

UČNI NAČRT PREDMETA / COURSE SYLLABUS							
Predmet:		Teorija programskih jezikov					
Course title:		Theory of programming languages					
Študijski program in stopnja		Študijska smer		Letnik		Semester	
Study programme and level		Study field		Academic year		Semester	
Magistrski študijski program Matematika		ni smeri		1 ali 2		prvi ali drugi	
Master's study programme Mathematics		none		1 or 2		first or second	
Vrsta predmeta / Course type				izbirni			
Univerzitetna koda predmeta / University course code:				M2820			
Predavanja	Seminar	Vaje	Klinične vaje	Druge oblike	Samost. delo	ECTS	
Lectures	Seminar	Tutorial	work	študija	Individ. work		
45		30			105	6	
Nosilec predmeta / Lecturer:		doc. Matija Pretnar, prof. Alexander Keith Simpson, prof. Andrej Bauer					
Jeziki / Languages:		Predavanja / Lectures: slovenski/Slovene, angleški/English					
		Vaje / Tutorial: slovenski/Slovene, angleški/English					
Pogoji za vključitev v delo oz. za opravljanje študijskih obveznosti:				Prerequisites:			
Vsebina:				Content (Syllabus outline):			

<p>Pri predmetu se obravnava teorija programskih jezikov s poudarkom na uporabi matematičnih metod pri podajanju jezikov in analizi njihovih lastnosti. Obravnavajo se naslednje teme:</p> <ul style="list-style-type: none"> - konkretna in abstraktna sintaksa, - induktivne definicije, definicije - dokazovanje s strukturno indukcijo - induktivni podatkovni tipi kot - operacijska semantika kot - funkcijski programski jeziki: - polimorfizem, parametrični - ukazni programski jeziki, - denotacijska semantika: domene in - izbirne vsebine: objektni <p>leksična in gramatična analiza kot prevajanje konkretne v abstraktno sintakso</p> <p>podane z sodbami in pravili sklepanja</p> <p>po abstraktni sintaksi ali po strukturi izpeljave</p> <p>primer uporabe strukturnih definicij in strukturne indukcije</p>	<p>The course covers the theory of programming languages with emphasis on mathematical methods for specification of programming languages and analysis of their properties. The following topics are covered:</p> <ul style="list-style-type: none"> - concrete and abstract syntax, lexical analysis and parsing as translation of concrete syntax to abstract syntax - inductive definitions, definitions given in terms of judgements and rules of inference - proofs by structural induction on abstract syntax and on the structure of a derivation - inductive datatypes as an example of use of structural definitions and structural induction - operational semantics as an inductively specified relation, small-step and big-step semantics - functional programming languages: recursive definitions, eager and lazy languages, static analysis, type checking,
---	---

<p>induktivno definirana relacija, semantika malih in velikih korakov rekurzivne definicije, neučakani in leni jeziki, statična analiza, preverjanje tipov, varnost kot posledica leme o napredku in leme o ohranitvi, pomen varnosti v praksi polimorfizem in Hindley-Milnerjeva izpeljava tipov specifikacije in dokazovanje pravilnosti programov zvezne funkcije, izrek o obstoju negibnih točk, denotacijska semantika funkcijskega programskega jezika, interpretacija rekurzije z negibnimi točkami programski jeziki, paralelno računanje, logično programiranje</p>	<p>safety as a consequence of progress and termination lemmas, significance of safety in practice - polymorphism, parametric polymorphism and Hindley-Milner type inference - imperative programming languages, specification and proofs of correctness - denotational semantics: domains and continuous maps, existence of fixed points, denotational semantics of a functional language, interpretation of recursion as fixed points - optional topics: object-oriented programming languages, parallel computing, logic programming</p>
---	---

Temeljni literatura in viri / Readings:

- B.C. Pierce: "Types and Programming Languages". The MIT Press 2002.
- J.C. Reynolds: "Theories of Programming Languages". Cambridge University Press 1998.
- R.M. Amadio & P.-L. Currien: "Domains and λ -calculi". Cambridge Tracts in Theoretical Computer Science 46. Cambridge University Press, 1998.

Cilji in kompetence:

Cilj predmeta je predstavitev modernega, matematičnega pristopa, k teoriji programskih jezikov. Študenti pridobijo sposobnost analize programskih jezikov ter osnovnih konceptov povezanih z njimi.

Objectives and competences:

The objective of the course is to present modern, mathematical approach to theory of programming languages. Students will attain the ability to analyze programming languages and the basic concepts related to them.

Predvideni študijski rezultati:

Znanje in razumevanje:
Slušatelji se naučijo, kako načrtujemo in analiziramo programske jezike s formalnimi matematičnimi metodami.

Intended learning outcomes:

Knowledge and understanding:
Students learn how to design and analyze programming languages with formal mathematical methods.

Metode poučevanja in učenja:

predavanja, vaje, domače naloge

Learning and teaching methods:

lectures, tutorials, homeworks

Delež (v %) /

Načini ocenjevanja:

domače naloge, kolokviji, projekti, pisni izpit, ustni izpit
ocene: 5 (negativno), 6-10 (pozitivno) (po Statutu UL)

Weight (in %) **Assessment:**

homework, midterm exams, projects, written exam, oral exam

grading: 5 (fail), 6-10 (pass) (according to the Statute of UL)

100%

Reference nosilca / Lecturer's references:

Andrej Bauer:

– LUKŠIČ, Primož, HORVAT, Boris, BAUER, Andrej, PISANSKI, Tomaž. Practical E-Learning for the Faculty of Mathematics and Physics at the University of Ljubljana. Interdisciplinary journal of knowledge & learning objects, ISSN 1552-2210, 2007, vol. 3, str. 73-83 [COBISS.SI-ID 14269529]

– BAUER, Andrej, STONE, Christopher A. RZ: a tool for bringing constructive and computable mathematics closer to programming practice. V: Computation and logic in the real world : Third Conference on Computability in Europe, CiE 2007, Siena, Italy, June 18-23, 2007 : proceedings, (Lecture notes in computer science, ISSN 0302-9743, 4497). Berlin, Heidelberg: Springer, cop. 2007, str. 28-42 [COBISS.SI-ID 14631769]

Matija Pretnar:

– PLOTKIN, Gordon, PRETNAR, Matija. Handling algebraic effects. Logical methods in computer science, ISSN 1860-5974, 2013, vol. 9, iss. 4, paper 23 (str. 1-36) [COBISS.SI-ID 16816729]

– PRETNAR, Matija. Inferring algebraic effects. Logical methods in computer science, ISSN 1860-5974, 2014, vol. 10, iss. 3, paper 21 (str. 1-43) [COBISS.SI-ID 17190745]

– BAUER, Andrej, PRETNAR, Matija. An effect system for algebraic effects and handlers. Logical methods in computer science, ISSN 1860-5974, 2014, vol. 10, iss. 4, paper 9 (str. 1-29). <http://arxiv.org/pdf/1306.6316> [COBISS.SI-ID 17191001]

Alexander Keith Simpson:

– AWODEY, Steve, BUTZ, Carsten, SIMPSON, Alex, STREICHER, Thomas. Relating first-order set theories and elementary toposes. Bulletin of symbolic logic, ISSN 1079-8986, 2007, vol. 13, no. 3, str. 340-358 [COBISS.SI-ID 17096537]

– SIMPSON, Alex. Computational adequacy for recursive types in models of intuitionistic set theory. Annals of pure and applied Logic, ISSN 0168-0072. [Print ed.], 2004, vol. 130, iss. 1-3, str. 207-275 [COBISS.SI-ID 17117017]

– SIMPSON, Alex. A characterization of the least-fixed-point operator by dinaturality. Theoretical computer science, ISSN 0304-3975, 1993, vol. 118, iss. 2, str. 301-314 [COBISS.SI-ID 17181017]