

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	Fortgeschrittene Programmierungsmethoden						
2.2 Lehrverantwortlicher – Vorlesung	Lect. dr. Cătălin Rusu						
2.3 Lehrverantwortlicher – Seminar	Lect. dr. Cătălin Rusu						
2.4 Studienjahr	2	2.5 Semester	3	2.6. Prüfungsform	P	2.7 Art der LV	Pflichtfach

3. Geschätzter Workload in Stunden

3.1 SWS	5	von denen: 3.2 Vorlesung	2	3.3 Seminar/Übung	3
3.4 Gesamte Stundenanzahl im Lehrplan	70	von denen: 3.5 Vorlesung	28	3.6 Seminar/Übung	42
Verteilung der Studienzeit:					Std.
Studium nach Handbücher, Kursbuch, Bibliographie und Mitschriften					20
Zusätzliche Vorbereitung in der Bibliothek, auf elektronischen Fachplattformen und durch Feldforschung					10
Vorbereitung von Seminaren/Übungen, Präsentationen, Referate, Portfolios und Essays					30
Tutorien					10
Prüfungen					10
Andere Tätigkeiten:					-
3.7 Gesamtstundenanzahl Selbststudium	80				
3.8 Gesamtstundenanzahl / Semester	150				
3.9 Leistungspunkte	6				

4. Voraussetzungen (falls zutreffend)

4.1 curricular	<ul style="list-style-type: none"> • Objektorientierte Programmierung • Datenstrukturen und Algorithmen
4.2 kompetenzbezogen	<ul style="list-style-type: none"> • Grundlegende Begriffe der objektorientierter Programmierung.

5. Bedingungen (falls zutreffend)

5.1 zur Durchführung der Vorlesung	<ul style="list-style-type: none">• Videoprojektor
5.2 zur Durchführung des Seminars / der Übung	<ul style="list-style-type: none">• Computers mit den Betriebssystemen Windows und Zugang mit einem individuellen Kennwort

6. Spezifische erworbene Kompetenzen

Berufliche Kompetenzen	<ul style="list-style-type: none"> • der Kurs Kompetenzen im Bereich Systemdesign an. • das Erlernen von Methoden selbständiger Arbeit • Erfahrung in objektorientierter Programmierung.
Transversale Kompetenzen	<ul style="list-style-type: none"> • die erworbene Erkenntnisse objektorientierter Programmierung anzuwenden • die Fähigkeiten im Bereich objektorientierter Programmierung.

7. Ziele (entsprechend der erworbenen Kompetenzen)

7.1 Allgemeine Ziele der Lehrveranstaltung	<ul style="list-style-type: none"> • Kenntnisse im Umgang mit Datenmodellen, insbesondere die der objektorientierter Programmierung. • Anwenden der Schablone • Anwenden von frameworks
7.2 Spezifische Ziele der Lehrveranstaltung	<ul style="list-style-type: none"> • Analyse und Design von Softwaresysteme • Anlernen grundlegender Kenntnisse über UML • Aneignen von Schablonen. • Java • C#

8. Inhalt

8.1 Vorlesung	Lehr- und Lernmethode	Anmerkungen
1. Einführung in Java und C#	Vortrag, Unterrichtsgespräch, Erklärungen, Beispiele	
2. Klassen und Objekte Java/C#	Vortrag, Unterrichtsgespräch, Erklärungen, Beispiele	
3-4. Abstrakte Klassen	Vortrag, Unterrichtsgespräch, Erklärungen, Beispiele	
5. Pakete, Zugänge, java.util	Vortrag, Unterrichtsgespräch, Erklärungen, Beispiele	
6-7. Bibliotheken	Vortrag, Unterrichtsgespräch, Erklärungen, Beispiele	
8-9. GUI und GRASP Schablone	Vortrag, Unterrichtsgespräch, Erklärungen, Beispiele	
10. Observer Schablon	Vortrag, Unterrichtsgespräch, Erklärungen, Beispiele	
11. Einführung in das objektorientiertes Design.	Vortrag, Unterrichtsgespräch, Erklärungen, Beispiele	
12-13. Factory patterns. Dependency Injection.	Vortrag, Unterrichtsgespräch,	

Inversion of Control	Erklärungen, Beispiele	
14. Reflektion	Vortrag, Unterrichtsgespräch, Erklärungen, Beispiele	

Literatur

1. Kamper, S., Grundkurs Programmieren mit Visual Basic, Vieweg Teubner, Wiesbaden, 2009.
2. Kaiser, R., Einführung in C++ und objektorientierte .NET Programmierung, Springer, 2009.
3. Rauber, Th., Runger, G., Parallele Programmierung, Springer, 2007.

Zusatzliteratur

1. Eckel, B., Thinking in Java, 4th Edition, New York: Prentice Hall, 2006.
2. Larman, C.: Applying UML and Design Patterns: An Introduction to OO Analysis and Design, Berlin: Prentice Hall, 2004.
3. Fowler, M., Patterns of Enterprise Application Architecture, Addison-Wesley, 2002.
4. E. Gamma, R. Helm, R. Johnson, J. Vlissides, Design Patterns – Elements of Reusable Object Oriented Software, Ed. Addison Wesley, 1994.
5. Walls, Craig, Spring in Action, Third Edition, Ed. O’Reilly, 2011.
6. Kent Beck, Test Driven Development: By Example, Ed. Addison-Wesley Professional, 2002.
7. ***, Tutoriale Java <http://download.oracle.com/javase/tutorial/>
8. ***, Tutoriale C# <http://msdn.microsoft.com/en-us/library/aa288436%28v=vs.71%29.aspx>

8.2 Seminar / Übung	Lehr- und Lernmethode	Anmerkungen
	Lösung der Aufgaben	
S1. Einfache Java/C# Programme		
S2. Polymorphie		
S3. Case studies		
S4. Graphische Oberflächen		
S5. Observer Schablon		
S6. Design Schablone		
S7. Persistence level		
Labor		
L1. Einfache Java/C# Programme		
L2-L4. Unit testing		
L5. Exception handling		
L6. I/O		
L7. Einfache graphische Oberflächen		
L8-L10. Mittlere komplexitätsstufe Programme mit graphischer Oberfläche		
L11-L12. Observer Schablon		
L13-L14. Anwendungen.		

Literatur

1. Eckel, B., Thinking in Java, 4th Edition, New York: Prentice Hall, 2006.
2. Larman, C.: Applying UML and Design Patterns: An Introduction to OO Analysis and Design, Berlin: Prentice Hall, 2004.
3. Fowler, M., Patterns of Enterprise Application Architecture, Addison-Wesley, 2002.
4. E. Gamma, R. Helm, R. Johnson, J. Vlissides, Design Patterns – Elements of Reusable Object Oriented Software, Ed. Addison Wesley, 1994.
5. Walls, Craig, Spring in Action, Third Edition, Ed. O’Reilly, 2011.
6. Kent Beck, Test Driven Development: By Example, Ed. Addison-Wesley Professional, 2002.

7. ***, Tutoriale Java <http://download.oracle.com/javase/tutorial/>
 8. ***, Tutoriale C# <http://msdn.microsoft.com/en-us/library/aa288436%28v=vs.71%29.aspx>

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

- Der Kurs folgt die IEEE und ACM Curricula Empfehlungen für das Informatikstudium.
- Der Kurs konzentriert sich auf die Lösung der Probleme die ein Absolvent an seiner zukünftigen Arbeitsstelle lösen muss.

10. Prüfungsform

Veranstaltungsart	10.1 Evaluationskriterien	10.2 Evaluationsmethoden	10.3 Anteil an der Gesamtnote
10.4 Vorlesung	Ein vertieftes Verständnis der im Kurs behandelten Themen	Klausur Prüfung Praktische Anwendung	30% 15% 20%
10.5 Seminar / Übung	Die Fähigkeit praktische Probleme direkt am Computer in begrenzter Zeit zu lösen	Praktische Prüfung am Ende des Semesters	35%
10.6 Minimale Leistungsstandards			
<ul style="list-style-type: none"> • Note 5 auf einer Skala von 1 bis 10. 			

Ausgefüllt am:

13.12.2013

Vorlesungsverantwortlicher

Ruff Laura

Seminarverantwortlicher

Gaceanu Radu

Genehmigt im Department am:

20.12.2013

Departmentdirektor

Prof. Dr.Bazil Parv