Study programme(s): Computer Science

Level: Bachelor Academic Studies

Course title: Object-Oriented Programming 2

Lecturer: Miloš M. Radovanović

Status: elective

ECTS: 6

Requirements: Object-oriented programming 1

Learning objectives

Acquainting the students with advanced methodologies and techniques of object-oriented programming, and enabling them to apply proven solutions and modern tools in the development of a wide range of complex applications.

Learning outcomes

Minimum: At the end of the course it is expected from a successful student to demonstrate the capability for understanding and analyzing complex problems, as well as designing and implementing advanced solutions in a concrete object-oriented programming language.

Desirable: At the end of the course it is expected from a successful student to demonstrate the capability for understanding, analyzing and defining complex problems on a logically-founded basis, as well as creative design and implementation of advanced solution using the most recent techniques of the object-oriented programming paradigm.

Syllabus

Theoretical instruction

Advanced object-oriented language features. Generics. Input/output (I/O). Multithreading. Advanced solutions for graphical user interfaces (GUIs). Lambda expressions and basics of functional programming, streams. Elements of network programming. Distributed programming: remote method invocation, serialization, reflection, class loaders. Web services. Advanced object-oriented design. Study examples using object-oriented programming.

Practical instruction

Practicing the understanding of object-oriented design. Generics, I/O, multithreading, GUIs, lambda expressions, streams, network programming, distributed programming, Web services. Complex application written in a concrete object-oriented language. Testing complete solutions, tools, discussion of possibilities of application, etc.

Literature

Recomended

1. J. Bloch. Effective Java. Addison-Welsey, 2nd edition, 2008.
2. R. Gallardo, S. Hommel, S. Kannan, J. Gordon, S. B. Zakhour. The Java Tutorial: A Short Course on the Basics. Addison-Wesley, 6th edition, 2015.

Weekly teaching load				
Lectures:	Exercises:	Practical Exercises:	Student research:	Other:
2	2	2	0	0

Teaching methodology

Lectures are organized using classic teaching methods with use of a projector. Advanced principles of object-oriented programming are explained and illustrated with appropriate examples. At theoretical exercises, the explained principles are practiced, illustrative complete examples are analyzed, as well as real-world projects, and own solutions are modeled. During practical exercises, students independently apply the acquired techniques by developing different applications, the complexity and capability of which increases during the semester.

Students' knowledge is checked through solving practical problems (first individually, and later in small groups), and through elective written tests. Practical exercise hours are planned in such a way to alternate between exercises where certain principles and techniques are practiced, solutions discussed, etc., with the help of the teaching assistant, and exercises where students work on projects and their performance is monitored and evaluated in detail. At the oral exam the student demonstrates understanding of advanced methodologies of object-oriented programming.

Grading method (maximal number of points 100)

Pre-exam oblications	**points**	**Final exam**	**points**
practical exercises – individual problems	25	oral examination (obligatory)	20-40
practical exercises – group problems	35	tests (elective)	0-20