



RTU Course "Evolution of Object-Oriented Software"

12306 Lietišķo datorzinātņu katedra

General data

Code	DPI503
Course title	Evolution of Object-Oriented Software
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	Rusakovs Pāvils
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	Students learn the implementation of object-oriented approach (OOA) principles in several programming languages. Both fundamental OOA principles (abstraction, encapsulation, inheritance, polymorphism) and some additional principles (parallelism, typing) are discussed. In the focus of attention there are three object-oriented programming languages: Ada, Java, Python. The course includes a survey of some specific questions: applet development, server script development, creation of complex data structures and others.
Goals and objectives of the course in terms of competences and skills	To understand implementation features of the object-oriented approach in several object-oriented programming languages, to be acquainted with notions and principles of parallel programming, to learn the development of applets and event handling in Java, to know about possibilities of creating server scripts.
Structure and tasks of independent studies	The foundation of the course is lectures and laboratory works. Instructor offers some additional themes for public discussion. Some additional experiments can be implemented in the laboratory.
Recommended literature	<ol style="list-style-type: none"> 1. Grady Booch. Object Oriented Analysis and Design With Applications. Addison-Wesley Publishing Company, 1994. 2. John Barnes. Programming in Ada 2005. Addison-Wesley, 2006. 3. John Barnes. Programming in Ada. Addison-Wesley, 1995. 4. Jānis Osis. Programmēšanas valoda Ada. Rīgas Tehniskā universitāte, Rīga, 1993. 5. Ira Pohl. Object-Oriented Programming Using C++. Addison-Wesley Publishing Company, 1997. 6. Patrick Naughton, Herbert Schildt. Java 2: The Complete Reference, Third Edition. Osborne Publishing, 3rd edition, 1999. 7. Herbert Schildt. Java 7: A Beginner's Guide, Fifth Edition. McGraw-Hill Osborne Media, 5 edition, 2010. 8. Bruce Eckel. Thinking in Java (4th Edition). Prentice Hall, 4 edition, 2006. 9. Harvey M. Deitel, Paul J. Deitel. Java How to Program, 7th Edition. Prentice Hall, 7 edition, 2007. 10. Mark Lutz. Learning Python: Powerful Object-Oriented Programming. O'Reilly Media, 4th edition, 2009. 11. Mark Summerfield. Programming in Python 3: A Complete Introduction to the Python Language (2nd Edition). Addison-Wesley Professional, 2 edition, 2009.
Course prerequisites	Principles of object-oriented programming, construction of algorithms.

Course outline

Theme	Hours
History, basic principles and evolution of the Ada language	2
Input/output of information in Ada. Generics and parametric polymorphism	4
Abstraction, encapsulation and inheritance in Ada. Exception handling	4
Parallel programming in Ada. Tasks. Rendezvous	4
Protected types in Ada. Synchronization. Semaphores	4
History, basic principles and evolution of Java language	2
Abstraction, encapsulation and inheritance in Java	4
Processing and sorting arrays in Java. Interfaces. Simulation of multiple inheritance	4
Reflection in Java. Run-Time Type Identification (RTTI)	4

Exception handling in Java	2
Usage of java.util package in Java (class, interfaces, collections)	4
Other aspects of programming in Java: classes-shells, parameter passing	2
Parallel programming in Java: basic principles. Threads, synchronization	4
Parallel programming in Java: additional features. The java.util.concurrent package	4
Programming applets in Java. Event handling. Layout managers	4
Conception and basic principles of the Python language. Data types, data type conversion, operators	2
Lists, tuples, arrays, dictionaries in Python. Multilevel structures. Exception handling	4
Subroutines in Python. File processing. Lambda expressions	2
Classes and objects in Python. Graphic features	4

Learning outcomes and assessment

Learning outcomes	Assessment methods
Is able to implement four basic principles of object-oriented programming in the Ada, Java and Python languages.	Execution and defence of laboratory works.
Is able to understand differences between Ada'83, Ada'95 and Ada'2005 standarts.	Execution and defence of laboratory works.
Is able to create parallel programs in Ada and Java.	Execution and defence of laboratory works.
Is able to use collections in Java.	Execution and defence of laboratory works.
Is able to develop Java applets and graphical programs.	Execution and defence of laboratory works.
Is able to develop multifunctional scripts in Python.	Execution and defence of laboratory works.
Is able to understand basic principles of graphical programs in Python .	Execution and defence of laboratory works.
Is able to demonstrate theoretical and practical knowledge of object-oriented features in several programming languages to another specialist.	Written examination.

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		*	