**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 in deutscher Sprache** |

# 2. Angaben zum Studienfach

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 2.1 LV-Bezeichnung  | **Überprüfung und Validierung von Softwaresystemen** |  |  |  |
| 2.2 Lehrverantwortlicher – Vorlesung  |  Dr.ing. Kuderna-Iulian Bența  |
| 2.3 Lehrverantwortlicher – Seminar  |  Dr.ing. Kuderna-Iulian Bența  |
| 2.4 Studienjahr  | 3  | 2.5 Semester  | 2  | 2.6. Prüfungsform  | P  | 2.7 Art der LV  | Verpflichtend  |
| 2.8 Modulnummer | **MLG5014** |

# 3. Geschätzter Workload in Stunden

3.1 SWS 4 3.2 von denen: 2 3.3 Übung+Labor 2

Vorlesung

3.4 Gesamte Stundenanzahl im 48 3.5 von denen: 24 3.6 Übung+Labor 24

Lehrplan Vorlesung

Verteilung der Studienzeit: Std.

Studium nach Handbüchern, Kursbuch, Bibliographie und Mitschriften 32

Zusätzliche Vorbereitung in der Bibliothek, auf elektronischen Fachplattformen und durch 22

Feldforschung

Vorbereitung von Seminaren/Übungen, Präsentationen, Referaten, Portfolios und Essays 22

Tutorien 35

Prüfungen 16

Andere Tätigkeiten: .................. 0

3.7 Gesamtstundenanzahl Selbststudium 127

3.8 Gesamtstundenanzahl / Semester 175

3.9 Leistungspunkte 7

# 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** | K2.1 Identifizierung geeigneter Methoden für die Entwicklung von Softwaresystemen K2.2 Identifizierung und Erklärung geeigneter Mechanismen für die Spezifizierung von Softwaresystemen K2.3 Benutzung der Methoden, Spezifizierungsmechanismen und Entwurfsmedien für die Entwicklung von Software-Anwendungen K2.4 Benutzung von geeigneten Kriterien und Methoden für die Auswertung von Software Anwendungen K2.5 Entwurf von spezifischen Software-Anwendungen   |
| **Transversale Kompetenzen** | **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 **TK2** Effizienter Ablauf der Tätigkeiten in einer interdisziplinären Gruppe, das Entwickeln der Kapazitäten für empathische zwischenmenschliche Kommunikation, Verknüpfung und Zusammenarbeit mit unterschiedlichen Gruppen **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  |

**7. Ziele** (entsprechend der erworbenen Kompetenzen)

|  |  |  |
| --- | --- | --- |
| 7.1 Allgemeine Ziele der Lehrveranstaltung  | •  | 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.  |
|  | •   | Programmierkenntnisse.  |
| 7.2 Spezifische Ziele der Lehrveranstaltung  | • •  | Erlernen der Schritte einer Inspektion Erlernen der Testbeispiele Aufbaus.  |
|   | •  | Anwenden der Instrumente zum Testmanagement.  |
|       | •  | Testkriterien (black-box, white-box).  |

# 8. Inhalt

|  |  |  |
| --- | --- | --- |
| 8.1 Vorlesung  | Lehr- und Lernmethode  | Anmerkungen  |
| 1. Softwaresystemen Verifikation und Validierung. Programm Inspektion | Interaktive Präsentation, Erklärungen, Konversation, didaktische Demonstration |   |
| 2. Programmtesten (1): das Konzept der Programm Inspektion. Testen Kriterien. White-box testen  | Interaktive Präsentation, Erklärungen, Konversation, didaktische Demonstration |   |
| 3. Programmtesten (2): Testen Kriterien. White-box testen.  | Interaktive Präsentation, Erklärungen, Konversation, didaktische Demonstration |   |
| 4. Levels testen. Test Typen.  | Interaktive Präsentation, Erklärungen, Konversation, didaktische Demonstration |   |
| 5. Web Anwendung Testen.  | Interaktive Präsentation, Erklärungen, Konversation, didaktische Demonstration |   |
| 6. Symbolische Ausführung.  | Interaktive Präsentation, Erklärungen, Konversation, didaktische Demonstration |   |
| 7. Modellen Verifikation.  | Interaktive Präsentation, Erklärungen, Konversation, didaktische Demonstration |   |
| 8. Die Theorie der Programmkorrektheit (I)- Korrektheit Konzept Entwicklung- Die Beiträge von Floyd, Hoare  | Interaktive Präsentation, Erklärungen, Konversation, didaktische Demonstration |   |
| 9. Die Theorie der Programmkorrektheit- Die Beiträge von Dijkstra  | Interaktive Präsentation, Erklärungen, Konversation, didaktische Demonstration |   |
| 10. Softwareprodukte Qualitätssicherung. Qualitätskontrolle.  | Interaktive Präsentation, Erklärungen, Konversation, didaktische Demonstration |   |
| 11. Testfähigkeiten und Aufgaben des Testers | Interaktive Präsentation, Erklärungen, Konversation, didaktische Demonstration |   |
| 12. Berichte Präsentationen  | Interaktive Präsentation, Erklärungen, Konversation, didaktische Demonstration |   |
| Literatur  In deutscher Sprache: 1. Kleuker, S., Formale Modelle der Softwareentwicklung, Vieweg Teubner, 2009.
2. Haubelt, C., Teich, J., Digitale Hardware/Software-Systeme, Spezifikation und Verifikation, Springer, 2010.
3. A. Spillner, M. Winter, A. Pietschker (Hrsg.), Test, Analyse und Verifikation von Software – gestern, heute, morgen, November 2017, 224 Seiten, Broschur, dpunkt.verlag, ISBN Print: 978-3-86490-470-7
4. R. Osherove, The Art of Unit Testing 2. Auflage 2015, 312 Seiten, MITP, ISBN: 9783826697128

Literatur in anderen Sprachen 1. Frentiu, M., Verificarea si validarea sistemelor soft, Presa Universitara Clujeana, 2010
2. R. S. Pressman, Software engineering: a practinioner’s approach, seventh edition, Higher Education, 2010
3. L. Crispin, J. Grecory, Agile testing: a practical guide for testers and agile teams, Addison-Wesley, 2009
4. M. Pezzand, M. Young, Software Testing and Analysis: Process, Principles and Techniques, John Wiley & Sons, 2008
 |
| 8.2 Labor  | Lehr- und Lernmethode  | Anmerkungen  |
| Ü1: Inspektion L1: InspektionInspektionswerkzeuge. Issue-Tracker-Tools. Testmanagement-Tools  | Projekte, Aufgabenlösen, Selbststudium, Gruppenübungen, Unterrichtsgespräch  |   |
| Ü2: Black-box Testing (BBT) SpezifikationL2: Design von Testfällen basierend auf Spezifikationen (BBT) Inspektionswerkzeuge. Issue-Tracker-Tools. Testmanagement-Tools | Projekte, Aufgabenlösen, Selbststudium, Gruppenübungen, Unterrichtsgespräch  |   |
| Ü3. White-box Testing (WBT) Spezifikation.L3: Design von Testfällen basierend auf Spezifikationen (WBT)Inspektionswerkzeuge. Issue-Tracker-Tools. Testmanagement-Tools | Projekte, Aufgabenlösen, Selbststudium,  |   |
| U4: Test level L4:Test level Inspektionswerkzeuge. Issue-Tracker-Tools. Testmanagement-Tools | Gruppenübungen, Unterrichtsgespräch  |  |
| Ü5: Korrektheit, Floyd, HoareL5: GBO und Web Testen Web-Test Wezeuge, Inspektionswerkzeuge. Issue-Tracker-Tools. Testmanagement-Tools | Projekte, Aufgabenlösen, Selbststudium, Gruppenübungen, Unterrichtsgespräch  |   |
| Ü6: Verfeinerung von SpezifikationenL6: Statische Analyse: JML, ESC2Java | Projekte, Aufgabenlösen, Selbststudium, Gruppenübungen, Unterrichtsgespräch  |   |
|  Die **Literatur** ist die gleiche wie für die Vorlesung |

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

- Die Studenten erwerben die Möglichkeit, die Tools zur Verwaltung des Testprozesses zu verwenden

- Den Studenten werden verschiedene Testmethoden vorgestellt, die sie auf Softwareprodukte anwenden werden

- Die Studenten erwerben die Möglichkeit, Testfälle anhand verschiedener Kriterien (Black-Box, White-Box) zu entwerfen.

# 10. Prüfungsform

|  |  |  |  |
| --- | --- | --- | --- |
| Veranstaltungsart  | 10.1 Evaluationskriterien  | 10.2 Evaluationsmethoden  | 10.3 Anteil an der Gesamtnote  |
| 10.4 Vorlesung  | Schriftliche Prüfung (die **E** Note)  | Schriftliche Prüfung  | 40%  |
| Verifikation und Validierung Berichte Präsentation | Mündliche Prüfung | Bonus (ab 0 bis 2 Punkte in die Schlussnote) |
| 10.5 Übung+Labor  | Die Teilnahme an Diskussionen und Übungen wird vermerkt (die **S** Note) | Seminaraktivität   | 10%    |
| Die Laboraktivität wird anhand der Anforderungen notiert (die **L** Note) | Bewertung von Laborthemen | 50% |
| Bemerkungen:* Die Laboraktivität wird während der Untersuchung oder der erneuten Untersuchung nicht neu bewertet.
* Die Laboraktivität für die verbleibende Studenten muss neu gemacht werden.
* Die Anwesenheit bei der Laboraktivität kann nur mit der Gruppe erfolgen, zu der der Student gehört.
* Verspätete Lieferung wird mit 2 Punkten aus dem angegebenen Hinweis bestraft.
* Nach Ablauf der Lieferzeit wird das Laborarbeit mit 1 gekennzeichnet.
 |
| 10.6 Minimale Leistungsstandards  |
| • Praktische Fähigkeit zur Verwendung von Testprozessmanagement-Tools.• Theoretisches Verständnis und praktische Fähigkeit, verschiedene Testfall-Designkriterien (Black-Box, White-Box) zu verwenden.• Kenntnis verschiedener Überprüfungsmethoden (Inspektion, Prüfung, Nachweis der Richtigkeit).• Teilnahmebedingungen an der Abschlussprüfung: 75% der Seminaraktivitäten (mindestens 4 Teilnehmer) und 90% der Laboraktivitäten (mindestens 5 Teilnehmer).• Um die Disziplin zu fördern, muss man an der Prüfung teilnehmen und den endgültigen Durchschnitt erreichen M >= 5.00, wo M=40%E+10%S+50%L+ Bonus, wie oben erklärt. |

 Ausgefüllt am: Vorlesungsverantwortlicher Seminarverantwortlicher

April.2024 Lekt. dr. ing. Kuderna-Iulian Bența Lekt. dr. ing. Kuderna-Iulian Bența

Genehmigt im Department am: Departmentleiter

April 2024 Dozent Dr. Sterca Adrian