



RTU Course "Object-Oriented System Analysis"

12306 Lietišķo datorzinātņu katedra

General data

Code	DPI502
Course title	Object-Oriented System Analysis
Course status in the programme	Compulsory/Courses of Limited Choice
Course level	Post-graduate Studies
Course type	Academic
Field of study	Computer Science
Responsible instructor	Nikiforova Oksana
Volume of the course: parts and credits points	1 part, 4.0 Credit Points, 6.0 ECTS credits
Language of instruction	LV, EN, RU
Possibility of distance learning	Not planned
Abstract	At the beginning of the course an introduction to the types of system requirements structuring is given. The forms of requirement presentation are shown to explain possible input information at the stage of system analysis. Further process of system analysis and design is presented from use case perspective and system modeling is based on system use case analysis. System analysis and design is performed using UML diagrams. System conceptual modeling and analysis of using the system are demonstrated within the course. Great attention is devoted to analysis of object interaction as well as identification and description of system classes. Practical system modeling in UML is shown based on the usage of CASE tool Sparx Enterprise Architect. System design process and the concept of system architecture are also mentioned in the syllabus. At the end of the course several methodologies of object-oriented system analysis and the role of system analysis in general software development process are considered.
Goals and objectives of the course in terms of competences and skills	The goal is to gain experience in using object-oriented system analysis, as well as to learn to use advanced tools for object-oriented system modeling. Objectives: - to learn to use methods and techniques of object-oriented analysis for identification of elements of a system model; - to learn to develop a system model in UML in analysis and design of software development - to gain practical experience in using an object-oriented system model for the implementation of a software system
Structure and tasks of independent studies	Students are asked to group into teams. All teams are assigned the same problem domain for analysis and modeling. UML diagrams and additional system analysis artefacts are developed at the practical lessons and laboratories. At the end of the course students exchange the models among the teams and have to evaluate the results. Students, who don't want to take part in team work, have the task to develop a system model individually.
Recommended literature	Nikiforova O. Objektorientēta sistēmanalīze, Drukatāva, 2007 Lano K. Advanced System Design with Java, UML and MDA, Elsevier, 2005 Dennis A. et al Systems Analysis and Design with UML 2.0, Wiley, 2005 Satzinger J.W. Object-Oriented Analysis and Design, Thomson Course Technology, 2005
Course prerequisites	It is preferred that students have initial knowledge in some object-oriented programming language

Course outline

Theme	Hours
Introduction to the course and course requirements. Introduction to object-oriented system analysis.	2
Definition of system requirements in terms of use-cases.	12
System conceptual modeling.	12
Analysis and modeling of object interaction.	12
Design of class static and dynamic aspects.	12
Introduction to system implementation.	6
Methodologies for object-oriented system analysis.	8

Learning outcomes and assessment

Learning outcomes	Assessment methods
Is able to use methods of object-oriented system analysis for identification of elements of system model.	The instructor checks task performance.
Is able to develop a system model in UML notation in accordance with system description.	The instructor checks task performance.
Is able to relate elements of system models from different kinds of diagrams, to analyze system models and to base new element derivation on existing elements.	The instructor checks task performance.
Is able to generate a program code from UML class diagram.	The instructor checks task performance.
Is able to use CASE tools for development of a system model in UML notation.	The instructor checks task performance.

Is able to analyze and evaluate system model created by other analyst.	Students evaluate models developed by other teams and the instructor checks the validity of the evaluation
--	--

Study subject structure

Part	CP	ECTS	Hours per Week			Tests		
			Lectures	Practical	Lab.	Test	Exam	Work
1.	4.0	6.0	2.0	1.0	1.0		*	