Study programme(s): Informatics (IM)

Level: master

Course title: Software evolution (IB331)

Lecturer: Miloš M. Radovanović

Status: elective

ECTS: 7,5

Requirements: none

Learning objectives

The goal of this course is to present and critically analyze the current techniques for software evolution and provide students with practical experience in using a set of tools known as FermaT.

Learning outcomes

Minimal: Students should be able to critically evaluate the current basics of software evolution, adopt reengineering techniques for software migration and abstraction, and develop an integrated approach for software evolution life cycles.

Desirable: Students should be able to demonstrate the ability to apply transformation rules in order to migrate a temporally and economically critical system, and acquire practical experience in the use of an industrial-strength tool such as FermaT.

Syllabus

Theoretical instruction

Theoretical basis and classifications of software evolution, evolution within software development life cycles, Lehman's laws of evolution, software comprehension techniques, abstraction, slicing, refactoring, Wide Spectrum Language (WSL) and software transformation, transformation theory and its implementation, tools, and migration of software.

Practical instruction

Acquaintance with tools such as FermaT and analysis of study examples.

Literature

1. H. Yang, M. Ward. *Successful Evolution of Software Systems*. Artech House, 2003
2. M. Fowler. *Refactoring: Improving the Design of Existing Programs*. Addison-Wesley, 1999

Weekly teaching load				Other: 0
Lectures: 2	Exercises: 3	Other forms of teaching: 0	Student research: 0	

Teaching methodology

Lectures are held using classical presentation methods involving a video-beam. Classical teaching methods involving a video-beam are used to analyze study examples in exercises. The principles of application of studied topics are practiced on the computer, through acquaintance with the use of recommended tools. Students complement their knowledge through research into selected topics, and are tested through a written test, solution of practical problems, and preparation of a seminar paper that is defended at the end of the course.

Grading (maximum number of points 100)

Pre-exam obligations	**points**	**Final exam**	**points**
test	30	seminar paper	50
practical problems	20		