Study programme(s): Informatics (IM), Teaching Informatics (IC)

Level: master

Course title: Compiler construction 2 (code IA422)

Lecturer: Vladimir Kurbalija

Status: elective

ECTS: 7

Requirements: Compiler construction 1

Learning objectives

Training students to design and build a compiler for declarative programming languages.

Learning outcomes

Minimum: Students should be able to write a parser for a subset of declarative programming language based on the given specification.

Optimal: Students should be able to develop software for parsing and interpretation of the declarative programming language, according to the given specifications.

Syllabus

Theoretical instruction

Advanced methods of syntax analysis, the application of LL and LR grammars. Advanced methods of semantic analysis, code generation and optimization. The construction of a compiler for declarative programming language. The main differences between declarative and procedural programming languages, from the point of realization. The mathematical basis of declarative programming languages.

Practical instruction

Source code translator, intermediate code and translator to the language of abstract machines. Abstract machine for the implementation of a declarative programming language.

Literature

Recommended:

1. Мирјана Ивановић, *Функционално програмирање,* скрипта, Универзитет у Новом Саду, Природно-математички факултет, Департман за математику и информатику, Нови Сад, 2000.
2. Andrew W. Appel, *Modern Compiler Implementation in Java*, Cambridge University Press, 1998.
3. V. Aho, R. Sethi, J. D. Ullman "Compilers, Principles, Techniques and Tools, Addison-Wesley, 1985

Alternative:

1. V. Aho, J. D. Ullman: "Principles of Compiler Design", Addison-Wesley, 1977

J. P. Trembley, P. G. Sorenson, "The Theory and Practice of Compiler Writing", McGraw Hill, 1985

Weekly teaching load				Other:
Lectures: 2	Exercises: 3	Other forms of teaching:	Student research:	

Teaching methodology

Lectures are held using conventional teaching methods with the use of the video beam. The main functioning principles of the compiler are illustrated through examples. During the lectures, the students’ knowledge is tested through three tests. On exercises, the programming languages Java and Modula-2 are used for compiler implementation. The software package "Svetovid", developed at the Department of Computer Science, is used to prevent cheating. During the

exercises, students’ knowledge is tested through the development of three projects covering the appropriate compiler phases. In the oral part of the exam, the student demonstrates understanding of the main principles of a compiler for the declarative programming languages.			
Grading (maximum number of points 100)			
Pre-exam obligations	**points**	**Final exam**	**points**
Active participation in lectures	**6**		
Practical instruction	**6**	Oral exam	40
Colloquia	**8, 8, 8**		
Seminar(s)	**8, 8, 8**		