschreibungen\MH11; Position: 14 - 14]

207.

"[Die Studierenden] [können Qualitätskriterien für lesbaren, wartbaren, wiederverwendbaren Quellcode] [...] beim Erstellen eigener Programme umsetzen."

[Modulbeschreibungen\MH11; Position: 15 - 15]

208.

"[Die Studierenden] können Programme automatisiert testen."

[Modulbeschreibungen\MH11; Position: 16 - 16]

209.

"[Die Studierenden können fortgeschrittene Programmier Techniken] [...] in kleinen Beispielprogrammen in einer aktuellen Programmiersprache einsetzen."

[Modulbeschreibungen\MH11; Position: 19 - 19]

210.

"[Die Studierenden] können technische Basis-Hilfsmittel für die teamorientierte SW-Entwicklung anwenden (z.B. Versionsverwaltung)."

[Modulbeschreibungen\MH11; Position: 20 - 20]

211.

"Die Studierenden können in praxisorientierten Problemen erfolgreiche graphentheoretischen Modellierungsparadigmen und Formalismen anwenden."

[Modulbeschreibungen\MH11; Position: 22 - 22]

212.

"[Die Studierenden] können eigenständig praxisorientierten Probleme mit graphentheoretischen Methoden Modellieren."

[Modulbeschreibungen\MH11; Position: 24 - 24]

213.

"[Die Studierenden] [können eigenständig praxisorientierten Probleme mit graphentheoretischen Methoden] Lösen."

[Modulbeschreibungen\MH11; Position: 25 - 25]

214.

"[Die Studierenden] besitzen die Fähigkeit zum eigenständigen Modellieren."

[Modulbeschreibungen\MH11; Position: 26 - 26]

215.

"[Die Studierenden] [besitzen die Fähigkeit zu] einem Redesign von nebenläufigen Prozessen mittels Petri-Netzen."

[Modulbeschreibungen\MH11; Position: 28 - 28]

216.

"[Die Studierenden] können Algorithmen entwerfen."

[Modulbeschreibungen\MH11; Position: 30 - 30]

217.

"[Die Studierenden] können Datenstrukturen entwerfen."

[Modulbeschreibungen\MH11; Position: 31 - 31]

218.

"[Die Studierenden] können Algorithmen analysieren."

[Modulbeschreibungen\MH11; Position: 32 - 32]

219.

"[Die Studierenden] können Datenstrukturen analysieren."

[Modulbeschreibungen\MH11; Position: 33 - 33]

220.

"[Die Studierenden] können Algorithmen anwenden."

[Modulbeschreibungen\MH11; Position: 34 - 34]

221.

"[Die Studierenden] können Datenstrukturen anwenden."

[Modulbeschreibungen\MH11; Position: 35 - 35]

222.

"Die Studierenden besitzen Kenntnisse über algorithmische Lösungen."

[Modulbeschreibungen\MH12; Position: 5 - 5]

223.

"[Die Studierenden] sind in der Lage, [...] [algorithmische Lösungen] im Hinblick auf Problemadäquatheit zu bewerten."

[Modulbeschreibungen\MH12; Position: 6 - 6]

224.

"[Die Studierenden] sind in der Lage, [...] [algorithmische Lösungen] im Hinblick auf Zeit- und Platzkomplexität zu bewerten."

[Modulbeschreibungen\MH12; Position: 7 - 7]

225.

"[Die Studierenden] sind in der Lage, [...] [algorithmische Lösungen] im Hinblick auf Korrektheit und Vollständigkeit zu bewerten."

[Modulbeschreibungen\MH12; Position: 8 - 8]

226.

"[Die Studierenden] verfügen über grundlegende Fertigkeiten für die Auswahl von Algorithmen vor dem Hintergrund konkreter Informationsverarbeitungsaufgaben."

[Modulbeschreibungen\MH12; Position: 9 - 9]

227.

"[Die Studierenden] [verfügen über grundlegende Fertigkeiten für die] Umsetzung [von Algorithmen vor dem Hintergrund konkreter Informationsverarbeitungsaufgaben]."

[Modulbeschreibungen\MH12; Position: 10 - 10]

228.

"[Die Studierenden] [verfügen über grundlegende Fertigkeiten für die] Modifikation [von Algorithmen vor dem Hintergrund konkreter Informationsverarbeitungsaufgaben]."

[Modulbeschreibungen\MH12; Position: 11 - 11]

229.

"Die Teilnehmer können sicher mit einem Rechner umgehen."

[Modulbeschreibungen\MH12; Position: 13 - 13]

230.

"[Die Teilnehmer] können Programmierwerkzeuge wie Compiler und Editoren nutzen."

[Modulbeschreibungen\MH12; Position: 16 - 16]

231.

"[Die Teilnehmer können] [die] Grenzen [von] [Programmierwerkzeugen wie Compiler und Editoren] einschätzen."

[Modulbeschreibungen\MH12; Position: 17 - 17]

232.

"[Die Teilnehmer] kennen grundlegende Datenstrukturen."

[Modulbeschreibungen\MH12; Position: 19 - 19]

233.

"[Die Teilnehmer] können die Tragweite von Tests abschätzen."

[Modulbeschreibungen\MH12; Position: 21 - 21]

234.

"Die Studierenden beherrschen die Grundlagen zur Entwicklung kleiner, gebrauchstauglicher Anwendungen mit Hilfe objektorientierter Konzepte."

[Modulbeschreibungen\MH12; Position: 23 - 23]

235.

"[Die Studierenden] kennen zentrale Konzepte zur Abstraktion und Modularisierung."

[Modulbeschreibungen\MH12; Position: 24 - 24]

236.

"[Weiterhin] können [die Studierenden] mit integrierten Entwicklungsumgebungen umgehen."

[Modulbeschreibungen\MH12; Position: 26 - 26]

237.

"[Die Studierenden] kennen geeignete Abstraktionen, Modellbildungen und Verfahren zur Beschreibung von Algorithmen und Prozessen."

[Modulbeschreibungen\MH12; Position: 29 - 29]

238.

"[Die Studierenden] [kennen geeignete Abstraktionen, Modellbildungen und Verfahren zur] Analyse von Algorithmen und Prozessen."

[Modulbeschreibungen\MH12; Position: 30 - 30]

239.

"[Die Studierenden] können einfache Funktionalitäten in Klassen kapseln, Objekte erzeugen."

[Modulbeschreibungen\MH13; Position: 8 - 8]

240.

"[Die Studierenden] können Methoden aufrufen."

[Modulbeschreibungen\MH13; Position: 9 - 9]

241.

"Darüber hinaus sind [die Studierenden] in der Lage, korrekten, lesbaren und wartbaren Code zu erzeugen."

[Modulbeschreibungen\MH13; Position: 10 - 10]

242.

"Einfache Problemstellungen zu strukturieren [ist] umsetzbar."

[Modulbeschreibungen\MH13; Position: 12 - 12]

243.

"Eine Lösung zu erarbeiten [für einfache Problemstellungen] [ist] umsetzbar."

[Modulbeschreibungen\MH13; Position: 13 - 13]

244.

"Die Qualität [...] [einer eigenen] Lösung anhand grundlegender Merkmale zu reflektieren [...] [ist] umsetzbar."

[Modulbeschreibungen\MH13; Position: 14 - 14]

245.

"[Die Studierenden] können [...] [grafische Benutzerschnittstellen] erstellen."

[Modulbeschreibungen\MH13; Position: 20 - 20]

246.

"[Die Studierenden] sind in der Lage, größere Anwendungen zu strukturieren."

[Modulbeschreibungen\MH13; Position: 21 - 21]

247.

"[Die Studierenden] [sind in der Lage, größere Anwendungen] zu erstellen."

[Modulbeschreibungen\MH13; Position: 22 - 22]

248.

"Die Studierenden kennen die wichtigsten statischen und dynamischen Datenstrukturen sowie Such- und Sortieralgorithmen."

[Modulbeschreibungen\MH13; Position: 25 - 25]

249.

"Darüber hinaus kennen [die Studierenden] die O-Notation."

[Modulbeschreibungen\MH13; Position: 26 - 26]

250.

"[Die Studierenden] können die Effizienz von Algorithmen mit Hilfe mathematischer Methoden analysieren."

[Modulbeschreibungen\MH13; Position: 27 - 27]

251.

"Die Studierenden kennen zudem die wichtigsten Programmierkonzepte von C."

[Modulbeschreibungen\MH13; Position: 29 - 29]

252.

"[Die Studierenden] sind in der Lage, ein prozedurales Programm zu entwerfen."

[Modulbeschreibungen\MH13; Position: 30 - 30]

253.

"[Die Studierenden] [sind in der Lage ein prozedurales Programm] zu implementieren."

[Modulbeschreibungen\MH13; Position: 31 - 31]

254.

"[Die Studierenden] [sind in der Lage ein prozedurales Programm] zu testen."

[Modulbeschreibungen\MH13; Position: 32 - 32]

255.

"Gründliche Kenntnisse einer Programmiersprache."

[Modulbeschreibungen\MH14; Position: 5 - 5]

256.

"Kenntnis grundlegender Algorithmen und Datenstrukturen."

[Modulbeschreibungen\MH14; Position: 11 - 11]

257.

"Design mit Hilfe von Szenarien, Objekt- und Klassendiagrammen."

[Modulbeschreibungen\MH14; Position: 18 - 18]

258.

"Implementierung."

[Modulbeschreibungen\MH14; Position: 19 - 19]

259.

"Validierung durch systematische Tests."

[Modulbeschreibungen\MH14; Position: 20 - 20]

260.

"Fach- und Methodenkompetenzen (Wissen und Verstehen sowie Anwendung und Erzeugung von Wissen): Die Studierenden kennen ein umfassendes Instrumentarium an Techniken und Lösungsmustern zur Softwareentwicklung in unterschiedlichen Programmiersprachen."

[Modulbeschreibungen\MH15; Position: 5 - 5]

261.

"[Fach- und Methodenkompetenzen (Wissen und Verstehen sowie Anwendung und Erzeugung von Wissen):] Die Entwicklung von eigenen Bibliotheken und komplexen Anwendungen in einzelnen Komponenten kann gezeigt werden."

[Modulbeschreibungen\MH15; Position: 6 - 6]

262.

"[Fach- und Methodenkompetenzen (Wissen und Verstehen sowie Anwendung und Erzeugung von Wissen):] Standardarchitekturmuster sind bekannt."

[Modulbeschreibungen\MH15; Position: 7 - 7]

263.

"Fach- und Methodenkompetenzen (Wissen und Verstehen sowie Anwendung und Erzeugung von Wissen): Entwurf von Algorithmen für gegebene Problemstellungen sind typische Aufgaben eines Informatikers."

[Modulbeschreibungen\MH15; Position: 23 - 23]

264.

"[Fach- und Methodenkompetenzen (Wissen und Verstehen sowie Anwendung und Erzeugung von Wissen):] [Entwurf von] Datenstrukturen für gegebene Problemstellungen sind typische Aufgaben eines Informatikers."

[Modulbeschreibungen\MH15; Position: 24 - 24]

265.

"[Fach- und Methodenkompetenzen (Wissen und Verstehen sowie Anwendung und Erzeugung von Wissen):] Implementierung von Algorithmen und Datenstrukturen für gegebene Problemstellungen sind typische Aufgaben eines Informatikers."

[Modulbeschreibungen\MH15; Position: 25 - 25]

266.

"[Fach- und Methodenkompetenzen (Wissen und Verstehen sowie Anwendung und Erzeugung von Wissen):] Auswahl von Algorithmen und Datenstrukturen für gegebene Problemstellungen sind typische Aufgaben eines Informatikers."

[Modulbeschreibungen\MH15; Position: 26 - 26]

267.

"Nach Beendigung dieses Moduls können die Studierenden selbstständig Algorithmen entwerfen."

[Modulbeschreibungen\MH15; Position: 27 - 27]

268.

"[Nach Beendigung dieses Moduls können die Studierenden selbstständig Algorithmen] bewerten (Laufzeit)."

[Modulbeschreibungen\MH15; Position: 28 - 28]

269.

"[Nach Beendigung dieses Moduls können die Studierenden selbstständig Algorithmen] implementieren."

[Modulbeschreibungen\MH15; Position: 29 - 29]

270.

"[Nach Beendigung dieses Moduls können die Studierenden] dynamische Datenstrukturen (objektorientiert) implementieren."

[Modulbeschreibungen\MH15; Position: 30 - 30]

271.

"[Nach Beendigung dieses Moduls können die Studierenden] für Problemstellungen passende Algorithmen und Datenstrukturen auswählen."

[Modulbeschreibungen\MH15; Position: 31 - 31]

272.

"[Nach Beendigung dieses Moduls können die Studierenden] bestehende Bibliotheken nutzen."

[Modulbeschreibungen\MH15; Position: 32 - 32]

273.

"[Nach Beendigung dieses Moduls können die Studierenden] graphentheoretische Konzepte für praktische Problemstellungen anwenden."

[Modulbeschreibungen\MH15; Position: 33 - 33]

274.

"Wissen und Verstehen: Die Studierenden sollen grundlegende Datenstrukturen mit deren Eigenschaften und Leistungsparametern kennen."

[Modulbeschreibungen\MH16; Position: 5 - 5]

275.

"[Wissen und Verstehen:] [Die Studierenden sollen grundlegende Datenstrukturen mit deren Eigenschaften und Leistungsparametern] [...] [in asymptotischer Notation] vergleichen können."

[Modulbeschreibungen\MH16; Position: 7 - 7]

276.

"Können: Die Studierenden lernen, Datenstrukturen für neue Problemstellungen eigenständig zu entwerfen (instrumentale Kompetenz)."

[Modulbeschreibungen\MH16; Position: 8 - 8]

277.

"[Können:] [Die Studierenden lernen,] [...] [Leistungsparameter für eigenständig entworfene Datenstrukturen zur Lösung neuer Problemstellungen] zu analysieren [(instrumentale Kompetenz)]."

[Modulbeschreibungen\MH16; Position: 9 - 9]

278.

"[Können:] [...] [Durch die Analyse von Leistungsparametern für eigenständig entworfene Datenstrukturen zur Lösung neuer Problemstellungen] sollen [die Studierenden] im Beruf z.B. in der Lage sein, bestehende Software durch geeignetere Datenstrukturen zu beschleunigen (systemische Kompetenz)."

[Modulbeschreibungen\MH16; Position: 10 - 10]

279.

"[Die Studierenden sollen] über Wissen zum strukturierten und objektorientierten Programmieren mit einer imperativen Programmiersprache verfügen (instrumentale Kompetenz)."

[Modulbeschreibungen\MH16; Position: 14 - 14]

280.

"[Die Studierenden] sollen die Prozesse und Methoden der Software-Entwicklung und des Algorithmenentwurfs sowie die Services des Betriebssystems kennen."

[Modulbeschreibungen\MH16; Position: 15 - 15]

281.

"[Die Studierenden sollen weiterhin] verteilte Systeme und paralleles Programmieren kennen (systemische Kompetenz)."

[Modulbeschreibungen\MH16; Position: 17 - 17]

282.

"[Die Studierenden sollen] einfache Programmieraufgaben lösen."

[Modulbeschreibungen\MH16; Position: 21 - 21]

283.

"Zur Erarbeitung instrumentaler und systemischer Kompetenzen sollen die Studierenden die verschiedenen Programmiersprachparadigmen und Konzepte zu Syntax und Semantik kennen."

[Modulbeschreibungen\MH16; Position: 25 - 25]

284.

"[Die Studierenden] sollen Wissen über funktionale Sprachen erwerben."

[Modulbeschreibungen\MH16; Position: 26 - 26]

285.

"[Zur Erarbeitung instrumentaler und systemischer Kompetenzen sollen die Studierenden] die grundlegenden Konzepte des Übersetzens und des Compilerentwurfs kennen."

[Modulbeschreibungen\MH16; Position: 28 - 28]

286.

"[Zur Erarbeitung instrumentaler und systemischer Kompetenzen sollen die Studierenden] die Modellierung, Verwaltung und Nutzung größerer Datenbestände kennen."

[Modulbeschreibungen\MH16; Position: 29 - 29]

287.

"[Zur Erarbeitung instrumentaler und systemischer Kompetenzen sollen die Studierenden] [die Modellierung, Verwaltung und Nutzung größerer Datenbestände] für kleinere Datenbanken implementieren können."

[Modulbeschreibungen\MH16; Position: 30 - 30]

288.

"Die Studierenden können [...] einfache Programmieraufgaben mit Hilfe von funktionalen und/oder objektorientierten Programmiersprachen systematisch lösen."

[Modulbeschreibungen\MH17; Position: 6 - 6]

289.

"[Die Studierenden können] [...] Qualitätssicherung mittels einfacher (Unit-) Tests durchführen."

[Modulbeschreibungen\MH17; Position: 7 - 7]

290.

"[Die Studierenden können] [...] basierend [auf den Komplexitätsklassen von Algorithmen] die Eignung selbiger für konkrete Aufgaben einschätzen."

[Modulbeschreibungen\MH17; Position: 9 - 9]

291.

"[Die Studierenden können] [...] Sourcecode grundlegend unter Zuhilfenahme von Standardwerkzeugen dokumentieren."

[Modulbeschreibungen\MH17; Position: 10 - 10]

292.

"In dieser Veranstaltung lernen Studierende grundlegende Datenstrukturen und Algorithmen kennen."

[Modulbeschreibungen\MH17; Position: 12 - 12]

293.

"[Die Studierenden] [erwerben die Fähigkeiten] asymptotische Komplexität einzuschätzen."

[Modulbeschreibungen\MH17; Position: 15 - 15]

294.

"[Die Studierenden] [erwerben die Fähigkeiten] [asymptotische Komplexität] zu bestimmen."

[Modulbeschreibungen\MH17; Position: 16 - 16]

295.

"[Außerdem] können [die Studierenden] [...] [bedeutende algorithmische Strategien] anwenden."

[Modulbeschreibungen\MH17; Position: 18 - 18]

296.

"[Die Studierenden] können einfache Anwendungen mittels systemnaher und/oder paralleler Programmierung auf ausgewählten Plattformen entwickeln."

[Modulbeschreibungen\MH17; Position: 21 - 21]

297.

"[Die Studierenden] [können einfache Anwendungen mittels systemnaher und/oder paralleler Programmierung auf ausgewählten Plattformen] [...] analysieren."

[Modulbeschreibungen\MH17; Position: 22 - 22]

298.

"Anwenden von Grundlagen der prozeduralen und der objektorientierten Programmierung am Beispiel der Sprachen C und C++."

[Modulbeschreibungen\MH18; Position: 6 - 6]

299.

"Entwickeln einfacher Algorithmen."

[Modulbeschreibungen\MH18; Position: 7 - 7]

300.

"Implementieren einfacher Algorithmen."

[Modulbeschreibungen\MH18; Position: 8 - 8]

301.

"Programmierung grafischer Oberflächen am Beispiel von MS Windows anhand objektorientierter Ansätze."

[Modulbeschreibungen\MH18; Position: 9 - 9]

302.

"[Wichtige Algorithmen] anwenden."

[Modulbeschreibungen\MH18; Position: 12 - 12]

303.

"[Wichtige Datenstrukturen] anwenden [(z.B. Arrays, Stapel, Bäume)]."

[Modulbeschreibungen\MH18; Position: 14 - 14]

304.

"Effizienz von Algorithmen analysieren."

[Modulbeschreibungen\MH18; Position: 15 - 15]

305.

"[Effizienz von Algorithmen] bewerten."

[Modulbeschreibungen\MH18; Position: 16 - 16]

306.

"Geeignete Algorithmen für neue Problemstellungen erarbeiten."

[Modulbeschreibungen\MH18; Position: 17 - 17]

307.

"Umsetzungsmöglichkeiten für die Programmiersprachen C++, Java und .NET in den Grundzügen kennen."

[Modulbeschreibungen\MH18; Position: 18 - 18]

308.

"Das objektorientierte Programmierparadigma kennen."

[Modulbeschreibungen\MH18; Position: 20 - 20]

309.

"[Das objektorientierte Programmierparadigma] mit Java richtig anwenden."

[Modulbeschreibungen\MH18; Position: 21 - 21]

310.

"[Prinzipien der Generischen Programmierung] anwenden (C++)."

[Modulbeschreibungen\MH18; Position: 23 - 23]

311.

"[Probleme und Lösungsansätze der nebenläufigen Programmierung] anwenden [(Java)]."

[Modulbeschreibungen\MH18; Position: 25 - 25]

312.

"[Methodenkompetenz:] Entwurf effizienter Datenstrukturen für einfache Probleme."

[Modulbeschreibungen\MH19; Position: 8 - 8]

313.

"[Selbst- und Sozialkompetenz:] Unabhängigkeit algorithmischer Ideen vom Programmierparadigma."

[Modulbeschreibungen\MH19; Position: 11 - 11]

314.

"Fachkompetenz: Kenntnis wichtiger Basisalgorithmen für Graph-, Codierungs-, Kommunikations- und Optimierungsprobleme."

[Modulbeschreibungen\MH19; Position: 13 - 13]

315.

"[Fachkompetenz:] Kenntnis der wichtigsten Datenstrukturen und Muster."

[Modulbeschreibungen\MH19; Position: 14 - 14]

316.

"[Methodenkompetenz:] Auswahl geeigneter Algorithmen."

[Modulbeschreibungen\MH19; Position: 17 - 17]

317.

"[Methodenkompetenz:] Anpassung von Algorithmen an spezielle Erfordernisse."

[Modulbeschreibungen\MH19; Position: 18 - 18]

318.

"[Methodenkompetenz] Anpassung von Datenstrukturen an spezielle Erfordernisse."

[Modulbeschreibungen\MH19; Position: 19 - 19]

319.

"Selbst- und Sozialkompetenz: Fähigkeit zur Argumentation über gewählte Ansätze und zur Begründung von Entwurfsentscheidungen."

[Modulbeschreibungen\MH19; Position: 20 - 20]

320.

"[Selbst- und Sozialkompetenz:] Informatiktypische Denk- und Herangehensweisen losgelöst von konkreten Sprachen und Paradigmen."

[Modulbeschreibungen\MH19; Position: 22 - 22]

321.

"Methodenkompetenz: Umgang mit Entwicklungsumgebungen und –werkzeugen."

[Modulbeschreibungen\MH19; Position: 27 - 27]

322.

"[Methodenkompetenz:] Auswahl geeigneter Prozesse und Methoden."

[Modulbeschreibungen\MH19; Position: 29 - 29]

323.

"Algorithmische Kompetenz: eine konkrete Problemstellung analysieren."

[Modulbeschreibungen\MH20; Position: 5 - 5]

324.

"[Algorithmische Kompetenz:] [eine konkrete Problemstellung] algorithmisch lösen können."

[Modulbeschreibungen\MH20; Position: 6 - 6]

325.

"[Algorithmische Kompetenz:] grundlegende Algorithmen zur Lösung von Problemen einsetzen."

[Modulbeschreibungen\MH20; Position: 7 - 7]

326.

"[Algorithmische Kompetenz:] grundlegende Datenstrukturen zur Lösung von Problemen einsetzen."

[Modulbeschreibungen\MH20; Position: 8 - 8]

327.

"[Realisierungs-Kompetenz:] Erstellen von Programmen unter Einsatz entsprechender Werkzeuge."

[Modulbeschreibungen\MH20; Position: 10 - 10]

328.

"[Realisierungs-Kompetenz:] Testen von Programmen unter Einsatz entsprechender Werkzeuge."

[Modulbeschreibungen\MH20; Position: 11 - 11]

329.

"Informatik-Kompetenz: kennen des Aufbaus und der Funktionsweise von Computern."

[Modulbeschreibungen\MH20; Position: 13 - 13]

330.

"[Informatik-Kompetenz:] effizientes Benutzen von UNIX-Systemen nur mit der Kommandozeile."

[Modulbeschreibungen\MH20; Position: 15 - 15]

331.

"[Informatik-Kompetenz:] Programmierung in Assembler."

[Modulbeschreibungen\MH20; Position: 16 - 16]

332.

"Kennen des Aufbaus und der Funktionsweise von Netzen wie dem Internet."

[Modulbeschreibungen\MH20; Position: 17 - 17]

333.

"Algorithmische Kompetenz: eine konkrete Problemstellung analysieren."

[Modulbeschreibungen\MH20; Position: 21 - 21]

334.

"[Algorithmische Kompetenz:] [eine konkrete Problemstellung] algorithmisch lösen können."

[Modulbeschreibungen\MH20; Position: 22 - 22]

335.

"Analyse-Kompetenz: ein informell dargestelltes Problem mithilfe einer Modellierungssprache (UML) (semi-)formal beschreiben können."

[Modulbeschreibungen\MH20; Position: 23 - 23]

336.

"Design-Kompetenz: UML-Modelle in ablauffähige Programme umsetzen und dabei grundlegende Entwurfsprinzipien berücksichtigen."

[Modulbeschreibungen\MH20; Position: 24 - 24]

337.

"[Realisierungs-Kompetenz:] Erstellen von Programmen unter Einsatz entsprechender Werkzeuge."

[Modulbeschreibungen\MH20; Position: 26 - 26]

338.

"[Realisierungs-Kompetenz:] Testen von Programmen unter Einsatz entsprechender Werkzeuge."

[Modulbeschreibungen\MH20; Position: 27 - 27]

339.

"Formale und algorithmische Kompetenzen: Aufwandsabschätzungen mit Hilfe der O-Notation kennen."

[Modulbeschreibungen\MH20; Position: 29 - 29]

340.

"[Formale und algorithmische Kompetenzen:] [Aufwandsabschätzungen mit Hilfe der O-Notation] [...] auf Algorithmen anwenden."

[Modulbeschreibungen\MH20; Position: 30 - 30]

341.

"[Formale und algorithmische Kompetenzen:] Algorithmen bezüglich deren erwarteter Laufzeit vergleichen können."

[Modulbeschreibungen\MH20; Position: 31 - 31]

342.

"[Formale und algorithmische Kompetenzen:] gute Algorithmen für wichtige Standardprobleme (z.B. Sortieren) kennen."

[Modulbeschreibungen\MH20; Position: 32 - 32]

343.

"[Formale und algorithmische Kompetenzen:] [gute Algorithmen für wichtige Standardprobleme (z.B. Sortieren)] vergleichen."

[Modulbeschreibungen\MH20; Position: 33 - 33]

344.

"Analyse-, Design- und Realisierungs-Kompetenzen: wichtige lineare und nicht-lineare Datenstrukturen kennen."

[Modulbeschreibungen\MH20; Position: 34 - 34]

345.

"[Analyse-, Design- und Realisierungs-Kompetenzen:] [wichtige lineare und nicht-lineare Datenstrukturen] sinnvoll für ein gegebenes Problem einsetzen können."

[Modulbeschreibungen\MH20; Position: 35 - 35]

346.

"[Analyse-, Design- und Realisierungs-Kompetenzen:] verschiedene Implementierungsvarianten für wichtige Datenstrukturen kennen."

[Modulbeschreibungen\MH20; Position: 36 - 36]

347.

"[Analyse-, Design- und Realisierungs-Kompetenzen:] [verschiedene Implementierungsvarianten für wichtige Datenstrukturen] vergleichen."

[Modulbeschreibungen\MH20; Position: 37 - 37]

348.

"[Analyse-, Design- und Realisierungs-Kompetenzen:] [verschiedene Implementierungsvarianten für wichtige Datenstrukturen] auswählen können."

[Modulbeschreibungen\MH20; Position: 38 - 38]

349.

"[Analyse-, Design- und Realisierungs-Kompetenzen:] effiziente Datenstrukturen für neue Probleme entwickeln können."

[Modulbeschreibungen\MH20; Position: 39 - 39]

350.

"[Analyse-, Design- und Realisierungs-Kompetenzen:] Algorithmen für neue Probleme entwickeln können."

[Modulbeschreibungen\MH20; Position: 40 - 40]

351.

"[Analyse-, Design- und Realisierungs-Kompetenzen:] Datenstrukturen aus Standardbibliotheken kennen."

[Modulbeschreibungen\MH20; Position: 41 - 41]

352.

"[Analyse-, Design- und Realisierungs-Kompetenzen:] [Datenstrukturen aus Standardbibliotheken] effizient einsetzen können."

[Modulbeschreibungen\MH20; Position: 42 - 42]

353.

"Technologische Kompetenzen: Studierende können mit den Werkzeugen gcc, g++ und make umgehen."

[Modulbeschreibungen\MH20; Position: 44 - 44]

354.

"[Design-, Realisierungs- und Methodenkompetenzen:] [Studierende] können typische Programmierprobleme in C90 lösen."

[Modulbeschreibungen\MH20; Position: 47 - 47]

355.

"[Design-, Realisierungs- und Methodenkompetenzen:] [Studierende] können einfache Programmierprobleme in C++98 lösen."

[Modulbeschreibungen\MH20; Position: 49 - 49]

356.

"[Design- und Realisierungskompetenzen:] Testen eines größeren Software-Programms im Team."

[Modulbeschreibungen\MH20; Position: 53 - 53]

357.

"Die Studierenden sollen nach Abschluss des Moduls selbstständig imperative Programme auf der Grundlage von Java für die Lösung kleinerer Probleme entwickeln."

[Modulbeschreibungen\MH21; Position: 8 - 8]

358.

"[Die Studierenden sollen nach Abschluss des Moduls] die Effizienz ihrer Programme einschätzen können."

[Modulbeschreibungen\MH21; Position: 9 - 9]

359.

"[Fachkompetenzen:] [Die Studierenden] erkennen die Terminologie der imperativen Programmierung."

[Modulbeschreibungen\MH21; Position: 11 - 11]

360.

"[Fachkompetenzen:] [Die Studierenden] beschreiben, was ihnen vorgelegte Programme tun."

[Modulbeschreibungen\MH21; Position: 13 - 13]

361.

"[Fachkompetenzen:] [Die Studierenden] entwickeln selbstständig Programme für die Lösung kleinerer Probleme."

[Modulbeschreibungen\MH21; Position: 14 - 14]

362.

"[Fachkompetenzen:] [Die Studierenden] untersuchen systematisch eigene und fremde Programme auf Fehler."

[Modulbeschreibungen\MH21; Position: 15 - 15]

363.

"[Fachkompetenzen:] [Die Studierenden] setzen moderne Programmierungsumgebungen zum Entwickeln und Testen von Programmen ein."

[Modulbeschreibungen\MH21; Position: 16 - 16]

364.

"[Fachkompetenzen:] [Die Studierenden] erstellen Algorithmen mit allgemeinen Entwurfskonzepten (z.B. Greedy-Verfahren, Divide-and-Conquer-Verfahren)."

[Modulbeschreibungen\MH21; Position: 17 - 17]

365.

"[Fachkompetenzen:] [Die Studierenden] benennen Algorithmen und Datenstrukturen zur Lösung von häufig vorkommenden Problemen."

[Modulbeschreibungen\MH21; Position: 18 - 18]

366.

"[Fachkompetenzen:] [Die Studierenden] bewerten [...] [Algorithmen und Datenstrukturen zur Lösung von häufig vorkommenden Problemen] in ihrer Anwendbarkeit."

[Modulbeschreibungen\MH21; Position: 19 - 19]

367.

"[Fachkompetenzen:] [Die Studierenden] bewerten [...] [Probleme der Effizienz von algorithmischen Lösungen konkreter Fragestellungen]."

[Modulbeschreibungen\MH21; Position: 21 - 21]

368.

"[Fachkompetenzen:] [Die Studierenden] wählen fundiert einen Algorithmus und eine Datenstruktur zur Lösung eines konkreten Problems aus."

[Modulbeschreibungen\MH21; Position: 22 - 22]

369.

"[Fachkompetenzen:] [Die Studierenden] wenden die gelernten Algorithmen sinnvoll auf gegebene und konkrete Probleme an."

[Modulbeschreibungen\MH21; Position: 23 - 23]

370.

"[Fachkompetenzen:] [Die Studierenden] wenden die gelernten Datenstrukturen sinnvoll auf gegebene und konkrete Probleme an."

[Modulbeschreibungen\MH21; Position: 24 - 24]

371.

"Gegebene Problemstellungen werden [...] [in der Softwareentwicklung] mit Hilfe objektorientierter Analyse- und Entwurfsverfahren in ein objektorientiertes Modell überführt."

[Modulbeschreibungen\MH21; Position: 33 - 33]

372.

"[Gegebene Problemstellungen werden [...] [in der Softwareentwicklung] mit Hilfe objektorientierter Analyse- und Entwurfsverfahren] in ein objektorientiertes Programm überführt."

[Modulbeschreibungen\MH21; Position: 34 - 34]

373.

"Ziel des Moduls „Objektorientierte Modellierung und Programmierung“ ist das Erlernen grundlegender Konzepte der objektorientierten Modellierung mit Hilfe der UML als Modellierungsnotation."

[Modulbeschreibungen\MH21; Position: 35 - 35]

374.

"[Ziel des Moduls „Objektorientierte Modellierung und Programmierung“ ist das Erlernen] der objektorientierten Programmierung mit der Programmiersprache Java."

[Modulbeschreibungen\MH21; Position: 36 - 36]

375.

"Die Studierenden sollen nach Abschluss des Moduls selbstständig objektorientierte Programme auf der Grundlage von Java für die Lösung mittelgroßer Probleme entwickeln können."

[Modulbeschreibungen\MH21; Position: 37 - 37]

376.

"Fachkompetenzen: Die Studierenden kennen grundlegende Konzepte der objektorientierten Modellierung und UML als Modellierungsnotation."

[Modulbeschreibungen\MH21; Position: 38 - 38]

377.

"[Fachkompetenzen:] [Die Studierenden] kennen grundlegende Konzepte der objektorientierten Programmierung mit Java."

[Modulbeschreibungen\MH21; Position: 39 - 39]

378.

"[Fachkompetenzen:] [Die Studierenden] kennen die Terminologie der objektorientierten Modellierung und Programmierung."

[Modulbeschreibungen\MH21; Position: 40 - 40]

379.

"[Fachkompetenzen:] [Die Studierenden] können beschreiben, was ihnen vorgelegte objektorientierte Programme tun."

[Modulbeschreibungen\MH21; Position: 42 - 42]

380.

"[Fachkompetenzen:] [Die Studierenden] entwickeln selbstständig Modelle für die Lösung mittelgroßer Probleme."

[Modulbeschreibungen\MH21; Position: 43 - 43]

381.

"[Fachkompetenzen:] [Die Studierenden] entwickeln selbstständig Programme für die Lösung mittelgroßer Probleme."

[Modulbeschreibungen\MH21; Position: 44 - 44]

382.

"[Fachkompetenzen:] [Die Studierenden] untersuchen systematisch eigene und fremde Modelle und Programme auf Fehler."

[Modulbeschreibungen\MH21; Position: 45 - 45]

383.

"[Fachkompetenzen:] [Die Studierenden] setzen moderne Entwicklungsumgebungen zum Modellieren und Entwickeln von Programmen ein."

[Modulbeschreibungen\MH21; Position: 46 - 46]

384.

"[Fachkompetenzen:] [Die Studierenden] kennen die Unterschiede zwischen dem imperativen, objektorientierten, funktionalen und logischen Programmierparadigma."

[Modulbeschreibungen\MH21; Position: 47 - 47]

385.

"Methodenkompetenzen: Die Studierenden entwickeln selbstständig Programme für gegebene Probleme durch konsequente Anwendung der Konzepte der objektorientierten Modellierung und Programmierung."

[Modulbeschreibungen\MH21; Position: 48 - 48]

386.

"[Selbstkompetenzen:] [Die Studierenden] beziehen die Konzepte des objektorientierten Programmentwurfs in ihr Handeln ein."

[Modulbeschreibungen\MH21; Position: 53 - 53]

387.

"Kenntnisse: Wesentliche Konzepte imperativer und objektorientierter Programmiersprachen sowie wichtige Programmiertechniken in diesen Sprachen."

[Modulbeschreibungen\MH22; Position: 5 - 5]

388.

"Kenntnis der Programmierkonzepte logischer und funktionaler Programmiersprachen."

[Modulbeschreibungen\MH22; Position: 6 - 6]

389.

"Kenntnis grundlegender Datenstrukturen in verschiedenen Programmierparadigmen."

[Modulbeschreibungen\MH22; Position: 7 - 7]

390.

"[Kenntnis] [...] der Realisierung [grundlegender Datenstrukturen] in verschiedenen Programmierparadigmen."

[Modulbeschreibungen\MH22; Position: 8 - 8]

391.

"Kenntnis grundlegender Beschreibungsformen für Programmiersprachen."

[Modulbeschreibungen\MH22; Position: 9 - 9]

392.

"Grundkenntnisse der Programmverifikation."

[Modulbeschreibungen\MH22; Position: 10 - 10]

393.

"[Fähigkeiten:] Verwendung logischer und funktionaler Programmiersprachen."

[Modulbeschreibungen\MH22; Position: 12 - 12]

394.

"[Fähigkeiten:] Realisierung von Datenstrukturen in verschiedenen Programmierparadigmen."

[Modulbeschreibungen\MH22; Position: 13 - 13]

395.

"[Fähigkeiten:] Programmverifikation."

[Modulbeschreibungen\MH22; Position: 15 - 15]

396.

"Kompetenzen: Selbstständige Entwicklung kleinerer Programme und ihrer Dokumentation unter Beachtung üblicher Programmierkonventionen."

[Modulbeschreibungen\MH22; Position: 16 - 16]

397.

"Beim erfolgreichen Abschluss des Moduls sollten die Studierenden Kenntnisse über folgende Themen haben: Grundlegende Entwurfsmethoden für Algorithmen."

[Modulbeschreibungen\MH22; Position: 18 - 18]

398.

"[Beim erfolgreichen Abschluss des Moduls sollten die Studierenden Kenntnisse über] [...] effiziente Algorithmen und Datenstrukturen für Standardprobleme [haben]."

[Modulbeschreibungen\MH22; Position: 20 - 20]

399.

"Die Studierenden sollten in der Lage sein, grundlegende Methoden zur Laufzeitanalyse von Algorithmen anzuwenden."

[Modulbeschreibungen\MH22; Position: 21 - 21]

400.

"[Die Studierenden sollten in der Lage sein,] algorithmischen Problemen formal zu modellieren."

[Modulbeschreibungen\MH22; Position: 22 - 22]

401.

"[Die Studierenden sollten in der Lage sein,] vorhandene Algorithmen an eine gegebene Problemstellung anzupassen."

[Modulbeschreibungen\MH22; Position: 23 - 23]

402.

"[Die Studierenden sollten in der Lage sein,] vorhandene Datenstrukturen an eine gegebene Problemstellung anzupassen."

[Modulbeschreibungen\MH22; Position: 24 - 24]

403.

"Basierend auf den Kenntnissen und den Fähigkeiten sollten die Studierenden in der Lage sein, Programme unter Verwendung der erlernten algorithmischen Methoden und unter Berücksichtigung programmiertechnischer Konzepte wie z.B. der Kapselung von Datenstrukturen zu implementieren."

[Modulbeschreibungen\MH22; Position: 25 - 25]

404.

"Kenntnisse: Elementare Kenntnisse über physikalischen Prinzipien, welche der Funktionsweise elektronischer Rechner zugrunde liegen."

[Modulbeschreibungen\MH22; Position: 27 - 27]

405.

"[Studierende sollen nach Absolvierung der Lehrveranstaltungen in der Lage sein,] [den] Ablauf [einfacher Algorithmen (iterativ und rekursiv)] zu beschreiben."

[Modulbeschreibungen\MH23; Position: 22 - 22]

406.

"[Studierende sollen nach Absolvierung der Lehrveranstaltungen in der Lage sein,] [einfache Algorithmen (iterativ und rekursiv)] selbst zu erstellen."

[Modulbeschreibungen\MH23; Position: 23 - 23]

407.

"[Studierende sollen nach Absolvierung der Lehrveranstaltungen in der Lage sein,] eigene Datentypen zu konzipieren."

[Modulbeschreibungen\MH23; Position: 24 - 24]

408.

"[Studierende sollen nach Absolvierung der Lehrveranstaltungen in der Lage sein,] eigene Datentypen anzuwenden."

[Modulbeschreibungen\MH23; Position: 25 - 25]

409.

"[Studierende sollen nach Absolvierung der Lehrveranstaltungen in der Lage sein,] einfache objektorientierte Programme mit Polymorphie, Vererbung und Schnittstellen zu entwickeln."

[Modulbeschreibungen\MH23; Position: 26 - 26]

410.

"[Studierende sollen nach Absolvierung der Lehrveranstaltungen in der Lage sein,] die behandelten dynamischen Datenstrukturen anzuwenden."

[Modulbeschreibungen\MH23; Position: 27 - 27]

411.

"[Studierende, die das Modul erfolgreich absolviert haben,] haben die Fähigkeit zur Problemspezifikation erworben."

[Modulbeschreibungen\MH23; Position: 40 - 40]

412.

"Studierende sollen nach erfolgreichem Abschluss des Moduls grundlegende Architekturen beschreiben können."

[Modulbeschreibungen\MH23; Position: 43 - 43]

413.

"[Studierende sollen nach erfolgreichem Abschluss des Moduls] selbständig Problemstellungen analysieren können."

[Modulbeschreibungen\MH23; Position: 44 - 44]

414.

"[Studierende sollen nach erfolgreichem Abschluss des Moduls] aus einfachen Problemstellungen ein objektorientiertes System zur Lösung entwerfen können."

[Modulbeschreibungen\MH23; Position: 45 - 45]

415.

"[Studierende sollen nach erfolgreichem Abschluss des Moduls] eine Lösung hinsichtlich der Wartbarkeit analysieren können."

[Modulbeschreibungen\MH23; Position: 46 - 46]

416.

"[Studierende sollen nach erfolgreichem Abschluss des Moduls] [eine Lösung hinsichtlich der Wartbarkeit] verbessern können."

[Modulbeschreibungen\MH23; Position: 47 - 47]

417.

"[Studierende sollen nach erfolgreichem Abschluss des Moduls] mit den gängigen Werkzeugen (z.B. IDE) umgehen können."

[Modulbeschreibungen\MH23; Position: 48 - 48]

418.

"Nach erfolgreichem Abschluss des Moduls sollen die Studierenden in der Lage sein, grundlegende Entwicklungsprozesse beschreiben zu können."

[Modulbeschreibungen\MH23; Position: 50 - 50]

419.

"[Nach erfolgreichem Abschluss des Moduls sollen die Studierenden in der Lage sein,] [eine gestellte komplexere Aufgabe] in Teilaufgaben strukturieren zu können."

[Modulbeschreibungen\MH23; Position: 52 - 52]

420.

"[Nach erfolgreichem Abschluss des Moduls sollen die Studierenden in der Lage sein,] die Teilaufgaben in einem Team entsprechend eines Entwicklungsprozesses implementieren zu können."

[Modulbeschreibungen\MH23; Position: 53 - 53]

421.

"[Nach erfolgreichem Abschluss des Moduls sollen die Studierenden in der Lage sein,] die einzelnen Bestandteile sowie das Gesamtprogramms mit geeigneten Verfahren testen zu können."

[Modulbeschreibungen\MH23; Position: 54 - 54]

422.

"[Nach erfolgreichem Abschluss des Moduls sollen die Studierenden in der Lage sein,] mit den Werkzeugen zur Softwareentwicklung im Team (Versionskontrolle, CI Systeme, Bugtracker) sicher umgehen zu können."

[Modulbeschreibungen\MH23; Position: 55 - 55]

423.

"[Die Studierenden sollen] den Aufbau und die Beschreibung von Programmiersprachen als formale Sprachen lernen."

[Modulbeschreibungen\MH24; Position: 6 - 6]

424.

"[Die Studierenden sollen] einfache Algorithmen und Datenstrukturen kennen."

[Modulbeschreibungen\MH24; Position: 7 - 7]

425.

"[Die Studierenden sollen] [einfache Algorithmen und Datenstrukturen] formal beschreiben können."

[Modulbeschreibungen\MH24; Position: 8 - 8]

426.

"[Die Studierenden sollen] unterschiedliche Paradigmen von Programmiersprachen kennen lernen."

[Modulbeschreibungen\MH24; Position: 9 - 9]

427.

"[Die Studierenden sollen] einfache Java-Programme analysieren können."

[Modulbeschreibungen\MH24; Position: 11 - 11]

428.

"[Die Studierenden sollen] einfache Java-Programme erstellen können."

[Modulbeschreibungen\MH24; Position: 12 - 12]

429.

"[Die Studierenden sollen] [die Grundkonzepte der objektorientierten Programmierung] mit der Standardnotation UML beschreiben können."

[Modulbeschreibungen\MH24; Position: 14 - 14]

430.

"[Die Studierenden sollen] die Prinzipien der objektorientierten Programmierung in Java umsetzen können."

[Modulbeschreibungen\MH24; Position: 15 - 15]

431.

"Die Studierenden sollen die grundlegenden Datenstrukturen und Algorithmen kennen lernen."

[Modulbeschreibungen\MH24; Position: 17 - 17]

432.

"[Die Studierenden sollen] [die] Wechselwirkungen [grundlegender Datenstrukturen und Algorithmen], insbesondere unter Laufzeit- und Speicherplatzbetrachtungen, auf praktische Beispiele anwenden können."

[Modulbeschreibungen\MH24; Position: 19 - 19]

433.

"[Die Studierenden sollen] die wesentlichen Such- und Sortieralgorithmen nach Anwendungsszenarien beurteilen."

[Modulbeschreibungen\MH24; Position: 21 - 21]

434.

"[Die Studierenden sollen] [die wesentlichen Such- und Sortieralgorithmen nach Anwendungsszenarien] auswählen können."

[Modulbeschreibungen\MH24; Position: 22 - 22]

435.

"[Die Studierenden sollen] die grundlegenden Datenstrukturen und Algorithmen in Java umsetzen können."

[Modulbeschreibungen\MH24; Position: 23 - 23]

436.

"Die Studierenden sollen die objektorientierten Konzepte in der Phase Analyse anwenden können."

[Modulbeschreibungen\MH24; Position: 25 - 25]

437.

"[Die Studierenden sollen die objektorientierten Konzepte in der Phase] Entwurf anwenden können."

[Modulbeschreibungen\MH24; Position: 26 - 26]

438.

"[Die Studierenden sollen] die objektorientierten Konzepte mit der Standardnotation UML beschreiben können."

[Modulbeschreibungen\MH24; Position: 27 - 27]

439.

"[Die Studierenden sollen] wissen, wie Sie am besten beim Erstellen objektorientierter Modelle vorgehen"

[Modulbeschreibungen\MH24; Position: 28 - 28]

440.

"[Die Studierenden sollen] wissen, wie sie gute von schlechten Modellen unterscheiden können."

[Modulbeschreibungen\MH24; Position: 29 - 29]

441.

"[Die Studierenden sollen] die objektorientierten Konzepte in Java umsetzen können."

[Modulbeschreibungen\MH24; Position: 30 - 30]

442.

"[Die Studierenden sollen] Analyse- und Entwurfsmuster anwenden können."

[Modulbeschreibungen\MH24; Position: 31 - 31]

443.

"[Die Studierenden sollen] die Bedeutung der Software-Prüfung kennen lernen."

[Modulbeschreibungen\MH24; Position: 32 - 32]

444.

"[Die Studierenden sollen] [die wesentlichen Testverfahren] anwenden können."

[Modulbeschreibungen\MH24; Position: 34 - 34]

445.

"[Studierende sollten nach der Veranstaltung in der Lage sein,] rekursive Algorithmen zu entwerfen."

[Modulbeschreibungen\MH24; Position: 39 - 39]

446.

"[Studierende sollten nach der Veranstaltung in der Lage sein,] rekursive Datenstrukturen zu entwerfen."

[Modulbeschreibungen\MH24; Position: 40 - 40]

447.

"[Studierende sollten nach der Veranstaltung in der Lage sein,] Funktionen höherer Ordnung zu erkennen."

[Modulbeschreibungen\MH24; Position: 41 - 41]

448.

"[Studierende sollten nach der Veranstaltung in der Lage sein,] [Funktionen höherer Ordnung] anzuwenden."

[Modulbeschreibungen\MH24; Position: 42 - 42]

449.

"[Studierende sollten nach der Veranstaltung in der Lage sein,] [funktionale Konzepte in anderen Programmiersprachen und Frameworks, z. B. Java Streams] effektiv zu nutzen."

[Modulbeschreibungen\MH24; Position: 44 - 44]

450.

"[Die Studierenden sollen] Algorithmen für vorgegebene Aufgabestellungen durch Einsatz der Algorithmen-Entwurfsmuster entwickeln können."

[Modulbeschreibungen\MH24; Position: 48 - 48]

451.

"[Die Studierenden sollen] Algorithmen hinsichtlich Korrektheit analysieren können."

[Modulbeschreibungen\MH24; Position: 49 - 49]

452.

"[Die Studierenden sollen] Algorithmen hinsichtlich Laufzeit analysieren können."

[Modulbeschreibungen\MH24; Position: 50 - 50]

453.

"[Die Studierenden sollen] Einsatz und Bedeutung effizienter Algorithmen in verschiedenen Anwendungsbereichen kennen lernen."

[Modulbeschreibungen\MH24; Position: 51 - 51]

454.

"Die Veranstaltung führt auch grundlegende Techniken ein, mit denen Programme verständlicher und einfacher wartbar gemacht werden, und Fehler während der Entwicklung erkannt werden können."

[Modulbeschreibungen\MH25; Position: 8 - 8]

455.

"In beiden Vorlesungen und Übungen werden Standard-Entwicklungswerkzeuge der verwendeten Programmiersprache verwendet und deren Verwendung eingeübt."

[Modulbeschreibungen\MH25; Position: 14 - 14]

456.

"Die Studierenden kennen mindestens zwei Programmierparadigmen (imperativ und objektorientiert)."

[Modulbeschreibungen\MH25; Position: 15 - 15]

457.

"Die Studierenden kennen grundlegende Datenstrukturen und Algorithmen."

[Modulbeschreibungen\MH25; Position: 17 - 17]

458.

"[Die Studierenden] können [...] [grundlegende Datenstrukturen und Algorithmen] umsetzen."

[Modulbeschreibungen\MH25; Position: 18 - 18]

459.

"[Die Studierenden] können [...] [grundlegende Datenstrukturen und Algorithmen] testen."

[Modulbeschreibungen\MH25; Position: 19 - 19]

460.

"Grundlegende Modellierungskonzepte sind bekannt."

[Modulbeschreibungen\MH25; Position: 20 - 20]

461.

"[Die Studierenden] kennen Zeiger- und Adresskonzepte in der Programmiersprache C/C++."

[Modulbeschreibungen\MH25; Position: 23 - 23]

462.

"[Die Studierenden] kennen das funktionale Paradigma."

[Modulbeschreibungen\MH25; Position: 24 - 24]

463.

"[Die Studierenden] können [...] [das funktionale Paradigma] praktisch umsetzen."

[Modulbeschreibungen\MH25; Position: 25 - 25]

464.

"[Die Studierenden] können Rekursion in verschiedenen Paradigmen anwenden."

[Modulbeschreibungen\MH25; Position: 26 - 26]

465.

"Laufzeitanalysen entwickeln."

[Modulbeschreibungen\MH26; Position: 7 - 7]

466.

"[Laufzeitanalysen] anwenden können."

[Modulbeschreibungen\MH26; Position: 9 - 9]

467.

"Kennenlernen von Algorithmen für die Programmiersprachen I-Vorlesung."

[Modulbeschreibungen\MH26; Position: 10 - 10]

468.

"Die Studierenden sollen am Ende des Semesters in der Lage sein, einfache Algorithmen und Datenstrukturen in Java umzusetzen."

[Modulbeschreibungen\MH26; Position: 14 - 14]

469.

"[Nach erfolgreicher Absolvierung dieses Moduls können die Studierenden fortgeschrittene Konzepte objektorientierter Programmierung in Java] anwenden."

[Modulbeschreibungen\MH26; Position: 19 - 19]

470.

"[Die Studierenden] sind in der Lage, objektorientierte Lösungen zu entwerfen."

[Modulbeschreibungen\MH26; Position: 20 - 20]

471.

"[Die Studierenden] [sind in der Lage, objektorientierte Lösungen] umzusetzen."

[Modulbeschreibungen\MH26; Position: 21 - 21]

472.

"Basierend auf dem im Modul Informatik 1 gewonnenen theoretischen Wissen zu Datenstrukturen können die Studierenden verschiedene Datenstrukturen implementieren."

[Modulbeschreibungen\MH26; Position: 22 - 22]

473.

"[Basierend auf dem im Modul Informatik 1 gewonnenen theoretischen Wissen zu Datenstrukturen können die Studierenden verschiedene Datenstrukturen] anwenden."

[Modulbeschreibungen\MH26; Position: 23 - 23]

474.

"[Die Studierenden] sind außerdem in der Lage, die für ein Anwendungsszenario geeignete Datenstruktur auszuwählen."

[Modulbeschreibungen\MH26; Position: 24 - 24]

475.

"Neben der Vertiefung der objektorientierten Programmierung erlernen die Studierenden die grundlegenden Konzepte der Programmiersprache C."

[Modulbeschreibungen\MH26; Position: 25 - 25]

476.

"Die Studierenden können Konzepte und Eigenheiten der imperativen Programmierung in C benennen."

[Modulbeschreibungen\MH26; Position: 26 - 26]

477.

"[Die Studierenden] sind in der Lage [...] Kenntnisse [zu Unterschieden sowie Vor- und Nachteilen zu objektorientierten und typsicheren Sprachen] in der Implementierung von C-Programmen anzuwenden."

[Modulbeschreibungen\MH26; Position: 28 - 28]

478.

"Die Studierenden sollen die wichtigsten Vorgehensmodelle der Softwareentwicklung kennen."

[Modulbeschreibungen\MH26; Position: 31 - 31]

479.

"Des Weiteren sollen [die Studierenden] lernen wie Anforderungsanalyse von komplexen Anwendungssystemen durchzuführen [ist], insbesondere in der Kommunikation mit den Anwendern."

[Modulbeschreibungen\MH26; Position: 33 - 33]

480.

"Des Weiteren sollen [die Studierenden] lernen wie objektorientierte Modellierung von komplexen Anwendungssystemen durchzuführen [ist], insbesondere in der Kommunikation mit den Anwendern."

[Modulbeschreibungen\MH26; Position: 34 - 34]

481.

"[Die Studierenden] sollen zudem die Hauptproblematiken bei der Durchführung von SW-Projekten kennen und umgehen lernen."

[Modulbeschreibungen\MH26; Position: 35 - 35]

482.

"Korrektheit beweisen."

[Modulbeschreibungen\MH27; Position: 7 - 7]

483.

"Laufzeit abschätzen."

[Modulbeschreibungen\MH27; Position: 8 - 8]

484.

"Einfache Programmiersprachen implementieren können."

[Modulbeschreibungen\MH27; Position: 12 - 12]

485.

"Routine im Umgang mit Interpretern und Übersetzern."

[Modulbeschreibungen\MH27; Position: 15 - 15]

486.

"Die Studierenden lernen die Grundprinzipien der imperativen/objektorientierten Programmierung kennen."

[Modulbeschreibungen\MH27; Position: 17 - 17]

487.

"In dieser Vorlesung lernen sie wie Rechner Programme ausführen."

[Modulbeschreibungen\MH27; Position: 19 - 19]

488.

"[In dieser Vorlesung lernen sie] kleinere, wohlstrukturierte Programme in C zu schreiben."

[Modulbeschreibungen\MH27; Position: 21 - 21]

489.

"[In dieser Vorlesung lernen sie] Mitttelgroße objektorientierte Systeme in Java zu implementieren."

[Modulbeschreibungen\MH27; Position: 22 - 22]

490.

"[In dieser Vorlesung lernen sie] [Mitttelgroße objektorientierte Systeme in Java] zu testen."

[Modulbeschreibungen\MH27; Position: 23 - 23]

491.

"Die Studierenden lernen die wichtigsten Methoden des Entwurfs von Algorithmen und Datenstrukturen kennen: Teile-und-Herrsche, Dynamische Programmierung, inkrementelle Konstruktion, „Greedy“, Dezimierung, Hierarchisierung, Randomisierung."

[Modulbeschreibungen\MH27; Position: 26 - 26]

492.

"[Die Studierenden] lernen Algorithmen und Datenstrukturen bzgl. Zeit- und Platzverbrauch für das übliche RAM Maschinenmodell zu analysieren."

[Modulbeschreibungen\MH27; Position: 27 - 27]

493.

"[Die Studierenden] lernen [...] [auf Basis der Analyse von] [Algorithmen und Datenstrukturen bzgl. Zeit- und Platzverbrauch für das übliche RAM Maschinenmodell] [diese Algorithmen] zu vergleichen."

[Modulbeschreibungen\MH27; Position: 28 - 28]

494.

"Sie lernen verschiedene Arten der Analyse (schlechtester Fall, amortisiert, erwartet) einzusetzen."

[Modulbeschreibungen\MH27; Position: 29 - 29]

495.

"Die Studierenden lernen wichtige effiziente Datenstrukturen und Algorithmen kennen."

[Modulbeschreibungen\MH27; Position: 30 - 30]

496.

"Sie sollen die Fähigkeit erwerben, vorhandene Methoden durch theoretische Analysen und Abwägungen für ihre Verwendbarkeit in tatsächlich auftretenden Szenarien zu prüfen."

[Modulbeschreibungen\MH27; Position: 31 - 31]

497.

"Ferner sollen die Studierenden die Fähigkeit trainieren, Algorithmen unter dem Aspekt von Performanzgarant zu entwickeln."

[Modulbeschreibungen\MH27; Position: 32 - 32]

498.

"Ferner sollen die Studierenden die Fähigkeit trainieren, Datenstrukturen unter dem Aspekt von Performanzgarant ein zu entwickeln."

[Modulbeschreibungen\MH27; Position: 33 - 33]

499.

"Ferner sollen die Studierenden die Fähigkeit trainieren, Algorithmen unter dem Aspekt von Performanzgarant anzupassen."

[Modulbeschreibungen\MH27; Position: 34 - 34]

500.

"Ferner sollen die Studierenden die Fähigkeit trainieren, Datenstrukturen unter dem Aspekt von Performanzgarant anzupassen."

[Modulbeschreibungen\MH27; Position: 35 - 35]

501.

"Am Ende dieses Moduls kennt jeder Kursteilnehmer den grundlegenden Aufbau und die Funktionsweise eines Rechnersystems."

[Modulbeschreibungen\MH28; Position: 5 - 5]

502.

"[Am Ende dieses Moduls] kann [jeder Kursteilnehmer] die Verfahren zur rechnerinternen Darstellung von Daten und Zahlen erläutern."

[Modulbeschreibungen\MH28; Position: 6 - 6]

503.

"Die Studierenden kennen darüber hinaus wesentliche Konzepte und Verfahren moderner Programmiersprachen, angefangen von einfachen Datentypen, über Kontrollstrukturen bis hin zu den Themen Klassen, Objekte und Vererbung."

[Modulbeschreibungen\MH28; Position: 7 - 7]

504.

"Somit ist es den Studierenden möglich, einfache praxisrelevante Problemstellungen selbständig zu analysieren."

[Modulbeschreibungen\MH28; Position: 9 - 9]

505.

"[Somit ist es den Studierenden möglich, einfache praxisrelevante Problemstellungen] anschließend programmiertechnisch umzusetzen."

[Modulbeschreibungen\MH28; Position: 10 - 10]

506.

"Die Studierenden verfügen insbesondere über vertiefte Kenntnisse bezogen auf die Bereiche interne und externe Datenverwaltung (einschließlich der Ein- und Ausgabe)."

[Modulbeschreibungen\MH28; Position: 16 - 16]

507.

"[Die Studierenden verfügen insbesondere über vertiefte Kenntnisse bezogen auf die Bereiche] Abschätzung des Aufwands einfacher Algorithmen (z.B. für Such- und Sortierprobleme)."

[Modulbeschreibungen\MH28; Position: 17 - 17]

508.

"Jeder Teilnehmer kann Unterschiede, Vor- und Nachteile von speziellen Datenstrukturen (Listen, Felder, Assoziativ-Speicher) benennen."

[Modulbeschreibungen\MH28; Position: 18 - 18]

509.

"Die Teilnehmer sind in der Lage, systematisch nach Fehlern in Programmen zu suchen bzw. diese zu validieren."

[Modulbeschreibungen\MH28; Position: 19 - 19]

510.

"[Die Teilnehmer] kennen Werkzeuge wie Debugger und Profiler."

[Modulbeschreibungen\MH28; Position: 20 - 20]

511.

"[Die Teilnehmer] verwenden Werkzeuge wie Debugger und Profiler."

[Modulbeschreibungen\MH28; Position: 21 - 21]

512.

"Gegenstand des Programmierbeleges ist der Entwurf eines Programms für eine umfangreichere, selbstgewählte Aufgabenstellung."

[Modulbeschreibungen\MH28; Position: 24 - 24]

513.

"Gegenstand des Programmierbeleges ist die Realisierung eines Programms für eine umfangreichere, selbstgewählte Aufgabenstellung."

[Modulbeschreibungen\MH28; Position: 25 - 25]

514.

"[Jeder Studierende kann präzise die von ihm bearbeitete Aufgabe] abgrenzen."

[Modulbeschreibungen\MH28; Position: 27 - 27]

515.

"[Jeder Studierende] kann er zu lösenden (Teil-) Probleme identifizieren."

[Modulbeschreibungen\MH28; Position: 28 - 28]

516.

"Darüber hinaus wird jeder Teilnehmer befähigt, einen einfachen Programm-Entwurf zu konstruieren."

[Modulbeschreibungen\MH28; Position: 29 - 29]

517.

"[Darüber hinaus wird jeder Teilnehmer befähigt, einen einfachen Programm-Entwurf] [...] selbständig programmtechnisch umzusetzen."

[Modulbeschreibungen\MH28; Position: 30 - 30]

518.

"Jeder Studierende führt eigenverantwortlich einen Programm-Test durch."

[Modulbeschreibungen\MH28; Position: 31 - 31]

519.

"[Jeder Studierende] erstellt eine Programm-Dokumentation."

[Modulbeschreibungen\MH28; Position: 32 - 32]

520.

"Die Teilnehmer dieses Moduls sind in der Lage, aus einer breiten Auswahl an Standardlösungen (Fachkompetenz) die am besten geeignete auszuwählen (Methodenkompetenz)."

[Modulbeschreibungen\MH28; Position: 38 - 38]

521.

"[Die Teilnehmer dieses Moduls sind in der Lage, aus einer breiten Auswahl an Standardlösungen (Fachkompetenz) die am besten geeignete] selbständig zu implementieren (Methodenkompetenz)."

[Modulbeschreibungen\MH28; Position: 39 - 39]

522.

"Die Studierenden können konkrete Algorithmen hinsichtlich ihrer unterschiedlichen Eigenschaften und Eignungen analysieren."

[Modulbeschreibungen\MH28; Position: 42 - 42]

523.

"[Die Studierenden können konkrete Algorithmen hinsichtlich ihrer unterschiedlichen Eigenschaften und Eignungen] vergleichen."

[Modulbeschreibungen\MH28; Position: 43 - 43]

524.

"[Die Studierenden können konkrete Algorithmen hinsichtlich ihrer unterschiedlichen Eigenschaften und Eignungen] auswählen."

[Modulbeschreibungen\MH28; Position: 44 - 44]

525.

"Die Lerneinheit ist darauf ausgerichtet, den Studierenden ein breites Spektrum informatischer Werkzeuge zu eröffnen, die die Teilnehmer [...] selbständig einsetzen (Befähigung Teamkompetenz)."

[Modulbeschreibungen\MH28; Position: 45 - 45]

526.

"[Die Absolventen] kennen die Konzepte der Programmiersprache C mit ihren Vorteilen und Schwächen."

[Modulbeschreibungen\MH28; Position: 51 - 51]

527.

"[Die Absolventen] können [...] [die C-Standardbibliothek und weitere ausgewählte Bibliotheken] (und auch andere) Funktionsbibliotheken selbständig einsetzen und anwenden."

[Modulbeschreibungen\MH28; Position: 54 - 54]

528.

"Die Studierenden besitzen Kenntnisse über Grundlagen der imperativen Programmierung (Syntaxdiagramme, EBNF, Funktionen, Module, Datenstrukturen)."

[Modulbeschreibungen\MH29; Position: 5 - 5]

529.

"[Die Studierenden] können [...] [Kenntnisse über Grundlagen der imperativen Programmierung (Syntaxdiagramme, EBNF, Funktionen, Module, Datenstrukturen)] zur Formulierung von Algorithmen für klassische Problemstellungen (Sortier- und Suchverfahren, Algorithmen auf Bäumen und Graphen) verwenden."

[Modulbeschreibungen\MH29; Position: 6 - 6]

530.

"Die Studierenden kennen verschiedene Klassen von Algorithmen (divide-and-conquer, dynamisches Programmieren, Iteration versus Rekursion, backtracking)."

[Modulbeschreibungen\MH29; Position: 7 - 7]

531.

"Als erste Schritte zu Komplexitätsanalysen können sie außerdem Algorithmen hinsichtlich ihres Laufzeitverhaltens analysieren."

[Modulbeschreibungen\MH29; Position: 8 - 8]

532.

"Die Studierenden besitzen Kenntnisse des funktionalen Programmierens."

[Modulbeschreibungen\MH29; Position: 10 - 10]

533.

"[Die Studierenden] können [...] [Kenntnisse des funktionalen Programmierens] praxisnah einsetzen."

[Modulbeschreibungen\MH29; Position: 11 - 11]

534.

"Die Studierenden besitzen die Fähigkeiten, formale Werkzeuge (Grundlagen der Berechnung, Übersetzung von Programmkonstruktoren, Programmtransformationen, Verifikation von Programmeigenschaften) zu benutzen."

[Modulbeschreibungen\MH29; Position: 12 - 12]

535.

"[Die Studierenden besitzen die Fähigkeiten, formale Werkzeuge (Grundlagen der Berechnung, Übersetzung von Programmkonstruktoren, Programmtransformationen, Verifikation von Programmeigenschaften)] zu entwickeln."

[Modulbeschreibungen\MH29; Position: 13 - 13]

536.

"[Die Studierenden] sind in der Lage, einfache Aufgaben mittels eines Rechnersimulationsprogramms zu lösen."

[Modulbeschreibungen\MH30; Position: 6 - 6]

537.

"[Einfache Codierungen können die Studierenden] bearbeiten."

[Modulbeschreibungen\MH30; Position: 8 - 8]

538.

"[Einfache Codierungen können die Studierenden] selber erstellen."

[Modulbeschreibungen\MH30; Position: 9 - 9]

539.

"Darüber hinaus kennen [die Studierenden] grundlegende Problemlösungs-und Programmkonstruktionsmethoden der imperativen Programmierung."

[Modulbeschreibungen\MH30; Position: 13 - 13]

540.

"[Darüber hinaus] können [...] [die Studierenden] [grundlegende Problemlösungs-und Programmkonstruktionsmethoden der imperativen Programmierung] anwenden."

[Modulbeschreibungen\MH30; Position: 14 - 14]

541.

"Die Studenten haben Kenntnisse im Erstellen von Internetsoftware."

[Modulbeschreibungen\MH30; Position: 17 - 17]

542.

"[Die Studierenden] können komplexe, rekursive Skripte in Unix schreiben."

[Modulbeschreibungen\MH30; Position: 20 - 20]

543.

"Die Studierenden kennen die wichtigsten Konzepte der objektorientierten Programmierung."

[Modulbeschreibungen\MH30; Position: 22 - 22]

544.

"Außerdem verfügen [die Studierenden] über Kenntnisse der wichtigsten dynamischen Datenstrukturen."

[Modulbeschreibungen\MH30; Position: 24 - 24]

545.

"[Die Studierenden] sind in der Lage [die wichtigsten dynamischen Datenstrukturen] zu implementieren."

[Modulbeschreibungen\MH30; Position: 25 - 25]

546.

"[Die Studierenden] sind in der Lage [die wichtigsten dynamischen Datenstrukturen] anzuwenden."

[Modulbeschreibungen\MH30; Position: 26 - 26]

547.

"Schließlich können [die Studierenden] auch die Datenstrom-Ein-und Ausgabe anwenden."

[Modulbeschreibungen\MH30; Position: 27 - 27]

548.

"Die Teilnehmer kennen grundlegende und wichtige Algorithmen."

[Modulbeschreibungen\MH30; Position: 29 - 29]

549.

"[Die Teilnehmer] sind in der Lage [...] [grundlegende und wichtige Algorithmen] anzuwenden."

[Modulbeschreibungen\MH30; Position: 30 - 30]

550.

"[Die Studierenden sind in der Lage, [...] die objektorientierte Programmierung in C++] [...] anzuwenden."

[Modulbeschreibungen\MH30; Position: 33 - 33]

551.

"Befähigung zur Lösung von algorithmischen Aufgaben."

[Modulbeschreibungen\MH31; Position: 6 - 6]

552.

"Befähigung zum Design von Datenstrukturen."

[Modulbeschreibungen\MH31; Position: 7 - 7]

553.

"Befähigung zur Lösung von algorithmischen Aufgaben."

[Modulbeschreibungen\MH31; Position: 11 - 11]

554.

"Befähigung zum Design von Datenstrukturen."

[Modulbeschreibungen\MH31; Position: 12 - 12]

555.

"Erlernen von Techniken für die Prozess- und Datenmodellierung auf fachkonzeptueller Ebene."

[Modulbeschreibungen\MH31; Position: 17 - 17]

556.

"Erlernen von objektorientierten Modellierungstechniken auf DV-konzeptueller Ebene."

[Modulbeschreibungen\MH31; Position: 18 - 18]

557.

"Kenntnisse in zwei (weiteren) Paradigmen."

[Modulbeschreibungen\MH31; Position: 22 - 22]

558.

"Fertigkeiten im Umgang mit deklarativen Programmierumgebungen."

[Modulbeschreibungen\MH31; Position: 23 - 23]

559.

"Kenntnis existierender Programmierparadigmen."

[Modulbeschreibungen\MH32; Position: 5 - 5]

560.

"[Kenntnis existierender] Laufzeitmodelle."

[Modulbeschreibungen\MH32; Position: 6 - 6]

561.

"Studierende sollen am Bsp. der Programmiersprache JAVA die folgenden Aspekte einer imperativen/prozeduralen Programmiersprache beherrschen: Sichere Anwendung des Konzepts primitiver Datentypen und Variablen."

[Modulbeschreibungen\MH32; Position: 7 - 7]

562.

"[Studierende sollen am Bsp. der Programmiersprache JAVA die folgenden Aspekte einer imperativen/prozeduralen Programmiersprache beherrschen:] Sichere Anwendung des Konzepts von Operatoren und Ausdrücken."

[Modulbeschreibungen\MH32; Position: 8 - 8]

563.

"[Studierende sollen am Bsp. der Programmiersprache JAVA die folgenden Aspekte einer imperativen/prozeduralen Programmiersprache beherrschen:] Sichere Anwendung von Kontrollstrukturen."

[Modulbeschreibungen\MH32; Position: 9 - 9]

564.

"[Studierende sollen am Bsp. der Programmiersprache JAVA die folgenden Aspekte einer imperativen/prozeduralen Programmiersprache beherrschen:] Fähigkeit zur problemorientierten Definition und Nutzung von Routinen/Methoden."

[Modulbeschreibungen\MH32; Position: 10 - 10]

565.

"[Studierende sollen am Bsp. der Programmiersprache JAVA die folgenden Aspekte einer imperativen/prozeduralen Programmiersprache beherrschen:] Fähigkeit zur Definition von problemspezifischen Referenztypen."

[Modulbeschreibungen\MH32; Position: 11 - 11]

566.

"[Studierende sollen am Bsp. der Programmiersprache JAVA die folgenden Aspekte einer imperativen/prozeduralen Programmiersprache beherrschen:] Kenntnis der Datenstrukturen Liste, Stack, Map und Set."

[Modulbeschreibungen\MH32; Position: 13 - 13]

567.

"[Studierende sollen am Bsp. der Programmiersprache JAVA die folgenden Aspekte einer imperativen/prozeduralen Programmiersprache beherrschen:] Sichere Anwendung der Datenstrukturen Liste, Stack, Map und Set."

[Modulbeschreibungen\MH32; Position: 14 - 14]

568.

"[Studierende sollen am Bsp. der Programmiersprache JAVA die folgenden Aspekte einer imperativen/prozeduralen Programmiersprache beherrschen:] Kenntnis des Prinzips rekursiver Programmierung."

[Modulbeschreibungen\MH32; Position: 15 - 15]

569.

"[Studierende sollen am Bsp. der Programmiersprache JAVA die folgenden Aspekte einer imperativen/prozeduralen Programmiersprache beherrschen:] Sichere Anwendung des Prinzips rekursiver Programmierung."

[Modulbeschreibungen\MH32; Position: 16 - 16]

570.

"[Studierende sollen am Bsp. der Programmiersprache JAVA die folgenden Aspekte einer imperativen/prozeduralen Programmiersprache beherrschen:] Kenntnis der rekursiven Datenstruktur Baum."

[Modulbeschreibungen\MH32; Position: 17 - 17]

571.

"[Studierende sollen am Bsp. der Programmiersprache JAVA die folgenden Aspekte einer imperativen/prozeduralen Programmiersprache beherrschen:] Sichere Anwendung der rekursiven Datenstruktur Baum."

[Modulbeschreibungen\MH32; Position: 18 - 18]

572.

"[Studierende sollen am Bsp. der Programmiersprache JAVA die folgenden Aspekte einer objektorientierten Programmiersprache beherrschen:] Darstellung objektorientierter Sachverhalte mittels UML Klassendiagrammen."

[Modulbeschreibungen\MH32; Position: 24 - 24]

573.

"Diese Lehrveranstaltung soll den Lernenden die engen inneren Zusammenhänge zwischen den scheinbar so verschiedenen Gebieten Mengenlehre, Logik, Relationen, Wahrscheinlichkeitsrechnung und Algorithmen klarmachen."

[Modulbeschreibungen\MH32; Position: 28 - 28]

574.

"[Die Studierenden] sind dann in der Lage, logische Zusammenhänge in logische Formeln umzusetzen und umgekehrt sowie formale Vereinfachungen von logischen Formeln vorzunehmen."

[Modulbeschreibungen\MH32; Position: 30 - 30]

575.

"[Die Studierenden] können einfachere Aufgabenstellungen aus [...] [den Gebieten der Kombinatorik und Wahrscheinlichkeitsrechnung] sicher lösen."

[Modulbeschreibungen\MH32; Position: 32 - 32]

576.

"Ganz allgemein sollen [die Studierenden] Problemstellungen aus dem Gebiet der Informatik analysieren und deren mathematischen Kern erkennen können."

[Modulbeschreibungen\MH32; Position: 34 - 34]

577.

"Studierende sollen am Bsp. der Programmiersprache JAVA die folgenden Aspekte einer objektorientierten Programmiersprache beherrschen: Fähigkeit das OO Konzept der Polymorphie zur Wiederverwendung OO Software einsetzen zu können."

[Modulbeschreibungen\MH32; Position: 38 - 38]

578.

"[Studierende sollen am Bsp. der Programmiersprache JAVA die folgenden Aspekte einer objektorientierten Programmiersprache beherrschen:] Fähigkeit vorgegebene generische Datentypen instanzieren und selber definieren zu können."

[Modulbeschreibungen\MH32; Position: 39 - 39]

579.

"[Studierende sollen am Bsp. der Programmiersprache JAVA die folgenden Aspekte einer objektorientierten Programmiersprache beherrschen:] Fähigkeit Testfälle entwickeln zu können, die eine vorgegebene Code Coverage erreichen."

[Modulbeschreibungen\MH32; Position: 41 - 41]

580.

"[Studierende sollen am Bsp. der Programmiersprache JAVA die folgenden Aspekte einer objektorientierten Programmiersprache beherrschen:] Fähigkeit Software unter Berücksichtigung objektorientierter Entwurfsprinzipien entwickeln zu können."

[Modulbeschreibungen\MH32; Position: 42 - 42]

581.

"Die Studierenden kennen effiziente Algorithmen und Datenstrukturen aus den wichtigsten Gebieten der Angewandten Informatik."

[Modulbeschreibungen\MH32; Position: 45 - 45]

582.

"[Die Studierenden sind in der Lage, Algorithmen] zu entwerfen."

[Modulbeschreibungen\MH32; Position: 48 - 48]

583.

"[Die Studierenden sind in der Lage, Datenstrukturen] zu entwerfen."

[Modulbeschreibungen\MH32; Position: 49 - 49]

584.

"[Die Studierenden sind in der Lage, Algorithmen und Datenstrukturen] zu implementieren."

[Modulbeschreibungen\MH32; Position: 50 - 50]

585.

"[Die Studierenden] [können Algorithmen und Datenstrukturen] [...] [auf Basis [...] von Analysen bzgl. ihrer Laufzeit und ihres Speicherverbrauchs] vergleichen."

[Modulbeschreibungen\MH32; Position: 52 - 52]

586.

"Die Studierenden verfügen über Kompetenzen zur Auswahl geeigneter Datenstrukturen und Algorithmen in der Softwareentwicklung zur Verbesserung der Effizienz und der Wartbarkeit der Programme sowie zur Verkürzung der Entwicklungszeit."

[Modulbeschreibungen\MH32; Position: 53 - 53]

587.

"[Die Studierenden verfügen über Kompetenzen] zur Anwendung geeigneter Datenstrukturen und Algorithmen in der Softwareentwicklung zur Verbesserung der Effizienz und der Wartbarkeit der Programme sowie zur Verkürzung der Entwicklungszeit."

[Modulbeschreibungen\MH32; Position: 54 - 54]

588.

"Kenntnisse verschiedener Programmierparadigmen."

[Modulbeschreibungen\MH33; Position: 6 - 6]

589.

"Fähigkeit, einfache Programme selbständig zu entwerfen"

[Modulbeschreibungen\MH33; Position: 9 - 9]

590.

"[Fähigkeit, einfache Programme selbständig] zu implementieren."

[Modulbeschreibungen\MH33; Position: 10 - 10]

591.

"Gute Java-Kenntnisse."

[Modulbeschreibungen\MH33; Position: 11 - 11]

592.

"Fähigkeit, Lösungen entsprechend allgemein anerkannter Qualitätsstandards zu entwerfen."

[Modulbeschreibungen\MH33; Position: 12 - 12]

593.

"Grundlegende Fähigkeit zur Anwendung von Techniken des sicheren Programmierens."

[Modulbeschreibungen\MH33; Position: 15 - 15]

594.

"[Beherrschen grundlegender Prinzipien und Methoden für] Implementierung von Algorithmen."

[Modulbeschreibungen\MH33; Position: 22 - 22]

595.

"[Beherrschen grundlegender Prinzipien und Methoden für] Analyse von Algorithmen."

[Modulbeschreibungen\MH33; Position: 23 - 23]

596.

"[Die Studierenden] [können syntaktische und semantische Beschreibungen von Programmiersprachen] anpassen."

[Modulbeschreibungen\MH33; Position: 28 - 28]

597.

"[Die Studierenden] [können syntaktische und semantische Beschreibungen von Programmiersprachen] erweitern."

[Modulbeschreibungen\MH33; Position: 29 - 29]

598.

"[Die Studierenden] können über die Unterstützung von Typsystemen für korrekte Programme argumentieren."

[Modulbeschreibungen\MH33; Position: 33 - 33]

599.

"[Die Studierenden] können zu vorgegebenen Aufgaben geeignete Programmiersprachen auswählen."

[Modulbeschreibungen\MH33; Position: 34 - 34]

600.

"Die Studierenden codieren Ganzzahlen, Gleitkommazahlen und textuelle Daten in Binärdarstellung oder decodieren sie aus dieser."

[Modulbeschreibungen\MH34; Position: 5 - 5]

601.

"[Die Studierenden] wählen geeignete Standarddatentypen (C) für Daten vorgegebener Anwendungsbeispiele aus."

[Modulbeschreibungen\MH34; Position: 6 - 6]

602.

"[Die Studierenden] beschreiben in eigenen Worten die numerischen Beschränkungen existierender Datentypen an Beispielen."

[Modulbeschreibungen\MH34; Position: 8 - 8]

603.

"[Die Studierenden] benennen die grundlegenden Kontrollstrukturen strukturierter Programmierung."

[Modulbeschreibungen\MH34; Position: 9 - 9]

604.

"[Die Studierenden] stellen [...] [die grundlegenden Kontrollstrukturen strukturierter Programmierung] korrekt im Struktogramm und in C-Syntax dar."

[Modulbeschreibungen\MH34; Position: 10 - 10]

605.

"Die Studierenden entwerfen einfache Algorithmen in Form von Struktogrammen zu vorgegebenen Aufgaben."

[Modulbeschreibungen\MH34; Position: 11 - 11]

606.

"[Die Studierenden] zerlegen komplexere verbale Aufgabenstellungen hierarchisch in Teilaufgaben (Top-Down-Entwurf)."

[Modulbeschreibungen\MH34; Position: 12 - 12]

607.

"[Die Studierenden] legen die [...] Datentypen, -strukturen und Funktionssignaturen [komplexerer verbaler Aufgabenstellungen und deren Teilaufgaben] sinnvoll fest."

[Modulbeschreibungen\MH34; Position: 13 - 13]

608.

"[Die Studierenden] implementieren arbeitsteilig eine Lösung."

[Modulbeschreibungen\MH34; Position: 14 - 14]

609.

"[Die Studierenden] halten bei der Lösung von Programmierproblemen die Konventionen der Quelldateiorganisation in C ein."

[Modulbeschreibungen\MH34; Position: 15 - 15]

610.

"[Die Studierenden] begründen die Auswahl geeigneter aggregierender Datenstrukturen (wie Arrays, verkettete Listen, Bäume) an Hand ihres Laufzeitverhaltens entsprechend dem Anforderungsprofil einer Anwendungsaufgabe."

[Modulbeschreibungen\MH34; Position: 16 - 16]

611.

"[Die Studierenden] passen [...] [aggregierender Datenstrukturen] dem konkreten Einsatzfall an."

[Modulbeschreibungen\MH34; Position: 17 - 17]

612.

"Die Studierenden können die grundlegenden Prinzipien der OOP benennen."

[Modulbeschreibungen\MH34; Position: 19 - 19]

613.

"[Die Studierenden können] Konzepte wie Referenzen, abstrakte Klassen, virtuelle Methoden, Klassenkonstanten, -attribute und -methoden richtig einordnen."

[Modulbeschreibungen\MH34; Position: 21 - 21]

614.

"[Die Studierenden können] Aggregations- und Kompositionsbeziehungen zwischen Klassen in geeigneter Weise implementieren."

[Modulbeschreibungen\MH34; Position: 23 - 23]

615.

"[Die Studierenden können] den Lebenszyklus von Objekten konsistent gestalten (Konstruktoren, Destruktor, Kopie, Wertzuweisung, dynamische Inhalte)."

[Modulbeschreibungen\MH34; Position: 24 - 24]

616.

"[Die Studierenden können] aus einer verbalen Aufgabenstellung ein sinnvolles System von Objektklassen ableiten."

[Modulbeschreibungen\MH34; Position: 25 - 25]

617.

"[Die Studierenden können] [aus einer verbalen Aufgabenstellung] passende Schnittstellen entwerfen."

[Modulbeschreibungen\MH34; Position: 26 - 26]

618.

"[Die Studierenden können] [aus einer verbalen Aufgabenstellung] [die Auswahl passender Schnittstellen] begründen."

[Modulbeschreibungen\MH34; Position: 27 - 27]

619.

"[Die Studierenden können] ein Projekt Sourcecode-verwaltet im Team implementieren."

[Modulbeschreibungen\MH34; Position: 28 - 28]

620.

"[Die Studierenden können] analysieren, wo Container-Templates der C++-Standardbibliotheken effizient eingesetzt werden können."

[Modulbeschreibungen\MH34; Position: 29 - 29]

621.

"[Die Studierenden können] [...] [Container-Templates der C++-Standardbibliotheken] in korrekter Weise umsetzen."

[Modulbeschreibungen\MH34; Position: 30 - 30]

622.

"Die Studierenden können eine Problemstellung in ein objektorientiertes Modell überführen."

[Modulbeschreibungen\MH34; Position: 32 - 32]

623.

"[Die Studierenden können eine Problemstellung in ein] korrektes Java-Programm überführen."

[Modulbeschreibungen\MH34; Position: 33 - 33]

624.

"[Die Studierenden können] [Konzepte objektorientierter Programmierung] mit den korrekten Fachbegriffen wiedergeben."

[Modulbeschreibungen\MH34; Position: 35 - 35]

625.

"[Die Studierenden können] [Konzepte objektorientierter Programmierung] in der Programmierung mit Java anwenden."

[Modulbeschreibungen\MH34; Position: 36 - 36]

626.

"[Die Studierenden können] Vor- und Nachteile von objektorientierten Konzepten benennen."

[Modulbeschreibungen\MH34; Position: 37 - 37]

627.

"[Die Studierenden können] [Vor- und Nachteile von objektorientierten Konzepten] in der Praxis entsprechend sinnvoll einsetzen."

[Modulbeschreibungen\MH34; Position: 38 - 38]

628.

"[Die Studierenden können] Gemeinsamkeiten und Unterschiede hinsichtlich der Sprachphilosophie und Konzepte von Java und C/C++ benennen."

[Modulbeschreibungen\MH34; Position: 39 - 39]

629.

"[Die Studierenden können] die gegebenen Datenstrukturen und Algorithmen der Java Collections API erläutern."

[Modulbeschreibungen\MH34; Position: 41 - 41]

630.

"[Die Studierenden können] [die gegebenen Datenstrukturen der Java Collections API] in eigenen Programmen verwenden."

[Modulbeschreibungen\MH34; Position: 42 - 42]

631.

"[Die Studierenden können] [die gegebenen Algorithmen der Java Collections API] in eigenen Programmen verwenden."

[Modulbeschreibungen\MH34; Position: 43 - 43]

632.

"[Die Studierenden können] gängige Entwicklungsumgebungen für die Entwicklung mit Java benennen."

[Modulbeschreibungen\MH34; Position: 44 - 44]

633.

"[Die Studierenden können] [gängige Entwicklungsumgebungen für die Entwicklung mit Java] verwenden."

[Modulbeschreibungen\MH34; Position: 45 - 45]

634.

"[Die Studierenden können] die Konzepte und Notwendigkeit von Unit-Tests erläutern."

[Modulbeschreibungen\MH34; Position: 46 - 46]

635.

"[Die Studierenden können] Unit-Tests erfolgreich bei der Software-Entwicklung einsetzen."

[Modulbeschreibungen\MH34; Position: 47 - 47]

636.

"[Die Studierenden können] die Notwendigkeit eines Versionskontrollsystems erläutern."

[Modulbeschreibungen\MH34; Position: 48 - 48]

637.

"[Die Studierenden können] ein [...] [Versionskontrollsystem] praktisch einsetzen."

[Modulbeschreibungen\MH34; Position: 49 - 49]

638.

"Fachkompetenz: Die Studierenden verfügen über Kenntnisse zu algorithmischen Modellen, Basisalgorithmen."

[Modulbeschreibungen\MH35; Position: 5 - 5]

639.

"[Fachkompetenz:] [Die Studierenden] können grundlegende Algorithmen nach einer Problembeschreibung systematisch durch Aufstellen eines Induktionsbeweises für die Lösbarkeit einer gegebenen Aufgabenstellung herleiten."

[Modulbeschreibungen\MH35; Position: 7 - 7]

640.

"[Fachkompetenz:] [Die Studierenden] [können grundlegende Algorithmen] [...] durch Angabe und Abschätzen von Rekurrenzgleichungen in Ihrer Laufzeitkomplexität bewerten."

[Modulbeschreibungen\MH35; Position: 8 - 8]

641.

"Methodenkompetenz: [Die Studierenden] sind in der Lage, Algorithmen hinsichtlich ihrer Eigenschaften und Anwendbarkeit für konkrete Problemstellungen zu identifizieren."

[Modulbeschreibungen\MH35; Position: 9 - 9]

642.

"[Methodenkompetenz:] [Die Studierenden] [sind in der Lage, Algorithmen hinsichtlich ihrer Eigenschaften und Anwendbarkeit für konkrete Problemstellungen] und bewerten."

[Modulbeschreibungen\MH35; Position: 10 - 10]

643.

"[Methodenkompetenz:] [Die Studierenden] [sind in der Lage, Algorithmen hinsichtlich ihrer Eigenschaften und Anwendbarkeit für konkrete Problemstellungen] in eigenen kleineren Programmierprojekten in der Programmiersprache Java umzusetzen."

[Modulbeschreibungen\MH35; Position: 11 - 11]

644.

"[Systemkompetenz:] [Die Studierenden] können Algorithmen für einfache Problemstellungen selbstständig entwerfen."

[Modulbeschreibungen\MH35; Position: 14 - 14]

645.

"Die Studierenden sind in der Lage, zu gegebenen Problemen geeignete Paradigmen kritisch auszuwählen."

[Modulbeschreibungen\MH35; Position: 21 - 21]

646.

"[Die Studierenden] können einfache Programme sowohl im funktionalen als auch im objektorientierten Programmierstil systematisch entwerfen (Methodenkompetenz)."

[Modulbeschreibungen\MH35; Position: 22 - 22]

647.

"[Die Studierenden] [können einfache Programme sowohl im funktionalen als auch im objektorientierten Programmierstil] implementieren (Methodenkompetenz)."

[Modulbeschreibungen\MH35; Position: 23 - 23]

4.4. Meta-kognitives Wissen

1.

"Des weiteren entwickeln die Studierenden ein Verständnis für die allgemeinen Prinzipien der Programmierung und der Programmiersprachen, das den Grundstein dafür legt, dass die Studierenden sich (nach weiteren Erfahrungen im Laufe des Studiums) in beliebige Programmiersprachen schnell und präzise einarbeiten können."

[Modulbeschreibungen\MH03; Position: 8 - 8]

2.

"Nach erfolgreicher Teilnahme am Softwareentwicklungspraktikum sollen sich die Teilnehmer trauen, Werkstudentenjobs in der IT-Industrie anzunehmen."

[Modulbeschreibungen\MH03; Position: 28 - 28]

3.

"Kompetenz zur Vernetzung unterschiedlicher Fachgebiete: Im Rahmen von Übungen setzen die Studierenden die erarbeiteten Grundlagen in verschiedenen Anwendungsbereichen ein."

[Modulbeschreibungen\MH04; Position: 47 - 47]

4.

"[Die Studierenden] sind in der Lage ihr Wissen vertikal, horizontal und lateral zu vertiefen."

[Modulbeschreibungen\MH08; Position: 29 - 29]

5.

"Die Studierenden sind in der Lage ihr Wissen und Verstehen auf ihre Tätigkeit oder ihren Beruf anzuwenden."

[Modulbeschreibungen\MH08; Position: 31 - 31]

6.

"Die Studierenden erschließen sich vertiefende Informationen und Konzepte aus der zugrunde liegenden Literatur bzw. Dokumentation."

[Modulbeschreibungen\MH13; Position: 23 - 23]

7.

"[Die Studierenden] sollen die Fähigkeit entwickeln, selbstständig in Programmierhandbüchern und -beschreibungen Details der Programmiersprache herauszufinden und nutzen zu können (instrumentale Kompetenz)."

[Modulbeschreibungen\MH16; Position: 22 - 22]

8.

"Die Studierenden erlernen den Umstieg auf eine neue Programmiersprache, die insbesondere auch für größere Programmierprojekte geeignet ist."

[Modulbeschreibungen\MH16; Position: 33 - 33]

9.

"Methodenkompetenz: Fähigkeit zur schnellen Einarbeitung in eine beliebige Programmiersprache."

[Modulbeschreibungen\MH19; Position: 15 - 15]

10.

"[Methodenkompetenz:] Fähigkeit zur schnellen Einarbeitung in neue Anwendungen."

[Modulbeschreibungen\MH19; Position: 28 - 28]

11.

"Allgemeine Kompetenz: Selbstständiges Erarbeiten (Lesen und Verstehen) von Informatik-Themen mit Hilfe wissenschaftlicher Literatur."

[Modulbeschreibungen\MH20; Position: 19 - 19]

12.

"[Methodenkompetenzen:] [Die Studierenden] übertragen praktische Erfahrungen in der Programmentwicklung auf neue Aufgaben."

[Modulbeschreibungen\MH21; Position: 26 - 26]

13.

"Selbstkompetenzen: Die Studierenden organisieren sich beim Finden von algorithmischen Lösungen für kleine und mittelgroße Probleme der Informatik."

[Modulbeschreibungen\MH21; Position: 29 - 29]

14.

"[Methodenkompetenzen:] [Die Studierenden] übertragen praktische Erfahrungen in der Programmentwicklung auf neue Aufgaben."

[Modulbeschreibungen\MH21; Position: 49 - 49]

15.

"Selbstkompetenzen: Die Studierenden organisieren sich beim Entwickeln von Programmen für kleine und mittelgroße Probleme der Informatik."

[Modulbeschreibungen\MH21; Position: 52 - 52]

16.

"Studierende sollten nach der Veranstaltung in der Lage sein, Probleme funktional zu abstrahieren."

[Modulbeschreibungen\MH24; Position: 36 - 36]

17.

"[In dieser Vorlesung lernen sie] sich in wenigen Tagen eine neue imperative/objektorientierte Sprache anzueignen, um sich in ein bestehendes Projekt einzuarbeiten."

[Modulbeschreibungen\MH27; Position: 24 - 24]

18.

"Gemeinsam können die Studierenden, begleitet durch die Diskussion mit Mitstudierenden im Rahmen des Seminars, Lösungen für neue unbekannte Problemstellungen aus dem Bereich der Programmierung erarbeiten."

[Modulbeschreibungen\MH28; Position: 11 - 11]

19.

"Studierende sollen in der Lage sein, die am Beispiel der Sprache JAVA erlernten Prinzipien auf andere Programmiersprachen imperativen/prozeduralen und objektorientierten Charakters übertragen zu können."

[Modulbeschreibungen\MH32; Position: 25 - 25]

20.

"Neben dem Erlernen von Rechenverfahren zur Lösung von gewissen mathematischen Problemen ist es ein Hauptziel diese Lehrveranstaltung, das für Informatiker und Informatikerinnen so wichtige Abstraktionsvermögen zu schulen."

[Modulbeschreibungen\MH32; Position: 29 - 29]

21.

"Studierende sollen am Bsp. der Programmiersprache C begreifen, erlernte generelle Konzepte imperativer/prozeduraler Programmiersprachen auf andere Sprachen zu übertragen (hier: Transfer von JAVA auf C)."

[Modulbeschreibungen\MH32; Position: 43 - 43]

22.

"[Die Studierenden] können neue Sprachkonstrukte selbstständig erlernen."

[Modulbeschreibungen\MH33; Position: 31 - 31]

23.

"[Systemkompetenz:] [Die Studierenden] können [...] [die Wirkungsweise von Standardalgorithmen und –datenstrukturen] in neuen Zusammenhängen einsetzen."

[Modulbeschreibungen\MH35; Position: 13 - 13]

5. kognitive Prozessdimensionen

5.1. Erinnern

5.1.1. Literale zu Datentypen zuordnen

1.

"[Die Studierenden] ordnen Literale der Sprache C sicher den entsprechenden Datentypen zu."

[Modulbeschreibungen\MH34; Position: 7 - 7]

5.1.2. Begriffe & Kategorien zu Komplexität & Effizienz von Algorithmen kennen

1.

"[Bei erfolgreichem Abschluss des Moduls sollten die Teilnehmer(innen) Kenntnisse über] wesentliche Komplexitätsklassen für das Laufzeitverhalten und den Speicherplatzbedarf von Algorithmen [erlangt haben]."

[Modulbeschreibungen\MH03; Position: 17 - 17]

2.

"[Studentinnen und Studenten] kennen die Definition [...] von NP-Vollständigkeit für die effiziente Lösbarkeit von Problemen."

[Modulbeschreibungen\MH05; Position: 40 - 40]

3.

"Churchsche These kennen."

[Modulbeschreibungen\MH08; Position: 7 - 7]

4.

"Kenntnis der Effizienzmaße auf Parallelrechner-systemen und von effizienten parallelen Algorithmen."

[Modulbeschreibungen\MH08; Position: 24 - 24]

5.

"[In dieser Veranstaltung lernen Studierende] die Komplexitätsklassen P, NP und NPC kennen."

[Modulbeschreibungen\MH17; Position: 13 - 13]

6.

"[Fachkompetenzen:] [Die Studierenden] benennen Probleme der Effizienz von algorithmischen Lösungen konkreter Fragestellungen."

[Modulbeschreibungen\MH21; Position: 20 - 20]

7.

"[Beim erfolgreichen Abschluss des Moduls sollten die Studierenden Kenntnisse über] [...] wesentliche Komplexitätskategorien für Laufzeit und Speicherbedarf von Algorithmen [haben]."

[Modulbeschreibungen\MH22; Position: 19 - 19]

5.1.3. Merkmale von Algorithmen kennen

1.

"Die Studentinnen und Studenten kennen die Grundbegriffe zu Algorithmen."

[Modulbeschreibungen\MH05; Position: 16 - 16]

2.

"Studentinnen und Studenten können die Grundbegriffe der Algorithmik definieren."

[Modulbeschreibungen\MH05; Position: 31 - 31]

3.

"Kenntnis des Algorithmusbegriffs, von Merkmalen von Algorithmen und Grenzen der Algorithmisierung."

[Modulbeschreibungen\MH08; Position: 5 - 5]

5.1.4. Funktionsbibliotheken für Algorithmen kennen

1.

"Kenntnis des Umfangs typischer Funktionsbibliotheken für Algorithmen in Programmiersprachen."

[Modulbeschreibungen\MH04; Position: 39 - 39]

2.

"[Die Studierenden] kennen einige grundlegende Klassen der Java-Bibliothek."

[Modulbeschreibungen\MH13; Position: 11 - 11]

3.

"Die Absolventen kennen die C-Standardbibliothek und weitere ausgewählte Bibliotheken."

[Modulbeschreibungen\MH28; Position: 53 - 53]

5.1.5. Elemente von GUIs kennen

1.

"[Die Studierenden] kennen grafische Benutzerschnittstellen."

[Modulbeschreibungen\MH13; Position: 19 - 19]

5.1.6. Grundprinzipien und Methoden von Programmiersprachen und –paradigmen kennen

1.

"Die Studierenden lernen aus dem Kanon der Grundlagen der Informatik die prozedurale Programmierung kennen sowie einführend auch Grundzüge der Objektorientierung."

[Modulbeschreibungen\MH04; Position: 11 - 11]

2.

"Die Studierenden lernen aus dem Kanon der Grundlagen der Informatik die objektorientierte Programmierung kennen."

[Modulbeschreibungen\MH04; Position: 19 - 19]

3.

"[Die Studierenden] kennen insbesondere die Unterschiede bei der objektorientierten Programmierung zwischen C++ und Java."

[Modulbeschreibungen\MH04; Position: 58 - 58]

4.

"[Die Studentinnen und Studenten kennen die Grundbegriffe zu] der funktionalen Programmierung."

[Modulbeschreibungen\MH05; Position: 17 - 17]

5.

"Die Studierenden erwerben Grundkenntnisse einer objektorientierten Programmiersprache."

[Modulbeschreibungen\MH06; Position: 20 - 20]

6.

"Die Studierenden haben ihre Kenntnisse einer objektorientierten Programmiersprache erweitert."

[Modulbeschreibungen\MH06; Position: 26 - 26]

7.

"Die Studierenden kennen die Konzepte der Objektorientierung und objektorientierten Programmierung am Beispiel der Programmiersprache Java."

[Modulbeschreibungen\MH07; Position: 16 - 16]

8.

"Die Studierenden kennen die Konzepte der Prozeduralen und objektorientierten Programmierung, wie sie in den Programmiersprache C und C++ umgesetzt sind."

[Modulbeschreibungen\MH07; Position: 23 - 23]

9.

"Grundprinzipien funktionaler Programmierung kennen."

[Modulbeschreibungen\MH08; Position: 12 - 12]

10.

"Programmierparadigmen und -sprachen, Syntax und Semantik bei Programmiersprachen definieren können."

[Modulbeschreibungen\MH08; Position: 14 - 14]

11.

"Konzepte einer imperativen Programmiersprache kennen."

[Modulbeschreibungen\MH10; Position: 7 - 7]

12.

"Konzepte und typische Merkmale des funktionalen Programmierens kennen."

[Modulbeschreibungen\MH10; Position: 41 - 41]

13.

"Die Studierenden können die Basiskonzepte einer modernen Programmiersprache benennen."

[Modulbeschreibungen\MH11; Position: 5 - 5]

14.

"Die Studierenden kennen zudem die wichtigsten Programmierkonzepte von C."

[Modulbeschreibungen\MH13; Position: 29 - 29]

15.

"Gründliche Kenntnisse einer Programmiersprache."

[Modulbeschreibungen\MH14; Position: 5 - 5]

16.

"[Die Studierenden sollen] über Wissen zum strukturierten und objektorientierten Programmieren mit einer imperativen Programmiersprache verfügen (instrumentale Kompetenz)."

[Modulbeschreibungen\MH16; Position: 14 - 14]

17.

"Zur Erarbeitung instrumentaler und systemischer Kompetenzen sollen die Studierenden die verschiedenen Programmiersprachparadigmen und Konzepte zu Syntax und Semantik kennen."

[Modulbeschreibungen\MH16; Position: 25 - 25]

18.

"[Die Studierenden] sollen Wissen über funktionale Sprachen erwerben."

[Modulbeschreibungen\MH16; Position: 26 - 26]

19.

"Umsetzungsmöglichkeiten für die Programmiersprachen C++, Java und .NET in den Grundzügen kennen."

[Modulbeschreibungen\MH18; Position: 18 - 18]

20.

"Das objektorientierte Programmierparadigma kennen."

[Modulbeschreibungen\MH18; Position: 20 - 20]

21.

"[Fachkompetenzen:] [Die Studierenden] erkennen die Terminologie der imperativen Programmierung."

[Modulbeschreibungen\MH21; Position: 11 - 11]

22.

"[Fachkompetenzen:] [Die Studierenden] kennen grundlegende Konzepte der objektorientierten Programmierung mit Java."

[Modulbeschreibungen\MH21; Position: 39 - 39]

23.

"[Fachkompetenzen:] [Die Studierenden] kennen die Terminologie der objektorientierten Modellierung und Programmierung."

[Modulbeschreibungen\MH21; Position: 40 - 40]

24.

"[Fachkompetenzen:] [Die Studierenden] kennen die Unterschiede zwischen dem imperativen, objektorientierten, funktionalen und logischen Programmierparadigma."

[Modulbeschreibungen\MH21; Position: 47 - 47]

25.

"Kenntnisse: Wesentliche Konzepte imperativer und objektorientierter Programmiersprachen sowie wichtige Programmier Techniken in diesen Sprachen."

[Modulbeschreibungen\MH22; Position: 5 - 5]

26.

"Kenntnis der Programmierkonzepte logischer und funktionaler Programmiersprachen."

[Modulbeschreibungen\MH22; Position: 6 - 6]

27.

"Kenntnis grundlegender Beschreibungsformen für Programmiersprachen."

[Modulbeschreibungen\MH22; Position: 9 - 9]

28.

"[Die Studierenden sollen] den Aufbau und die Beschreibung von Programmiersprachen als formale Sprachen lernen."

[Modulbeschreibungen\MH24; Position: 6 - 6]

29.

"[Die Studierenden sollen] [einfache Algorithmen und Datenstrukturen] formal beschreiben können."

[Modulbeschreibungen\MH24; Position: 8 - 8]

30.

"[Die Studierenden sollen] unterschiedliche Paradigmen von Programmiersprachen kennen lernen."

[Modulbeschreibungen\MH24; Position: 9 - 9]

31.

"Die Studierenden kennen mindestens zwei Programmierparadigmen (imperativ und objektorientiert)."

[Modulbeschreibungen\MH25; Position: 15 - 15]

32.

"[Die Studierenden] kennen Zeiger- und Adresskonzepte in der Programmiersprache C/C++."

[Modulbeschreibungen\MH25; Position: 23 - 23]

33.

"[Die Studierenden] kennen das funktionale Paradigma."

[Modulbeschreibungen\MH25; Position: 24 - 24]

34.

"Neben der Vertiefung der objektorientierten Programmierung erlernen die Studierenden die grundlegenden Konzepte der Programmiersprache C."

[Modulbeschreibungen\MH26; Position: 25 - 25]

35.

"Die Studierenden können Konzepte und Eigenheiten der imperativen Programmierung in C benennen."

[Modulbeschreibungen\MH26; Position: 26 - 26]

36.

"Die Studierenden lernen die Grundprinzipien der imperativen/objektorientierten Programmierung kennen."

[Modulbeschreibungen\MH27; Position: 17 - 17]

37.

"Die Studierenden kennen darüber hinaus wesentliche Konzepte und Verfahren moderner Programmiersprachen, angefangen von einfachen Datentypen, über Kontrollstrukturen bis hin zu den Themen Klassen, Objekte und Vererbung."

[Modulbeschreibungen\MH28; Position: 7 - 7]

38.

"[Die Absolventen] kennen die Konzepte der Programmiersprache C mit ihren Vorteilen und Schwächen."

[Modulbeschreibungen\MH28; Position: 51 - 51]

39.

"Die Studierenden besitzen Kenntnisse über Grundlagen der imperativen Programmierung (Syntaxdiagramme, EBNF, Funktionen, Module, Datenstrukturen)."

[Modulbeschreibungen\MH29; Position: 5 - 5]

40.

"Die Studierenden besitzen Kenntnisse des funktionalen Programmierens."

[Modulbeschreibungen\MH29; Position: 10 - 10]

41.

"Darüber hinaus kennen [die Studierenden] grundlegende Problemlösungs-und Programmkonstruktionsmethoden der imperativen Programmierung."

[Modulbeschreibungen\MH30; Position: 13 - 13]

42.

"Die Studierenden kennen die wichtigsten Konzepte der objektorientierten Programmierung."

[Modulbeschreibungen\MH30; Position: 22 - 22]

43.

"Kenntnisse in zwei (weiteren) Paradigmen."

[Modulbeschreibungen\MH31; Position: 22 - 22]

44.

"Kenntnis existierender Programmierparadigmen."

[Modulbeschreibungen\MH32; Position: 5 - 5]

45.

"Kenntnisse verschiedener Programmierparadigmen."

[Modulbeschreibungen\MH33; Position: 6 - 6]

46.

"Gute Java-Kenntnisse."

[Modulbeschreibungen\MH33; Position: 11 - 11]

47.

"Die Studierenden können die grundlegenden Prinzipien der OOP benennen."

[Modulbeschreibungen\MH34; Position: 19 - 19]

48.

"[Die Studierenden können] [Konzepte objektorientierter Programmierung] mit den korrekten Fachbegriffen wiedergeben."

[Modulbeschreibungen\MH34; Position: 35 - 35]

49.

"[Die Studierenden können] Vor- und Nachteile von objektorientierten Konzepten benennen."

[Modulbeschreibungen\MH34; Position: 37 - 37]

50.

"[Die Studierenden können] Gemeinsamkeiten und Unterschiede hinsichtlich der Sprachphilosophie und Konzepte von Java und C/C++ benennen."

[Modulbeschreibungen\MH34; Position: 39 - 39]

5.1.7. Kenntnis grundlegender Algorithmen & Datenstrukturen

1.

"Die Studierenden kennen nach engagierter Mitarbeit in dieser Veranstaltung diverse zentrale Algorithmen auf geeigneten Datenstrukturen, die für eine effiziente Nutzung von Computern unverzichtbar sind."

[Modulbeschreibungen\MH02; Position: 17 - 17]

2.

"Kenntnis der Eigenschaften elementarer und häufig benötigter Algorithmen."

[Modulbeschreibungen\MH02; Position: 20 - 20]

3.

"[Bei erfolgreichem Abschluss des Moduls sollten die Teilnehmer(innen) Kenntnisse über] effiziente Algorithmen und Datenstrukturen für grundlegende Probleme [erlangt haben]."

[Modulbeschreibungen\MH03; Position: 16 - 16]

4.

"Datenstrukturen benennen."

[Modulbeschreibungen\MH04; Position: 28 - 28]

5.

"Such- und Sortierverfahren benennen."

[Modulbeschreibungen\MH04; Position: 30 - 30]

6.

"Grundlagen Informatik: Die Studierenden lernen grundlegende Algorithmen und Datenstrukturen kennen."

[Modulbeschreibungen\MH04; Position: 42 - 42]

7.

"[Die Studentinnen und Studenten] kennen grundlegende Datenstrukturen und Algorithmen."

[Modulbeschreibungen\MH05; Position: 8 - 8]

8.

"[Die Studentinnen und Studenten] kennen grundlegende Datenstrukturen und Algorithmen."

[Modulbeschreibungen\MH05; Position: 11 - 11]

9.

"[Studentinnen und Studenten] wissen, was ein abstrakter Datentyp ist."

[Modulbeschreibungen\MH05; Position: 32 - 32]

10.

"Die Studierenden kennen wichtige Algorithmen und die zugehörigen Datenstrukturen."

[Modulbeschreibungen\MH06; Position: 11 - 11]

11.

"Die Studierenden kennen Standardalgorithmen für typische Problemstellungen aus den Bereichen Suchen, Sortieren, Mustererkennung, Rekursion, Bäume und Graphen."

[Modulbeschreibungen\MH07; Position: 5 - 5]

12.

"[Die Studierenden] kennen die Datenstrukturen Liste, Array, verkettete Liste, Stapel, Schlange, Baum, Graph."

[Modulbeschreibungen\MH07; Position: 11 - 11]

13.

"Die Studierenden besitzen Kenntnisse über algorithmische Lösungen."

[Modulbeschreibungen\MH12; Position: 5 - 5]

14.

"[Die Teilnehmer] kennen grundlegende Datenstrukturen."

[Modulbeschreibungen\MH12; Position: 19 - 19]

15.

"Die Studierenden kennen die wichtigsten statischen und dynamischen Datenstrukturen sowie Such- und Sortialgorithmen."

[Modulbeschreibungen\MH13; Position: 25 - 25]

16.

"Kenntnis grundlegender Algorithmen und Datenstrukturen."

[Modulbeschreibungen\MH14; Position: 11 - 11]

17.

"Wissen und Verstehen: Die Studierenden sollen grundlegende Datenstrukturen mit deren Eigenschaften und Leistungsparametern kennen."

[Modulbeschreibungen\MH16; Position: 5 - 5]

18.

"In dieser Veranstaltung lernen Studierende grundlegende Datenstrukturen und Algorithmen kennen."

[Modulbeschreibungen\MH17; Position: 12 - 12]

19.

"Fachkompetenz: Kenntnis wichtiger Basisalgorithmen für Graph-, Codierungs-, Kommunikations- und Optimierungsprobleme."

[Modulbeschreibungen\MH19; Position: 13 - 13]

20.

"[Fachkompetenz:] Kenntnis der wichtigsten Datenstrukturen und Muster."

[Modulbeschreibungen\MH19; Position: 14 - 14]

21.

"[Formale und algorithmische Kompetenzen:] gute Algorithmen für wichtige Standardprobleme (z.B. Sortieren) kennen."

[Modulbeschreibungen\MH20; Position: 32 - 32]

22.

"Analyse-, Design- und Realisierungs-Kompetenzen: wichtige lineare und nicht-lineare Datenstrukturen kennen."

[Modulbeschreibungen\MH20; Position: 34 - 34]

23.

"[Analyse-, Design- und Realisierungs-Kompetenzen:] Datenstrukturen aus Standardbibliotheken kennen."

[Modulbeschreibungen\MH20; Position: 41 - 41]

24.

"[Fachkompetenzen:] [Die Studierenden] benennen Algorithmen und Datenstrukturen zur Lösung von häufig vorkommenden Problemen."

[Modulbeschreibungen\MH21; Position: 18 - 18]

25.

"Kenntnis grundlegender Datenstrukturen in verschiedenen Programmierparadigmen."

[Modulbeschreibungen\MH22; Position: 7 - 7]

26.

"[Beim erfolgreichen Abschluss des Moduls sollten die Studierenden Kenntnisse über] [...] effiziente Algorithmen und Datenstrukturen für Standardprobleme [haben]."

[Modulbeschreibungen\MH22; Position: 20 - 20]

27.

"[Die Studierenden sollen] einfache Algorithmen und Datenstrukturen kennen."

[Modulbeschreibungen\MH24; Position: 7 - 7]

28.

"Die Studierenden sollen die grundlegenden Datenstrukturen und Algorithmen kennen lernen."

[Modulbeschreibungen\MH24; Position: 17 - 17]

29.

"[Die Studierenden sollen] Einsatz und Bedeutung effizienter Algorithmen in verschiedenen Anwendungsbereichen kennen lernen."

[Modulbeschreibungen\MH24; Position: 51 - 51]

30.

"Die Studierenden kennen grundlegende Datenstrukturen und Algorithmen."

[Modulbeschreibungen\MH25; Position: 17 - 17]

31.

"Kennenlernen von Algorithmen für die Programmiersprachen I-Vorlesung."

[Modulbeschreibungen\MH26; Position: 10 - 10]

32.

"Die Studierenden lernen wichtige effiziente Datenstrukturen und Algorithmen kennen."

[Modulbeschreibungen\MH27; Position: 30 - 30]

33.

"Jeder Teilnehmer kann Unterschiede, Vor- und Nachteile von speziellen Datenstrukturen (Listen, Felder, Assoziativ-Speicher) benennen."

[Modulbeschreibungen\MH28; Position: 18 - 18]

34.

"Außerdem verfügen [die Studierenden] über Kenntnisse der wichtigsten dynamischen Datenstrukturen."

[Modulbeschreibungen\MH30; Position: 24 - 24]

35.

"Die Teilnehmer kennen grundlegende und wichtige Algorithmen."

[Modulbeschreibungen\MH30; Position: 29 - 29]

36.

"[Studierende sollen am Bsp. der Programmiersprache JAVA die folgenden Aspekte einer imperativen/prozeduralen Programmiersprache beherrschen:] Kenntnis der Datenstrukturen Liste, Stack, Map und Set."

[Modulbeschreibungen\MH32; Position: 13 - 13]

37.

"[Studierende sollen am Bsp. der Programmiersprache JAVA die folgenden Aspekte einer imperativen/prozeduralen Programmiersprache beherrschen:] Kenntnis der rekursiven Datenstruktur Baum."

[Modulbeschreibungen\MH32; Position: 17 - 17]

38.

"Die Studierenden kennen effiziente Algorithmen und Datenstrukturen aus den wichtigsten Gebieten der Angewandten Informatik."

[Modulbeschreibungen\MH32; Position: 45 - 45]

39.

"Fachkompetenz: Die Studierenden verfügen über Kenntnisse zu algorithmischen Modellen, Basisalgorithmen."

[Modulbeschreibungen\MH35; Position: 5 - 5]

5.1.8. Methoden der Softwareentwicklung kennen

1.

"Die Studierenden kennen die Grundlagen der professionellen Softwareentwicklung und des Konfigurationsmanagements (ALM)."

[Modulbeschreibungen\MH04; Position: 50 - 50]

2.

"[Die Studierenden] kennen die [...] nötigen Methoden und Werkzeuge für [professionellen Code, automatisierte Tests und testgetriebene Softwareentwicklung]."

[Modulbeschreibungen\MH04; Position: 55 - 55]

3.

"[Die Studierenden] kennen zentrale Konzepte zur Abstraktion und Modularisierung."

[Modulbeschreibungen\MH12; Position: 24 - 24]

4.

"Fach- und Methodenkompetenzen (Wissen und Verstehen sowie Anwendung und Erzeugung von Wissen): Die Studierenden kennen ein umfassendes Instrumentarium an Techniken und Lösungsmustern zur Softwareentwicklung in unterschiedlichen Programmiersprachen."

[Modulbeschreibungen\MH15; Position: 5 - 5]

5.

"[Fach- und Methodenkompetenzen (Wissen und Verstehen sowie Anwendung und Erzeugung von Wissen):] Standardarchitekturmuster sind bekannt."

[Modulbeschreibungen\MH15; Position: 7 - 7]

6.

"[Die Studierenden] sollen die Prozesse und Methoden der Software-Entwicklung und des Algorithmenentwurfs sowie die Services des Betriebssystems kennen."

[Modulbeschreibungen\MH16; Position: 15 - 15]

7.

"Grundkenntnisse der Programmverifikation."

[Modulbeschreibungen\MH22; Position: 10 - 10]

8.

"[Die Studierenden sollen] die Bedeutung der Software-Prüfung kennen lernen."

[Modulbeschreibungen\MH24; Position: 32 - 32]

9.

"Die Studierenden sollen die wichtigsten Vorgehensmodelle der Softwareentwicklung kennen."

[Modulbeschreibungen\MH26; Position: 31 - 31]

10.

"Des Weiteren sollen [die Studierenden] lernen wie Anforderungsanalyse von komplexen Anwendungssystemen durchzuführen [ist], insbesondere in der Kommunikation mit den Anwendern."

[Modulbeschreibungen\MH26; Position: 33 - 33]

11.

"[Die Studierenden] sollen zudem die Hauptproblematiken bei der Durchführung von SW-Projekten kennen und umgehen lernen."

[Modulbeschreibungen\MH26; Position: 35 - 35]

12.

"Die Studenten haben Kenntnisse im Erstellen von Internetsoftware."

[Modulbeschreibungen\MH30; Position: 17 - 17]

5.1.9. Konzepte zur Modellierung von Algorithmen und Prozessen kennen

1.

"[Die Studierenden] kennen geeignete Abstraktionen, Modellbildungen und Verfahren zur Beschreibung von Algorithmen und Prozessen."

[Modulbeschreibungen\MH12; Position: 29 - 29]

2.

"[Die Studierenden] [kennen geeignete Abstraktionen, Modellbildungen und Verfahren zur] Analyse von Algorithmen und Prozessen."

[Modulbeschreibungen\MH12; Position: 30 - 30]

3.

"Ziel des Moduls „Objektorientierte Modellierung und Programmierung“ ist das Erlernen grundlegender Konzepte der objektorientierten Modellierung mit Hilfe der UML als Modellierungsnotation."

[Modulbeschreibungen\MH21; Position: 35 - 35]

4.

"Fachkompetenzen: Die Studierenden kennen grundlegende Konzepte der objektorientierten Modellierung und UML als Modellierungsnotation."

[Modulbeschreibungen\MH21; Position: 38 - 38]

5.

"[Die Studierenden sollen] wissen, wie Sie am besten beim Erstellen objektorientierter Modelle vorgehen"

[Modulbeschreibungen\MH24; Position: 28 - 28]

6.

"[Die Studierenden sollen] wissen, wie sie gute von schlechten Modellen unterscheiden können."

[Modulbeschreibungen\MH24; Position: 29 - 29]

7.

"Grundlegende Modellierungskonzepte sind bekannt."

[Modulbeschreibungen\MH25; Position: 20 - 20]

8.

"Erlernen von Techniken für die Prozess- und Datenmodellierung auf fachkonzeptueller Ebene."

[Modulbeschreibungen\MH31; Position: 17 - 17]

9.

"Erlernen von objektorientierten Modellierungstechniken auf DV-konzeptueller Ebene."

[Modulbeschreibungen\MH31; Position: 18 - 18]

5.1.10. Entwurfsmöglichkeiten von Algorithmen & Datenstrukturen kennen

1.

"[Die Teilnehmer] kennen die Möglichkeiten, Daten- und Ablaufstrukturen zu entwerfen."

[Modulbeschreibungen\MH02; Position: 10 - 10]

2.

"[Die Teilnehmer] [kennen die Möglichkeiten, Daten- und Ablaufstrukturen] zu beschreiben."

[Modulbeschreibungen\MH02; Position: 11 - 11]

3.

"Bei erfolgreichem Abschluss des Moduls sollten die Teilnehmer(innen) Kenntnisse über grundlegende Eigenschaften und Entwurfsmethoden für Algorithmen [erlangt haben]."

[Modulbeschreibungen\MH03; Position: 15 - 15]

4.

"Die Studierenden erwerben Kenntnisse der grundlegenden Techniken zur algorithmischen Problemlösung."

[Modulbeschreibungen\MH04; Position: 6 - 6]

5.

"Beim erfolgreichen Abschluss des Moduls sollten die Studierenden Kenntnisse über folgende Themen haben: Grundlegende Entwurfsmethoden für Algorithmen."

[Modulbeschreibungen\MH22; Position: 18 - 18]

6.

"Die Studierenden lernen die wichtigsten Methoden des Entwurfs von Algorithmen und Datenstrukturen kennen: Teile-und-Herrsche, Dynamische Programmierung, inkrementelle Konstruktion, „Greedy“, Dezimierung, Hierarchisierung, Randomisierung."

[Modulbeschreibungen\MH27; Position: 26 - 26]

7.

"Die Studierenden kennen verschiedene Klassen von Algorithmen (divide-and-conquer, dynamisches Programmieren, Iteration versus Rekursion, backtracking)."

[Modulbeschreibungen\MH29; Position: 7 - 7]

8.

"[Studierende sollen am Bsp. der Programmiersprache JAVA die folgenden Aspekte einer imperativen/prozeduralen Programmiersprache beherrschen:] Kenntnis des Prinzips rekursiver Programmierung."

[Modulbeschreibungen\MH32; Position: 15 - 15]

5.1.11. Aufbau und Funktionsweise des Computers kennen

1.

"Die Studierenden erlernen die prinzipiellen Beschränkungen heutiger Computer bei der Lösung von wichtigen Problemen."

[Modulbeschreibungen\MH01; Position: 5 - 5]

2.

"Informatik-Kompetenz: kennen des Aufbaus und der Funktionsweise von Computern."

[Modulbeschreibungen\MH20; Position: 13 - 13]

3.

"Kenntnisse: Elementare Kenntnisse über physikalischen Prinzipien, welche der Funktionsweise elektronischer Rechner zugrunde liegen."

[Modulbeschreibungen\MH22; Position: 27 - 27]

4.

"In dieser Vorlesung lernen sie wie Rechner Programme ausführen."

[Modulbeschreibungen\MH27; Position: 19 - 19]

5.

"Am Ende dieses Moduls kennt jeder Kursteilnehmer den grundlegenden Aufbau und die Funktionsweise eines Rechnersystems."

[Modulbeschreibungen\MH28; Position: 5 - 5]

5.1.12. Implementierungsmöglichkeiten von Datentypen & -strukturen kennen

1.

"[Die Teilnehmer] [kennen die Möglichkeiten, Daten- und Ablaufstrukturen] zu codieren."

[Modulbeschreibungen\MH02; Position: 12 - 12]

2.

"[Studentinnen und Studenten] kennen die wichtigsten abstrakten Datentypen und die Datenstrukturen zu deren Implementierung."

[Modulbeschreibungen\MH05; Position: 34 - 34]

3.

"[Analyse-, Design- und Realisierungs-Kompetenzen:] verschiedene Implementierungsvarianten für wichtige Datenstrukturen kennen."

[Modulbeschreibungen\MH20; Position: 36 - 36]

4.

"[Kenntnis] [...] der Realisierung [grundlegender Datenstrukturen] in verschiedenen Programmierparadigmen."

[Modulbeschreibungen\MH22; Position: 8 - 8]

5.1.13. Prozesskenntnis der Laufzeitanalyse

1.

"Darüber hinaus kennen [die Studierenden] die O-Notation."

[Modulbeschreibungen\MH13; Position: 26 - 26]

2.

"Formale und algorithmische Kompetenzen: Aufwandsabschätzungen mit Hilfe der O-Notation kennen."

[Modulbeschreibungen\MH20; Position: 29 - 29]

3.

"[Die Studierenden verfügen insbesondere über vertiefte Kenntnisse bezogen auf die Bereiche] Abschätzung des Aufwands einfacher Algorithmen (z.B. für Such- und Sortierprobleme)."

[Modulbeschreibungen\MH28; Position: 17 - 17]

4.

"[Kenntnis existierender] Laufzeitmodelle."

[Modulbeschreibungen\MH32; Position: 6 - 6]

5.1.14. Werkzeuge zur Softwareentwicklung, deren Funktionen und Anwendungszwecke kennen (IDEs, Debugger, Profiler)

1.

"Editor, Refactoring, Build, Versionsverwaltungssystem und Test der Software in einer Applikation zeigen Ihnen den aktuellen Stand moderner Entwicklungsumgebungen."

[Modulbeschreibungen\MH04; Position: 53 - 53]

2.

"[Die Teilnehmer] kennen Werkzeuge wie Debugger und Profiler."

[Modulbeschreibungen\MH28; Position: 20 - 20]

3.

"[Die Studierenden können] gängige Entwicklungsumgebungen für die Entwicklung mit Java benennen."

[Modulbeschreibungen\MH34; Position: 44 - 44]

5.1.15. Qualitätskriterien für Quellcode nennen

1.

"Die Studierenden entwickeln zu Beginn ihrer Ausbildung ein Verständnis für Stilistik und Ästhetik der Programmierung."

[Modulbeschreibungen\MH04; Position: 5 - 5]

2.

"[Die Studierenden] können Qualitätskriterien für lesbaren, wartbaren, wiederverwendbaren Quellcode nennen."

[Modulbeschreibungen\MH11; Position: 14 - 14]

3.

"Die Veranstaltung führt auch grundlegende Techniken ein, mit denen Programme verständlicher und einfacher wartbar gemacht werden, und Fehler während der Entwicklung erkannt werden können."

[Modulbeschreibungen\MH25; Position: 8 - 8]

5.1.16. Mathematische Grundlagen von Algorithmen kennen

1.

"[Die Studierenden] kennen in konkreten Problemstellungen der Informatik das nötige mathematische Handwerkszeug."

[Modulbeschreibungen\MH07; Position: 13 - 13]

2.

"Diese Lehrveranstaltung soll den Lernenden die engen inneren Zusammenhänge zwischen den scheinbar so verschiedenen Gebieten Mengenlehre, Logik, Relationen, Wahrscheinlichkeitsrechnung und Algorithmen klarmachen."

[Modulbeschreibungen\MH32; Position: 28 - 28]

5.1.17. Konzepte zur Datenverwaltung kennen

1.

"[Zur Erarbeitung instrumentaler und systemischer Kompetenzen sollen die Studierenden] die Modellierung, Verwaltung und Nutzung größerer Datenbestände kennen."

[Modulbeschreibungen\MH16; Position: 29 - 29]

2.

"Die Studierenden verfügen insbesondere über vertiefte Kenntnisse bezogen auf die Bereiche interne und externe Datenverwaltung (einschließlich der Ein- und Ausgabe)."

[Modulbeschreibungen\MH28; Position: 16 - 16]

5.1.18. Begriff der Kontrollstrukturen kennen

1.

"[Die Studierenden] benennen die grundlegenden Kontrollstrukturen strukturierter Programmierung."

[Modulbeschreibungen\MH34; Position: 9 - 9]

2.

"[Die Studierenden] stellen [...] [die grundlegenden Kontrollstrukturen strukturierter Programmierung] korrekt im Struktogramm und in C-Syntax dar."

[Modulbeschreibungen\MH34; Position: 10 - 10]

5.1.19. Aufbau und Funktionsweise von Netzen kennen

1.

"Kennen des Aufbaus und der Funktionsweise von Netzen wie dem Internet."

[Modulbeschreibungen\MH20; Position: 17 - 17]

5.1.20. Funktionsweise des Compilers kennen

1.

"[Zur Erarbeitung instrumentaler und systemischer Kompetenzen sollen die Studierenden] die grundlegenden Konzepte des Übersetzens und des Compilerentwurfs kennen."

[Modulbeschreibungen\MH16; Position: 28 - 28]

5.1.21. Verteilte Systeme & paralleles Programmieren kennen

1.

"[Die Studierenden sollen weiterhin] verteilte Systeme und paralleles Programmieren kennen (systemische Kompetenz)."

[Modulbeschreibungen\MH16; Position: 17 - 17]

5.1.22. Techniken zur formalen Definition von Programmiersprachen kennen

1.

"[Die Teilnehmer] kennen die Techniken und Notationen zur Definition kontextfreier Programmiersprachen."

[Modulbeschreibungen\MH02; Position: 14 - 14]

5.2. Verstehen

5.2.1. Konzepte von Programmierparadigmen (und –sprachen) beschreiben

1.

"Fachkompetenzen: Die Studierenden beschreiben grundlegende Konzepte der imperativen Programmierung mit Java."

[Modulbeschreibungen\MH21; Position: 10 - 10]

2.

"[Fachkompetenzen:] [Die Studierenden] verwenden die [...] Begriffe [der imperativen Programmierung] präzise bei Diskussionen."

[Modulbeschreibungen\MH21; Position: 12 - 12]

3.

"[Fachkompetenzen:] [Die Studierenden] verwenden die entsprechenden Begriffe [der objektorientierten Modellierung und Programmierung] präzise bei Diskussionen."

[Modulbeschreibungen\MH21; Position: 41 - 41]

4.

"[Fähigkeiten:] Verwendung grundlegender Beschreibungsformen für Programmiersprachen."

[Modulbeschreibungen\MH22; Position: 14 - 14]

5.

"Nach erfolgreicher Absolvierung dieses Moduls können die Studierenden fortgeschrittene Konzepte objektorientierter Programmierung in Java erklären."

[Modulbeschreibungen\MH26; Position: 18 - 18]

6.

"[Die Studierenden] können Unterschiede sowie Vor- und Nachteile zu objektorientierten und typsicheren Sprachen erläutern."

[Modulbeschreibungen\MH26; Position: 27 - 27]

7.

"Die Studierenden können wichtige Programmiersprachen charakterisieren."

[Modulbeschreibungen\MH33; Position: 25 - 25]

8.

"[Die Studierenden] können [...] Anwendungsgebiete [wichtiger Programmiersprachen] gegenüberstellen."

[Modulbeschreibungen\MH33; Position: 26 - 26]

9.

"[Die Studierenden können die grundlegenden Prinzipien der OOP] mit eigenen Worten am Beispiel beschreiben."

[Modulbeschreibungen\MH34; Position: 20 - 20]

10.

"[Die Studierenden können] Anwendungsmöglichkeiten [von Konzepten wie Referenzen, abstrakte Klassen, virtuelle Methoden, Klassenkonstanten, -attribute und -methoden] aufzeigen."

[Modulbeschreibungen\MH34; Position: 22 - 22]

11.

"[Die Studierenden können] [Gemeinsamkeiten und Unterschiede hinsichtlich der Sprachphilosophie und Konzepte von Java und C/C++] erklären."

[Modulbeschreibungen\MH34; Position: 40 - 40]

5.2.2. Algorithmische Probleme und Aufgabenstellungen beschreiben

1.

"[Studierende sollten nach der Veranstaltung in der Lage sein, Probleme] zu beschreiben."

[Modulbeschreibungen\MH24; Position: 37 - 37]

2.

"[Die Studierenden sollen algorithmische Aufgabenstellungen] formulieren können."

[Modulbeschreibungen\MH24; Position: 47 - 47]

3.

"Jeder Studierende kann präzise die von ihm bearbeitete Aufgabe formulieren."

[Modulbeschreibungen\MH28; Position: 26 - 26]

5.2.3. Begriffe formaler Verifikationstechniken erläutern

1.

"[Die Studentinnen und Studenten] können [...] [die Begriffe Laufzeit und Korrektheit] einsetzen."

[Modulbeschreibungen\MH05; Position: 28 - 28]

2.

"Grundbegriffe der formalen Verifikation erläutern können."

[Modulbeschreibungen\MH10; Position: 35 - 35]

5.2.4. Algorithmen, Datenstrukturen und Datentypen charakterisieren

1.

"Weiterhin können [die Studierenden] abschätzen, wie [...] [komplexe Datentypen] funktionieren."

[Modulbeschreibungen\MH01; Position: 23 - 23]

2.

"[Datenstrukturen] bzgl. ihrer Leistungsmerkmale charakterisieren können."

[Modulbeschreibungen\MH04; Position: 29 - 29]

3.

"[Such- und Sortiervverfahren] bzgl. ihrer Leistung und Anwendbarkeit charakterisieren können."

[Modulbeschreibungen\MH04; Position: 32 - 32]

4.

"Elementare Datentypen und Datentypkonstruktoren mit ihren mathematischen Konzepten beschreiben."

[Modulbeschreibungen\MH08; Position: 10 - 10]

5.

"Wesentliche Algorithmen der Informatik erklären können."

[Modulbeschreibungen\MH10; Position: 31 - 31]

6.

"[Studierende sollen nach Absolvierung der Lehrveranstaltungen in der Lage sein,] [den] Ablauf [einfacher Algorithmen (iterativ und rekursiv)] zu beschreiben."

[Modulbeschreibungen\MH23; Position: 22 - 22]

7.

"[Am Ende dieses Moduls] kann [jeder Kursteilnehmer] die Verfahren zur rechnerinternen Darstellung von Daten und Zahlen erläutern."

[Modulbeschreibungen\MH28; Position: 6 - 6]

8.

"[Die Studierenden] beschreiben in eigenen Worten die numerischen Beschränkungen existierender Datentypen an Beispielen."

[Modulbeschreibungen\MH34; Position: 8 - 8]

9.

"[Die Studierenden können] die gegebenen Datenstrukturen und Algorithmen der Java Collections API erläutern."

[Modulbeschreibungen\MH34; Position: 41 - 41]

5.2.5. Softwarearchitektur & deren Entwicklung beschreiben

1.

"Studierende sollen nach erfolgreichem Abschluss des Moduls grundlegende Architekturen beschreiben können."

[Modulbeschreibungen\MH23; Position: 43 - 43]

2.

"Nach erfolgreichem Abschluss des Moduls sollen die Studierenden in der Lage sein, grundlegende Entwicklungsprozesse beschreiben zu können."

[Modulbeschreibungen\MH23; Position: 50 - 50]

5.2.6. Notwendigkeit von Werkzeugen zur Softwareentwicklung begründen

1.

"[Die Studierenden können] die Konzepte und Notwendigkeit von Unit-Tests erläutern."

[Modulbeschreibungen\MH34; Position: 46 - 46]

2.

"[Die Studierenden können] die Notwenigkeit eines Versionskontrollsystems erläutern."

[Modulbeschreibungen\MH34; Position: 48 - 48]

5.3. Anwenden

5.3.1. Werkzeuge zur Software-Entwicklung nutzen

1.

"Die Benutzung einer Entwicklungsumgebung wie Eclipse fördert die Professionalisierung."

[Modulbeschreibungen\MH03; Position: 7 - 7]

2.

"[Die Studierenden] [wenden] Notationen und Werkzeugen gemäß des Stands der Forschung und der industriellen Praxis [an]."

[Modulbeschreibungen\MH03; Position: 34 - 34]

3.

"[Darüber hinaus erlernen die Studierenden das eigenständige] Testen objektorientierter Java-Programme mit einer geeigneten Erstellungsumgebung (z.B. Eclipse)."

[Modulbeschreibungen\MH04; Position: 24 - 24]

4.

"[Die Studierenden] können Werkzeuge zur Build-Automatisierung in größeren Softwareprojekten anwenden."

[Modulbeschreibungen\MH04; Position: 51 - 51]

5.

"Die Teilnehmenden lernen den gekonnten Einsatz einer IDE (Integrated Development Environment)."

[Modulbeschreibungen\MH04; Position: 52 - 52]

6.

"Die Studierenden beherrschen die Techniken: Projektorganisation, Editieren, Debuggen und Fehlersuche."

[Modulbeschreibungen\MH07; Position: 29 - 29]

7.

"[Die Studierenden] können [...] [die Techniken: Projektorganisation, Editieren, Debuggen und Fehlersuche] beispielhaft in der integrierten Entwicklungsumgebung Visual Studio anwenden."

[Modulbeschreibungen\MH07; Position: 30 - 30]

8.

"Neben der Fähigkeit zur selbstständigen Entwicklung programm-technischer Lösungen wird insbesondere der Umgang mit modernen Programmierumgebungen erlernt und geübt."

[Modulbeschreibungen\MH09; Position: 11 - 11]

9.

"Neben der Fähigkeit zur selbstständigen Entwicklung komplexer programmtechnischer Lösungen wird insbesondere der Umgang mit modernen Spezifikationstools und Programmierumgebungen erlernt und geübt."

[Modulbeschreibungen\MH09; Position: 22 - 22]

10.

"Eine Entwicklungsumgebung nutzen können."

[Modulbeschreibungen\MH10; Position: 21 - 21]

11.

"Versionsverwaltungssysteme einsetzen können."

[Modulbeschreibungen\MH10; Position: 23 - 23]

12.

"Die Studierenden können eine moderne Entwicklungsumgebung bedienen (Editor, Debugger)."

[Modulbeschreibungen\MH11; Position: 10 - 10]

13.

"[Die Studierenden] können technische Basis-Hilfsmittel für die teamorientierte SW-Entwicklung anwenden (z.B. Versionsverwaltung)."

[Modulbeschreibungen\MH11; Position: 20 - 20]

14.

"[Die Teilnehmer] können Programmierwerkzeuge wie Compiler und Editoren nutzen."

[Modulbeschreibungen\MH12; Position: 16 - 16]

15.

"[Weiterhin] können [die Studierenden] mit integrierten Entwicklungsumgebungen umgehen."

[Modulbeschreibungen\MH12; Position: 26 - 26]

16.

"Methodenkompetenz: Umgang mit Entwicklungsumgebungen und –werkzeugen."

[Modulbeschreibungen\MH19; Position: 27 - 27]

17.

"Technologische Kompetenzen: Studierende können mit den Werkzeugen gcc, g++ und make umgehen."

[Modulbeschreibungen\MH20; Position: 44 - 44]

18.

"[Fachkompetenzen:] [Die Studierenden] setzen moderne Programmentwicklungsumgebungen zum Entwickeln und Testen von Programmen ein."

[Modulbeschreibungen\MH21; Position: 16 - 16]

19.

"[Fachkompetenzen:] [Die Studierenden] setzen moderne Entwicklungsumgebungen zum Modellieren und Entwickeln von Programmen ein."

[Modulbeschreibungen\MH21; Position: 46 - 46]

20.

"[Studierende sollen nach erfolgreichem Abschluss des Moduls] mit den gängigen Werkzeugen (z.B. IDE) umgehen können."

[Modulbeschreibungen\MH23; Position: 48 - 48]

21.

"[Nach erfolgreichem Abschluss des Moduls sollen die Studierenden in der Lage sein,] mit den Werkzeugen zur Softwareentwicklung im Team (Versionskontrolle, CI Systeme, Bugtracker) sicher umgehen zu können."

[Modulbeschreibungen\MH23; Position: 55 - 55]

22.

"In beiden Vorlesungen und Übungen werden Standard-Entwicklungswerkzeuge der verwendeten Programmiersprache verwendet und deren Verwendung eingeübt."

[Modulbeschreibungen\MH25; Position: 14 - 14]

23.

"Routine im Umgang mit Interpretern und Übersetzern."

[Modulbeschreibungen\MH27; Position: 15 - 15]

24.

"[Die Teilnehmer] verwenden Werkzeuge wie Debugger und Profiler."

[Modulbeschreibungen\MH28; Position: 21 - 21]

25.

"Die Lerneinheit ist darauf ausgerichtet, den Studierenden ein breites Spektrum informatischer Werkzeuge zu eröffnen, die die Teilnehmer [...] selbständig einsetzen (Befähigung Teamkompetenz)."

[Modulbeschreibungen\MH28; Position: 45 - 45]

26.

"Die Studierenden besitzen die Fähigkeiten, formale Werkzeuge (Grundlagen der Berechnung, Übersetzung von Programmkonstruktoren, Programmtransformationen, Verifikation von Programmeigenschaften) zu benutzen."

[Modulbeschreibungen\MH29; Position: 12 - 12]

27.

"[Die Studierenden] sind in der Lage, einfache Aufgaben mittels eines Rechnersimulationsprogramms zu lösen."

[Modulbeschreibungen\MH30; Position: 6 - 6]

28.

"Fertigkeiten im Umgang mit deklarativen Programmierumgebungen."

[Modulbeschreibungen\MH31; Position: 23 - 23]

29.

"[Die Studierenden können] [gängige Entwicklungsumgebungen für die Entwicklung mit Java] verwenden."

[Modulbeschreibungen\MH34; Position: 45 - 45]

30.

"[Die Studierenden können] ein [...] [Versionskontrollsystem] praktisch einsetzen."

[Modulbeschreibungen\MH34; Position: 49 - 49]

5.3.2. Mathematische Berechnungen und Codierungen durchführen

1.

"[Die Studierenden] können [...] [das nötige mathematische Handwerkszeug in konkreten Problemstellungen der Informatik] anwenden."

[Modulbeschreibungen\MH07; Position: 14 - 14]

2.

"[Einfache Codierungen können die Studierenden] bearbeiten."

[Modulbeschreibungen\MH30; Position: 8 - 8]

3.

"[Die Studierenden] sind dann in der Lage, logische Zusammenhänge in logische Formeln umzusetzen und umgekehrt sowie formale Vereinfachungen von logischen Formeln vorzunehmen."

[Modulbeschreibungen\MH32; Position: 30 - 30]

4.

"[Die Studierenden] können einfachere Aufgabenstellungen aus [...] [den Gebieten der Kombinatorik und Wahrscheinlichkeitsrechnung] sicher lösen."

[Modulbeschreibungen\MH32; Position: 32 - 32]

5.

"Die Studierenden codieren Ganzzahlen, Gleitkommazahlen und textuelle Daten in Binärdarstellung oder decodieren sie aus dieser."

[Modulbeschreibungen\MH34; Position: 5 - 5]

5.3.3. Bestehende Bibliotheken einsetzen

1.

"Befähigung zum Einsatz typischer Funktionsbibliotheken für Algorithmen in Programmiersprachen."

[Modulbeschreibungen\MH04; Position: 40 - 40]

2.

"[Nach Beendigung dieses Moduls können die Studierenden] bestehende Bibliotheken nutzen."

[Modulbeschreibungen\MH15; Position: 32 - 32]

3.

"[Die Absolventen] können [...] [die C-Standardbibliothek und weitere ausgewählte Bibliotheken] (und auch andere) Funktionsbibliotheken selbständig einsetzen und anwenden."

[Modulbeschreibungen\MH28; Position: 54 - 54]

4.

"Schließlich können [die Studierenden] auch die Datenstrom-Ein-und Ausgabe anwenden."

[Modulbeschreibungen\MH30; Position: 27 - 27]

5.

"[Die Studierenden können] [...] [Container-Templates der C++-Standardbibliotheken] in korrekter Weise umsetzen."

[Modulbeschreibungen\MH34; Position: 30 - 30]

5.3.4. Programme professionell dokumentieren

1.

"[Anhand kleinerer Problemstellungen lernen] [die Studierenden], professionell zu dokumentieren."

[Modulbeschreibungen\MH09; Position: 10 - 10]

2.

"[Die Strukturierung von Problemen in Teilprobleme] [mit Mitteln von Java] aussagekräftig dokumentieren können."

[Modulbeschreibungen\MH10; Position: 16 - 16]

3.

"[Die Studierenden können] [...] Sourcecode grundlegend unter Zuhilfenahme von Standardwerkzeugen dokumentieren."

[Modulbeschreibungen\MH17; Position: 10 - 10]

4.

"[Jeder Studierende] erstellt eine Programm-Dokumentation."

[Modulbeschreibungen\MH28; Position: 32 - 32]

5.3.5. Qualitätskriterien und Programmierkonventionen auf Quellcode anwenden

1.

"[Die Studierenden] sind in der Lage, Programme in einem guten Programmierstil zu programmieren."

[Modulbeschreibungen\MH07; Position: 20 - 20]

2.

"[Die Studierenden] [können Qualitätskriterien für lesbaren, wartbaren, wiederverwendbaren Quellcode] [...] beim Erstellen eigener Programme umsetzen."

[Modulbeschreibungen\MH11; Position: 15 - 15]

3.

"Fähigkeit, Lösungen entsprechend allgemein anerkannter Qualitätsstandards zu entwerfen."

[Modulbeschreibungen\MH33; Position: 12 - 12]

4.

"[Die Studierenden] halten bei der Lösung von Programmierproblemen die Konventionen der Quelldateiorganisation in C ein."

[Modulbeschreibungen\MH34; Position: 15 - 15]

5.3.6. Computer bedienen

1.

"Die Teilnehmer können sicher mit einem Rechner umgehen."

[Modulbeschreibungen\MH12; Position: 13 - 13]

5.3.7. LaTeX benutzen

1.

"LaTeX zur Erstellung einfacher Dokumente nutzen können."

[Modulbeschreibungen\MH10; Position: 22 - 22]

5.3.8. UNIX-Systeme bedienen mit Hilfe der Kommandozeile

1.

"[Informatik-Kompetenz:] effizientes Benutzen von UNIX-Systemen nur mit der Kommandozeile."

[Modulbeschreibungen\MH20; Position: 15 - 15]

5.4. Analysieren

5.4.1. Programmiersprachen & -paradigmen differenzieren

1.

"Die Studierenden sind in der Lage die verschiedenen Sprachmerkmale (imperative, objekt-orientiert) einzuordnen."

[Modulbeschreibungen\MH01; Position: 29 - 29]

2.

"Einfache Programmiersprachen formal beschreiben können."

[Modulbeschreibungen\MH27; Position: 11 - 11]

3.

"[Die Studierenden] können den Aufbau und die Prinzipien von Programmiersprachen analysieren."

[Modulbeschreibungen\MH33; Position: 30 - 30]

4.

"[Die Studierenden] [können neue Sprachkonstrukte selbstständig] einordnen."

[Modulbeschreibungen\MH33; Position: 32 - 32]

5.

"[Die Studierenden können] Konzepte wie Referenzen, abstrakte Klassen, virtuelle Methoden, Klassenkonstanten, -attribute und -methoden richtig einordnen."

[Modulbeschreibungen\MH34; Position: 21 - 21]

6.

"[Die Studierenden] können Programmiersprachen und deren Konzepte nach wesentlichen Paradigmen klassifizieren (Fachkompetenz)."

[Modulbeschreibungen\MH35; Position: 20 - 20]

5.4.2. Komplexität von Algorithmen analysieren

1.

"Weiterhin können [die Studierenden] abschätzen, welchen Laufzeitaufwand [...] [komplexe Datentypen] besitzen."

[Modulbeschreibungen\MH01; Position: 24 - 24]

2.

"Teilnehmer(innen) des Moduls erlernen die Fähigkeit zur Analyse des Laufzeitverhaltens und des Speicherplatzbedarfs für gegebene Algorithmen."

[Modulbeschreibungen\MH03; Position: 18 - 18]

3.

"[Fertigkeit zur Analyse und Strukturierung technischer Problemstellungen: Die Studierenden lernen, wie sie für vorgegebene Anwendungsfälle geeignete Datenstrukturen und Algorithmen] bzgl. ihrer Leistung analysieren."

[Modulbeschreibungen\MH04; Position: 44 - 44]

4.

"[Studentinnen und Studenten] können die asymptotische Laufzeit von Algorithmen bestimmen."

[Modulbeschreibungen\MH05; Position: 39 - 39]

5.

"[Die Studierenden können die Laufzeit von Algorithmen] vergleichen."

[Modulbeschreibungen\MH06; Position: 14 - 14]

6.

"[Die Studierenden können die Laufzeit von Algorithmen] in die zugehörigen Komplexitätsklassen einordnen."

[Modulbeschreibungen\MH06; Position: 15 - 15]

7.

"[Die Studierenden] können die Leistungsfähigkeit von Algorithmen abschätzen."

[Modulbeschreibungen\MH07; Position: 9 - 9]

8.

"Die Komplexität von einfachen Algorithmen analysieren können."

[Modulbeschreibungen\MH10; Position: 36 - 36]

9.

"[Die Studierenden] können die Effizienz von Algorithmen mit Hilfe mathematischer Methoden analysieren."

[Modulbeschreibungen\MH13; Position: 27 - 27]

10.

"[Die Studierenden] [erwerben die Fähigkeiten] asymptotische Komplexität einzuschätzen."

[Modulbeschreibungen\MH17; Position: 15 - 15]

11.

"[Die Studierenden] [erwerben die Fähigkeiten] [asymptotische Komplexität] zu bestimmen."

[Modulbeschreibungen\MH17; Position: 16 - 16]

12.

"Effizienz von Algorithmen analysieren."

[Modulbeschreibungen\MH18; Position: 15 - 15]

13.

"[Formale und algorithmische Kompetenzen:] [Aufwandsabschätzungen mit Hilfe der O-Notation] [...] auf Algorithmen anwenden."

[Modulbeschreibungen\MH20; Position: 30 - 30]

14.

"[Formale und algorithmische Kompetenzen:] Algorithmen bezüglich deren erwarteter Laufzeit vergleichen können."

[Modulbeschreibungen\MH20; Position: 31 - 31]

15.

"Die Studierenden sollten in der Lage sein, grundlegende Methoden zur Laufzeitanalyse von Algorithmen anzuwenden."

[Modulbeschreibungen\MH22; Position: 21 - 21]

16.

"[Die Studierenden sollen] Algorithmen hinsichtlich Laufzeit analysieren können."

[Modulbeschreibungen\MH24; Position: 50 - 50]

17.

"Laufzeitanalysen entwickeln."

[Modulbeschreibungen\MH26; Position: 7 - 7]

18.

"[Laufzeitanalysen] anwenden können."

[Modulbeschreibungen\MH26; Position: 9 - 9]

19.

"Laufzeit abschätzen."

[Modulbeschreibungen\MH27; Position: 8 - 8]

20.

"[Die Studierenden] lernen Algorithmen und Datenstrukturen bzgl. Zeit- und Platzverbrauch für das übliche RAM Maschinenmodell zu analysieren."

[Modulbeschreibungen\MH27; Position: 27 - 27]

21.

"[Die Studierenden] lernen [...] [auf Basis der Analyse von] [Algorithmen und Datenstrukturen bzgl. Zeit- und Platzverbrauch für das übliche RAM Maschinenmodell] [diese Algorithmen] zu vergleichen."

[Modulbeschreibungen\MH27; Position: 28 - 28]

22.

"Sie lernen verschiedene Arten der Analyse (schlechtester Fall, amortisiert, erwartet) einzusetzen."

[Modulbeschreibungen\MH27; Position: 29 - 29]

23.

"Als erste Schritte zu Komplexitätsanalysen können sie außerdem Algorithmen hinsichtlich ihres Laufzeitverhaltens analysieren."

[Modulbeschreibungen\MH29; Position: 8 - 8]

24.

"[Die Studierenden] [können Algorithmen und Datenstrukturen] [...] [auf Basis [...] von Analysen bzgl. ihrer Laufzeit und ihres Speicherverbrauchs] vergleichen."

[Modulbeschreibungen\MH32; Position: 52 - 52]

5.4.3. Problemstellung zerlegen

1.

"Diverse Probleme erkennen [die Studierenden] von vorneherein als durch Computer unlösbare Aufgabenstellungen."

[Modulbeschreibungen\MH01; Position: 11 - 11]

2.

"Die Studierenden lernen, kleine Probleme zu analysieren."

[Modulbeschreibungen\MH01; Position: 14 - 14]

3.

"[Die Studierenden] wenden bestehende Lösungskonzepte an, um komplexere Probleme in kleinere aufzuteilen."

[Modulbeschreibungen\MH01; Position: 16 - 16]

4.

"Um diese Lösungsstrategien umzusetzen, lernen die Studenten zunächst, wie man an einfache mathematische und technische Problemstellungen herangeht (Analyse)."

[Modulbeschreibungen\MH04; Position: 14 - 14]

5.

"[Um diese Lösungsstrategien umzusetzen, lernen die Studenten zunächst,] wie man einfache Probleme strukturiert."

[Modulbeschreibungen\MH04; Position: 15 - 15]

6.

"Für die Umsetzung lernen Studenten am Beispiel der Programmiersprache Java, wie man mit analytischem und konzeptionellem Denken die Problemstellungen zerlegt."

[Modulbeschreibungen\MH04; Position: 16 - 16]

7.

"Sie erwerben die Fertigkeit zur Analyse technischer Problemstellungen an größeren geeigneten Beispielen."

[Modulbeschreibungen\MH04; Position: 21 - 21]

8.

"[Sie erwerben die Fertigkeit] zur Strukturierung technischer Problemstellungen an größeren geeigneten Beispielen."

[Modulbeschreibungen\MH04; Position: 22 - 22]

9.

"Probleme in Teilprobleme zerlegen."

[Modulbeschreibungen\MH10; Position: 14 - 14]

10.

"Einfache Problemstellungen zu strukturieren [ist] umsetzbar."

[Modulbeschreibungen\MH13; Position: 12 - 12]

11.

"Algorithmische Kompetenz: eine konkrete Problemstellung analysieren."

[Modulbeschreibungen\MH20; Position: 5 - 5]

12.

"Algorithmische Kompetenz: eine konkrete Problemstellung analysieren."

[Modulbeschreibungen\MH20; Position: 21 - 21]

13.

"[Studierende sollen nach erfolgreichem Abschluss des Moduls] selbständig Problemstellungen analysieren können."

[Modulbeschreibungen\MH23; Position: 44 - 44]

14.

"[Nach erfolgreichem Abschluss des Moduls sollen die Studierenden in der Lage sein,] [eine gestellte komplexere Aufgabe] in Teilaufgaben strukturieren zu können."

[Modulbeschreibungen\MH23; Position: 52 - 52]

15.

"Die Studierenden sollen die objektorientierten Konzepte in der Phase Analyse anwenden können."

[Modulbeschreibungen\MH24; Position: 25 - 25]

16.

"Somit ist es den Studierenden möglich, einfache praxisrelevante Problemstellungen selbständig zu analysieren."

[Modulbeschreibungen\MH28; Position: 9 - 9]

17.

"[Jeder Studierende kann präzise die von ihm bearbeitete Aufgabe] abgrenzen."

[Modulbeschreibungen\MH28; Position: 27 - 27]

18.

"[Jeder Studierende] kann er zu lösenden (Teil-) Probleme identifizieren."

[Modulbeschreibungen\MH28; Position: 28 - 28]

19.

"Ganz allgemein sollen [die Studierenden] Problemstellungen aus dem Gebiet der Informatik analysieren und deren mathematischen Kern erkennen können."

[Modulbeschreibungen\MH32; Position: 34 - 34]

20.

"[Die Studierenden] zerlegen komplexere verbale Aufgabenstellungen hierarchisch in Teilaufgaben (Top-Down-Entwurf)."

[Modulbeschreibungen\MH34; Position: 12 - 12]

5.4.4. Algorithmen auf Eigenschaften und Eignung hin analysieren

1.

"[Für vorgegebene Anwendungsfälle geeignete Algorithmen] analysieren."

[Modulbeschreibungen\MH04; Position: 35 - 35]

2.

"[Die Studentinnen und Studenten] können am Ende des Moduls Beschreibungen und Quelltexte elementarer Algorithmen lesen."

[Modulbeschreibungen\MH05; Position: 18 - 18]

3.

"[Die Studierenden] können Algorithmen analysieren."

[Modulbeschreibungen\MH11; Position: 32 - 32]

4.

"[Formale und algorithmische Kompetenzen:] [gute Algorithmen für wichtige Standardprobleme (z.B. Sortieren)] vergleichen."

[Modulbeschreibungen\MH20; Position: 33 - 33]

5.

"[Studierende sollen nach erfolgreichem Abschluss des Moduls] eine Lösung hinsichtlich der Wartbarkeit analysieren können."

[Modulbeschreibungen\MH23; Position: 46 - 46]

6.

"[Studierende sollten nach der Veranstaltung in der Lage sein,] Funktionen höherer Ordnung zu erkennen."

[Modulbeschreibungen\MH24; Position: 41 - 41]

7.

"Die Studierenden können konkrete Algorithmen hinsichtlich ihrer unterschiedlichen Eigenschaften und Eignungen analysieren."

[Modulbeschreibungen\MH28; Position: 42 - 42]

8.

"[Die Studierenden können konkrete Algorithmen hinsichtlich ihrer unterschiedlichen Eigenschaften und Eignungen] vergleichen."

[Modulbeschreibungen\MH28; Position: 43 - 43]

9.

"[Beherrschen grundlegender Prinzipien und Methoden für] Analyse von Algorithmen."

[Modulbeschreibungen\MH33; Position: 23 - 23]

5.4.5. (Fremde) Programme beschreiben können

1.

"[Die Teilnehmer] sind in der Lage, kleine Programme (bis zu einigen hundert Zeilen) zu analysieren."

[Modulbeschreibungen\MH02; Position: 7 - 7]

2.

"[Die Studierenden] sind in der Lage fremde Programme zu lesen."

[Modulbeschreibungen\MH07; Position: 27 - 27]

3.

"[Die Studierenden] können fremden Quellcode analysieren."

[Modulbeschreibungen\MH11; Position: 12 - 12]

4.

"[Die Studierenden] sind in der Lage, größere Anwendungen zu strukturieren."

[Modulbeschreibungen\MH13; Position: 21 - 21]

5.

"[Die Studierenden] [können einfache Anwendungen mittels systemnaher und/oder paralleler Programmierung auf ausgewählten Plattformen] [...] analysieren."

[Modulbeschreibungen\MH17; Position: 22 - 22]

6.

"[Fachkompetenzen:] [Die Studierenden] beschreiben, was ihnen vorgelegte Programme tun."

[Modulbeschreibungen\MH21; Position: 13 - 13]

7.

"[Fachkompetenzen:] [Die Studierenden] können beschreiben, was ihnen vorgelegte objektorientierte Programme tun."

[Modulbeschreibungen\MH21; Position: 42 - 42]

8.

"[Die Studierenden sollen] einfache Java-Programme analysieren können."

[Modulbeschreibungen\MH24; Position: 11 - 11]

5.4.6. Datenstrukturen auf Eigenschaften und Eignung hin analysieren

1.

"[Für vorgegebene Anwendungsfälle geeignete Datenstrukturen] analysieren."

[Modulbeschreibungen\MH04; Position: 34 - 34]

2.

"Typische Datenstrukturen identifizieren."

[Modulbeschreibungen\MH10; Position: 28 - 28]

3.

"[Die Studierenden] können Datenstrukturen analysieren."

[Modulbeschreibungen\MH11; Position: 33 - 33]

4.

"[Wissen und Verstehen:] [Die Studierenden sollen grundlegende Datenstrukturen mit deren Eigenschaften und Leistungsparametern] [...] [in asymptotischer Notation] vergleichen können."

[Modulbeschreibungen\MH16; Position: 7 - 7]

5.

"[Können:] [Die Studierenden lernen,] [...] [Leistungsparameter für eigenständig entworfene Datenstrukturen zur Lösung neuer Problemstellungen] zu analysieren [(instrumentale Kompetenz)]."

[Modulbeschreibungen\MH16; Position: 9 - 9]

6.

"[Analyse-, Design- und Realisierungs-Kompetenzen:] [verschiedene Implementierungsvarianten für wichtige Datenstrukturen] vergleichen."

[Modulbeschreibungen\MH20; Position: 37 - 37]

5.4.7. Formale Sprachen durch Chomsky-Hierarchie differenzieren

1.

"[Die Studierenden] klassifizieren formale Sprachen mit Hilfe der sog. Chomsky-Hierarchie."

[Modulbeschreibungen\MH01; Position: 7 - 7]

5.4.8. Grenzen der Berechnungskraft von Rechnermodellen aufzeigen

1.

"[Weiterhin] können [die Studierenden] mit exakten logischen Argumenten [...] Grenzen [der Berechnungskraft gängiger Rechnermodelle durch endliche Automaten] aufzeigen."

[Modulbeschreibungen\MH01; Position: 10 - 10]

5.5. Bewerten

5.5.1. Angemessenheit von Algorithmen, Datenstrukturen und Datentypen beurteilen

1.

"Weiterhin können [die Studierenden] abschätzen, in welcher Situation bestimmte komplexe Datentypen eingesetzt werden."

[Modulbeschreibungen\MH01; Position: 22 - 22]

2.

"[Unter Anwendung der erlangten Fähigkeiten und Kenntnisse sollen die Teilnehmer in die Lage versetzt werden,] für ein gegebenes Problem die Verwendung verschiedener Lösungsmöglichkeiten nach formalen Kriterien zu beurteilen."

[Modulbeschreibungen\MH03; Position: 24 - 24]

3.

"Für vorgegebene Anwendungsfälle geeignete Datenstrukturen und Algorithmen finden."

[Modulbeschreibungen\MH04; Position: 33 - 33]

4.

"[Für vorgegebene Anwendungsfälle geeignete Datenstrukturen und Algorithmen] bewerten können."

[Modulbeschreibungen\MH04; Position: 36 - 36]

5.

"Kenntnisse von praxisrelevanten Aufgabenstellungen: Anhand praktischer Beispiele werden die Einsatzszenarien für verschiedene Algorithmen erarbeitet."

[Modulbeschreibungen\MH04; Position: 48 - 48]

6.

"[Studentinnen und Studenten] können [...] [die wichtigsten abstrakten Datentypen und die Datenstrukturen zu deren Implementierung] in Bezug auf ihre Eigenschaften beurteilen."

[Modulbeschreibungen\MH05; Position: 35 - 35]

7.

"[Studentinnen und Studenten] können [...] [die wichtigsten abstrakten Datentypen und die Datenstrukturen zu deren Implementierung] [in Bezug auf ihre Eigenschaften] geeignet auswählen."

[Modulbeschreibungen\MH05; Position: 36 - 36]

8.

"[Die Studierenden] können für grundlegende Aufgabenstellungen passende Algorithmen und Datenstrukturen auswählen."

[Modulbeschreibungen\MH06; Position: 12 - 12]

9.

"Die Studentinnen sollen die in der Veranstaltung vermittelten Begriffe und Konzepte zu Algorithmen und Datenstrukturen derart auf beispielhafte Problemstellungen anwenden können, dass sie geeignete Standard-Datenstrukturen auswählen."

[Modulbeschreibungen\MH09; Position: 24 - 24]

10.

"[Die Studentinnen sollen die in der Veranstaltung vermittelten Begriffe und Konzepte zu Algorithmen und Datenstrukturen derart auf beispielhafte Problemstellungen anwenden können, dass sie] diese auf Eleganz beurteilen können."

[Modulbeschreibungen\MH09; Position: 29 - 29]

11.

"Algorithmische Alternativen bezüglich der Eignung für ein Problem beurteilen können."

[Modulbeschreibungen\MH10; Position: 34 - 34]

12.

"[Die Studierenden] sind in der Lage, [...] [algorithmische Lösungen] im Hinblick auf Problemadäquatheit zu bewerten."

[Modulbeschreibungen\MH12; Position: 6 - 6]

13.

"[Die Studierenden] verfügen über grundlegende Fertigkeiten für die Auswahl von Algorithmen vor dem Hintergrund konkreter Informationsverarbeitungsaufgaben."

[Modulbeschreibungen\MH12; Position: 9 - 9]

14.

"[Fach- und Methodenkompetenzen (Wissen und Verstehen sowie Anwendung und Erzeugung von Wissen):] Auswahl von Algorithmen und Datenstrukturen für gegebene Problemstellungen sind typische Aufgaben eines Informatikers."

[Modulbeschreibungen\MH15; Position: 26 - 26]

15.

"[Nach Beendigung dieses Moduls können die Studierenden] für Problemstellungen passende Algorithmen und Datenstrukturen auswählen."

[Modulbeschreibungen\MH15; Position: 31 - 31]

16.

"[Methodenkompetenz:] Auswahl geeigneter Algorithmen."

[Modulbeschreibungen\MH19; Position: 17 - 17]

17.

"Selbst- und Sozialkompetenz: Fähigkeit zur Argumentation über gewählte Ansätze und zur Begründung von Entwurfsentscheidungen."

[Modulbeschreibungen\MH19; Position: 20 - 20]

18.

"[Methodenkompetenz:] Auswahl geeigneter Prozesse und Methoden."

[Modulbeschreibungen\MH19; Position: 29 - 29]

19.

"[Analyse-, Design- und Realisierungs-Kompetenzen:] [verschiedene Implementierungsvarianten für wichtige Datenstrukturen] auswählen können."

[Modulbeschreibungen\MH20; Position: 38 - 38]

20.

"[Fachkompetenzen:] [Die Studierenden] bewerten [...] [Algorithmen und Datenstrukturen zur Lösung von häufig vorkommenden Problemen] in ihrer Anwendbarkeit."

[Modulbeschreibungen\MH21; Position: 19 - 19]

21.

"[Fachkompetenzen:] [Die Studierenden] wählen fundiert einen Algorithmus und eine Datenstruktur zur Lösung eines konkreten Problems aus."

[Modulbeschreibungen\MH21; Position: 22 - 22]

22.

"[Die Studierenden sollen] die wesentlichen Such- und Sortieralgorithmen nach Anwendungsszenarien beurteilen."

[Modulbeschreibungen\MH24; Position: 21 - 21]

23.

"[Die Studierenden sollen] [die wesentlichen Such- und Sortieralgorithmen nach Anwendungsszenarien] auswählen können."

[Modulbeschreibungen\MH24; Position: 22 - 22]

24.

"[Die Studierenden] sind außerdem in der Lage, die für ein Anwendungsszenario geeignete Datenstruktur auszuwählen."

[Modulbeschreibungen\MH26; Position: 24 - 24]

25.

"Sie sollen die Fähigkeit erwerben, vorhandene Methoden durch theoretische Analysen und Abwägungen für ihre Verwendbarkeit in tatsächlich auftretenden Szenarien zu prüfen."

[Modulbeschreibungen\MH27; Position: 31 - 31]

26.

"Die Teilnehmer dieses Moduls sind in der Lage, aus einer breiten Auswahl an Standardlösungen (Fachkompetenz) die am besten geeignete auszuwählen (Methodenkompetenz)."

[Modulbeschreibungen\MH28; Position: 38 - 38]

27.

"[Die Studierenden können konkrete Algorithmen hinsichtlich ihrer unterschiedlichen Eigenschaften und Eignungen] auswählen."

[Modulbeschreibungen\MH28; Position: 44 - 44]

28.

"Die Studierenden verfügen über Kompetenzen zur Auswahl geeigneter Datenstrukturen und Algorithmen in der Softwareentwicklung zur Verbesserung der Effizienz und der Wartbarkeit der Programme sowie zur Verkürzung der Entwicklungszeit."

[Modulbeschreibungen\MH32; Position: 53 - 53]

29.

"[Die Studierenden] können über die Unterstützung von Typsystemen für korrekte Programme argumentieren."

[Modulbeschreibungen\MH33; Position: 33 - 33]

30.

"[Die Studierenden] wählen geeignete Standarddatentypen (C) für Daten vorgegebener Anwendungsbeispiele aus."

[Modulbeschreibungen\MH34; Position: 6 - 6]

31.

"[Die Studierenden] legen die [...] Datentypen, -strukturen und Funktionssignaturen [komplexerer verbaler Aufgabenstellungen und deren Teilaufgaben] sinnvoll fest."

[Modulbeschreibungen\MH34; Position: 13 - 13]

32.

"[Die Studierenden können] [aus einer verbalen Aufgabenstellung] [die Auswahl passender Schnittstellen] begründen."

[Modulbeschreibungen\MH34; Position: 27 - 27]

33.

"Methodenkompetenz: [Die Studierenden] sind in der Lage, Algorithmen hinsichtlich ihrer Eigenschaften und Anwendbarkeit für konkrete Problemstellungen zu identifizieren."

[Modulbeschreibungen\MH35; Position: 9 - 9]

34.

"[Methodenkompetenz:] [Die Studierenden] [sind in der Lage, Algorithmen hinsichtlich ihrer Eigenschaften und Anwendbarkeit für konkrete Problemstellungen] und bewerten."

[Modulbeschreibungen\MH35; Position: 10 - 10]

5.5.2. Testen von Algorithmen und Programmen auf Fehler

1.

"[Anhand kleinerer Problemstellungen lernen] [die Studierenden], [programmiersprachlich] zu testen."

[Modulbeschreibungen\MH09; Position: 9 - 9]

2.

"Einfache in Java realisierte Algorithmen systematisch testen können."

[Modulbeschreibungen\MH10; Position: 13 - 13]

3.

"[Die Studierenden] können Programme automatisiert testen."

[Modulbeschreibungen\MH11; Position: 16 - 16]

4.

"[Die Teilnehmer] können die Tragweite von Tests abschätzen."

[Modulbeschreibungen\MH12; Position: 21 - 21]

5.

"[Die Studierenden] [sind in der Lage ein prozedurales Programm] zu testen."

[Modulbeschreibungen\MH13; Position: 32 - 32]

6.

"Validierung durch systematische Tests."

[Modulbeschreibungen\MH14; Position: 20 - 20]

7.

"[Die Studierenden können] [...] Qualitätssicherung mittels einfacher (Unit-) Tests durchführen."

[Modulbeschreibungen\MH17; Position: 7 - 7]

8.

"[Realisierungs-Kompetenz:] Testen von Programmen unter Einsatz entsprechender Werkzeuge."

[Modulbeschreibungen\MH20; Position: 11 - 11]

9.

"[Realisierungs-Kompetenz:] Testen von Programmen unter Einsatz entsprechender Werkzeuge."

[Modulbeschreibungen\MH20; Position: 27 - 27]

10.

"[Design- und Realisierungskompetenzen:] Testen eines größeren Software-Programms im Team."

[Modulbeschreibungen\MH20; Position: 53 - 53]

11.

"[Fachkompetenzen:] [Die Studierenden] untersuchen systematisch eigene und fremde Programme auf Fehler."

[Modulbeschreibungen\MH21; Position: 15 - 15]

12.

"[Fachkompetenzen:] [Die Studierenden] untersuchen systematisch eigene und fremde Modelle und Programme auf Fehler."

[Modulbeschreibungen\MH21; Position: 45 - 45]

13.

"[Nach erfolgreichem Abschluss des Moduls sollen die Studierenden in der Lage sein,] die einzelnen Bestandteile sowie das Gesamtprogramms mit geeigneten Verfahren testen zu können."

[Modulbeschreibungen\MH23; Position: 54 - 54]

14.

"[Die Studierenden sollen] [die wesentlichen Testverfahren] anwenden können."

[Modulbeschreibungen\MH24; Position: 34 - 34]

15.

"[Die Studierenden] können [...] [grundlegende Datenstrukturen und Algorithmen] testen."

[Modulbeschreibungen\MH25; Position: 19 - 19]

16.

"[In dieser Vorlesung lernen sie] [Mittelgroße objektorientierte Systeme in Java] zu testen."

[Modulbeschreibungen\MH27; Position: 23 - 23]

17.

"Die Teilnehmer sind in der Lage, systematisch nach Fehlern in Programmen zu suchen bzw. diese zu validieren."

[Modulbeschreibungen\MH28; Position: 19 - 19]

18.

"Jeder Studierende führt eigenverantwortlich einen Programm-Test durch."

[Modulbeschreibungen\MH28; Position: 31 - 31]

19.

"[Die Studierenden können] Unit-Tests erfolgreich bei der Software-Entwicklung einsetzen."

[Modulbeschreibungen\MH34; Position: 47 - 47]

5.5.3. Komplexität von Algorithmen beurteilen

1.

"[Die Studierenden] erkennen [...] [durch die Klassifikation formaler Sprachen] ihre algorithmische Komplexität."

[Modulbeschreibungen\MH01; Position: 8 - 8]

2.

"Die Studierenden können die Laufzeit von Algorithmen bewerten."

[Modulbeschreibungen\MH06; Position: 13 - 13]

3.

"[Die Studierenden] [können die Leistungsfähigkeit von Algorithmen] beurteilen."

[Modulbeschreibungen\MH07; Position: 10 - 10]

4.

"Bewertung [einfacher Algorithmen] hinsichtlich Zeit- und Platzkomplexität."

[Modulbeschreibungen\MH08; Position: 18 - 18]

5.

"Bewertung [einfacher Algorithmen] hinsichtlich Zeit- und Platzkomplexität."

[Modulbeschreibungen\MH08; Position: 20 - 20]

6.

"[Die Studentinnen sollen die in der Veranstaltung vermittelten Begriffe und Konzepte zu Algorithmen und Datenstrukturen derart auf beispielhafte Problemstellungen anwenden können, dass sie] diese auf Komplexität beurteilen können."

[Modulbeschreibungen\MH09; Position: 28 - 28]

7.

"[Die Studierenden] sind in der Lage, [...] [algorithmische Lösungen] im Hinblick auf Zeit- und Platzkomplexität zu bewerten."

[Modulbeschreibungen\MH12; Position: 7 - 7]

8.

"[Nach Beendigung dieses Moduls können die Studierenden selbstständig Algorithmen] bewerten (Laufzeit)."

[Modulbeschreibungen\MH15; Position: 28 - 28]

9.

"[Die Studierenden können] [...] basierend [auf den Komplexitätsklassen von Algorithmen] die Eignung selbiger für konkrete Aufgaben einschätzen."

[Modulbeschreibungen\MH17; Position: 9 - 9]

10.

"[Effizienz von Algorithmen] bewerten."

[Modulbeschreibungen\MH18; Position: 16 - 16]

11.

"[Die Studierenden sollen nach Abschluss des Moduls] die Effizienz ihrer Programme einschätzen können."

[Modulbeschreibungen\MH21; Position: 9 - 9]

12.

"[Fachkompetenzen:] [Die Studierenden] bewerten [...] [Probleme der Effizienz von algorithmischen Lösungen konkreter Fragestellungen]."

[Modulbeschreibungen\MH21; Position: 21 - 21]

13.

"[Die Studierenden] begründen die Auswahl geeigneter aggregierender Datenstrukturen (wie Arrays, verkettete Listen, Bäume) an Hand ihres Laufzeitverhaltens entsprechend dem Anforderungsprofil einer Anwendungsaufgabe."

[Modulbeschreibungen\MH34; Position: 16 - 16]

14.

"[Fachkompetenz:] [Die Studierenden] [können grundlegende Algorithmen] [...] durch Angabe und Abschätzen von Rekurrenzgleichungen in Ihrer Laufzeitkomplexität bewerten."

[Modulbeschreibungen\MH35; Position: 8 - 8]

5.5.4. Eigenschaften von Algorithmen und Programmen beweisen

1.

"Die vorgestellten Ergebnisse können die Studierenden durch den sicheren Umgang mit verschiedenen Beweistechniken belegen."

[Modulbeschreibungen\MH01; Position: 12 - 12]

2.

"[Die Studentinnen und Studenten] können die Eigenschaften von kleinen Programmen formal und informell beweisen."

[Modulbeschreibungen\MH05; Position: 12 - 12]

3.

"[Die Studentinnen und Studenten] [können am Ende des Moduls] Eigenschaften von funktionalen Programmen formal beweisen."

[Modulbeschreibungen\MH05; Position: 26 - 26]

4.

"[Studentinnen und Studenten] können die Korrektheit von Algorithmen nachweisen."

[Modulbeschreibungen\MH05; Position: 38 - 38]

5.

"[Die Studentinnen sollen die in der Veranstaltung vermittelten Begriffe und Konzepte zu Algorithmen und Datenstrukturen derart auf beispielhafte Problemstellungen anwenden können, dass sie] diese auf Korrektheit beurteilen können."

[Modulbeschreibungen\MH09; Position: 27 - 27]

6.

"[Operationelle Semantik einfacher While-Sprachen] zum Nachweis einfacher Programmeigenschaften anwenden können."

[Modulbeschreibungen\MH10; Position: 20 - 20]

7.

"[Die Studierenden] sind in der Lage, [...] [algorithmische Lösungen] im Hinblick auf Korrektheit und Vollständigkeit zu bewerten."

[Modulbeschreibungen\MH12; Position: 8 - 8]

8.

"[Fähigkeiten:] Programmverifikation."

[Modulbeschreibungen\MH22; Position: 15 - 15]

9.

"[Die Studierenden sollen] Algorithmen hinsichtlich Korrektheit analysieren können."

[Modulbeschreibungen\MH24; Position: 49 - 49]

10.

"Korrektheit beweisen."

[Modulbeschreibungen\MH27; Position: 7 - 7]

5.5.5. Angemessenheit programmiersprachlicher Lösungen beurteilen

1.

"[Die Studierenden sind in der Lage die verschiedenen Sprachmerkmale (imperative, objekt-orientiert)] je nach Anwendungsfall geeignet einzusetzen."

[Modulbeschreibungen\MH01; Position: 30 - 30]

2.

"[Fähigkeit, kleine Algorithmen] [...] im Vergleich mit imperativen Lösungen zu bewerten."

[Modulbeschreibungen\MH03; Position: 12 - 12]

3.

"Die Qualität [...] [einer eigenen] Lösung anhand grundlegender Merkmale zu reflektieren [...] [ist] umsetzbar."

[Modulbeschreibungen\MH13; Position: 14 - 14]

4.

"[Die Studierenden] können zu vorgegebenen Aufgaben geeignete Programmiersprachen auswählen."

[Modulbeschreibungen\MH33; Position: 34 - 34]

5.

"Die Studierenden sind in der Lage, zu gegebenen Problemen geeignete Paradigmen kritisch auszuwählen."

[Modulbeschreibungen\MH35; Position: 21 - 21]

5.5.6. Beurteilen von Programmierwerkzeugen & Templates

1.

"[Die Teilnehmer können] [die] Grenzen [von] [Programmierungswerkzeugen wie Compiler und Editoren] einschätzen."

[Modulbeschreibungen\MH12; Position: 17 - 17]

2.

"[Die Studierenden können] analysieren, wo Container-Templates der C++-Standardbibliotheken effizient eingesetzt werden können."

[Modulbeschreibungen\MH34; Position: 29 - 29]

5.5.7. Angemessenheit des Selbstkonzepts bewerten

1.

"Nach erfolgreicher Teilnahme am Softwareentwicklungspraktikum sollen sich die Teilnehmer trauen, Werkstudentenjobs in der IT-Industrie anzunehmen."

[Modulbeschreibungen\MH03; Position: 28 - 28]

5.6. Erzeugen

5.6.1. Programmiersprachliche, lauffähige Lösungen für Probleme schreiben

1.

"[Die Studierenden] können am Ende zu gängigen Problemen geeignete programmiersprachliche Lösungen angeben."

[Modulbeschreibungen\MH02; Position: 18 - 18]

2.

"[Die Studierenden] [können am Ende zu gängigen Problemen geeignete programmiersprachliche Lösungen] [...] in einer konkreten Programmiersprache formulieren."

[Modulbeschreibungen\MH02; Position: 19 - 19]

3.

"[Die Studierenden werden in die Lage versetzt, Lösungen für kleinere und überschaubare Probleme] mit einer höheren Programmiersprache als ausführbare Programme zu realisieren."

[Modulbeschreibungen\MH03; Position: 6 - 6]

4.

"Fähigkeit, kleine Algorithmen funktional zu programmieren."

[Modulbeschreibungen\MH03; Position: 11 - 11]

5.

"[Die Studierenden] sind in der Lage, die Modelle [struktureller und dynamischer Eigenschaften komplexer Softwaresysteme] in Software zu überführen."

[Modulbeschreibungen\MH03; Position: 36 - 36]

6.

"[Die Studierenden sind in der Lage eigenständig eine Lösungsstrategie zum Schreiben kleiner prozeduraler und objektorientierter Java-Programme nach einer vorgegebenen Entwurfsidee] umzusetzen."

[Modulbeschreibungen\MH04; Position: 13 - 13]

7.

"Darüber hinaus erlernen die Studierenden das eigenständige Schreiben objektorientierter Java-Programme mit einer geeigneten Erstellungsumgebung (z.B. Eclipse)."

[Modulbeschreibungen\MH04; Position: 23 - 23]

8.

"[Die Studierenden] können die in C++ typischen Sprachkonstrukte erfolgreich anwenden."

[Modulbeschreibungen\MH04; Position: 59 - 59]

9.

"[Die Studierenden] haben gelernt, ihre Problemlösungen [...] [in einer Programmiersprache] auszudrücken."

[Modulbeschreibungen\MH06; Position: 8 - 8]

10.

"[Die Studierenden] lernen, ihre Problemlösungen [...] [in einer objektorientierten Programmiersprache] auszudrücken."

[Modulbeschreibungen\MH06; Position: 21 - 21]

11.

"[Die Studierenden haben] gelernt, Problemlösungen [...] [in einer objektorientierten Programmiersprache] auszudrücken."

[Modulbeschreibungen\MH06; Position: 27 - 27]

12.

"[Die Studierenden] können [...] [die Entwurfsvorgaben in Klassendiagrammen] in Programme umsetzen."

[Modulbeschreibungen\MH07; Position: 19 - 19]

13.

"[Die Studierenden] sind in der Lage, selbständig Programme in C und C++ zu schreiben."

[Modulbeschreibungen\MH07; Position: 26 - 26]

14.

"Einfache Algorithmen in Programme funktionaler und imperativer Notation (z.B. Python) umsetzen können."

[Modulbeschreibungen\MH08; Position: 8 - 8]

15.

"Kleinere funktionale Programme schreiben können."

[Modulbeschreibungen\MH08; Position: 13 - 13]

16.

"Die Studierenden erwerben grundlegende theoretische und praktische Fähigkeiten zur Entwicklung von Software mit Hilfe höherer Programmiersprachen."

[Modulbeschreibungen\MH09; Position: 5 - 5]

17.

"[Anhand kleinerer Problemstellungen lernen] [die Studierenden], programmiersprachlich zu formulieren."

[Modulbeschreibungen\MH09; Position: 7 - 7]

18.

"Die Studierenden erwerben grundlegende theoretische und praktische Fähigkeiten zur Entwicklung von Software mit Hilfe objektorientierter Programmiersprachen."

[Modulbeschreibungen\MH09; Position: 18 - 18]

19.

"[Einfache Algorithmen] in Java umsetzen können."

[Modulbeschreibungen\MH10; Position: 12 - 12]

20.

"[Die Strukturierung von Problemen in Teilprobleme] mit Mitteln von Java umsetzen"

[Modulbeschreibungen\MH10; Position: 15 - 15]

21.

"Datenstrukturen und Algorithmen in Java umsetzen können."

[Modulbeschreibungen\MH10; Position: 30 - 30]

22.

"[Konzepte und typische Merkmale des funktionalen Programmierens] anwenden können."

[Modulbeschreibungen\MH10; Position: 43 - 43]

23.

"Datenstrukturen und Algorithmen in einer funktionalen Programmiersprache umsetzen können."

[Modulbeschreibungen\MH10; Position: 44 - 44]

24.

"[Datenstrukturen und Algorithmen in einer funktionalen Programmiersprache] auf einfachere praktische Probleme anwenden können."

[Modulbeschreibungen\MH10; Position: 45 - 45]

25.

"[Die Studierenden] können einfache Programme in einer modernen Programmiersprache entwickeln."

[Modulbeschreibungen\MH11; Position: 11 - 11]

26.

"[Die Studierenden können fortgeschrittene Programmier Techniken] [...] in kleinen Beispielprogrammen in einer aktuellen Programmiersprache einsetzen."

[Modulbeschreibungen\MH11; Position: 19 - 19]

27.

"Die Studierenden beherrschen die Grundlagen zur Entwicklung kleiner, gebrauchstauglicher Anwendungen mit Hilfe objektorientierter Konzepte."

[Modulbeschreibungen\MH12; Position: 23 - 23]

28.

"Darüber hinaus sind [die Studierenden] in der Lage, korrekten, lesbaren und wartbaren Code zu erzeugen."

[Modulbeschreibungen\MH13; Position: 10 - 10]

29.

"[Die Studierenden sollen] einfache Programmieraufgaben lösen."

[Modulbeschreibungen\MH16; Position: 21 - 21]

30.

"Die Studierenden können [...] einfache Programmieraufgaben mit Hilfe von funktionalen und/oder objektorientierten Programmiersprachen systematisch lösen."

[Modulbeschreibungen\MH17; Position: 6 - 6]

31.

"Anwenden von Grundlagen der prozeduralen und der objektorientierten Programmierung am Beispiel der Sprachen C und C++."

[Modulbeschreibungen\MH18; Position: 6 - 6]

32.

"[Das objektorientierte Programmierparadigma] mit Java richtig anwenden."

[Modulbeschreibungen\MH18; Position: 21 - 21]

33.

"[Prinzipien der Generischen Programmierung] anwenden (C++)."

[Modulbeschreibungen\MH18; Position: 23 - 23]

34.

"[Probleme und Lösungsansätze der nebenläufigen Programmierung] anwenden [(Java)]."

[Modulbeschreibungen\MH18; Position: 25 - 25]

35.

"[Realisierungs-Kompetenz:] Erstellen von Programmen unter Einsatz entsprechender Werkzeuge."

[Modulbeschreibungen\MH20; Position: 10 - 10]

36.

"[Informatik-Kompetenz:] Programmierung in Assembler."

[Modulbeschreibungen\MH20; Position: 16 - 16]

37.

"Design-Kompetenz: UML-Modelle in ablauffähige Programme umsetzen und dabei grundlegende Entwurfsprinzipien berücksichtigen."

[Modulbeschreibungen\MH20; Position: 24 - 24]

38.

"[Realisierungs-Kompetenz:] Erstellen von Programmen unter Einsatz entsprechender Werkzeuge."

[Modulbeschreibungen\MH20; Position: 26 - 26]

39.

"[Design-, Realisierungs- und Methodenkompetenzen:] [Studierende] können typische Programmierprobleme in C90 lösen."

[Modulbeschreibungen\MH20; Position: 47 - 47]

40.

"[Design-, Realisierungs- und Methodenkompetenzen:] [Studierende] können einfache Programmierprobleme in C++98 lösen."

[Modulbeschreibungen\MH20; Position: 49 - 49]

41.

"Die Studierenden sollen nach Abschluss des Moduls selbstständig imperative Programme auf der Grundlage von Java für die Lösung kleinerer Probleme entwickeln."

[Modulbeschreibungen\MH21; Position: 8 - 8]

42.

"[Gegebene Problemstellungen werden [...] [in der Softwareentwicklung] mit Hilfe objektorientierter Analyse- und Entwurfsverfahren] in ein objektorientiertes Programm überführt."

[Modulbeschreibungen\MH21; Position: 34 - 34]

43.

"[Ziel des Moduls „Objektorientierte Modellierung und Programmierung“ ist das Erlernen] der objektorientierten Programmierung mit der Programmiersprache Java."

[Modulbeschreibungen\MH21; Position: 36 - 36]

44.

"Die Studierenden sollen nach Abschluss des Moduls selbstständig objektorientierte Programme auf der Grundlage von Java für die Lösung mittelgroßer Probleme entwickeln können."

[Modulbeschreibungen\MH21; Position: 37 - 37]

45.

"[Fähigkeiten:] Verwendung logischer und funktionaler Programmiersprachen."

[Modulbeschreibungen\MH22; Position: 12 - 12]

46.

"[Studierende sollen nach Absolvierung der Lehrveranstaltungen in der Lage sein,] einfache objektorientierte Programme mit Polymorphie, Vererbung und Schnittstellen zu entwickeln."

[Modulbeschreibungen\MH23; Position: 26 - 26]

47.

"[Die Studierenden sollen] einfache Java-Programme erstellen können."

[Modulbeschreibungen\MH24; Position: 12 - 12]

48.

"[Die Studierenden sollen] die Prinzipien der objektorientierten Programmierung in Java umsetzen können."

[Modulbeschreibungen\MH24; Position: 15 - 15]

49.

"[Die Studierenden sollen] die grundlegenden Datenstrukturen und Algorithmen in Java umsetzen können."

[Modulbeschreibungen\MH24; Position: 23 - 23]

50.

"[Die Studierenden sollen] die objektorientierten Konzepte in Java umsetzen können."

[Modulbeschreibungen\MH24; Position: 30 - 30]

51.

"[Studierende sollten nach der Veranstaltung in der Lage sein,] [funktionale Konzepte in anderen Programmiersprachen und Frameworks, z. B. Java Streams] effektiv zu nutzen."

[Modulbeschreibungen\MH24; Position: 44 - 44]

52.

"[Die Studierenden] können [...] [grundlegende Datenstrukturen und Algorithmen] umsetzen."

[Modulbeschreibungen\MH25; Position: 18 - 18]

53.

"[Die Studierenden] können [...] [das funktionale Paradigma] praktisch umsetzen."

[Modulbeschreibungen\MH25; Position: 25 - 25]

54.

"[Die Studierenden] können Rekursion in verschiedenen Paradigmen anwenden."

[Modulbeschreibungen\MH25; Position: 26 - 26]

55.

"Die Studierenden sollen am Ende des Semesters in der Lage sein, einfache Algorithmen und Datenstrukturen in Java umzusetzen."

[Modulbeschreibungen\MH26; Position: 14 - 14]

56.

"[Nach erfolgreicher Absolvierung dieses Moduls können die Studierenden fortgeschrittene Konzepte objektorientierter Programmierung in Java] anwenden."

[Modulbeschreibungen\MH26; Position: 19 - 19]

57.

"[Die Studierenden] [sind in der Lage, objektorientierte Lösungen] umzusetzen."

[Modulbeschreibungen\MH26; Position: 21 - 21]

58.

"[In dieser Vorlesung lernen sie] kleinere, wohlstrukturierte Programme in C zu schreiben."

[Modulbeschreibungen\MH27; Position: 21 - 21]

59.

"[In dieser Vorlesung lernen sie] Mittelgroße objektorientierte Systeme in Java zu implementieren."

[Modulbeschreibungen\MH27; Position: 22 - 22]

60.

"[Somit ist es den Studierenden möglich, einfache praxisrelevante Problemstellungen] anschließend programmiertechnisch umzusetzen."

[Modulbeschreibungen\MH28; Position: 10 - 10]

61.

"Gegenstand des Programmierbeleges ist die Realisierung eines Programms für eine umfangreichere, selbstgewählte Aufgabenstellung."

[Modulbeschreibungen\MH28; Position: 25 - 25]

62.

"[Darüber hinaus wird jeder Teilnehmer befähigt, einen einfachen Programm-Entwurf] [...] selbständig programmtechnisch umzusetzen."

[Modulbeschreibungen\MH28; Position: 30 - 30]

63.

"[Die Studierenden] können [...] [Kenntnisse des funktionalen Programmierens] praxisnah einsetzen."

[Modulbeschreibungen\MH29; Position: 11 - 11]

64.

"[Einfache Codierungen können die Studierenden] selber erstellen."

[Modulbeschreibungen\MH30; Position: 9 - 9]

65.

"[Die Studierenden] können komplexe, rekursive Skripte in Unix schreiben."

[Modulbeschreibungen\MH30; Position: 20 - 20]

66.

"[Die Studierenden sind in der Lage, [...] die objektorientierte Programmierung in C++] [...] anzuwenden."

[Modulbeschreibungen\MH30; Position: 33 - 33]

67.

"Studierende sollen am Bsp. der Programmiersprache JAVA die folgenden Aspekte einer imperativen/prozeduralen Programmiersprache beherrschen: Sichere Anwendung des Konzepts primitiver Datentypen und Variablen."

[Modulbeschreibungen\MH32; Position: 7 - 7]

68.

"[Studierende sollen am Bsp. der Programmiersprache JAVA die folgenden Aspekte einer imperativen/prozeduralen Programmiersprache beherrschen:] Sichere Anwendung des Konzepts von Operatoren und Ausdrücken."

[Modulbeschreibungen\MH32; Position: 8 - 8]

69.

"[Studierende sollen am Bsp. der Programmiersprache JAVA die folgenden Aspekte einer imperativen/prozeduralen Programmiersprache beherrschen:] Sichere Anwendung von Kontrollstrukturen."

[Modulbeschreibungen\MH32; Position: 9 - 9]

70.

"[Studierende sollen am Bsp. der Programmiersprache JAVA die folgenden Aspekte einer imperativen/prozeduralen Programmiersprache beherrschen:] Fähigkeit zur problemorientierten Definition und Nutzung von Routinen/Methoden."

[Modulbeschreibungen\MH32; Position: 10 - 10]

71.

"[Studierende sollen am Bsp. der Programmiersprache JAVA die folgenden Aspekte einer imperativen/prozeduralen Programmiersprache beherrschen:] Fähigkeit zur Definition von problemspezifischen Referenztypen."

[Modulbeschreibungen\MH32; Position: 11 - 11]

72.

"[Studierende sollen am Bsp. der Programmiersprache JAVA die folgenden Aspekte einer imperativen/prozeduralen Programmiersprache beherrschen:] Sichere Anwendung des Prinzips rekursiver Programmierung."

[Modulbeschreibungen\MH32; Position: 16 - 16]

73.

"Studierende sollen am Bsp. der Programmiersprache JAVA die folgenden Aspekte einer objektorientierten Programmiersprache beherrschen: Fähigkeit das OO Konzept der Polymorphie zur Wiederverwendung OO Software einsetzen zu können."

[Modulbeschreibungen\MH32; Position: 38 - 38]

74.

"[Studierende sollen am Bsp. der Programmiersprache JAVA die folgenden Aspekte einer objektorientierten Programmiersprache beherrschen:] Fähigkeit vorgegebene generische Datentypen instanziierten und selber definieren zu können."

[Modulbeschreibungen\MH32; Position: 39 - 39]

75.

"Grundlegende Fähigkeit zur Anwendung von Techniken des sicheren Programmierens."

[Modulbeschreibungen\MH33; Position: 15 - 15]

76.

"[Die Studierenden können eine Problemstellung in ein] korrektes Java-Programm überführen."

[Modulbeschreibungen\MH34; Position: 33 - 33]

77.

"[Die Studierenden können] [Konzepte objektorientierter Programmierung] in der Programmierung mit Java anwenden."

[Modulbeschreibungen\MH34; Position: 36 - 36]

78.

"[Die Studierenden können] [Vor- und Nachteile von objektorientierten Konzepten] in der Praxis entsprechend sinnvoll einsetzen."

[Modulbeschreibungen\MH34; Position: 38 - 38]

79.

"[Methodenkompetenz:] [Die Studierenden] [sind in der Lage, Algorithmen hinsichtlich ihrer Eigenschaften und Anwendbarkeit für konkrete Problemstellungen] in eigenen kleineren Programmierprojekten in der Programmiersprache Java umzusetzen."

[Modulbeschreibungen\MH35; Position: 11 - 11]

80.

"[Die Studierenden] [können einfache Programme sowohl im funktionalen als auch im objektorientierten Programmierstil] implementieren (Methodenkompetenz)."

[Modulbeschreibungen\MH35; Position: 23 - 23]

5.6.2. Algorithmen mit Hilfe von Entwurfsmustern problemadäquat entwerfen

1.

"Erweiterung der Kompetenz im Entwurf von Algorithmen."

[Modulbeschreibungen\MH02; Position: 22 - 22]

2.

"Die Studierenden werden in die Lage versetzt, Lösungen für kleinere und überschaubare Probleme algorithmisch umzusetzen."

[Modulbeschreibungen\MH03; Position: 5 - 5]

3.

"Algorithmen entwickeln können."

[Modulbeschreibungen\MH04; Position: 37 - 37]

4.

"Fertigkeit zur Entwicklung und zum Umsetzen von Lösungsstrategien: Die Studierenden lernen, für praktische Problemstellungen algorithmische Lösungen zu entwickeln."

[Modulbeschreibungen\MH04; Position: 45 - 45]

5.

"[Die Studentinnen und Studenten] [...] können Entwurfsmuster [objektorientierter Programme] problemadäquat einsetzen."

[Modulbeschreibungen\MH05; Position: 14 - 14]

6.

"[Die Studentinnen und Studenten] [können am Ende des Moduls] elementare Algorithmen funktional entwerfen."

[Modulbeschreibungen\MH05; Position: 20 - 20]

7.

"Die Studierenden können einfache Probleme in Form von Algorithmen lösen."

[Modulbeschreibungen\MH06; Position: 5 - 5]

8.

"[Die Studierenden] [erwerben die Fähigkeit, Algorithmen] zu konstruieren."

[Modulbeschreibungen\MH07; Position: 7 - 7]

9.

"Einfache Algorithmen in einer halbformalen Notation erstellen können."

[Modulbeschreibungen\MH08; Position: 6 - 6]

10.

"Entwurf einfacher Algorithmen."

[Modulbeschreibungen\MH08; Position: 17 - 17]

11.

"Anhand kleinerer Problemstellungen lernen [die Studierenden], problemorientiert Algorithmen zu entwerfen."

[Modulbeschreibungen\MH09; Position: 6 - 6]

12.

"[Die Studentinnen] formulieren für kleinere Problemstellungen selbständig Algorithmen."

[Modulbeschreibungen\MH09; Position: 15 - 15]

13.

"[Die Studentinnen] stellen [...] [selbstständig formulierte Algorithmen für kleinere Problemstellungen] geeignet dar."

[Modulbeschreibungen\MH09; Position: 16 - 16]

14.

"[Die Studentinnen sollen die in der Veranstaltung vermittelten Begriffe und Konzepte zu Algorithmen und Datenstrukturen derart auf beispielhafte Problemstellungen anwenden können, dass sie] Algorithmen zur Bearbeitung entwickeln."

[Modulbeschreibungen\MH09; Position: 26 - 26]

15.

"Einfache Algorithmen entwickeln."

[Modulbeschreibungen\MH10; Position: 11 - 11]

16.

"[Die Studierenden] können eine abstrakte Problembeschreibung in einen programmierbaren Algorithmus übertragen."

[Modulbeschreibungen\MH11; Position: 7 - 7]

17.

"[Die Studierenden] können Algorithmen entwerfen."

[Modulbeschreibungen\MH11; Position: 30 - 30]

18.

"Eine Lösung zu erarbeiten [für einfache Problemstellungen] [ist] umsetzbar."

[Modulbeschreibungen\MH13; Position: 13 - 13]

19.

"Fach- und Methodenkompetenzen (Wissen und Verstehen sowie Anwendung und Erzeugung von Wissen): Entwurf von Algorithmen für gegebene Problemstellungen sind typische Aufgaben eines Informatikers."

[Modulbeschreibungen\MH15; Position: 23 - 23]

20.

"Nach Beendigung dieses Moduls können die Studierenden selbstständig Algorithmen entwerfen."

[Modulbeschreibungen\MH15; Position: 27 - 27]

21.

"Entwickeln einfacher Algorithmen."

[Modulbeschreibungen\MH18; Position: 7 - 7]

22.

"Geeignete Algorithmen für neue Problemstellungen erarbeiten."

[Modulbeschreibungen\MH18; Position: 17 - 17]

23.

"[Selbst- und Sozialkompetenz:] Unabhängigkeit algorithmischer Ideen vom Programmierparadigma."

[Modulbeschreibungen\MH19; Position: 11 - 11]

24.

"[Selbst- und Sozialkompetenz:] Informatiktypische Denk- und Herangehensweisen losgelöst von konkreten Sprachen und Paradigmen."

[Modulbeschreibungen\MH19; Position: 22 - 22]

25.

"[Algorithmische Kompetenz:] [eine konkrete Problemstellung] algorithmisch lösen können."

[Modulbeschreibungen\MH20; Position: 6 - 6]

26.

"[Algorithmische Kompetenz:] [eine konkrete Problemstellung] algorithmisch lösen können."

[Modulbeschreibungen\MH20; Position: 22 - 22]

27.

"[Analyse-, Design- und Realisierungs-Kompetenzen:] Algorithmen für neue Probleme entwickeln können."

[Modulbeschreibungen\MH20; Position: 40 - 40]

28.

"[Fachkompetenzen:] [Die Studierenden] erstellen Algorithmen mit allgemeinen Entwurfskonzepten (z.B. Greedy-Verfahren, Divide-and-Conquer-Verfahren)."

[Modulbeschreibungen\MH21; Position: 17 - 17]

29.

"[Studierende sollen nach Absolvierung der Lehrveranstaltungen in der Lage sein,] [einfache Algorithmen (iterativ und rekursiv)] selbst zu erstellen."

[Modulbeschreibungen\MH23; Position: 23 - 23]

30.

"[Die Studierenden sollen] [die] Wechselwirkungen [grundlegender Datenstrukturen und Algorithmen], insbesondere unter Laufzeit- und Speicherplatzbetrachtungen, auf praktische Beispiele anwenden können."

[Modulbeschreibungen\MH24; Position: 19 - 19]

31.

"[Die Studierenden sollen] Analyse- und Entwurfsmuster anwenden können."

[Modulbeschreibungen\MH24; Position: 31 - 31]

32.

"[Studierende sollten nach der Veranstaltung in der Lage sein,] rekursive Algorithmen zu entwerfen."

[Modulbeschreibungen\MH24; Position: 39 - 39]

33.

"[Die Studierenden sollen] Algorithmen für vorgegebene Aufgabestellungen durch Einsatz der Algorithmen-Entwurfsmuster entwickeln können."

[Modulbeschreibungen\MH24; Position: 48 - 48]

34.

"Ferner sollen die Studierenden die Fähigkeit trainieren, Algorithmen unter dem Aspekt von Performanzgarant zu entwickeln."

[Modulbeschreibungen\MH27; Position: 32 - 32]

35.

"[Die Studierenden] können [...] [Kenntnisse über Grundlagen der imperativen Programmierung (Syntaxdiagramme, EBNF, Funktionen, Module, Datenstrukturen)] zur Formulierung von Algorithmen für klassische Problemstellungen (Sortier- und Suchverfahren, Algorithmen auf Bäumen und Graphen) verwenden."

[Modulbeschreibungen\MH29; Position: 6 - 6]

36.

"Befähigung zur Lösung von algorithmischen Aufgaben."

[Modulbeschreibungen\MH31; Position: 6 - 6]

37.

"Befähigung zur Lösung von algorithmischen Aufgaben."

[Modulbeschreibungen\MH31; Position: 11 - 11]

38.

"[Studierende sollen am Bsp. der Programmiersprache JAVA die folgenden Aspekte einer objektorientierten Programmiersprache beherrschen:] Fähigkeit Software unter Berücksichtigung objektorientierter Entwurfsprinzipien entwickeln zu können."

[Modulbeschreibungen\MH32; Position: 42 - 42]

39.

"[Die Studierenden sind in der Lage, Algorithmen] zu entwerfen."

[Modulbeschreibungen\MH32; Position: 48 - 48]

40.

"[Fachkompetenz:] [Die Studierenden] können grundlegende Algorithmen nach einer Problembeschreibung systematisch durch Aufstellen eines Induktionsbeweises für die Lösbarkeit einer gegebenen Aufgabenstellung herleiten."

[Modulbeschreibungen\MH35; Position: 7 - 7]

41.

"[Systemkompetenz:] [Die Studierenden] können Algorithmen für einfache Problemstellungen selbstständig entwerfen."

[Modulbeschreibungen\MH35; Position: 14 - 14]

5.6.3. Algorithmen implementieren

1.

"Die Studierenden lernen objekt-orientierte Techniken kennen, um Software-Entwicklungs-Projekte eigenständig zu implementieren."

[Modulbeschreibungen\MH01; Position: 20 - 20]

2.

"[Die Teilnehmer] [sind in der Lage, kleine Programme (bis zu einigen hundert Zeilen)] zu implementieren."

[Modulbeschreibungen\MH02; Position: 9 - 9]

3.

"[Unter Anwendung der erlangten Fähigkeiten und Kenntnisse sollen die Teilnehmer in die Lage versetzt werden, Programme unter Verwendung der erlernten algorithmischen Techniken] in einer Programmiersprache zu implementieren."

[Modulbeschreibungen\MH03; Position: 23 - 23]

4.

"[Algorithmen] implementieren können."

[Modulbeschreibungen\MH04; Position: 38 - 38]

5.

"[Die Studentinnen und Studenten] können objektorientierte Programme implementieren."

[Modulbeschreibungen\MH05; Position: 13 - 13]

6.

"[Die Studierenden können] [...] [die Lösungen zu einfachen Problemen in Form von Algorithmen] als strukturierte Programme implementieren."

[Modulbeschreibungen\MH06; Position: 6 - 6]

7.

"[Die Studierenden können Probleme in Form von Klassendiagrammen] [...] objektorientiert implementieren."

[Modulbeschreibungen\MH06; Position: 19 - 19]

8.

"[Die Studierenden können anspruchsvolle Probleme in Form von Klassendiagrammen] [...] objektorientiert implementieren."

[Modulbeschreibungen\MH06; Position: 25 - 25]

9.

"Fähigkeit zur Implementierung von Programmen mit wahlweise textueller bzw. grafischer Benutzeroberfläche."

[Modulbeschreibungen\MH06; Position: 28 - 28]

10.

"[Die Studierenden] [erwerben die Fähigkeit, Algorithmen] zu implementieren."

[Modulbeschreibungen\MH07; Position: 8 - 8]

11.

"Implementierung einfacher Algorithmen"

[Modulbeschreibungen\MH08; Position: 19 - 19]

12.

"[Anhand kleinerer Problemstellungen lernen] [die Studierenden], [programmiersprachlich] zu implementieren."

[Modulbeschreibungen\MH09; Position: 8 - 8]

13.

"[Die Studierenden] [verfügen über grundlegende Fertigkeiten für die] Umsetzung [von Algorithmen vor dem Hintergrund konkreter Informationsverarbeitungsaufgaben]."

[Modulbeschreibungen\MH12; Position: 10 - 10]

14.

"[Die Studierenden] [sind in der Lage ein prozedurales Programm] zu implementieren."

[Modulbeschreibungen\MH13; Position: 31 - 31]

15.

"Implementierung."

[Modulbeschreibungen\MH14; Position: 19 - 19]

16.

"[Fach- und Methodenkompetenzen (Wissen und Verstehen sowie Anwendung und Erzeugung von Wissen):] Implementierung von Algorithmen und Datenstrukturen für gegebene Problemstellungen sind typische Aufgaben eines Informatikers."

[Modulbeschreibungen\MH15; Position: 25 - 25]

17.

"[Nach Beendigung dieses Moduls können die Studierenden selbstständig Algorithmen] implementieren."

[Modulbeschreibungen\MH15; Position: 29 - 29]

18.

"[Nach Beendigung dieses Moduls können die Studierenden] dynamische Datenstrukturen (objektorientiert) implementieren."

[Modulbeschreibungen\MH15; Position: 30 - 30]

19.

"[Zur Erarbeitung instrumentaler und systemischer Kompetenzen sollen die Studierenden] [die Modellierung, Verwaltung und Nutzung größerer Datenbestände] für kleinere Datenbanken implementieren können."

[Modulbeschreibungen\MH16; Position: 30 - 30]

20.

"Implementieren einfacher Algorithmen."

[Modulbeschreibungen\MH18; Position: 8 - 8]

21.

"[Fähigkeiten:] Realisierung von Datenstrukturen in verschiedenen Programmierparadigmen."

[Modulbeschreibungen\MH22; Position: 13 - 13]

22.

"Basierend auf den Kenntnissen und den Fähigkeiten sollten die Studierenden in der Lage sein, Programme unter Verwendung der erlernten algorithmischen Methoden und unter Berücksichtigung programmiertechnischer Konzepte wie z.B. der Kapselung von Datenstrukturen zu implementieren."

[Modulbeschreibungen\MH22; Position: 25 - 25]

23.

"[Nach erfolgreichem Abschluss des Moduls sollen die Studierenden in der Lage sein,] die Teilaufgaben in einem Team entsprechend eines Entwicklungsprozesses implementieren zu können."

[Modulbeschreibungen\MH23; Position: 53 - 53]

24.

"Basierend auf dem im Modul Informatik 1 gewonnenen theoretischen Wissen zu Datenstrukturen können die Studierenden verschiedene Datenstrukturen implementieren."

[Modulbeschreibungen\MH26; Position: 22 - 22]

25.

"[Die Studierenden] sind in der Lage [...] Kenntnisse [zu Unterschieden sowie Vor- und Nachteilen zu objektorientierten und typsicheren Sprachen] in der Implementierung von C-Programmen anzuwenden."

[Modulbeschreibungen\MH26; Position: 28 - 28]

26.

"Einfache Programmiersprachen implementieren können."

[Modulbeschreibungen\MH27; Position: 12 - 12]

27.

"[Die Teilnehmer dieses Moduls sind in der Lage, aus einer breiten Auswahl an Standardlösungen (Fachkompetenz) die am besten geeignete] selbständig zu implementieren (Methodenkompetenz)."

[Modulbeschreibungen\MH28; Position: 39 - 39]

28.

"[Die Studierenden sind in der Lage, Algorithmen und Datenstrukturen] zu implementieren."

[Modulbeschreibungen\MH32; Position: 50 - 50]

29.

"[Fähigkeit, einfache Programme selbständig] zu implementieren."

[Modulbeschreibungen\MH33; Position: 10 - 10]

30.

"[Beherrschen grundlegender Prinzipien und Methoden für] Implementierung von Algorithmen."

[Modulbeschreibungen\MH33; Position: 22 - 22]

31.

"[Die Studierenden] implementieren arbeitsteilig eine Lösung."

[Modulbeschreibungen\MH34; Position: 14 - 14]

32.

"[Die Studierenden können] Aggregations- und Kompositionsbeziehungen zwischen Klassen in geeigneter Weise implementieren."

[Modulbeschreibungen\MH34; Position: 23 - 23]

33.

"[Die Studierenden können] ein Projekt Sourcecode-verwaltet im Team implementieren."

[Modulbeschreibungen\MH34; Position: 28 - 28]

5.6.4. Modellieren von Problemen und Programmen

1.

"[Die Studierenden] nutzen weitergehende Analyse-, Design- und Realisierungskompetenzen, um das Zusammenspiel und die architektonische Organisation vieler Klassen in einem Projekt anhand der Programmiersprache Java sowie der UML zu formulieren."

[Modulbeschreibungen\MH01; Position: 21 - 21]

2.

"[Teilnehmer(innen) des Moduls erlernen die Fähigkeit zur] formalen Modellierung von Algorithmen Problemstellungen."

[Modulbeschreibungen\MH03; Position: 19 - 19]

3.

"[Die Studierenden] sind in der Lage, strukturelle und dynamische Eigenschaften komplexer Softwaresysteme zu modellieren."

[Modulbeschreibungen\MH03; Position: 35 - 35]

4.

"[Die Studentinnen und Studenten] können objektorientierte Modelle erstellen."

[Modulbeschreibungen\MH05; Position: 7 - 7]

5.

"Die Studierenden können Probleme in Form von Klassendiagrammen formulieren."

[Modulbeschreibungen\MH06; Position: 18 - 18]

6.

"Die Studierenden können anspruchsvolle Probleme in Form von Klassendiagrammen formulieren."

[Modulbeschreibungen\MH06; Position: 24 - 24]

7.

"Anschauliche Sachverhalte im Modell der Objektorientierung ausdrücken können."

[Modulbeschreibungen\MH10; Position: 10 - 10]

8.

"[Die Studierenden] können objektorientierte Problemlösungen modellieren."

[Modulbeschreibungen\MH11; Position: 8 - 8]

9.

"Die Studierenden können in praxisorientierten Problemen erfolgreiche graphentheoretischen Modellierungsparadigmen und Formalismen anwenden."

[Modulbeschreibungen\MH11; Position: 22 - 22]

10.

"[Die Studierenden] können eigenständig praxisorientierten Probleme mit graphentheoretischen Methoden Modellieren."

[Modulbeschreibungen\MH11; Position: 24 - 24]

11.

"[Die Studierenden] besitzen die Fähigkeit zum eigenständigen Modellieren."

[Modulbeschreibungen\MH11; Position: 26 - 26]

12.

"Design mit Hilfe von Szenarien, Objekt- und Klassendiagrammen."

[Modulbeschreibungen\MH14; Position: 18 - 18]

13.

"Analyse-Kompetenz: ein informell dargestelltes Problem mithilfe einer Modellierungssprache (UML) (semi-)formal beschreiben können."

[Modulbeschreibungen\MH20; Position: 23 - 23]

14.

"Gegebene Problemstellungen werden [...] [in der Softwareentwicklung] mit Hilfe objektorientierter Analyse- und Entwurfsverfahren in ein objektorientiertes Modell überführt."

[Modulbeschreibungen\MH21; Position: 33 - 33]

15.

"[Fachkompetenzen:] [Die Studierenden] entwickeln selbstständig Modelle für die Lösung mittelgroßer Probleme."

[Modulbeschreibungen\MH21; Position: 43 - 43]

16.

"[Die Studierenden sollten in der Lage sein,] algorithmischen Problemen formal zu modellieren."

[Modulbeschreibungen\MH22; Position: 22 - 22]

17.

"[Studierende sollen nach erfolgreichem Abschluss des Moduls] aus einfachen Problemstellungen ein objektorientiertes System zur Lösung entwerfen können."

[Modulbeschreibungen\MH23; Position: 45 - 45]

18.

"[Die Studierenden sollen] [die Grundkonzepte der objektorientierten Programmierung] mit der Standardnotation UML beschreiben können."

[Modulbeschreibungen\MH24; Position: 14 - 14]

19.

"[Die Studierenden sollen] die objektorientierten Konzepte mit der Standardnotation UML beschreiben können."

[Modulbeschreibungen\MH24; Position: 27 - 27]

20.

"[Die Studierenden] sind in der Lage, objektorientierte Lösungen zu entwerfen."

[Modulbeschreibungen\MH26; Position: 20 - 20]

21.

"Des Weiteren sollen [die Studierenden] lernen wie objektorientierte Modellierung von komplexen Anwendungssystemen durchzuführen [ist], insbesondere in der Kommunikation mit den Anwendern."

[Modulbeschreibungen\MH26; Position: 34 - 34]

22.

"[Studierende sollen am Bsp. der Programmiersprache JAVA die folgenden Aspekte einer objektorientierten Programmiersprache beherrschen:] Darstellung objektorientierter Sachverhalte mittels UML Klassendiagrammen."

[Modulbeschreibungen\MH32; Position: 24 - 24]

23.

"Die Studierenden entwerfen einfache Algorithmen in Form von Struktogrammen zu vorgegebenen Aufgaben."

[Modulbeschreibungen\MH34; Position: 11 - 11]

24.

"[Die Studierenden können] den Lebenszyklus von Objekten konsistent gestalten (Konstruktoren, Destruktor, Kopie, Wertzuweisung, dynamische Inhalte)."

[Modulbeschreibungen\MH34; Position: 24 - 24]

25.

"[Die Studierenden können] aus einer verbalen Aufgabenstellung ein sinnvolles System von Objektklassen ableiten."

[Modulbeschreibungen\MH34; Position: 25 - 25]

26.

"Die Studierenden können eine Problemstellung in ein objektorientiertes Modell überführen."

[Modulbeschreibungen\MH34; Position: 32 - 32]

5.6.5. kleinere Programme strukturiert entwerfen

1.

"[Die Teilnehmer] [sind in der Lage, kleine Programme (bis zu einigen hundert Zeilen)] selbst zu konzipieren."

[Modulbeschreibungen\MH02; Position: 8 - 8]

2.

"Unter Anwendung der erlangten Fähigkeiten und Kenntnisse sollen die Teilnehmer in die Lage versetzt werden, Programme unter Verwendung der erlernten algorithmischen Techniken eigenständig zu konzipieren."

[Modulbeschreibungen\MH03; Position: 22 - 22]

3.

"Die Studierenden sind in der Lage eigenständig eine Lösungsstrategie zum Schreiben kleiner prozeduraler und objektorientierter Java-Programme nach einer vorgegebenen Entwurfsidee zu entwickeln."

[Modulbeschreibungen\MH04; Position: 12 - 12]

4.

"[Die Studentinnen und Studenten] [können am Ende des Moduls] strukturierte Programme entwickeln."

[Modulbeschreibungen\MH05; Position: 25 - 25]

5.

"Schwerpunkt der praktischen Arbeit ist der Entwurf von Applikationen unter Nutzung von Standard-APIs."

[Modulbeschreibungen\MH09; Position: 20 - 20]

6.

"[Die Studierenden] sind in der Lage, ein prozedurales Programm zu entwerfen."

[Modulbeschreibungen\MH13; Position: 30 - 30]

7.

"[Fachkompetenzen:] [Die Studierenden] entwickeln selbstständig Programme für die Lösung kleinerer Probleme."

[Modulbeschreibungen\MH21; Position: 14 - 14]

8.

"[Fachkompetenzen:] [Die Studierenden] entwickeln selbstständig Programme für die Lösung mittelgroßer Probleme."

[Modulbeschreibungen\MH21; Position: 44 - 44]

9.

"Methodenkompetenzen: Die Studierenden entwickeln selbstständig Programme für gegebene Probleme durch konsequente Anwendung der Konzepte der objektorientierten Modellierung und Programmierung."

[Modulbeschreibungen\MH21; Position: 48 - 48]

10.

"[Selbstkompetenzen:] [Die Studierenden] beziehen die Konzepte des objektorientierten Programmentwurfs in ihr Handeln ein."

[Modulbeschreibungen\MH21; Position: 53 - 53]

11.

"Kompetenzen: Selbstständige Entwicklung kleinerer Programme und ihrer Dokumentation unter Beachtung üblicher Programmierkonventionen."

[Modulbeschreibungen\MH22; Position: 16 - 16]

12.

"[Die Studierenden sollen die objektorientierten Konzepte in der Phase] Entwurf anwenden können."

[Modulbeschreibungen\MH24; Position: 26 - 26]

13.

"Gegenstand des Programmierbeleges ist der Entwurf eines Programms für eine umfangreichere, selbstgewählte Aufgabenstellung."

[Modulbeschreibungen\MH28; Position: 24 - 24]

14.

"Darüber hinaus wird jeder Teilnehmer befähigt, einen einfachen Programm-Entwurf zu konstruieren."

[Modulbeschreibungen\MH28; Position: 29 - 29]

15.

"[Darüber hinaus] können [...] [die Studierenden] [grundlegende Problemlösungs-und Programmkonstruktionsmethoden der imperativen Programmierung] anwenden."

[Modulbeschreibungen\MH30; Position: 14 - 14]

16.

"Fähigkeit, einfache Programme selbständig zu entwerfen"

[Modulbeschreibungen\MH33; Position: 9 - 9]

17.

"[Die Studierenden] können einfache Programme sowohl im funktionalen als auch im objektorientierten Programmierstil systematisch entwerfen (Methodenkompetenz)."

[Modulbeschreibungen\MH35; Position: 22 - 22]

5.6.6. Standard-Algorithmen anwenden

1.

"Darüber hinaus lernen [die Studierenden] existierende Algorithmen anzuwenden."

[Modulbeschreibungen\MH01; Position: 18 - 18]

2.

"[Such- und Sortierverfahren] einsetzen."

[Modulbeschreibungen\MH04; Position: 31 - 31]

3.

"[Fertigkeit zur Entwicklung und zum Umsetzen von Lösungsstrategien: Die Studierenden lernen, für praktische Problemstellungen] vorhandene Algorithmen einzusetzen."

[Modulbeschreibungen\MH04; Position: 46 - 46]

4.

"[Die Studierenden] erwerben die Fähigkeit, Algorithmen anzuwenden."

[Modulbeschreibungen\MH07; Position: 6 - 6]

5.

"Suchen kürzester Wege und minimaler Spannbäume auf Graphen."

[Modulbeschreibungen\MH08; Position: 22 - 22]

6.

"Suchen kürzester Abstände und Bilden konvexer Hüllen auf Punktmengen."

[Modulbeschreibungen\MH08; Position: 23 - 23]

7.

"[Wesentliche Algorithmen der Informatik] anwenden können."

[Modulbeschreibungen\MH10; Position: 32 - 32]

8.

"[Die Studierenden] [können eigenständig praxisorientierten Probleme mit graphentheoretischen Methoden] Lösen."

[Modulbeschreibungen\MH11; Position: 25 - 25]

9.

"[Die Studierenden] können Algorithmen anwenden."

[Modulbeschreibungen\MH11; Position: 34 - 34]

10.

"[Nach Beendigung dieses Moduls können die Studierenden] graphentheoretische Konzepte für praktische Problemstellungen anwenden."

[Modulbeschreibungen\MH15; Position: 33 - 33]

11.

"[Außerdem] können [die Studierenden] [...] [bedeutende algorithmische Strategien] anwenden."

[Modulbeschreibungen\MH17; Position: 18 - 18]

12.

"[Wichtige Algorithmen] anwenden."

[Modulbeschreibungen\MH18; Position: 12 - 12]

13.

"[Algorithmische Kompetenz:] grundlegende Algorithmen zur Lösung von Problemen einsetzen."

[Modulbeschreibungen\MH20; Position: 7 - 7]

14.

"[Studierende sollten nach der Veranstaltung in der Lage sein,] [Funktionen höherer Ordnung] anzuwenden."

[Modulbeschreibungen\MH24; Position: 42 - 42]

15.

"[Die Teilnehmer] sind in der Lage [...] [grundlegende und wichtige Algorithmen] anzuwenden."

[Modulbeschreibungen\MH30; Position: 30 - 30]

16.

"[Die Studierenden können] [die gegebenen Algorithmen der Java Collections API] in eigenen Programmen verwenden."

[Modulbeschreibungen\MH34; Position: 43 - 43]

5.6.7. Standard-Datenstrukturen anwenden

1.

"Wichtige Datenstrukturen (z.B. Sequenz, Baum, File) in Programmiersprachen (z.B. Python) definieren können."

[Modulbeschreibungen\MH08; Position: 11 - 11]

2.

"[Die Studierenden] können Datenstrukturen anwenden."

[Modulbeschreibungen\MH11; Position: 35 - 35]

3.

"[Wichtige Datenstrukturen] anwenden [(z.B. Arrays, Stapel, Bäume)]."

[Modulbeschreibungen\MH18; Position: 14 - 14]

4.

"[Algorithmische Kompetenz:] grundlegende Datenstrukturen zur Lösung von Problemen einsetzen."

[Modulbeschreibungen\MH20; Position: 8 - 8]

5.

"[Analyse-, Design- und Realisierungs-Kompetenzen:] [wichtige lineare und nicht-lineare Datenstrukturen] sinnvoll für ein gegebenes Problem einsetzen können."

[Modulbeschreibungen\MH20; Position: 35 - 35]

6.

"[Analyse-, Design- und Realisierungs-Kompetenzen:] [Datenstrukturen aus Standardbibliotheken] effizient einsetzen können."

[Modulbeschreibungen\MH20; Position: 42 - 42]

7.

"[Studierende sollen nach Absolvierung der Lehrveranstaltungen in der Lage sein,] die behandelten dynamischen Datenstrukturen anzuwenden."

[Modulbeschreibungen\MH23; Position: 27 - 27]

8.

"[Basierend auf dem im Modul Informatik 1 gewonnenen theoretischen Wissen zu Datenstrukturen können die Studierenden verschiedene Datenstrukturen] anwenden."

[Modulbeschreibungen\MH26; Position: 23 - 23]

9.

"[Die Studierenden] sind in der Lage [die wichtigsten dynamischen Datenstrukturen] zu implementieren."

[Modulbeschreibungen\MH30; Position: 25 - 25]

10.

"[Die Studierenden] sind in der Lage [die wichtigsten dynamischen Datenstrukturen] anzuwenden."

[Modulbeschreibungen\MH30; Position: 26 - 26]

11.

"[Studierende sollen am Bsp. der Programmiersprache JAVA die folgenden Aspekte einer imperativen/prozeduralen Programmiersprache beherrschen:] Sichere Anwendung der Datenstrukturen Liste, Stack, Map und Set."

[Modulbeschreibungen\MH32; Position: 14 - 14]

12.

"[Studierende sollen am Bsp. der Programmiersprache JAVA die folgenden Aspekte einer imperativen/prozeduralen Programmiersprache beherrschen:] Sichere Anwendung der rekursiven Datenstruktur Baum."

[Modulbeschreibungen\MH32; Position: 18 - 18]

13.

"[Die Studierenden verfügen über Kompetenzen] zur Anwendung geeigneter Datenstrukturen und Algorithmen in der Softwareentwicklung zur Verbesserung der Effizienz und der Wartbarkeit der Programme sowie zur Verkürzung der Entwicklungszeit."

[Modulbeschreibungen\MH32; Position: 54 - 54]

14.

"[Die Studierenden können] [die gegebenen Datenstrukturen der Java Collections API] in eigenen Programmen verwenden."

[Modulbeschreibungen\MH34; Position: 42 - 42]

5.6.8. Datenstrukturen entwerfen

1.

"Erweiterung der Kompetenz im Entwurf von [...] zu Algorithmen zugehörigen Datenstrukturen."

[Modulbeschreibungen\MH02; Position: 23 - 23]

2.

"[Die Studentinnen sollen die in der Veranstaltung vermittelten Begriffe und Konzepte zu Algorithmen und Datenstrukturen derart auf beispielhafte Problemstellungen anwenden können, dass sie] eigene Datenstrukturen passend definieren können."

[Modulbeschreibungen\MH09; Position: 25 - 25]

3.

"[Die Studierenden] können Datenstrukturen entwerfen."

[Modulbeschreibungen\MH11; Position: 31 - 31]

4.

"[Fach- und Methodenkompetenzen (Wissen und Verstehen sowie Anwendung und Erzeugung von Wissen):] [Entwurf von] Datenstrukturen für gegebene Problemstellungen sind typische Aufgaben eines Informatikers."

[Modulbeschreibungen\MH15; Position: 24 - 24]

5.

"Können: Die Studierenden lernen, Datenstrukturen für neue Problemstellungen eigenständig zu entwerfen (instrumentale Kompetenz)."

[Modulbeschreibungen\MH16; Position: 8 - 8]

6.

"[Methodenkompetenz:] Entwurf effizienter Datenstrukturen für einfache Probleme."

[Modulbeschreibungen\MH19; Position: 8 - 8]

7.

"[Analyse-, Design- und Realisierungs-Kompetenzen:] effiziente Datenstrukturen für neue Probleme entwickeln können."

[Modulbeschreibungen\MH20; Position: 39 - 39]

8.

"[Studierende sollten nach der Veranstaltung in der Lage sein,] rekursive Datenstrukturen zu entwerfen."

[Modulbeschreibungen\MH24; Position: 40 - 40]

9.

"Ferner sollen die Studierenden die Fähigkeit trainieren, Datenstrukturen unter dem Aspekt von Performanzgarant ein zu entwickeln."

[Modulbeschreibungen\MH27; Position: 33 - 33]

10.

"Befähigung zum Design von Datenstrukturen."

[Modulbeschreibungen\MH31; Position: 7 - 7]

11.

"Befähigung zum Design von Datenstrukturen."

[Modulbeschreibungen\MH31; Position: 12 - 12]

12.

"[Die Studierenden sind in der Lage, Datenstrukturen] zu entwerfen."

[Modulbeschreibungen\MH32; Position: 49 - 49]

5.6.9. Datenstrukturen anpassen

1.

"[Teilnehmer(innen) des Moduls erlernen die Fähigkeit zur] Adaption bekannter Datenstrukturen an modifizierte Problemstellungen."

[Modulbeschreibungen\MH03; Position: 20 - 20]

2.

"[Typische Datenstrukturen] problemadäquat einsetzen können."

[Modulbeschreibungen\MH10; Position: 29 - 29]

3.

"[Können:] [...] [Durch die Analyse von Leistungsparametern für eigenständig entworfene Datenstrukturen zur Lösung neuer Problemstellungen] sollen [die Studierenden] im Beruf z.B. in der Lage sein, bestehende Software durch geeignetere Datenstrukturen zu beschleunigen (systemische Kompetenz)."

[Modulbeschreibungen\MH16; Position: 10 - 10]

4.

"[Methodenkompetenz] Anpassung von Datenstrukturen an spezielle Erfordernisse."

[Modulbeschreibungen\MH19; Position: 19 - 19]

5.

"[Fachkompetenzen:] [Die Studierenden] wenden die gelernten Datenstrukturen sinnvoll auf gegebene und konkrete Probleme an."

[Modulbeschreibungen\MH21; Position: 24 - 24]

6.

"[Die Studierenden sollten in der Lage sein,] vorhandene Datenstrukturen an eine gegebene Problemstellung anzupassen."

[Modulbeschreibungen\MH22; Position: 24 - 24]

7.

"Ferner sollen die Studierenden die Fähigkeit trainieren, Datenstrukturen unter dem Aspekt von Performanzgarant anzupassen."

[Modulbeschreibungen\MH27; Position: 35 - 35]

8.

"[Die Studierenden] passen [...] [aggregierender Datenstrukturen] dem konkreten Einsatzfall an."

[Modulbeschreibungen\MH34; Position: 17 - 17]

5.6.10. Algorithmen anpassen

1.

"[Teilnehmer(innen) des Moduls erlernen die Fähigkeit zur] Adaption bekannter Algorithmen an modifizierte Problemstellungen."

[Modulbeschreibungen\MH03; Position: 21 - 21]

2.

"[Wesentliche Algorithmen der Informatik] modifizieren können."

[Modulbeschreibungen\MH10; Position: 33 - 33]

3.

"[Die Studierenden] [verfügen über grundlegende Fertigkeiten für die] Modifikation [von Algorithmen vor dem Hintergrund konkreter Informationsverarbeitungsaufgaben]."

[Modulbeschreibungen\MH12; Position: 11 - 11]

4.

"[Methodenkompetenz:] Anpassung von Algorithmen an spezielle Erfordernisse."

[Modulbeschreibungen\MH19; Position: 18 - 18]

5.

"[Fachkompetenzen:] [Die Studierenden] wenden die gelernten Algorithmen sinnvoll auf gegebene und konkrete Probleme an."

[Modulbeschreibungen\MH21; Position: 23 - 23]

6.

"[Die Studierenden sollten in der Lage sein,] vorhandene Algorithmen an eine gegebene Problemstellung anzupassen."

[Modulbeschreibungen\MH22; Position: 23 - 23]

7.

"Ferner sollen die Studierenden die Fähigkeit trainieren, Algorithmen unter dem Aspekt von Performanzgarant anzupassen."

[Modulbeschreibungen\MH27; Position: 34 - 34]

5.6.11. (Abstrakte) Datentypen konzipieren & einsetzen

1.

"[Die Studentinnen und Studenten] sind in der Lage, abstrakte Datentypen zu spezifizieren."

[Modulbeschreibungen\MH05; Position: 9 - 9]

2.

"[Die Studentinnen und Studenten] [sind in der Lage, abstrakte Datentypen] zu implementieren."

[Modulbeschreibungen\MH05; Position: 10 - 10]

3.

"[Studentinnen und Studenten] können [...] [die wichtigsten abstrakten Datentypen und die Datenstrukturen zu deren Implementierung] [in Bezug auf ihre Eigenschaften] einsetzen."

[Modulbeschreibungen\MH05; Position: 37 - 37]

4.

"[Studierende sollen nach Absolvierung der Lehrveranstaltungen in der Lage sein,] eigene Datentypen zu konzipieren."

[Modulbeschreibungen\MH23; Position: 24 - 24]

5.

"[Studierende sollen nach Absolvierung der Lehrveranstaltungen in der Lage sein,] eigene Datentypen anzuwenden."

[Modulbeschreibungen\MH23; Position: 25 - 25]

5.6.12. Gegebene Programm(-komponenten) anpassen & verändern

1.

"[Die Studentinnen und Studenten] [können am Ende des Moduls] [Beschreibungen von Programmkomponenten] [...] in eigenen Programmen benutzen."

[Modulbeschreibungen\MH05; Position: 23 - 23]

2.

"[Die Studentinnen und Studenten] [können am Ende des Moduls] einfache Programme an geänderte Anforderungen anpassen."

[Modulbeschreibungen\MH05; Position: 24 - 24]

3.

"[Die Studierenden] [sind in der Lage fremde Programme] zu modifizieren."

[Modulbeschreibungen\MH07; Position: 28 - 28]

4.

"[Die Studierenden] [können fremden Quellcode] in eigene Programme integrieren."

[Modulbeschreibungen\MH11; Position: 13 - 13]

5.

"[Studierende sollen nach erfolgreichem Abschluss des Moduls] [eine Lösung hinsichtlich der Wartbarkeit] verbessern können."

[Modulbeschreibungen\MH23; Position: 47 - 47]

5.6.13. Formale Syntaxbeschreibungen entwerfen

1.

"[Die Teilnehmer] können mit den Techniken und Notationen zur Definition kontextfreier Programmiersprachen [...] arbeiten."

[Modulbeschreibungen\MH02; Position: 15 - 15]

2.

"[Formale Syntaxbeschreibungen] für einfache Sprachen entwickeln können."

[Modulbeschreibungen\MH10; Position: 18 - 18]

3.

"[Die Studierenden] [können syntaktische und semantische Beschreibungen von Programmiersprachen] anpassen."

[Modulbeschreibungen\MH33; Position: 28 - 28]

4.

"[Die Studierenden] [können syntaktische und semantische Beschreibungen von Programmiersprachen] erweitern."

[Modulbeschreibungen\MH33; Position: 29 - 29]

5.6.14. Programm-Spezifikation entwerfen

1.

"[Die Studentinnen und Studenten] [können am Ende des Moduls] Anforderungen an funktionale Programme spezifizieren."

[Modulbeschreibungen\MH05; Position: 21 - 21]

2.

"Funktionale Spezifikationen zu einfachen Problemen angeben können."

[Modulbeschreibungen\MH08; Position: 9 - 9]

3.

"[Studierende, die das Modul erfolgreich absolviert haben,] haben die Fähigkeit zur Problemspezifikation erworben."

[Modulbeschreibungen\MH23; Position: 40 - 40]

5.6.15. größere Anwendungen/Applikationen entwickeln

1.

"[Schwerpunkt der praktischen Arbeit] ist die selbständige Realisierung [von Applikationen unter Nutzung von Standard- APIs]."

[Modulbeschreibungen\MH09; Position: 21 - 21]

2.

"[Die Studierenden] [sind in der Lage, größere Anwendungen] zu erstellen."

[Modulbeschreibungen\MH13; Position: 22 - 22]

5.6.16. Formale Werkzeuge & Testfälle entwickeln

1.

"[Die Studierenden besitzen die Fähigkeiten, formale Werkzeuge (Grundlagen der Berechnung, Übersetzung von Programmkonstruktoren, Programmtransformationen, Verifikation von Programmeigenschaften)] zu entwickeln."

[Modulbeschreibungen\MH29; Position: 13 - 13]

2.

"[Studierende sollen am Bsp. der Programmiersprache JAVA die folgenden Aspekte einer objektorientierten Programmiersprache beherrschen:] Fähigkeit Testfälle entwickeln zu können, die eine vorgegebene Code Coverage erreichen."

[Modulbeschreibungen\MH32; Position: 41 - 41]

5.6.17. GUIs programmieren

1.

"[Die Studierenden] können [...] [grafische Benutzerschnittstellen] erstellen."

[Modulbeschreibungen\MH13; Position: 20 - 20]

2.

"Programmierung grafischer Oberflächen am Beispiel von MS Windows anhand objektorientierter Ansätze."

[Modulbeschreibungen\MH18; Position: 9 - 9]

5.6.18. nebenläufige und parallele Prozesse entwerfen

1.

"[Die Studierenden] [besitzen die Fähigkeit zu] einem Redesign von nebenläufigen Prozessen mittels Petri-Netzen."

[Modulbeschreibungen\MH11; Position: 28 - 28]

2.

"[Die Studierenden] können einfache Anwendungen mittels systemnaher und/oder paralleler Programmierung auf ausgewählten Plattformen entwickeln."

[Modulbeschreibungen\MH17; Position: 21 - 21]

5.6.19. Bibliotheken entwickeln

1.

"[Fach- und Methodenkompetenzen (Wissen und Verstehen sowie Anwendung und Erzeugung von Wissen):] Die Entwicklung von eigenen Bibliotheken und komplexen Anwendungen in einzelnen Komponenten kann gezeigt werden."

[Modulbeschreibungen\MH15; Position: 6 - 6]

5.6.20. Methoden aufrufen

1.

"[Die Studierenden] können Methoden aufrufen."

[Modulbeschreibungen\MH13; Position: 9 - 9]

5.6.21. Objekte erzeugen

1.

"[Die Studierenden] können einfache Funktionalitäten in Klassen kapseln, Objekte erzeugen."

[Modulbeschreibungen\MH13; Position: 8 - 8]

5.6.22. Schnittstellen entwerfen

1.

"[Die Studierenden können] [aus einer verbalen Aufgabenstellung] passende Schnittstellen entwerfen."

[Modulbeschreibungen\MH34; Position: 26 - 26]

5.6.23. Transfer von Wissen und Erfahrung auf neue Aufgaben und Programmiersprachen

1.

"Des weiteren entwickeln die Studierenden ein Verständnis für die allgemeinen Prinzipien der Programmierung und der Programmiersprachen, das den Grundstein dafür legt, dass die Studierenden sich (nach weiteren Erfahrungen im Laufe des Studiums) in beliebige Programmiersprachen schnell und präzise einarbeiten können."

[Modulbeschreibungen\MH03; Position: 8 - 8]

2.

"Kompetenz zur Vernetzung unterschiedlicher Fachgebiete: Im Rahmen von Übungen setzen die Studierenden die erarbeiteten Grundlagen in verschiedenen Anwendungsbereichen ein."

[Modulbeschreibungen\MH04; Position: 47 - 47]

3.

"Die Studierenden sind in der Lage ihr Wissen und Verstehen auf ihre Tätigkeit oder ihren Beruf anzuwenden."

[Modulbeschreibungen\MH08; Position: 31 - 31]

4.

"Die Studierenden erlernen den Umstieg auf eine neue Programmiersprache, die insbesondere auch für größere Programmierprojekte geeignet ist."

[Modulbeschreibungen\MH16; Position: 33 - 33]

5.

"Methodenkompetenz: Fähigkeit zur schnellen Einarbeitung in eine beliebige Programmiersprache."

[Modulbeschreibungen\MH19; Position: 15 - 15]

6.

"[Methodenkompetenz:] Fähigkeit zur schnellen Einarbeitung in neue Anwendungen."

[Modulbeschreibungen\MH19; Position: 28 - 28]

7.

"[Methodenkompetenzen:] [Die Studierenden] übertragen praktische Erfahrungen in der Programmentwicklung auf neue Aufgaben."

[Modulbeschreibungen\MH21; Position: 26 - 26]

8.

"[Methodenkompetenzen:] [Die Studierenden] übertragen praktische Erfahrungen in der Programmentwicklung auf neue Aufgaben."

[Modulbeschreibungen\MH21; Position: 49 - 49]

9.

"[In dieser Vorlesung lernen sie] sich in wenigen Tagen eine neue imperative/objektorientierte Sprache anzueignen, um sich in ein bestehendes Projekt einzuarbeiten."

[Modulbeschreibungen\MH27; Position: 24 - 24]

10.

"Gemeinsam können die Studierenden, begleitet durch die Diskussion mit Mitstudierenden im Rahmen des Seminars, Lösungen für neue unbekannte Problemstellungen aus dem Bereich der Programmierung erarbeiten."

[Modulbeschreibungen\MH28; Position: 11 - 11]

11.

"Studierende sollen in der Lage sein, die am Beispiel der Sprache JAVA erlernten Prinzipien auf andere Programmiersprachen imperativen/prozeduralen und objektorientierten Charakters übertragen zu können."

[Modulbeschreibungen\MH32; Position: 25 - 25]

12.

"Studierende sollen am Bsp. der Programmiersprache C begreifen, erlernte generelle Konzepte imperativer/prozeduraler Programmiersprachen auf andere Sprachen zu übertragen (hier: Transfer von JAVA auf C)."

[Modulbeschreibungen\MH32; Position: 43 - 43]

13.

"[Systemkompetenz:] [Die Studierenden] können [...] [die Wirkungsweise von Standardalgorithmen und –datenstrukturen] in neuen Zusammenhängen einsetzen."

[Modulbeschreibungen\MH35; Position: 13 - 13]

5.6.24. Selbstständige Organisation des Lernprozesses (unter Anwendung externer Ressourcen)

1.

"[Die Studierenden] sind in der Lage ihr Wissen vertikal, horizontal und lateral zu vertiefen."

[Modulbeschreibungen\MH08; Position: 29 - 29]

2.

"Die Studierenden erschließen sich vertiefende Informationen und Konzepte aus der zugrunde liegenden Literatur bzw. Dokumentation."

[Modulbeschreibungen\MH13; Position: 23 - 23]

3.

"[Die Studierenden] sollen die Fähigkeit entwickeln, selbstständig in Programmierhandbüchern und -beschreibungen Details der Programmiersprache herauszufinden und nutzen zu können (instrumentale Kompetenz)."

[Modulbeschreibungen\MH16; Position: 22 - 22]

4.

"Allgemeine Kompetenz: Selbstständiges Erarbeiten (Lesen und Verstehen) von Informatik-Themen mit Hilfe wissenschaftlicher Literatur."

[Modulbeschreibungen\MH20; Position: 19 - 19]

5.

"Selbstkompetenzen: Die Studierenden organisieren sich beim Finden von algorithmischen Lösungen für kleine und mittelgroße Probleme der Informatik."

[Modulbeschreibungen\MH21; Position: 29 - 29]

6.

"Selbstkompetenzen: Die Studierenden organisieren sich beim Entwickeln von Programmen für kleine und mittelgroße Probleme der Informatik."

[Modulbeschreibungen\MH21; Position: 52 - 52]

7.

"[Die Studierenden] können neue Sprachkonstrukte selbstständig erlernen."

[Modulbeschreibungen\MH33; Position: 31 - 31]

5.6.25. Abstraktion von Problemen

1.

"Studierende sollten nach der Veranstaltung in der Lage sein, Probleme funktional zu abstrahieren."

[Modulbeschreibungen\MH24; Position: 36 - 36]

2.

"Neben dem Erlernen von Rechenverfahren zur Lösung von gewissen mathematischen Problemen ist es ein Hauptziel diese Lehrveranstaltung, das für Informatiker und Informatikerinnen so wichtige Abstraktionsvermögen zu schulen."

[Modulbeschreibungen\MH32; Position: 29 - 29]