



Univerza v Mariboru

Fakulteta za naravoslovje
in matematiko

UČNI NAČRT PREDMETA / SUBJECT SPECIFICATION

Predmet:	Principi programskih jezikov
Subject Title:	Principles of Programming Languages

Študijski program Study programme	Študijska smer Study option	Letnik Year	Semester Semester
Enovit magistrski študijski program druge stopnje Predmetni učitelj		5.	9
Five-year master's degree program Subject Teacher		5.	9

Univerzitetna koda predmeta / University subject code:

Predavanja Lectures	Seminar Seminar	Seminarske vaje Tutorial	Laborat. vaje Lab work	Terenske vaje Field work	Samostojno delo Individual work	ECTS
30		2	28		120	6

Nosilec predmeta / Lecturer:

Jeziki / Languages:

Predavanja / Lecture:	slovenski / Slovene
Vaje / Tutorial:	slovenski / Slovene

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

Prerequisites:

Vsebina:

- Uvod v programske jezike: neformalna definicija programskih jezikov, delitev programskih jezikov, kratka zgodovina programskih jezikov.
- Vrednosti in tipi: delitev tipov, statično in dinamično preverjanje tipov, ekvivalenca tipov, vrste izrazov.
- Pomnilnik: spremenljivka, shranjive vrednosti, življenjska doba spremenljivk, vrste ukazov, izrazi s stranskimi učinki.
- Povezovanje: povezljive vrednosti, statični in dinamični doseg, vrste deklaracij, bločni ukazi in bločni izrazi, kvalifikacijski princip.
- Abstrakcije: princip abstrakcije, vrste

Content (Syllabus outline):

- Introduction to programming languages: informal definition of programming languages, classification of programming languages, brief history of programming languages.
- Values and types: type classification, static and dynamic type checking, type equivalence, kind of expressions.
- Storage: variable, storable values, variable lifetime, kind of commands, expressions with side effects.
- Binding: bindable values, static and dynamic scope, kind of declarations, block commands and block expressions, the

abstrakcij, mehanizmi prenosa parametrov, korespondenčni princip, dosledni in normalni izračun.

- Ograjevanje: paketi, abstraktni tipi, objekti in razredi, generični moduli.
- Sistemi tipov: monomorfni in polimorfni sistem tipov, vrste polimorfizma. Generiki v programskem jeziku java.
- Objektno usmerjeno programiranje: objekt, razred, meta-razred. Razredni objektno usmerjeni jeziki in prototipni objektno usmerjeni jeziki. Vrste dedovanja: enkratno/večkratno, razredno/objektno, striktno/nestriktno, urejeno/neurejeno, dinamično/selektivno, običajno/mixin. Vgnezdjeni razredi v programskem jeziku java.
- Funkcijsko programiranje: značilnosti funkcijskih jezikov, uvod v programski jezik lisp/haskell.

qualification principle.

- Abstractions: abstraction principle, kind of abstractions, parameter passing mechanisms, the correspondence principle, eager and normal evaluation.
- Encapsulation: packages, abstract types, objects and classes, generic modules.
- Type systems: monomorphic and polymorphic type systems, kind of polymorphisms. Java generics.
- Object-Oriented programming: object, class, meta-class. Class-based object-oriented languages and prototype-based object-oriented languages. Kinds of inheritance: single/multiple, class-based/object-based, strict/non-strict, ordered/un-ordered, dynamic/selective, ordinary/mixin. Nested Java classes.
- Functional programming: characteristics of functional languages, introduction to Lisp/Haskell programming language.

Temeljni študijski viri / Textbooks:

- K.C. Louden: *Programming Languages: Principles & Practices*, Third Edition, Cengage Learning, 2013.
- P. Sestoft: *Programming Language Concepts*, Springer, Berlin, 2012.
- D. A. Watt: *Programming Language Design Concepts*, John Wiley, Chichester, 2004.

Cilji:

Cilj tega predmeta je razumevanje osnovnih konceptov programskih jezikov in razumeti bistvene razlike med funkcijskim, proceduralnim in objektno usmerjenim programiranjem.

Objectives:

The objective of this course is to understand the basic concepts of programming languages and to understand essential differences between functional, procedural and object-oriented programming.

Predvideni študijski rezultati:

Znanje in razumevanje:

Po zaključku tega predmeta bo študent sposoben

- identificirati slabosti in prednosti posameznega programskega vzorca,
- izbrati primeren programski jezik za rešitev dane naloge,

Intended learning outcomes:

Knowledge and understanding:

On completion of this course the student will be able to

- identify shortcomings and advantages of particular programming language,
- select suitable programming language to solve particular problem,
- understand concepts of programming

<ul style="list-style-type: none"> • razumeti koncepte programskih jezikov, s pomočjo katerih se bo hitreje naučil novega programskega jezika, • razumeti razlike med statičnim in dinamičnim tipiziranjem, • razumeti različne oblike dodeljevanja pomnilnika, • razumeti različne tehnike prenosa parametrov, • razumeti različne oblike polimorfizma. 	<p>languages with the aim to quicker learn new programming language,</p> <ul style="list-style-type: none"> • understand differences between static and dynamic typing, • understand different ways of memory allocation, • understand different parameter passing techniques, • understand different forms of polymorphisms.
<p><u>Prenosljive/ključne spretnosti in drugi atributi:</u></p> <ul style="list-style-type: none"> • <i>Spretnosti komuniciranja:</i> ustni zagovor laboratorijskih vaj, pisno izražanje pri pisnem izpitu. • <i>Uporaba informacijske tehnologije:</i> uporaba različnih prevajalnikov in interpreterjev. • <i>Reševanje problemov:</i> načrtovanje in implementacija programov z uporabo različnih programskih vzorcev. 	<p><u>Transferable/Key skills and other attributes:</u></p> <ul style="list-style-type: none"> • <i>Communication skills:</i> oral lab work defence, manner of expression at written examination. • <i>Use of information technology:</i> use of different compilers and interpreters. • <i>Problem solving:</i> program design and implementation using different programming paradigms.

Metode poučevanja in učenja:

- predavanja,
- laboratorijske vaje,
-

Teaching and learning methods:

- lectures,
- lab work,
-

Načini ocenjevanja:	Delež (v %) / Weight (in %)	Assessment methods:
<p>Sprotni način</p> <ul style="list-style-type: none"> • laboratorijske vaje, • 1. kolokvij, • 2. kolokvij. 	<p>50 25 25</p>	<p>Constant assessment methods</p> <ul style="list-style-type: none"> • lab work, • 1st midterm examination, • 2nd midterm examination.

Opomba:

Kolokvija se lahko nadomestita s pisnim izpitom v deležu 50 %..

Note:

The midterm examinations may be replaced by written exam in the weight of 50%..

Reference nosilca / Lecturer's references:

1. KOS, Tomaž, MERNIK, Marjan, KOSAR, Tomaž. A tool support for model-driven development :



Fakulteta za naravoslovje
in matematiko

- an industrial case study from a measurement domain. Applied sciences, ISSN 2076-3417, 2019, vol. 9, iss. 21, str. 1-20, ilustr., doi: 10.3390/app9214553. [COBISS.SI-ID 22712342]
2. KOSAR, Tomaž, GABERC, Sašo, CARVER, Jeffrey C., MERNIK, Marjan. Program comprehension of domain-specific and general-purpose languages: replication of a family of experiments using integrated development environments. Empirical software engineering, ISSN 1382-3256, 2018, vol. 23, iss. 5, str. 2734-2763, doi: 10.1007/s10664-017-9593-2. [COBISS.SI-ID 21123606]
 3. KOSAR, Tomaž, BOHRA, Sudev, MERNIK, Marjan. A systematic mapping study driven by the margin of error. The Journal of Systems and Software, ISSN 0164-1212. [Print ed.], Oct. 2018, vol. 144, str. 439-449, doi: 10.1016/j.jss.2018.06.078. [COBISS.SI-ID 21580566]