

DESCRIPTION OF A STUDY COURSE – SYLLABUS

Title of a course	Modular Software Engineering				
Head of course	Associate Professor, PhD Alen Jakupović				
Study programme	Specialist professional graduate study of Information Technology in Business Systems- Major: Software Engineering in Business Systems				
Status of a course	Obligatory				
Year of study	2.	Semester	III	ECTS credits	4
Teaching plan (L + E + S+ Pr)	1+2+0+0				
Goals of a course					
Acquire competencies for the development of computer programs in a modular multilayer architecture.					
Conditions for enrolling course					
No conditions					
Learning outcomes on a level of a study programme which includes course					
<p>Outcome 1: Apply information and communication systems design methods.</p> <p>Outcome 3: Apply software engineering principles in the development of information systems.</p> <p>Outcome 6: Apply appropriate tools in the implementation of information and communication systems.</p> <p>Outcome 7: Apply methods and techniques for creating and managing databases.</p> <p>Outcome 8: Apply methods and techniques for managing security and data protection in information and communication systems.</p> <p>Outcome 12: Use modular software engineering in the implementation of business information systems.</p> <p>Outcome 14: Organize and lead teamwork in the field of business information systems.</p> <p>Outcome 17: Present development and software solutions within a business organization.</p>					
Expected learning outcomes on a level of a course					
<ol style="list-style-type: none"> 1. Apply object-oriented approach in the software solution analysis phase. 2. Apply object-oriented approach in the software solution design phase. 3. Create a modular multi-layer software solution architecture. 4. Create a software solution based on models from the analysis and design phase, and with a modular multi-layer architecture. 5. Compile project documentation of software solution development with modular multi-layer architecture. 					
Content of a course					
Principles and technologies of modular software engineering. Object-oriented methodology. Modelling: object, dynamic, functional. Development of program support. Unified Modelling Language (UML). Component-oriented programming architecture. Goals of component-oriented architecture. Multi-layer programming architecture. Component-standards, link to UML, development of object-oriented components. Development of software system with modular software engineering. Comparative analysis of traditional and object paradigm. Common Object Request Broker Architecture (CORBA). Advanced concepts of distributed computing. JavaBeans (Enhanced JavaBeans). Principles of designing JavaBeans.					
Teaching modes	<input checked="" type="checkbox"/> lectures <input type="checkbox"/> auditory exercises <input checked="" type="checkbox"/> seminars and workshops <input type="checkbox"/> distance learning <input type="checkbox"/> field classes		<input checked="" type="checkbox"/> individual assignments <input type="checkbox"/> multimedia and network <input type="checkbox"/> laboratory <input type="checkbox"/> supervisor's work <input type="checkbox"/> other _____		
Comments					
Students' obligations					

Grading, evaluation and monitoring of students' work continuously during lectures and exams

Grading is based upon evaluation of course's learning outcomes' adoption. Grading is performed continuously during lectures and/or during exam, in compliance with the provisions of Regulation on the assessment of students.

Continuous check-up:

Outcomes	Project documentation	Project defense	Threshold	Max
Outcome 1	5 %	5 %	5 %	10 %
Outcome 2	5 %	5 %	5 %	10 %
Outcome 3	5 %	15 %	10 %	20 %
Outcome 4	5 %	45 %	25 %	50 %
Outcome 5	10 %		5 %	10 %
Percentage of ECTS	1,2	2,8	-	-
Total	30 %	70 %	50 %	100 %

A student has passed the exam if he has acquired a percentage of credits for each learning outcome higher or equal to defined threshold.

Exam term:

Outcomes	Written exam	Oral exam	Max
Outcome 1	5 %	5 %	10 %
Outcome 2	5 %	5 %	10 %
Outcome 3	5 %	15 %	20 %
Outcome 4	5 %	45 %	50 %
Outcome 5	10 %		10 %
Percentage of ECTS	1,2	2,8	-
Total	30 %	70 %	100 %

A student has passed the exam if he has acquired a percentage of credits for each learning outcome higher or equal to defined threshold.

Grading:

A student has passed the exam if he has acquired at least 50% of anticipated credits of a specific learning outcome.

If a student has passed learning outcomes of all courses, the accomplished credits (percentages) of all passed learning outcomes are being added, while the final grade is defined upon following table:

Range of credits (percentages)	Numerical grade	ECTS grade
90,00 – 100,00	Excellent (5)	A
75,00 – 89,99	Very good (4)	B
60,00 – 74,99	Good (3)	C
50,00 – 59,99	Sufficient (2)	D
0,00 – 49,99	Insufficient (1)	F

Obligatory literature

1. Authorized Lectures
2. Heineman, G.T., Councill, W.T.: Component-Based Software Engineering: Putting the Pieces Together. Addison-Wesley, 2001

Additional literature

1. Cheesman, J., Daniels, J.: UML Components: A Simple Process for Specifying Component-Based Software, Addison-Wesley, 2000.

