



UČNI NAČRT PREDMETA / SUBJECT SPECIFICATION

Predmet: Subject Title:	Programski vzorci Programming Paradigms
----------------------------	--

Študijski program Study programme	Študijska smer Study field	Letnik Year	Semester Semester
Izobraževano računalništvo, 1. stopnja		3.	poletni

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			45		105	6

Nosilec predmeta / Lecturer:

Jeziki /

Languages:

Predavanja / Lecture:

slovenski / Slovene

Vaje / Tutorial:

slovenski / Slovene

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

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

Prerequisites:

Knowledge included in the courses Fundamentals of Computer Science and Informatics, Programming II, and Discrete Mathematics.

Vsebina:

- Uvod: neformalna definicija programskih jezikov, delitve in zgodovina programskih jezikov.
- Vrednosti in tipi: delitev tipov, preverjanje tipov, ekvivalenca tipov, vrste izrazov, vrednosti prvega in drugega razreda, princip polnosti tipa.
- Pomnilnik: spremenljivka, model pomnilnika, selektivno ali popolno ažuriranje, shranljive vrednosti, življenjska doba, vrste ukazov, izrazi s stranskimi učinki.
- Povezovanje: povezovanje, okolje in blok, povezljive vrednosti, vrste deklaracij, statični in dinamični doseg, bločni izrazi in bločni ukazi, kvalifikacijski princip.
- Abstrakcije: princip abstrakcije, funkcijske abstrakcije, proceduralne abstrakcije, izbirne abstrakcije in generične abstrakcije, kopirni mehanizmi parametrov, korespondenčni princip, dosledni in normalni izračun.
- Pojem "kapsuliranja", koncepti, ki podpirajo modularnost.
- Sistemi tipov: monomorfni in polimorfni sistemi tipov, vrste polimorfizma, pojem "sekvencer", vrste sekvencerjev.
- Osnovni koncepti objektno usmerjenega programiranja, sistemi prvega, drugega, tretjega in petega nivoja, hierarhija objektnih jezikov, vrste dedovanja.
- Funkcijsko programiranje, programski jezik Lisp/Haskell.
- Logično programiranje, programski jezik prolog.
- Paralelno programiranje, konstrukti za podporo

Content (Syllabus outline):

- Introduction: definition of programming languages, classification and history of programming languages.
- Values and types: type classification, type checking, type equivalence, types of expressions, first-class values, second-class values, type completeness principle.
- Storage: variable, storage, total updating or selective updating, storable value, lifetime, types of commands, and expressions with side effects.
- Binding: binding, environment and block, bindable values, types of declarations, static scope, dynamic scope, block expression, qualification principle.
- Abstraction: abstraction principle, functional abstraction, procedural abstraction, selector abstraction, generic abstraction, copy parameter mechanism, correspondence principle, applicative-order evaluation, normal-order evaluation.
- Term "encapsulate", concepts of modularity.
- Type systems: monomorphic and polymorphic type systems, types of polymorphism, term "sequencer", types of sequencers.
- Basic concepts of object oriented programming, first, second, third, and fifth level systems, hierarchy of object languages, types of inheritance.
- Functional programming, programming language Lisp/Haskell.
- Logic programming, programming language prolog.

paralelnosti, programski jeziki za paralelno programiranje.

- Parallel programming, parallelism constructs, programming languages for parallel programming.

Temeljni študijski viri / Textbooks:

- V. Žumer, M. Mernik, *Principi programskih jezikov*, Univerza v Mariboru, Fakulteta za elektrotehniko, računalništvo in informatiko, Maribor 2003.
- R. Sethi: *Programming Languages: Concepts and Constructions*, Second Edition, Addison-Wesley, Reading, 1996.
- D. A. Watt, *Programming Language Concepts and Paradigms*. Prentice-Hall International, New York, 1990.
- D. A. Watt, *Programming Language Design Concepts*, John Wiley, Chichester, 2004.

Cilji:

Cilj predmeta je seznaniti študente s koncepti programskih jezikov in spoznati programske vzorce funkcijskega, logičnega in objektno usmerjeno programiranja.

Objectives:

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:

Znanje in razumevanje:

Po zaključku tega predmeta bo študent sposoben

- identificirati prednosti in slabosti posameznega vzorca,
- izbrati primeren programski jezik za rešitev dane naloge,
- razumeti koncepte programskih jezikov s pomočjo katerih se bodo hitreje naučili novega programskega jezika,
- razumeti razlike med statičnim in dinamičnim tipiziranjem,
- razumeti različne oblike dodeljevanja pomnilnika, dosega,
- razumeti različne oblike prenosa parametrov,
- razumeti pomen abstrakcij,
- razumeti različne oblike polimorfizma.

Prenosljive/ključne spretnosti in drugi atributi:

- *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.

Intended learning outcomes:

Knowledge and understanding:

On completion of this course the student will be able to

- identify advantages and disadvantages of some paradigms,
- to choose suitable program language for solving defined problem,
- understand concepts of programming languages for quickly learning a new programming language,
- understand differences between static and dynamic typing
- understand different storage and scope model,
- understand parameter mechanisms,
- understand term abstraction,
- understand different types of polymorphism.

Transferable/Key skills and other attributes:

- *Communication skills:* oral lab work defence, 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:

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

Teaching and learning methods:

- lectures,
- lab work,
- homework assignments.

Načini ocenjevanja:

- domače naloge,
- kvizi,
- laboratorijske vaje,
- pisni izpit.

Delež (v %) /
Weight (in %)

Assessment methods:

- homeworks,
- quizzes,
- lab work,
- written examination.

Opomba:

Navedene sestavine so obvezna sestavina učnega načrta predmeta kot ga določajo Merila za akreditacijo visokošolskih zavodov in študijskih programov v 7. členu (Ur. l. RS, št. 101/2004).