



## RTU Course "CASE Tools for Object-Oriented System Development"

12306 Lietišķo datorzinātņu katedra

### General data

Code	DPI504
Course title	CASE Tools for Object-Oriented System Development
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
Academic staff	Pavlova Nataļja Gorbiks Oļegs
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	The task of software development. Stages of software development. Organization of software development based on object-oriented technology. Software architecture. Unified software development by Rational unified process. Classification of software development tools. Analysis of CASE-tools for software development from the point of view of stage supporting level.
Goals and objectives of the course in terms of competences and skills	The goal is to gain experience in performing different stages of object-oriented software development and in using software development tools. Objectives: - to learn to perform different stages of software development process; - to learn to use different tools for software development; - to learn to prepare the report about the performed activities and to deliver a presentation about them; - to learn to evaluate the work performed by other students in a certain software development phase.
Structure and tasks of independent studies	Students work individually on performing a concrete task and demonstrate their results to other students at the workshop sessions.
Recommended literature	1. Jacobson I., Booch G., Rumbaugh J. The Unified Software Development Process, Addison Wesley, 1999 2. Schach S. R. Object-Oriented & Classical Software Engineering, McGraw-Hill International Edition, Seventh Edition, 2007 3. Pressman R. S. Software Engineering A Practitioner's Approach, McGraw-Hill International Edition, Sixth Edition, 2005 4. Cockburn A. Agile Software Development. The Cooperative Game, Second Edition, Pearson Education Inc, 2007
Course prerequisites	It is preferred that students have initial knowledge in object-oriented technology for system modeling and implementation.

### Course outline

Theme	Hours
Introduction to the course and course requirements.	4
Software Development Life-Cycle.	4
Organization of software development based on object-oriented technology.	4
Rational Unified Process (RUP) - unified process for software development.	8
Software development phase "Inception" and its supporting tools.	8
Software development phase "Elaboration" and its supporting tools.	8
Software development phase "Construction" and its supporting tools.	8
Tools supporting Model Driven software development.	12
Software development phase "Transition" and its supporting tools.	8

### Learning outcomes and assessment

Learning outcomes	Assessment methods
Is able to perform inception phase of software development process.	The student develops and demonstrates a presentation about the performance of the required task. Other students and the instructor evaluate it.
Is able to perform elaboration phase of software development process.	The student develops and demonstrates a presentation about the performance of the required task. Other students and the instructor evaluate it.

Is able to perform construction phase of software development process.	The student develops and demonstrates a presentation about the performance of the required task. Other students and the instructor evaluate it.
Is able to perform transition phase of software development process.	The student develops and demonstrates a presentation about the performance of the required task. Other students and the instructor evaluate it.
Is able to use tools to support different stages of software development process.	The student develops and demonstrates a presentation about the performance of the required task. Other students and the instructor evaluate it.
Is able to use tools for model-driven software development.	The student develops and demonstrates a presentation about the performance of the required task. Other students and the instructor evaluate it.

***Study subject structure***

Part	CP	ECTS	Hours per Week			Tests		
			Lectures	Practical	Lab.	Test	Exam	Work
1.	4.0	6.0	2.0	0.0	2.0		*	