

LEHRVERANSTALTUNGSBESCHREIBUNG

1. Angaben zum Programm

1.1 Hochschuleinrichtung	Babes-Bolyai Universität
1.2 Fakultät	Mathematik und Informatik
1.3 Department	Informatik
1.4 Fachgebiet	Informatik
1.5 Studienform	Bachelor
1.6 Studiengang / Qualifikation	Informatik

2. Angaben zum Studienfach

2.1 LV-Bezeichnung	Software Systementwurf						
2.2 Lehrverantwortlicher – Vorlesung							
2.3 Lehrverantwortlicher – Seminar							
2.4 Studienjahr	3	2.5 Semester	6	2.6 Prüfungsform	P	2.7 Art der LV	Obligatorisch

3. Geschätzter Workload in Stunden

3.1 SWS	4	3.2 von denen: Vorlesung	2	3.3 Übung+Labor	2
3.4 Gesamte Stundenanzahl im Lehrplan	48	3.5 von denen: Vorlesung	24	3.6 Übung+Labor	24
Verteilung der Studienzeit:					Std.
Studium nach Handbüchern, Kursbuch, Bibliographie und Mitschriften					22
Zusätzliche Vorbereitung in der Bibliothek, auf elektronischen Fachplattformen und durch Feldforschung					22
Vorbereitung von Seminaren/Übungen, Präsentationen, Referaten, Portfolios und Essays					22
Tutorien					3
Prüfungen					8
Andere Tätigkeiten:					0
3.7 Gesamtstundenanzahl Selbststudium	77				
3.8 Gesamtstundenanzahl / Semester	125				
3.9 Leistungspunkte	5				

4. Voraussetzungen (falls zutreffend)

4.1 curricular	
4.2 kompetenzbezogen	

5. Bedingungen (falls zutreffend)

5.1 zur Durchführung der Vorlesung	Projektor
5.2 zur Durchführung des Seminars / der Übung	Labor

6. Spezifische erworbene Kompetenzen

Berufliche Kompetenzen	<p>K2.1 Identifizierung geeigneter Methoden für die Entwicklung von Softwaresystemen</p> <p>K2.2 Identifizierung und Erklärung geeigneter Mechanismen für die Spezifizierung von Softwaresystemen</p> <p>K2.3 Benutzung der Methoden, Spezifizierungsmechanismen und Entwurfsmedien für die Entwicklung von Software-Anwendungen</p> <p>K2.4 Benutzung von geeigneten Kriterien und Methoden für die Auswertung von Software-Anwendungen</p> <p>K2.5 Entwurf von spezifischen Software-Anwendungen</p>
Transversale Kompetenzen	<p>TK1 Anwendung der Regeln für gut organisierte und effiziente Arbeit, für verantwortungsvolle Einstellungen gegenüber der Didaktik und der Wissenschaft, für kreative Förderung des eigenen Potentials, mit Rücksicht auf die Prinzipien und Normen der professionellen Ethik</p> <p>TK2 Effizienter Ablauf der Tätigkeiten in einer interdisziplinären Gruppe, das Entwickeln der Kapazitäten für empathische zwischenmenschliche Kommunikation, Verknüpfung und Zusammenarbeit mit unterschiedlichen Gruppen</p> <p>TK3 Anwendung von effizienten Methoden und Techniken für Lernen, Informieren und Recherchieren, für das Entwickeln der Kapazitäten der praktischen Umsetzung der Kenntnisse, der Anpassung an die Bedürfnisse einer dynamischen Gesellschaft, der Kommunikation in rumänischer Sprache und in einer internationalen Verkehrssprache</p>

7. Ziele (entsprechend der erworbenen Kompetenzen)

7.1 Allgemeine Ziele der Lehrveranstaltung	<ul style="list-style-type: none"> • Verstehen der Begriffe eines vollständig korrekten Algorithmus, sowie eines partiell korrekten Algorithmus. • Entwurf paralleler Algorithmen, sowie der Beweis deren Korrektheit. • Testen und Verifikation von Softwaresysteme. • Erlernen des korrekten Programmentwurfs anhand der Spezifikation. • Moderne Programmierkenntnisse.
7.2 Spezifische Ziele der Lehrveranstaltung	<ul style="list-style-type: none"> • Erlernen der Schritte einer Inspektion • Erlernen der Testbeispiele Aufbau. • Anwenden der Instrumente zum Testmanagement. • Testcriteria (black-box, white-box).

8. Inhalt

8.1 Vorlesung	Lehr- und Lernmethode	Anmerkungen
1. Verifikation und Validierung. Begriffe, Programmtesten, black-box testen.	Vortrag, Unterrichtsgespräch, Problematisierung	
2. Verifikation und Validierung. White-box testen, Test Typen.	Vortrag, Unterrichtsgespräch, Problematisierung	
3. Instrumente für das Testmanagement	Vortrag, Unterrichtsgespräch, Problematisierung	
4. Symbolische Ausführung.	Vortrag, Unterrichtsgespräch, Problematisierung	
5. Model checking	Vortrag, Unterrichtsgespräch, Problematisierung	
6. Die Theorie der Programmkorrektheit. Die Beiträge von Flyd, Hoare, Dijkstra, Droomey, Morgen.	Vortrag, Unterrichtsgespräch, Problematisierung	
7. Qualitätssicherung.	Vortrag, Unterrichtsgespräch, Problematisierung	
8. Performance testing und WebDriver.	Vortrag, Unterrichtsgespräch, Problematisierung	
9. Testen der GBO und Web Anwendungen mit Selenium und WebDriver.	Vortrag, Unterrichtsgespräch, Problematisierung	
10. COCOMO	Vortrag, Unterrichtsgespräch, Problematisierung	
11. Inspektion	Vortrag, Unterrichtsgespräch, Problematisierung	
12. SPI, SQA, CMM.	Vortrag, Unterrichtsgespräch, Problematisierung	
Literatur In deutscher Sprache: <ol style="list-style-type: none"> 1. Kleuker, S., Formale Modelle der Softwareentwicklung, Vieweg Teubner, 2009. 2. Haubelt, C., Teich, J., Digitale Hardware/Software-Systeme, Spezifikation und Verifikation, Springer, 2010. <ol style="list-style-type: none"> 1. BALANESCU T., Corectitudinea programelor, Editura tehnica, Bucuresti 1995. 2. DIJKSTRA, E., A constructive approach to the problem of program correctness, BIT, 8(1968), pg.174-186. 3. DIJKSTRA, E., Guarded commands, nondeterminacy and formal derivation of programs, CACM, 18(1975), 8, pg.453-457. 4. DROMEY G., Program Derivation. The Development of Programs From Specifications, Addison Wesley Publishing Company, 1989. 5. FRENTIU, M., Verificarea corectitudinii programelor, Ed.Univ."Petru-Maior", 2001. 6. GRIES, D., The Science of Programming, Springer-Verlag, Berlin, 1981. 7. HOARE, C.A.R., An axiomatic basis for computer programming, CACM, 12(1969), pg.576-580, 583. 8. Morgan, C., Programing from Specifications, Prentice Hall, NewYork, 1990. B. Internet		
8.2 Labor	Lehr- und Lernmethode	Anmerkungen
Ü1. Spezifikation. Black-box Testing (BBT) L1. Black-box Testing	Projekte, Aufgabenlösen, Selbststudium, Gruppenübungen, Unterrichtsgespräch	
Ü2. White-box Testing (WBT). L2. White-box Testing (WBT).	Projekte, Aufgabenlösen, Selbststudium, Gruppenübungen, Unterrichtsgespräch	
Ü3. Klausur.	Projekte, Aufgabenlösen, Selbststudium,	

L3. Test level	Gruppenübungen, Unterrichtsgespräch	
Ü4. Korrektheit, Floyd, Hoare. L4. Instrumente für das Testmanagement.	Projekte, Aufgabenlösen, Selbststudium, Gruppenübungen, Unterrichtsgespräch	
Ü5. GBO und Web Testen. L5. GBO und Web Testen	Projekte, Aufgabenlösen, Selbststudium, Gruppenübungen, Unterrichtsgespräch	
Ü6. Inspektion. L6. Inspektion.	Projekte, Aufgabenlösen, Selbststudium, Gruppenübungen, Unterrichtsgespräch	

9. Verbindung der Inhalte mit den Erwartungen der Wissensgemeinschaft, der Berufsverbände und der für den Fachbereich repräsentativen Arbeitgeber

Die Vorlesung entspricht den IEEE und ACM Richtlinien für Informatik Curricula.
Die Vorlesung ist wichtig für die Software Firmen.

10. Prüfungsform

Veranstaltungsart	10.1 Evaluationskriterien	10.2 Evaluationsmethoden	10.3 Anteil an der Gesamtnote
10.4 Vorlesung	der Kenntnisstand in Bezug auf den Lernstoff	Schriftliche Prüfung	50%
10.5 Übung+Labor	die Fertigkeit, die in den Vorlesungen und Übungen erworbenen Kenntnisse für das Lösen konkreter Aufgaben einzusetzen	Klausur	25%
		Projekt	25%
10.6 Minimale Leistungsstandards			
Die Gesamtnote muss mindestens 5 (auf einer Skala von 1 bis 10) betragen, damit die für diese Lehrveranstaltung vorgesehenen ECTS-Punkte vergeben werden.			

Ausgefüllt am:

22 April 2018

Vorlesungsverantwortlicher

Lect. Dr. Camelia Serban

Seminarverantwortlicher

Lect. Dr. Camelia Serban

Genehmigt im Department am:

Departmentleiter

Prof. PhD. Anca Andreica