



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:

Marjan Mernik

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, delitve programskih jezikov, kratka zgodovina programskih jezikov.
- Vrednosti in tipi: delitev tipov, statično in dinamično preverjanje tipov, ekvivalenca tipov, vrste izrazov.
- Pomnilnik: spremenljivka, shranljive 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 abstrakcij, mehanizmi prenosa parametrov, korespondenčni princip, dosledni in normalni izračun.
- Ograjevanje: paketi, abstraktni tipi, objekti in

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 qualification principle.
- Abstractions: abstraction principle, kind of abstractions, parameter passing mechanisms, the correspondence principle, eager and normal evaluation.

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.

- 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. Loudon: *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 seznaniti študente z osnovnimi koncepti programskih jezikov in razumeti bistvene razlike med funkcijskim, proceduralnim in objektno usmerjenim programiranjem.

Objectives:

The objective of this course is to acquaint students with 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,
 - 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.

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 languages with the aim to quicker learn new programming language,
 - understand differences between static and dynamic typing,
 - understand different ways of memory allocation,
 - understand different parameter passing techniques,
 - understand different forms of polymorphisms.

Prenosljive/ključne spretnosti in drugi atributi:

- *Spretnosti komuniciranja:* ustni zagovor laboratorijskih vaj, pisno izražanje pri pisnem izpitu.
- *Uporaba informacijske tehnologije:* uporaba različnih prevajalnikov in interpreterjev.
- *Reševanje problemov:* načrtovanje in implementacija programov z uporabo različnih programskih vzorcev.

Transferable/Key skills and other attributes:

- *Communication skills:* oral lab work defence, manner of expression at written examination.
- *Use of information technology:* use of different compilers and interpreters.
- *Problem solving:* program design and implementation using different programming paradigms.

Metode poučevanja in učenja:

- predavanja,
- laboratorijske vaje,
- reševanje domačih nalog.

Teaching and learning methods:

- lectures,
- lab work,
- homework assignments.

Načini ocenjevanja:	Delež (v %) / Weight (in %)	Assessment methods:
Sprotni način	Delež (Pogoj)	Constant assessment methods
• domače naloge,	10 (0)	• homeworks,
• laboratorijske vaje,	50 (25)	• lab work,
• 1. vmesni izpit,	13	• 1st midterm examination,
• 2. vmesni izpit,	13 } (20)	• 2nd midterm examination,
• 3. vmesni izpit,	14	• 3rd midterm examination.
Klasični način	Delež (Pogoj)	Classical assessment methods
• domače naloge,	10 (0)	• homeworks,
• laboratorijske vaje,	50 (25)	• lab work,
• pisni izpit.	40 (20)	written examination.

Reference nosilca / Lecturer's references:

1. KOSAR, Tomaž, MERNIK, Marjan, GRAY, Jeffrey G., KOS, Tomaž. Debugging measurement systems using a domain-specific modeling language. *Computers in industry*, ISSN 0166-3615. [Print ed.], 2014, vol. 65, iss. 4, str. 622-635.
2. MERNIK, Marjan. An object-oriented approach to language compositions for software language engineering. *The Journal of Systems and Software*, ISSN 0164-1212. [Print ed.], 2013, vol. 86, iss. 9, str. 2451-2464.
3. KOS, Tomaž, KOSAR, Tomaž, MERNIK, Marjan. Development of data acquisition systems by using a domain-specific modeling language. *Computers in industry*, ISSN 0166-3615. [Print ed.], Apr. 2012, vol. 63, no. 3, str. 181-192.
4. KOSAR, Tomaž, MERNIK, Marjan, CARVER, Jeffrey C. Program comprehension of domain-specific and general-purpose languages : comparison using a family of experiments. *Empirical software engineering*, ISSN 1382-3256, 2012, vol. 17, no. 3, str. 276-304.
5. ARORA, Ritu, BANGALORE, Purushotham, MERNIK, Marjan. Raising the level of abstraction for developing message passing applications. *The journal of supercomputing*, ISSN 0920-8542, 2012, vol. 59, no. 2, str. 1079-1100.