



MODULARES INNOVATIVES
NETZWERK FÜR DURCHLÄSSIGKEIT

ZERTIFIKATSKURSBESCHREIBUNG

1. VERSION UND GÜLTIGKEIT

Zertifikatskursbeschreibung gültig ab: 09. Februar 2017

Erstellt von: Simone Six, M.A., Wissenschaftliche Mitarbeiterin OTH mind, OTH Regensburg

Verantwortlich: Simone Six, M.A., Wissenschaftliche Mitarbeiterin OTH mind, OTH Regensburg

Wissenschaftliche Leitung: Prof. Dr. Clarissa Rudolph, Wissenschaftliche Leiterin „OTH mind“
Prof. Dr. Klaus Volbert, Fakultät Informatik und Mathematik

2. ANGABEN ZUR QUALIFIKATION

Bezeichnung der Qualifikation

Moderne Informatik

Name der Einrichtung, die die Qualifikation verliehen hat

Ostbayerische Technische Hochschule Regensburg,
93025 Regensburg; Bundesrepublik Deutschland

Name der Einrichtung, die die Weiterbildung durchgeführt hat

Zentrum für Weiterbildung und Wissensmanagement (ZWW) der Ostbayerischen Technischen Hochschule Regensburg (Organisation und Zertifizierung)

3. ANGABEN ZU STRUKTUR UND UMFANG DER AUSBILDUNG

Umfang

Insgesamt ca. 250 Unterrichtseinheiten Aufwand, davon mindestens 70 Unterrichtseinheiten Kontakt-/Präsenzzeit.

Mit dem Zertifikat werden 10 Credits, Leistungspunkte nach dem European Credit Transfer and Accumulation System (ECTS), vergeben.

Struktur

Die modularisierte Ausbildung (insgesamt vier Teilmodule) strukturiert sich wie folgt:

1. Algorithmik und Programmierung
2. Fortgeschrittene Konzepte im Software Engineering
3. Optimierung
4. Einführung in Maschinelles Lernen

Prüfungsleistung: Schriftliche Prüfung 90 Minuten (über die Teilmodule 1+2), Schriftliche Prüfung 90 Minuten (über die Teilmodule 3+4)

Inhalte der Teilmodule

Im Modul „Moderne Informatik“ werden sowohl Grundlagen als auch fortgeschrittenes Wissen in den Gebieten Algorithmik, Programmierung, Software Engineering, Optimierung und maschinelles Lernen in Form von seminaristischem Unterricht vermittelt. Anhand von Praxisbeispielen wird die Theorie anschaulich diskutiert und mit Hilfe von unterschiedlichen Übungsaufgaben eingeübt.

Vorbemerkung

- Sämtliche Teilmodule enthalten Inhalte zu folgenden Themen:
 - (1) Algorithmik und Programmierung
 - (2) Fortgeschrittene Konzepte im Software Engineering
 - (3) Optimierung
 - (4) Einführung in maschinelles Lernen

Teilmodul I: Algorithmik und Programmierung

Inhalte

- Algorithmische Grundkonzepte (Komplexität, Komplexitätsklassen, P und NP, NP-Vollständigkeit, Entwurfsmethoden)
- Basisalgorithmen mit Beispielen (Sortieren, Suchen, Hashing, Bäume, Graphen)
- Fortgeschrittene Algorithmen und Analysemethoden (Amortisierte Analyse, Approximationsalgorithmen, weitere ausgewählte Algorithmen mit Beispielen)
- Programmiersprachen: Konzepte und Beispiele
- Konzepte der objektorientierten Programmierung
- Fortgeschrittene Programmierkonzepte

Lernziele / Lernergebnisse / Kompetenzen

- Grundlagen zur Algorithmik (Entwurfsmethoden, Komplexität, Basisalgorithmen)
- Kennenlernen der Aspekte fortgeschrittener algorithmischer Methoden
- Fähigkeit zur Charakterisierung und Analyse von Algorithmen mit mathematischen Methoden
- Die Studierenden können fortgeschrittene algorithmische Verfahren zur Lösung praktischer Probleme analysieren und implementieren.
- Die Studierenden lernen fortgeschrittene Konzepte der objektorientierten Programmierung kennen und können diese durch eigene Implementierungen anwenden.

Literatur

- Cormen, T. H., Leiserson, C. E., Rivest, R.L., Stein, C.: Introduction to Algorithms, MIT Press, 2001
- Inden, M.: Der Weg zum Java-Profi: Konzepte und Techniken für die professionelle Java-Entwicklung, dpunkt.verlag, 3. Auflage, 2015
- Kleinberg, J., Tardos, E.: Algorithm Design, Addison Wesley, 2005
- Ottmann, T., Widmayer, P.: Algorithmen und Datenstrukturen, Spektrum Akademischer Verlag, 2002
- Pomberger, G., Dobler, H.: Algorithmen und Datenstrukturen, Pearson Studium 2008
- Schöning, U.: Algorithmik, Spektrum Akademischer Verlag, 2001
- Sedgewick, R.: Algorithmen in C++, Pearson Studium 2002
- Solymosi, A., Grude, U.: Grundkurs Algorithmen und Datenstrukturen in JAVA: Eine Einführung in die praktische Informatik, Vieweg, 2000

Teilmodul II: Fortgeschrittene Konzepte im Software Engineering

Inhalte

- Wiederholung Software Engineering 1 (Phasen im Software Engineering, agile Vorgehensmodelle)
- Modellierung mit UML (z.B. Klassendiagramme und OCL)
- Fortgeschrittene Modellierungskonzepte (z.B. EPK, BPMN, Process Engines und Webservice-Integration, Petrinetze, etc.)
- Software-Design (z.B. Analyse-, Architektur-, Design-Patterns)
- Software-Qualitätssicherung (Testtechniken und automatisiertes Testen mithilfe von Unit Tests, JUnit, TDD)
- Toolunterstützung im Software Engineering (z.B. Versionskontrolle, Buildsysteme, Trackingsysteme)

Lernziele / Lernergebnisse / Kompetenzen

- Die Studierenden können ihr Wissen aus den Grundlagenveranstaltungen zu Software Engineering benennen.
- Die Studierenden können fortgeschrittene Vorgehensmodelle, Methoden, Muster und Tools im Software Engineering erklären und diese auf konkrete Problemstellungen anwenden.
- Die Studierenden können mit fortgeschrittenen Modellierungstechniken ausführbare Modelle bewerten, optimieren und mit selbst erstellten Web Services automatisieren.

Literatur

- UML 2 glasklar, C. Rupp, S. Queins, 4. Auflage, Carl Hanser Verlag, 2012
- OCL language specification, Object Management Group, Version 2.4, 2014
- Praxishandbuch BPMN, J.Freund, B. Rücker, 5. Auflage, Carl Hanser Verlag, 2016
- Effektive Softwarearchitekturen, G. Starke, 7. Auflage, Carl Hanser Verlag, 2015
- Head First Design Patterns, E. Freeman et al., 1. Auflage, O'Reilly, 2004

Teilmodul III: Optimierung

Inhalte

- Optimierungsprobleme und Problemklassen
- Modellierung von Optimierungsproblemen, Anwendungsbeispiele
- Lineare Optimierung, Simplex-Algorithmus
- Linear-ganzzahlige Probleme, Branch-and-Bound-Lösungsverfahren, Schnittebenenverfahren, Heuristiken
- Optimalitätsbedingungen für nichtlineare Optimierungsprobleme
- Verfahren von Abstiegstyp, Gradientenverfahren
- Verfahren vom Newton-Typ, SQP-Verfahren
- Optimierung unter Differentialgleichungsnebenbedingungen, Shooting-Verfahren
- Optimale Steuerung und Parameterschätzung
- Praktische Übungen mit Software und Beispielproblemen

Lernziele / Lernergebnisse / Kompetenzen

- Kennenlernen der wichtigsten Klassen von Optimierungsproblemen
- Verständnis der mathematischen Grundlagen der Lösungsalgorithmen
- Fähigkeit zur Wahl von geeigneten Lösungsverfahren
- Praktische Erfahrung im Umgang mit Optimierungssoftware
- Übung in Modellierung und Lösung von Beispielproblemen und praktischen Anwendungsmöglichkeiten

Literatur

- Nocedal, J., Wright, S.: Numerical Optimization, Springer
- Jarre, F., Stoer, J.: Optimierung, Springer
- Ulbricht, M., Ulbricht, S.: Nichtlineare Optimierung, Birkhäuser
- Padberg, M., Linear Optimization and Extensions, Springer
- Nemhauser, G. L., Wolsey L. A.: Integer and Combinatorial Optimization, Wiley

Teilmodul IV: Einführung in Maschinelles Lernen**Inhalte**

Die Studierenden lernen grundlegende mathematische Konzepte und Algorithmen aus dem Bereich der statistischen Lerntheorie und des maschinellen Lernens. Die Themen sind eine Auswahl aus:

- Wiederholung Grundlagen Wahrscheinlichkeitstheorie
- Mathematische Prinzipien der (statistischen) Lerntheorie
- Überwachtes und Unüberwachtes Lernen
- K-Nearest und K-Means Klassifikation
- Bayes Klassifikation (Naive Bayes)
- Entscheidungsbäume
- Lineare Regression
- Neuronale Netze
- Support Vector Machines

Lernziele / Lernergebnisse / Kompetenzen

- Kennenlernen grundlegender Fragestellungen der Datenanalyse (großer Datensätze) und ihrer praktischen Relevanz
- Fähigkeit, praktische Probleme der Datenanalyse (großer Datensätze) mathematisch zu formulieren
- Kennenlernen und Verstehen grundlegender mathematischer Konzepte und Methoden des maschinellen Lernens
- Kennenlernen und Verstehen grundlegender Algorithmen des maschinellen Lernens, samt ihrer Stärken und Schwächen
- Fähigkeit zu entscheiden welche Algorithmen für welche Problemklasse angewendet werden können
- Fähigkeit die Algorithmen auf Probleme (mittlerer Komplexität) anzuwenden
- Fähigkeit selbständig weiterführende Literatur zu lesen und zu verstehen

Literatur

- C. M. Bishop, Pattern Recognition and Machine Learning, Springer, 2006
- G. James, D. Witten, T. Hastie, R. Tibshirani, An Introduction to Statistical Learning, Springer, 2013
- V. N. Vapnik, The Nature of Statistical Learning Theory, Springer, 1998

Übersicht zum Zertifikatskurs „Moderne Informatik“

Zuständige Fakultät	Fakultät Informatik und Mathematik
Spezielle Studienziele	Kenntnisse und Beherrschung grundlegender und fortgeschrittener Methoden der modernen Informatik, insbesondere in den Gebieten Algorithmik, Programmierung, Software Engineering, Optimierung und maschinelles Lernen
Spezielle Qualifikationsvoraussetzungen	Voraussetzung für die Zulassung ist ein einschlägiger erster Studienabschluss. Als einschlägig gelten Informatik-, Informationstechnologie- und Ingenieursstudiengänge sowie naturwissenschaftliche Studiengänge deutscher Hochschulen. Berufliche Praxis im IT-Bereich wünschenswert.
Spezielle Studienorganisation	Berufsbegleitend, in Blockveranstaltung
Regelstudiendauer	Ein Studiensemester

Übersicht über Kursmodule, Leistungsnachweise und Credits

1	2	3	4	5	6	7	8	9
Modul Nr.	Kursmodulbezeichnung	UE*)	Credits*)	Art der Lehrveranstaltung	Prüfungen			Ergänzende Regelungen
					Mündlich Schriftlich Dauer in Min.	Studien- begleitende Leistungs- nachweise	Fremd- sprachige Prüfungen	
1	Moderne Informatik	250	10					
1.1	Algorithmik und Programmierung	(75)	3	SU	Schriftliche Prüfung 90 Minuten			
1.2	Fortgeschrittene Konzepte im Software Engineering	(75)	3	SU				
1.3	Optimierung	(50)	2	SU	Schriftliche Prüfung 90 Minuten			
1.4	Einführung in Maschinelles Lernen	(50)	2	SU				

*) Angaben in Klammern geben den jeweiligen Anteil eines Teilmoduls am Gesamtmodul an.

Ein Credit entspricht im Durchschnitt einer Arbeitsbelastung für Präsenz und Selbststudium von 25 Unterrichtseinheiten.

Abkürzungen

m.E. Bewertung mit/ohne Erfolg

StA Studienarbeit

SU

Seminaristischer Unterricht ggf. mit
Übungen

TN Teilnahmenachweis

UE Unterrichtseinheit