

UČNI NAČRT PREDMETA / COURSE SYLLABUS (leto / year 2017/18)						
Predmet:	Principi programskih jezikov					
Course title:	Principles of programming languages					
Študijski program in stopnja Study programme and level	Študijska smer Study field		Letnik Academic year		Semester Semester	
Interdisciplinarni univerzitetni študijski program Računalništvo in matematika	ni smeri		2		drugi	
Interdisciplinary first cycle academic study programme Computer Science and Mathematics	none		2		second	
Vrsta predmeta / Course type			obvezni / compulsory			
Univerzitetna koda predmeta / University course code:			63220			
Predavanja Lectures	Seminar Seminar	Vaje Tutorial	Klinične vaje work	Druge oblike študija	Samost. delo Individ. work	ECTS
45		30			105	6
Nosilec predmeta / Lecturer:			prof. dr. Ivan Bratko			
Jeziki / Languages:	Predavanja / Lectures:		slovenski / Slovene			
	Vaje / Tutorial:		slovenski / Slovene			
Pogoji za vključitev v delo oz. za opravljanje študijskih obveznosti:			Prerequisites:			
Vpis v letnik študija.			Enrolment in the programme.			
Vsebina:			Content (Syllabus outline):			

<p>Modeli računanja in paradigme programiranja: imperativno, postopkovno programiranje, deklarativno, nepostopkovno, programiranje, objektno programiranje, funkcijsko programiranje, logično in relacijsko programiranje, programiranje z omejitvami, paralelno programiranje, genetsko programiranje, programiranje s primeri, itd.</p> <p>Pregled programskih jezikov za razne paradigme programiranja</p> <p>Elementi jezikov postopkovnega programiranja</p> <p>Nepostopkovno programiranje, logično programiranje in programski jezik prolog: logika kot programski jezik, postopkovni pomen programa kot avtomatsko dokazovanje izrekov, primeri simboličnega programiranja in deklarativnega snovanja programov</p> <p>Programiranje z omejitvami: ideje, principi in primeri, logično programiranje z omejitvami (CLP)</p> <p>Obravnavanje sintakse in semantike programskih jezikov: gramatike, operativna, prevajalska, denotacijska in aksiomska semantika</p> <p>Denotacijska semantika, povezava s gramatiko jezika, primeri denotacijskih definicij</p> <p>Aksiomska semantika in dokazovanje pravilnosti programov: parcialna in totalna pravilnost, invariantni pogoji, tehnike dokazovanja pravilnosti programov, uporaba najšibkejših predpogojev, avtomatsko dokazovanje pravilnosti</p>	<p>Computational models and programming paradigms: imperative, procedural programming, declarative, non-procedural programming, functional programming, logic and relational programming, programming with constraints, parallel programming, genetic programming, programming by examples, etc.</p> <p>Overview of programming languages for various programming paradigms</p> <p>Elements of languages for imperative programming</p> <p>Declarative programming, logic programming and the Prolog language: logic as a programming language, procedural meaning of programs as automatic theorem proving, examples of symbolic programming and declarative program design</p> <p>Programming with constraints: ideas, principles and examples, constraint logic programming (CLP)</p> <p>Handling of syntax and semantics of programming languages: grammars, operational, translational, denotational and axiomatic semantics</p> <p>Denotational semantics, relation to the the grammar of a language, examples of denotational definitions</p> <p>Axiomatic semantics and proving correctness of programs: partial and total correctness, invariant conditions, techniques of proving program correctness, using weakest preconditions, automatic correctness proving</p>
--	---

Temeljni literatura in viri / Readings:

Robert W. Sebesta, Concepts of Programming Languages, 8th edition, Addison-Wesley 2007.

Peter van Roy, Seif Haridi, Concepts, Techniques, and Models of Computer Programming, MIT Press 2004.

Ivan Bratko, Prolog Programming for Artificial Intelligence, 4th edition, Pearson Education, Addison-Wesley, 2001.

Ivan Bratko, Prolog in umetna inteligenca, Založba FE in FRI, ponatis 2011.

Cilji in kompetence:

Cilj je predstaviti principe in pregled vrst programskih jezikov, vključno z različnimi modeli računanja, formalnim obravnavanjem sintakse in semantike jezikov ter pravilnosti programov,

Razumevanje različnih vzorcev oz. paradigem programiranja ter njihove uporabe v ustreznih programskih jezikih,

Praktična uporaba simboličnega programiranja, nepostopkovnega programiranja in programiranja z omejitvami

Objectives and competences:

To introduce the principles and types of programming languages, including models of computation, formal treatment of the syntax and semantics of languages and program correctness,

Understanding of various programming paradigms and their use in corresponding programming languages,

Practical applications of symbolic, declarative and constraint programming

Predvideni študijski rezultati:

Znanje in razumevanje:

Sodobno razumevanja pojmov algoritem, računanje, izračunljivost, računski zahtevnost in obvladljivost ter povezav med njimi. Sposobnost samostojnega analiziranja računski zahtevnosti problemov in možnosti za učinkovit izračun (kakovostnih) rešitev.

Uporaba:

Uporaba naučenih pojmov, principov in tehnik pri reševanju konkretnih računskih problemov v praksi.

Intended learning outcomes:

Knowledge and understanding:

Understanding of various approaches to programming and suitability of these approaches to solving various problems,

Overview of the principles and mechanisms of types of programming languages,

Understanding ways of defining the syntax and semantics of languages, and formal proofs of program correctness.

Application:

<p>Refleksija:</p> <p>Razumevanje postopkov za analizo zahtevnosti računskih problemov in poznavanje strategij in metod za njihovo reševanje.</p> <p>Prenosljive spretnosti - niso vezane le na en predmet.</p>	<p>Skill of symbolic programming, logic and constrain programming</p> <p>Reflection:</p> <p>Ability of thinking about alternative formulations of problems and approaches to their solution,</p> <p>How different computational models, programming paradigms and languages, support alternative approaches to computer problem solving</p> <p>Transferable skills:</p> <p>Enhanced skills of program design</p>
---	--

Metode poučevanja in učenja:

<p>Predavanja, avditorne vaje and exercises, domače naloge</p>
--

Learning and teaching methods:

<p>Lectures, practical work and exercises, home work</p>
--

Načini ocenjevanja:

Delež (v %) /
Weight (in %)

Assessment:

<p>Način (pisni izpit, ustno izpraševanje, naloge, projekt):</p> <p>Sprotno preverjanje (domače naloge)</p> <p>Končno preverjanje (pisni in ustni izpit)</p> <p>Ocene: 6-10 pozitivno, 1-5 negativno</p> <p>(v skladu s Statutom UL)</p>	<p>50%</p> <p>50%</p>	<p>Type: exam, oral, coursework, project</p> <p>Continuing: homework, project work</p> <p>Final: written and oral exam</p> <p>Grading: 6-10 pass, 1-5 fail.</p>
--	-----------------------	---

Reference nosilca / Lecturer's references:

<p>BRATKO, Ivan. Prolog programming for artificial intelligence. 4th ed. Harlow (England) [etc.]: Addison-Wesley: Pearson, cop. 2012. XXI, 673 str., ilustr. ISBN 978-0-321-41746-6. ISBN 0-321-41746-1. [COBISS.SI-ID 8577364]</p>

MOŽINA, Martin, ŽABKAR, Jure, BRATKO, Ivan. Argument based machine learning. Artificial intelligence, ISSN 0004-3702. [Print ed.], 2007, vol. 171, no. 10/15, str. 922-937. [COBISS.SI-ID 6240084]

LUŠTREK, Mitja, GAMS, Matjaž, BRATKO, Ivan. Is real-valued minimax pathological?. Artificial intelligence, ISSN 0004-3702. [Print ed.], 2006, vol. 170, str. 620-642. [COBISS.SI-ID 19805735]

ŠUC, Dorian, VLADUŠIČ, Daniel, BRATKO, Ivan. Qualitatively faithful quantitative prediction. Artificial intelligence, ISSN 0004-3702. [Print ed.], 2004, vol. 158, no. 2, str. [189]-214, ilustr. [COBISS.SI-ID 4422740]

BRATKO, Ivan, MOZETIČ, Igor, LAVRAČ, Nada. Kardio : a study in deep and qualitative knowledge for expert systems. Cambridge (Mass.), London: The MIT Press, 1989. XIV, 260 str., graf. prikazi. ISBN 0-262-02273-7. [COBISS.SI-ID 19925760]