NOVO

Univerza v Mariboru

Fakulteta za naravoslovje in matematiko

UČNI NAČRT PREDMETA / COURSE SYLLABUS	
Predmet:	**Programski vzorci**
Course title:	**Programming Paradigms**

Študijski program in stopnja Study programme and level	Študijska smer Study field	Letnik Academic year	Semester Semester
Enovit magistrski študijski program druge stopnje Predmetni učitelj	**/**	**5.**	**9**
Five-year master's degree program Subject Teacher	**/**		

Vrsta predmeta / Course type Izbirni / Elective

Univerzitetna koda predmeta / University course code:

Predavanja Lectures	Seminar Seminar	Sem. vaje Tutorial	Lab. vaje Laboratory work	Teren. vaje Field work	Samost. delo Individ. work	ECTS
30		**3**	**42**		**105**	**6**

Nosilec predmeta / Lecturer: Janez Brest

Jeziki / Languages:		
	Predavanja / Lectures:	**slovenščina / Slovenian**
	Vaje / Tutorial:	**slovenščina / Slovenian**

Pogoji za vključitev v delo oz. za opravljanje študijskih obveznosti:

Znanje, vključeno v predmete Osnove računalništva in informatike, Programiranje II in Diskretne strukture.

Prerequisits:

Knowledge included in the courses Fundamentals of Computer science and Informatics, Programming 2 and Discrete Mathematics.

Vsebina:

- Uvod: neformalna definicija programskih jezikov, delitve in zgodovina programskih jezikov.
- Vzorci imperativno/proceduralnega programiranja.
- Vzorci objektno usmerjenega

Content (Syllabus outline):

- Introduction: definition of programming languages, classification and history of programming languages.
- Imperative/procedural paradigms.
- Paradigms of object oriented programming. Functional programming: Clojure

programiranja. Vzorci funkcijskega programiranja: programski jezik Clojure. • Objekti in seznami v funkcijskem programskem jeziku, predstavitev seznamov, konstruktorji za delo s seznamom, osnovne operacija nad seznamom. • Funkcije, opcijski parametri, spremenljivke, anonimne funkcije, funkcije višjega reda. • Paralelno programiranje, konstrukti za podporo paralelnosti, programski jeziki za paralelno programiranje. • Paralelizacija sekvenčnih programov. • Niti, kritične sekcije, deljeni pomnilnik. • Logično programiranje, programski jezik prolog. • Nepostopkovno programiranje, predstavitev znanja, dejstva in pravila, povpraševanja, preiskovanje alternativnih odgovorov. • Seznami, predstavitev in operacije nad seznami v programskem jeziku prolog.	programming language. • Object and lists in functional programming language, list representation, constructors and basic operations for lists. • Functions, optional parameters, variables, anonymous functions, higher-order functions. • Parallel programming, parallelism constructs, programming languages for parallel programming. • Parallelization of sequential programs. • Threads, critical sections, shared memory. • Logic programming, programming language prolog. • Declarative programming, knowledge representation, facts and rules, queries, searching for alternative answers. • Lists, representation of lists and basic operations on lists in prolog.

Temeljni literatura in viri / Readings:

- D. A. Watt, Programming Language Concepts and Paradigms. Prentice-Hall International, New York, 1990.
- M. Gabbrielli, S. Martini, Programming Languages: Principles and Paradigms, Springer, 2010.
- Akhil Wali, Mastering Clojure, Packt Publishing, 2016.
- Anthony Williams, C++ Concurrency in Action, Practical Multithreading, Manning Publications, 2012.
- Ulf Nilsson, Jan Maluszynski, Logic, Programming and Prolog (2ed), Wiley, 1995.

Cilji in kompetence:	**Objectives and competences:**
Cilj predmeta je seznaniti študente z vzorci programskih jezikov in spoznati programske vzorce funkcijskega, logičnega in objektno usmerjenega programiranja.	The objective of this course is to acquaint students with concepts of programming languages and programming paradigms of functional, logical, imperative and object oriented programming.

Predvideni študijski rezultati:	**Intended learning outcomes:**
<u>Znanje in razumevanje:</u> Po zaključku tega predmeta bo študent sposoben • identificirati prednosti in slabosti	<u>Knowledge and understanding:</u> On completion of this course the student will be able to • identify advantages and disadvantages of

posameznega vzorca, • izbrati primeren programski jezik za rešitev dane naloge, • razumeti programske vzorce jezikov s pomočjo katerih se bodo hitreje naučili novega programskega jezika, • razumeti pomen abstrakcij. <u>Prenosljive/ključne spretnosti in drugi atributi:</u> • *Spretnosti komuniciranja:* ustni zagovor laboratorijskih vaj, pisno izražanje pri pisnem izpitu. • *Uporaba informacijske tehnologije:* uporaba objektno usmerjenih, funkcijskih in logičnih programskih jezikov. • *Reševanje nalog:* uporaba programskih vzorcev pri načrtovanju in implementaciji programov.	some paradigms, • to choose suitable program language for solving defined problem, • understand programming paradigms for quickly learning a new programming language, • understand term abstraction. <u>Transferable/Key skills and other attributes:</u> • *Communication skills:* oral lab work defense, manner of expression at written examination. • *Use of information technology:* use object oriented, functional and logic programming languages. • *Problem solving:* using programming paradigms for program design and implementation.

Metode poučevanja in učenja:	**Learning and teaching methods:**
• predavanja, • seminarske vaje, • laboratorijske vaje, • kvizi.	• lectures, • tutorials, • lab work, • quizzes.

Načini ocenjevanja:	Delež (v %) / Weight (in %)	**Assessment:**
• kvizi, • laboratorijske vaje, • 1. vmesni pisni izpit, • 2. vmesni pisni izpit, • 3. vmesni pisni izpit.	15 % 35 % 16 % 17 % 17 %	• quizzes, • lab work, • 1st midterm written exam, • 2nd midterm written exam, • 3rd midterm written exam.

Opomba: Če študent ni uspešno opravil vseh treh vmesnih izpitov, jih nadomesti s pisnim izpitom v deležu 50%.
Note: If a student has not completed all three midterm exams, he replaces them with a written exam in the weight of 50%.

Reference nosilca / Lecturer's references:

- ZAMUDA, Aleš, BREST, Janez. Vectorized procedural models for animated trees reconstruction using differential evolution. *Information sciences*, ISSN 0020-0255. 2014, vol. 278, str. 1-21, [COBISS.SI-ID 17793558].
- BREST, Janez, SEPESY MAUČEC, Mirjam. Self-adaptive differential evolution algorithm using population size reduction and three strategies. *Soft computing*, ISSN 1432-7643. 2011, vol. 15,

no. 11, str. 2157-2174, [COBISS.SI-ID 14398230], [JCR, SNIP, WoS do 5. 8. 2014: št. citatov (TC): 19, Scopus : št. citatov (TC): 32]

- FISTER, Iztok, FISTER, Iztok, YANG, Xin-She, BREST, Janez. A comprehensive review of firefly algorithms. *Swarm and evolutionary computation*, ISSN 2210-6502, Dec. 2013, vol. 13, str. 34-46, doi: 10.1016/j.swevo.2013.06.001. [COBISS.SI-ID 17010454], [SNIP, Scopus do 8. 10. 2014: št. citatov (TC): 29]
- ZAMUDA, Aleš, BREST, Janez. Environmental framework to visualize emergent artificial forest ecosystems. *Information sciences*, ISSN 0020-0255. 2013, vol. 220, str. 522-540, doi: 10.1016/j.ins.2012.07.031. [COBISS.SI-ID 16157206], [JCR, SNIP, WoS do 6. 8. 2014: št. citatov (TC): 1, čistih citatov (CI): Scopus do 13. 8. 2014: št. citatov (TC): 3]
- BREST, Janez, GREINER, Sašo, BOŠKOVIĆ, Borko, MERNIK, Marjan, ŽUMER, Viljem. Self-adapting control parameters in differential evolution: a comparative study on numerical benchmark problems. *IEEE transactions on evolutionary computation*, ISSN 1089-778X. 2006, vol. 10, no. 6, str. 646-657. [COBISS.SI-ID 10376982], [JCR, SNIP, WoS do 2. 11. 2014: št. citatov (TC): 352, Scopus: št. citatov (TC): 857]