ALFA BK UNIVERSITY

Faculty of Mathematics and Computer Sciences
(ALFABK-FMCS)

STUDY PROGRAMME

Computer Science

MASTER ACADEMIC STUDIES

Course:	**Research methods**
Course id:	
Number of ECTS: 6	
Teacher:	**Lazar S. Kopanja**
Course status:	Mandatory first year, first semester

Precondition courses: Mathematical knowledge from the previous level of study

Educational goal

Introduction to the basic elements of professional and scientific work, research methods in the field of computer science, especially with the methodology of writing and presenting research results in the master thesis.

Educational outcomes (acquired knowledge):

Upon completion of the course, the student is introduced to the basic methods of scientific and professional work in the field of computer science and informatics, is trained in the use of theoretical algorithms and the results of experimental analysis in computer simulation.

Course content/structure

Theory classes:

Disciplinary and multidisciplinary approaches to research. Research methods and algorithms. Analysis and synthesis. Method of abstraction and concretization. Method of generalization and specialization. Inductive and deductive method. Observation. Experiment. Counting. Measurement. Fundamentals of statistical procedures for scientific analysis of experimental data. Methods of causal induction. Deductive method. Axiomatic method. Modelling method. Classification method. Comparative method. Generic method. Method of description. The method of ideal types. General system theory as a scientific method. Case study method. Content analysis method. Special methods of computer science. Writing professional and / or scientific paper: structure, content, composition, conclusion, references. Technique of making and defending a master's thesis.

Practical classes: Exercises, Other forms of teaching, Study research work: Work in a computer room in a simulation analysis of the considered model.

Literature

- Zoran Popović: *Kako napisati i publikovati naučno delo*, Akademska misao, Beograd, 1999.
- Richie J., Lewis J., (2003): *Qualitative Research Practice*, Sage Publications.
- Ristić, Ž., (2006): „O istraživanju, metodu i znanju“, Institut za pedagoška istraživanja, Beograd

Number of active teaching classes (weekly)	Lectures: 2	Practical classes: 2

Teaching methods

Methods of working with text, verbal methods, discussion methods, workshops, seminars, dialogue methods, methods of practical activities, written papers, group seminar papers and online discussions.

Knowledge evaluation (maximum 100 points)

Pre-examination obligations:	**Final exam:**
Lecture attendance and activity: 10 Practical classes Colloquium exam: 25 Seminary: 25	Theoretical part of the exam: 40

Course:	**Theory of Algorithms, Automata and Languages**
Course id:	
Number of ECTS: 6	
Teacher:	**Duško Bogdanić**
Course status:	Mandatory first year, first semester

Precondition courses: none

Educational goal

The aim of the course is to acquire general and specific knowledge of the theory of formal languages and automata through the analysis and application of specific algorithms.

Educational outcomes (acquired knowledge):

Upon completion of the course, the student has acquired general and specific knowledge about generators and recognizers of formal languages, knowledge about automata and is able to apply the acquired knowledge algorithmically in solving new problems.

Course content/structure

Theory classes:

Concepts of decision theory and computability. Formal models of algorithmic computation. Turing machine. Recursive functions. Turing-computable and partially recursive functions. Church's thesis. Classes of solvable and unsolvable problems. Language and formulas of predicate logic. Valid predicate calculus formulas. First-order predicate calculus. Erban's theorem. Resolution method and method of analytical boards in predicate logic. Concepts of formal language theory. Presentation of language, grammar. Regular languages and finite automata. Context-free languages and push automata. The relationship between automata and formal languages.

Practical classes:

Application and practice of the stated theoretical concepts and algorithms on solving specific tasks and problems.

Literature

- R. S. Madrasz, Crvenkovic, S. Uvod u teoriju automata i formalnih jezika, Novi Sad, 1995.
- Ognjanović, Z., Krdžavac, N. (2004): «Uvod u teorijsko računarstvo“, Fakultet organizacionih nauka u Beogradu
- Hopcroft, J., Motwani, R.,Ulman J., (2007): „Introduction to Automata Theory, Languages and Computations“, Addison Wesley (3rd edition)
- Kozen, D.C., (1997): „Automata and Computability“, Springer
- Spasić, I., Janičić, P., (2000): «Teorija algoritama, jezika i automata: zbirka zadataka“, Matematički fakultet u Beogradu

Number of active teaching classes	Lectures: 2	Practical classes: 2

Teaching methods

Frontal, group, individual and practical method.

Knowledge evaluation (maximum 100 points)

Pre-examination obligations:	**Final exam:**
Lecture attendance and activity: 10 Practical classes Colloquium exam: 25 Seminary: 25	Theoretical part of the exam: 40

Course:	**Data Mining**
Course id:	
Number of ECTS: 6	
Teacher:	**Marija V. Paunović, Danijela Karaklić**
Course status:	Mandatory first year, first semester

Precondition courses: Mathematical knowledge from the previous level of study

Educational goal

The aim of the course is to present the application of mathematical methods in the field of linear algebra, differential and difference equations, probability and statistics, and others, as well as special methods in the field of shape recognition, machine learning, etc., to find useful data in data sets that are too large or too diverse, in which even the existence of such data is not obvious; to get acquainted with the ways of graphical - visual representation of data sets in order to highlight and view useful data in them.

Educational outcomes (acquired knowledge):

Upon successful completion of the course, the student will be able to:

- applies methods of linear and nonlinear transformation and projection in vector spaces to large numerical data sets;
- applies methods in the field of statistics and statistical machine learning, in order to remove uncertainties and noise from numerical data sets;
- designs and implements programs for searching and recognizing shapes on given data sets;
- purposefully displays data in visual, graphic or animated form.

Course content/structure

Theoretical classes

Theoretical classes. Vector spaces, dimensionality, functions, mappings, projections; differential and difference equations, notion of path, convergence, sensitivity of the solution to the parameter; statistical methods of interpolation, machine learning, optimization of model parameters; graphic presentation and visual animation of multidimensional data;

Practical classes.

Computational exercises. Laboratory exercises. Project.

Literature

1. Glenn J. Myatt: Making Sense of Data: A Practical Guide to Exploratory Data Analysis and Data Mining : Wiley-Interscience, 2006
2. Glenn J. Myatt, Wayne P. Johnson: Making Sense of Data II: A Practical Guide to Data Visualization, Advanced Data Mining Methods, and Applications: Wiley, 2009
3. Glenn J. Myatt, Wayne P. Johnson: Making Sense of Data III: A Practical Guide to Designing Interactive Data Visualizations : Wiley,2011
4. Brian S. Everitt, Graham Dunn: Applied Multivariate Data Analysis : Wiley, 2009
5. Charu C. Aggarwal: Data Mining The Textbook, Springer, 2015.
6. Xindong Wu, Vipin Kumar (eds.): The Top Ten Algorithms in Data Mining, CRC Press, 2009.
7. Pang-Ning Tan, Michael Steinbach, Anuj Karpatne, Vipin Kumar: Introduction to Data Mining, 2nd ed, Pearson Education, 2019.

Number of active teaching classes (weekly)	Lectures: 2	Practical classes: 2

Teaching methods

Classical teaching methods (frontal, group method, laboratory-experimental methods) with the use of modern technology. The seminar paper is performed in the form of an independent or group development project.

Knowledge evaluation (maximum 100 points)

Pre-examination obligations:	**Final exam:**
Activity:10; Exercise attendance: Colloquium exam: 25; Seminar paper: 25;	Writing exam: 20 Oral part of exam: 20

<table>
<tr><td>Course:
Course id:
Number of ECTS: 6</td><td colspan="3">Software testing</td></tr>
<tr><td>Teacher:</td><td colspan="3">Milan Đorđević</td></tr>
<tr><td>Course status:</td><td colspan="3">Mandatory, first year, first semester</td></tr>
<tr><td colspan="4">Precondition courses: none</td></tr>
<tr><td colspan="4">Educational goal
Introduction to basic software testing concepts. Introducing students to the importance of software testing throughout the software development process. Application of various software tools for writing automatic tests in the Java programming language.</td></tr>
<tr><td colspan="4">Educational outcomes (acquired knowledge):
Upon completion of the course, the student is expected to be able to understand the role and basic settings of software testing, to be able to select and apply appropriate techniques according to technology and application domain, to design and implement test cases, and to understand their role in teamwork. testing activities.</td></tr>
<tr><td colspan="4">Course content/structure
Theory classes:
Basic concepts. Manual and automated testing. Black box techniques (division into equivalence classes, limit values ...). White box techniques (code coverage). Code coverage techniques based on control flow. Data flow techniques. Unit testing. Integration testing. System testing. OO testing. Testing parallel and web applications. Testing process management. Risk assessment. Using the JUnit software tool. Using the Selenium software tool.
Practical classes:
Auditory exercises that illustrate certain concepts and techniques covered in lectures. Tools introduction exercises. A practical project in the field of testing the given software that the student creates independently.</td></tr>
<tr><td colspan="4">Literature
● Software Testing – A Craftsman Approach, Paul Jorgensen, 1995.
● The Art of Software Testing, G. Myers, John Wiley and Sons, 2001, second edition
● Foundations of Software Testing, A. Mathur, Addison-Wesley Professional. 2008.
● Miodrag Živković (2018): Testiranje softvera, Univerzitet Singidunum</td></tr>
<tr><td>Number of active teaching classes</td><td>Lectures: 30</td><td colspan="2">Practical classes: 30</td></tr>
<tr><td colspan="4">Teaching methods
Lectures, auditory exercises, laboratory exercises, independent development of two projects.</td></tr>
<tr><td colspan="4">Knowledge evaluation (maximum 100 points)</td></tr>
<tr><td colspan="2">Pre-examination obligations:
Lecture attendance and activity: 10
Practical classes
Colloquium exam: 30
Project: 30</td><td colspan="2">Final exam:
Theoretical part of the exam: 30</td></tr>
</table>

Course: **Course id: 19.MR0010** **Number of ECTS: 6**	**Numerical methods**
Teacher:	Ivan Pavkov
Course status:	Elective

Precondition courses: None

Educational goal

Acquisition of general and professional knowledge in selected areas of numerical analysis.

Educational outcomes (acquired knowledge):

Upon completion of the course, the student has basic knowledge of selected chapters of numerical analysis. He is able to follow courses in professional fields in which the concepts and techniques he has mastered are applied and to identify problems to which he can apply the acquired knowledge. He is able to solve practical tasks from the exposed areas using the Matlab software package and to evaluate the reliability of the obtained results.

Course content/structure

Theoretical classes: Hermit's interpolation. Spline interpolation. Approximation of functions. Discrete Fourier transform. Cauchy problems for ordinary differential equations. Runge-Kuta type methods. Multilayer methods. Boundary value problems for ordinary differential equations. Shooting method. Finite difference method. Variation methods. Finite element method. Numerical methods for solving integral equations. Basic ideas on solving partial differential equations. Finite difference method. Variation methods. Finite element method. Stability. Economy.
Practical teaching: Tasks from the stated theoretical areas.

Literature

1) B. Jovanović, D. Radunović: Numerical Analysis, Faculty of Mathematics, Belgrade 2003.
2) D. Radunović: Numerical Methods, Academic Thought, 2004.

Number of active teaching classes	Lectures: 30	Practical classes: 30

Teaching methods

Frontal, group

Knowledge evaluation (maximum 100 points)

Pre-examination obligations: Colloquium exam: 50 Lecture attendance: 5 Exercise attendance:5	**Final exam:40**

Course:	**Mathematical foundations of coding theory**
Number of ECTS: 6	
Teacher:	Ivan Pavkov
Course status:	Elective

Precondition courses: None

Educational goal

Acquisition of basic knowledge of information and coding theory in communication systems and mathematical foundations of coding and decoding theory.

Educational outcomes (acquired knowledge):

Upon completion of the course, students are capable to understand professional courses in which information theory and coding are studied.

Course content/structure

Motivation and history of coding theory. Basic concepts of coding theory. The concept of communication channel. Noisy communication channel. Redundancy. Shannon communication system (encoder, communication channel, decoder). Code function features.

Block codes. Code space. Code length. Codeword. Distance of codewords. Good code. Shannon's theorem on the existence of good code.

Linear codes. Distance of codewords of linear code. Generator matrix of linear code. Coding of linear codes. Decoding of linear codes. Dual and self-dual code. Syndromic decoding. Hamming code. Encoding and decoding of Hamming codes. Reed-Solomon codes.

Literature

Hall J. I. (2010), *Notes on Coding Theory*, Department of Mathematics Michigan State University

Huffman W. C., Pless V. (2003), *Fundametals of Error Correcting Codes*, Cambridge University Press, New York

van Lint J. H. (1992), *Introduction to Coding Theory*, Springer-Verlag Berlin Heidelberg

Number of active teaching classes (weekly)	Lectures: 2	Practical classes: 2

Teaching methods

Classes are conducted frontally in a group

Knowledge evaluation (maximum 100 points)

Pre-examination obligations:	**Final exam: 40**
Activity:10; Exercise attendance: 10; Colloquium exam: 40; Seminary:	

Course: **Course id:** **Number of ECTS: 7**	**Microprocessor software**
Teacher:	Dejan Đukić
Course status:	Elective, , first year, second semester

Precondition courses: none

Educational goal

The aim of the course is to introduce and train students to program microcontrollers in assembler and C language. In addition, the aim of the course is to enable students to understand the function of the accompanying peripherals, the connection with the microprocessors of the MCS-51 family, as well as their software control.

Educational outcomes (acquired knowledge):

After taking the course, students will be able to independently design simple hardware modules based on modern microcontrollers compatible with the MCS-51 family, to use software tools for program development and development to work with microcontrollers.

Course content/structure

Theory classes:

History of microprocessors and microcontrollers. Architecture of microcomputer systems. Classifications. Scope. Microcontrollers. An overview of the families of modern microcontrollers. Microcontrollers with improved performance. Embedded systems. Higher programming languages for microcontrollers. C51 compiler. A51 assembler. BL51 linker.OHS51object-hex converter.

Practical classes:

Computational exercises. Laboratory exercises. Demonstration exercises.

Literature

- Karakanov, Z., Christensen, K., Embedded Systems Design with 8051 Microcontrollers,Marcel Dekker, New York, 1999.
- Keil, Cx51 Compiler, Keil Elektronic, 2000.
- Milivojević, Z., Mikrokontroleri - Arhitektura 8051, Punta, Niš, 2005.

Number of active teaching classes	Lectures: 3	Practical classes: 3

Teaching methods

Knowledge evaluation (maximum 100 points)

Pre-examination obligations:	**Final exam:**
Lecture attendance and activity: 10 Practical classes: 20 Colloquium exam: 30 Seminary: 10	Theoretical part of the exam: 30

Course:	**Software design**
Course id:	
Number of ECTS: 7	
Teacher:	**Nenad Gligorić**
Course status:	Elective

Precondition courses: None

Educational goal

The goal of this course is to provide knowledge in the domain of the programme support for large business projects; to explain methodology for modelling of business processes and providing education in the domain of requirements definition and delivery of the quantitively software projects; to explain software design, planning, organisation and execution using latest software development methods.

Educational outcomes (acquired knowledge):

The students completing the Software design course will:

- have the ability to apply the process of problem definition and description of business process and the need for use of information technology
- describe methods for execution of the software project
- design software taking into account defined requirements and arranged commitments
- estimate and plan design and execution of the organised management support for the project

Course content/structure

Theoretical lectures

Development of software for business systems, scope, components and systems activities; team composition, roles of the team members; methods of software project management, classic, agile; software reliability, methods for testing of software systems; approached for project design, methodologies for development of project plan; design of programme support for decentralized systems, clients, servers, cloud infrastructure systems.

Practical

Computer exercises. Laboratory course. Development project.

Literature

1. Len Bass, Paul Clements, Rick Kazman : Software Architecture in Practice (3rd Edition) Addison-Wesley, 2012
2. Martin Fowler, Rebecca Parsons : Domain Specific Languages : Addison-Wesley, 2010
3. Mike Cohn : Agile Estimating and Planning : Prentice Hall, 2005
4. Nicolai M. Josuttis : SOA in Practice: The Art of Distributed System Design : O'Reilly, 2009

Number of active teaching classes	Lectures: 45	Practical classes: 45

Teaching methods

Lectures and laboratory practice are based on frontal, group methods, as well as using laboratory-experimental methods facilitating information communication technologies.

Knowledge evaluation (maximum 100 points)

Pre-examination obligations: Activity:10; Exercise attendance: 20; Colloquium exam: 30; Seminary: 10;	**Final exam: 30**

Course:	**Translators and interpreters**
Course id:	
Number of ECTS:	
Teacher:	Duško Bogdanić
Course status:	Elective, first year, second semester

Precondition courses: Theory of Algorithms, Automata and Languages

Educational goal

The main goal of this course is to introduce students to the basic tasks of different phases of compilation, as well as enabling students to participate in larger projects and implement compilers for simpler procedural and object-oriented languages.

Educational outcomes (acquired knowledge):

Upon completion of the course, the student has mastered concepts related to theoretical and practical aspects of finite state automation theory and their various applications, mastered concepts related to the application of formal language theory in analysis and synthesis of programming languages, as well as methods for solving specific translation problems and their implementation, gained experience in tools used in lexical analysis of programming languages, as well as the application of regular expressions through various scripting languages.

Course content/structure

Theory classes:

Compiler structure and functions. Techniques for specifying programming language rules. Syntax diagrams, Bakus-Naur form and extended Bakus-Naur form for grammar specification of programming languages. Context-free grammars, LL, LR and attributive grammars. Lexical analysis. Parsing. Top-down analysis. Grammar transformations, normal forms. Recursive descent parsing. semantic analysis (type checking) and maintenance of the symbol table, code generation (using a virtual machine). Description of the complete compiler implementation for a simple procedural (with the addition of some basic object-oriented principles) programming language. Intercode generation. Optimization. Code generation.

Practical classes:

Graphical representation of automata and expression of appropriate languages by grammar. Implementations of automata in lexical analysis of programming languages using the constructor of lexical analyzers Lex. Implementations of lexical analysis of programming languages using the methods of downward syntactic analysis and upward syntactic analysis. Compiler constructions for a given programming language using the Yacc software system.

Literature

- Vitas, D., (2006): „Prevodioci i interpretatori“, Matematički fakultet u Beogradu
- Aho, A., Sethi, R., Ulman J., (2007): „Compilers - Principles, Techniques and Tools“, Addison Wesley (2nd edition)
- Bennett, J.P.,(1990): „Introduction to compiling techiques: a first course using ANSI C, LEX and YACC“, McGraw-Hill
- Levine et al, J.,(1992): „Lex & Yacc“, O'Reilly Associates

Number of active teaching classes	Lectures: 30	Practical classes: 45

Teaching methods

Lectures and exercises use frontal, group methods as well as laboratory-experimental teaching methods with the use of modern technology.

Knowledge evaluation (maximum 100 points)

Pre-examination obligations:	**Final exam:**
Lecture attendance and activity: 10 Practical classes: 50 Colloquium exam: Seminary:	Theoretical part of the exam: 40

Course: **Course id:** **Number of ECTS: 7**	**Knowledge presentation and automatic reasoning**
Teacher:	Dragiša Žunić
Course status:	Elective, first year, second semester

Precondition courses: none

Educational goal

Acquisition of basic knowledge of mathematical logic; presentation of formal systems in logical systems with an emphasis on the theory and practice of automated reasoning. Developing abstract thinking and getting acquainted with the techniques of formal reasoning.

Educational outcomes (acquired knowledge):

Students are introduced to the basics of logic, formal systems and formalisms for the representation of knowledge and other systems, as well as the principles and tools for (partially) automated reasoning about systems. The student's ability to use the acquired knowledge and skills in further education and practice.

Course content/structure

Theory classes:

Basic concepts of logic. Classical logic. Intuitionist logic. Axiomatic system, natural deduction, sequential calculus. Basic concepts of predicate calculus. Basic concepts of semantics of logical systems. DPLL method, board method, resolution method. Computational interpretations of logic: lambda calculus and combinator theory. Basic accounts without types and with types. Curry-Howard correspondence between logical systems and formal accounts. Applications of logical theories in knowledge modelling. Automatic proof of theorems and the most famous tools.

Practical classes:

Tasks from the mentioned areas of theoretical teaching.

Literature

- J. Harrison, Handbook of practical logic and automated reasoning, Cambridge University Press, 2009.
- P. Janičić, Matematička logika u računarstvu, Matematički Fakultet u Beogradu, 2007.
- M. Huth and M. Ryan, Logic in computer science: modelling and reasoning about systems, Cambridge University Press, 2012.
- Z. Ognjanović i S. Gilezan, Uvod u teorijsko računarstvo, Fakultet Tehničkih Nauka, Novi Sad, 2014.
- Z. Petrić, Uvod u Matematičku Logiku (skripta), Matematički Institut SANU, 2016.

Number of active teaching classes	Lectures: 30	Practical classes: 45

Teaching methods

frontal, group

Knowledge evaluation (maximum 100 points)

Pre-examination obligations:	**Final exam:**
Lecture attendance and activity: 10 Practical classes Colloquium exam: 50 Seminary:	Written exam: 20 Oral exam: 20

Course:	**Professional practice**	
Course id:		
Number of ECTS: 3		
Teacher:	All teachers in the study program, guest lecturers, lecturers out of employment	
Course status:	Mandatory	
Precondition courses: Research methods		
Educational goal Gaining experience in practical work in a real work environment.		
Educational outcomes (acquired knowledge): Upon completion of the internship, the student is introduced to how a specific real work environment works and how one or more specific IT jobs are performed in that environment. It connects the knowledge acquired in different subjects and uses them in solving practical problems.		
Course content/structure - Getting to know the work environment, subjects and business process. - Getting acquainted with the way of issuing tasks, performing work and reporting on the work done. - Taking over specific tasks, solving them and reporting on the work done, with monitoring of the business process, use of established methodologies and communication with other entities in the work environment. - The internship is performed in the duration of 90 working hours, with the prior conclusion of an internship contract and submission of a report on the internship. - The internship is done one day a week during the semester. - The student is guided by a teacher-coordinator from the faculty and a mentor-manager, who is assigned to him by the employer. If the student has already applied for the topic of the master's thesis, then the role of teacher-coordinator is played by the mentor of the master's thesis, and if not, then he is assigned a teacher.		
Literature ● Contemporary professional literature and existing documentation of the company in which the internship is performed.		
Number of active teaching classes		Classes: 6
Teaching methods Practical work.		
Knowledge evaluation (maximum 100 points)		
- Professional practice is not evaluated numerically but only descriptively: completed / not performed. - It is considered that the internship has not been performed if the company submits a report with a negative opinion after the internship. - There are no "pre-examination obligations", but the company can maintain a test of knowledge of the technological procedure before involving the student in some sensitive procedures. - At the end of the internship, the student submits and defends his / her report on the completed internship.		

<table>
<tr><td>Course:</td><td rowspan="3">Master thesis - study research work</td></tr>
<tr><td>Course id:</td></tr>
<tr><td>Number of ECTS: 6</td></tr>
<tr><td>Teacher:</td><td>Mentor</td></tr>
<tr><td>Course status:</td><td>Mandatory, first year, second semester</td></tr>
<tr><td colspan="2">Precondition courses: Passed the exam in the field of which the independent research work is defined</td></tr>
<tr><td colspan="2">Educational goal
Preparation of the student for independent professional and / or scientific-research work on the preparation of study-research work, as well as on the preparation of the following master's thesis. With the help of a mentor, the student observes, presents the methodology and solves a specific current problem with scientific research methods, with the application of theoretical and applied knowledge acquired during the study.</td></tr>
<tr><td colspan="2">Educational outcomes (acquired knowledge):
Successful first steps have been taken in training students for independent professional and / or scientific research work in the elective field of computer science.</td></tr>
<tr><td colspan="2">Course content/structure
Theory and practical classes:
With the support of the mentor, the student uses the acquired applied and theoretical knowledge, but still studies and researches the selected professional and / or scientific field, the content of which depends on the specifically considered problem.</td></tr>
<tr><td colspan="2">Literature
● Relevant literature should indicate the systematic approach of the student in writing a study research paper, which should be the starting point in the definition of the topic and area in the development of the master thesis.</td></tr>
<tr><td>Number of active teaching classes</td><td colspan="2">Study research work: 9</td></tr>
<tr><td colspan="2">Teaching methods
Consultations with the mentor in all phases of the study research work: collection of references, their systematic study, definition of the field of work and the preparation of the study research work itself.</td></tr>
<tr><td colspan="2">Knowledge evaluation (maximum 100 points)</td></tr>
<tr><td>Pre-examination obligations:
Lecture attendance and activity:
Practical classes
Colloquium exam:
Seminary:</td><td>Final exam:
Completed and accepted work: 45
Oral defence: 55</td></tr>
</table>

Course: **Course id:** **Number of ECTS: 7**	**Master thesis - development and defence**
Teacher:	All teachers in the study program
Course status:	Mandatory, first year, second semester

Precondition courses: Registration during the second semester, and submission and defense is possible after passing all exam obligations in master studies.

Educational goal

The aim of the master's thesis is for the student to show the ability to engage in professional (or scientific) research, apply adequate methods of data collection and processing, the ability to independently write professional or scientific papers, as well as the student's ability to independently present professional problems and represent certain professional ideas.

Educational outcomes (acquired knowledge):

By preparing and defending a master's thesis, students are able to solve real application problems, as well as to consider and analyse theoretical solutions. This includes developed critical thinking, the ability to analyse problems, synthesize solutions, predict the consequences of the chosen solution, using scientific methods and procedures. Especially important is the ability to adopt relevant innovations in the profession, their connection with basic knowledge, application in practice and clear transfer to the professional and general public.

Course content/structure

The supervisor - mentor of the master's thesis assigns a topic from the area covered by the curriculum, which the candidate can successfully process within three months, based on the acquired knowledge, literature studies, practical or experimental work.

Literature

- Relevant literature in the field of research, which is directly related to the topic of the master's thesis.

Number of active teaching classes		Classes: 4

Teaching methods

The student can take a master's thesis from all already professional subjects passed with the lowest grade 8. The Rulebook on taking master's thesis at undergraduate and master's studies at Alfa BK University in Belgrade defines the procedure for applying, writing and defending a master's thesis.

Knowledge evaluation (maximum 100 points)

The Master's thesis and oral defence are evaluated by the Commission (mentor and two members) with a single grade.