UČNI NAČRT PREDMETA / COURSE SYLLABUS

Predmet:	Funkcijsko programiranje
Course title:	Functional programming

Študijski program in stopnja **Study programme and level**	**Študijska smer** **Study field**	**Letnik** **Academic year**	**Semester** **Semester**
Interdisciplinarni magistrski študijski program Računalništvo in matematika	ni smeri	1	prvi
Interdisciplinary Masters study programme Computer Science and Mathematics	none	1	first

Vrsta predmeta / Course type	obvezni
Univerzitetna koda predmeta / University course code:	M2850

Predavanja **Lectures**	**Seminar** **Seminar**	**Vaje** **Tutorial**	**Klinične vaje** **work**	**Druge oblike** **študija**	**Samost. delo** **Individ. work**	**ECTS**
45	**10**	**20**			**105**	**6**

Nosilec predmeta / Lecturer:	Zoran Bosnić

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):**

Predmet poučuje koncept in uporabo paradigme funkcijskega programiranja, skozi katero se dotika teorije programskih jezikov in poglobljenega razumevanja njihovih lastnosti. Poglavja pri predmetu vsebujejo: 1. Uvod v funkcijsko programiranje. 2. Pojem okolja, leksikalnega in semantičnega dosega. 3. Osnove funkcijskega jezika Standard ML (sintaksa, semantika, enostavni in sestavljeni podatkovni tipi, opcije, lastni tipi) in osvajanje naslednjih pojmov: -ujemanje vzorcev, -funkcije višjega reda, currying, -delo z moduli. 4. Osnove funkcijskega jezika Racket in osvajanje naslednjih pojmov: -takojšnja in lena evalvacija, -tokovi, -zakasnitev in sprožitev, -gradnja podatkovnih tipov, -funkcije z dinamičnim številom argumentov, -izdelava interpreterja. 5. Primerjava funkcijskega in objektno usmerjega programiranja. 6. Vrste tipiziranj (statično/dinamično, močno/šibko, implicitno/eksplicitno) in trdnost/polnost sistema tipov.		The course teaches the concept and use of a functional programming paradigm and connects it to the programming language theory through a deeper understanding of programming language concepts. The content contains: 1. Introduction to functional programming. 2. Concepts of: environment, lexical and semantic scope. 3. Basics of Standard ML (syntax, semantics, basic and complex data types, options, custom types) and concepts: -pattern matching, -higher order functions, currying, -working with modules. 4. Basics of Racket programming language and concepts: -eager and lazy evaluation, -streams, -delay and force, -building custom datatypes, -functions with variable number of arguments, -making an interpreter. 5. Comparison of functional and object-oriented programming. 6. Different types of typing (static/dynamic, weak/strong, implicit/explicit) and soundness/completeness of a type system.

Temeljni literatura in viri / Readings:

1. R. Pucella: Notes on Programming SML/NJ, Cornell, 2001
2. Matthew Flatt, Robert Bruce Findler et al.: The Racket Guide, 2015.
3. Ravi Sethi: Programming Languages: concepts &, constructs. Addison-Wesley, 1996.
4. A. Tucker, R. Noonan: Programming Languages: Principles and Paradigms. McGraw-Hill, 2007.

Cilji in kompetence:

Študenti, ki so dokončali prvostopenjski študij RI, so opravili predmete s področja osnov programiranja in pretežno spoznali objektno-usmerjeno paradigmo programiranja. Cilj tega predmeta predstaviti drugačne tehnike programiranja s poudarkom na funkcijskem programiranju. Predmet bo študentom omogočil razvoj veščin kritičnega, analitičnega in sintetičnega mišljenja pri uporabi in razumevanju delovanja programskih jezikov kot temeljnih orodij vsakega programerja.

Objectives and competences:

Students who finished the undergraduate study of computer science already completed courses on basics of programming and mostly used the object-oriented programming paradigm. The objective of this course is to present alternative programming techniques with the emphasis on functional programming. The course will help develop students' skills in critical, analytical and synthetic thinking for use and understanding of programming languages as basic tools of each programmer.

Predvideni študijski rezultati:

Po uspešnem zaključku predmeta bo študent:
- razlikoval paradigme objektno-usmerjenega in funkcijskega programiranja,

- znal opisati prednosti izogibanja mutaciji in stranskim učinkom,

- sposoben uporabljati ujemanje vzorcev, funkcije višjega reda, lastne podatkovne tipe, zakasnjeno evalvacijo,

- razločeval med statično/dinamično, impliticno/eksplicitno, šibko/močno tipiziranimi programskimi jeziki,

- sposoben načrtovati lastni preprost programski jezik,

- sposoben argumentirati, katera programerska paradigma je bolj primerna za reševanje danega problema.

Intended learning outcomes:

After the completion of the course the student will be able to:
- differentiate between the object-oriented and functional programming paradigms,

- describe advantages of avoiding mutation and program side-effects,

- use pattern matching, higher-order functions, own data types and lazy evaluation,

- differentiate between statically/dynamically, implicitly/explicitly, weakly/strongly typed programming languages,

- design own simple programming language,

- argue which programming paradigm is the most suitable for solving a given problem.

Metode poučevanja in učenja:	Learning and teaching methods:
Predavanja, domače naloge in seminarske naloge. Poseben poudarek je na individualnem delu študentov.	Lectures, homeworks and seminar works with special emphasis on individual work.

Načini ocenjevanja:	Delež (v %) / Weight (in %)	Assessment:
Način (pisni izpit, ustno izpraševanje, naloge, projekt): Sprotno preverjanje (seminarske nal.) Končno preverjanje (pisni ali ustni izpit) Ocene: 6-10 pozitivno, 5 negativno (v skladu s Statutom UL).	50% 50%	Type (examination, oral, coursework, project): Continuing (homework) Final (written or oral exam) Grading: 6-10 pass, 5 fail (according to the rules of University of Ljubljana).

Reference nosilca / Lecturer's references:

Zoran Bosnić:

– ČEHOVIN, Luka, BOSNIĆ, Zoran. Empirical evaluation of feature selection methods in classification. Intelligent data analysis, ISSN 1088-467X. [Print ed.], 2010, vol. 14, no. 3, str. 265-281, ilustr [COBISS.SI-ID 7732564]

– BOSNIĆ, Zoran, KONONENKO, Igor. Correction of regression predictions using the secondary learner on the sensitivity analysis outputs. Computing and informatics, ISSN 1335-9150, 2010, vol. 29, no. 6, str. 929-946, graf. prikazi [COBISS.SI-ID 8027988]

– VOGRINČIČ, Sergeja, BOSNIĆ, Zoran. Ontology-based multi-label classification of economic articles. Computer science and information systems, ISSN 1820-0214. [Print ed.], 2011, vol. 8, no. 1, str. 101-119, ilustr [COBISS.SI-ID 8235860]

– POGORELC, Bogdan, BOSNIĆ, Zoran, GAMS, Matjaž. Automatic recognition of gait-related health problems in the elderly using machine learning. Multimedia tools and applications, ISSN 1380-7501, 2012, vol. 58, no. 2, str. 333-354, graf. prikazi [COBISS.SI-ID 8773460]

– BOSNIĆ, Zoran, VRAČAR, Petar, RADOVIĆ, Miloš D., DEVEDŽIĆ, Goran, FILIPOVIĆ, Nenad D., KONONENKO, Igor. Mining data from hemodynamic simulations for generating prediction and explanation models. IEEE transactions on information technology in biomedicine, ISSN 1089-7771. [Print ed.], Mar. 2012, vol. 16, no. 2, str. 248-254, ilustr [COBISS.SI-ID 9026900]