

**RTU Course "Functional Programming"**

12308 null

**General data**

Code	DIP330
Course title	Functional Programming
Course status in the programme	Compulsory/Courses of Limited Choice
Course level	Undergraduate Studies
Course type	Academic
Field of study	Computer Science
Responsible instructor	Marina Uhanova
Volume of the course: parts and credits points	1 part, 2.0 Credit Points, 3.0 ECTS credits
Language of instruction	LV, EN
Annotation	Functional programming is an alternative to imperative programming languages. Lisp - the most popular functional programming language. The language overview. Basic functions. Recursion. Declaration of new functions. List processing functions. Predicates. Conditional expressions. Recursive functions. Recursive list processing. Sublists and recursion. Recursive predicates. Logical operators. PROG-expressions. List creation. Predicates written using PROG. Recursion in PROG-expressions. Function joining. Evaluation rule. Evaluating function EVAL. Programs as data. Input-output of data. Practical usage of LISP language.
Goals and objectives of the course in terms of competences and skills	Objective of the course - to understand and learn a new programming paradigm based on problem-solving "from the end to the beginning". Tasks - to learn how to work with objects represented in the form of lists, learn the concept of recursion and be able to apply it to processing of non-determined structures, define new functions, realize interaction between objects with complex regulations. After completing the course students will have the competencies and skills in the use of non-imperative programming.
Structure and tasks of independent studies	The instructor during the lectures presents theoretical concepts and examples of their practical implementation to the students. Students must independently in accordance with the individual tasks carry out three laboratory works and prepare a report for each of them. Reports should include: task, software source code, proof of programs operational capabilities and conclusions. 1st work is on recursive list processing, 2nd - on modelling tables and 3rd - on interactive intellectual game.
Recommended literature	1.Šitikovs V. Programmēšanas valoda LISP. Lekciju konspekts//RTU,2002,36 lpp. 2.Graham P. On Lisp. Advanced Techniques for Common Lisp//Prentice Hall, 1993, 432 p. (free download from <a href="http://www.paulgraham.com/onlisp.html">http://www.paulgraham.com/onlisp.html</a> ) 3.Schwarz M. LISP Tutorial / Internet - <a href="http://cs1.cs.nyu.edu/phd_students/schwarz/NLCP/lisp.html">http://cs1.cs.nyu.edu/phd_students/schwarz/NLCP/lisp.html</a>
Course prerequisites	Imperative programming language(s).

**Course outline**

Theme	Hours
Introduction. LISP language description	2
Recursion	2
Definition of new functions	2
Functions for list processing	2
Predicates	2
Conditional expressions	1
Recursive functions	2
Logical operators	1
Recursive functions for two lists	2
Functions for type recognition	1
PROG - expressions	4
Functions executions	1
Predicates written using PROG	2
Recursions in PROG-expressions	2
Function combination	1
Rules of calculations. Computational function	2
Programs as data	3

**Learning outcomes and assessment**

Learning outcomes	Assessment methods
Knows and understands the content of the course. The student can apply functional programming as an alternative to imperative programming language, including:	the obtained skills are tested at the exam
- Define new functions for list processing	laboratory work on recursive list processing
- Provide multidimensional recursive processing	laboratory work on modelling tables
- Provide multi-user interactive interface and simulate collaborative virtual member	laboratory work on interactive intellectual game

***Study subject structure***

Part	CP	Hours per Week			Tests		
		Lectures	Practical	Lab.	Test	Exam	Work
1.	2.0	1.0	0.0	1.0		*	