	UČNI NAČRT PREDMETA/COURSE SYLLABUS
Predmet	Programiranje 2
Course title	Programming 2

Študijski program in stopnja **Study programme and level**	**Študijska smer** **Study field**	**Letnik** **Academic year**	**Semester** **Semester**
Poslovna informatika 1	Poslovna informatika	2.	3.
Business Informatics 1	Business informatics	2nd	3rd

Vrsta predmeta/Course type	obvezni/obligatory
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**
45			30		75	6

Nosilec predmeta/Lecturer:	Doc. dr. Sebastian Lahajnar
Jeziki/ Languages: **Predavanja/Lectures:**	slovenski/Slovenian
Vaje/Tutorial:	slovenski/Slovenian

Pogoji za vključitev v delo oz. za opravljanje študijskih obveznosti:	**Prerequisites:**
• Pogoj za vključitev v delo je vpis v 2. letnik študija. • Opravljen izpit pri predmetu Programiranje 1. • Pogoj za prijavo k izpitu: opravljene domače naloge.	• The prerequisite for participation is enrolment in the second year of study and completed examination at the subject Programming 1. • Students have to successfully do all the homework assignments before the examination.

Vsebina:	**Content (Syllabus outline):**
• *Uvod:* ponovitev osnovnega poznavanja programskega jezika java – osnovno o objektno orientiranem programiranju, temeljni elementi jave (spremenljivke, izrazi, stavčni blok, ključne besede, dobesedne vrednosti), podatkovni tipi, nadzor izvajanja programa, objekti, paketi, dedovanje, obravnavanje izjem. • *Dokumentiranje javanskih programov:* namen in cilji, orodje Java Doc. • *Refleksija:* koncept refleksije, razred Class, dostop do elementov razreda preko uporabe razreda Class, uporaba koncepta. • *Optimizacija kode - nitno programiranje:*	• *Introduction:* revision of the basic knowledge of the Java programming language - general information about objectively oriented programming, fundamental elements of Java (variables, expressions, statement block, key words, literal values), data types, supervision of program implementation, objects, batches, inheritance, dealing with exceptions. • *Documenting Java programs:* purpose and goals, the Java Doc tool. • *Reflection:* concept of reflection, Class, access to class elements using Class, use of concept.

uporaba glavne niti, izdelava in uporaba lastnih niti, nadzor nad večnitnim delovanjem, prioritete in sinhronizacija izvajanja niti. • *Datoteke in tokovi:* razredi za delo z datotekami, delo z datotekami, tokovi števil in znakov, delo s tokovi. • *Serializacija in deserealizacija:* splošno o konceptu, namen in delo s konceptom. • *Applet:* skelet, inicializacija, uporaba in delo z Appletom, vprašanje varnosti. • Grafični uporabniški vmesnik (GUI): splošno, sestava. • *Abstract Windowing Toolkit (AWT):* splošno o AWT, paket awt., dogodkovni model, sestava, gradniki, uporaba v programu. • *SWING:* splošno o SWING-u, sorodstvo z AWT, paket javax., sestava, gradniki, uporaba v programu, izdelava GUI z uporabo grafičnega vmesnika IRO (integriranega razvojnega okolja).	• *Optimisation of code - threaded programming:* use of main thread, preparation and use of own threads, supervision of multi-thread functioning and synchronisation of using threads. • *Files and flows:* classes for working with files, working with files, flows of numbers and characters, working with flows. • *Serialisation and deserialisation:* general information about the concept, purpose and working with the concept. • *Applet:* frame, initiation, use of and work with Applet, security questions. • *Graphical user interface (GUI):* general information, structure. • *Abstract Windowing Toolkit (AWT):* general information about AWT, AWT package, event model, structure, building blocks, use in the program. • *SWING:* general information about SWING, similarities with AWT, the javax package, structure, building blocks, use in the program, preparation of GUI using as graphical user interface.

Temeljna literatura in viri/Readings:

Temeljna literatura/Basic literature

Mesojedec, U. in Fabjan, B. (2004). Java 2: Temelji programiranja. Pasadena.
Schildt, H. (2011). Java: the complete reference. New York [etc.]: McGraw-Hill: Oracle Press.

Priporočljiva literatura/Recommended literature

Friesen, J. (2011). Beginning Java 7. Apress.

Cilji in kompetence:	**Objectives and competences:**
Učna enota prispeva predvsem k razvoju naslednjih splošnih in specifičnih kompetenc: • usposobljenost za raziskovanje na področju informatike v upravljanju in poslovanju ter razvoj kritične in samokritične presoje; • fleksibilna uporaba znanja v praksi; • sposobnost za reševanje konkretnih delovnih problemov z uporabo znanstvenih metod in postopkov; • znanje o načinih predstavitve, zapisa in modeliranja informacije; • zmožnost zapisa problema v obliki algoritma; • razvoj programske opreme; • razumevanje računalniških sistemov in	*The learning unit mainly contributes to the development of the following general and specific competences:* • the ability to carry out research in the field of informatics in business and management, and the development of critical and self-critical assessment; • flexible use of knowledge in practice; • the ability to solve concrete work problems using scientific methods and procedures; • knowledge of the methods of presenting, recording and modelling information; • the ability to record a problem in the form of an algorithm; • development of software;

arhitektur; • osveščenost o zmožnostih in omejitvah informacijskih tehnologij.	• understanding of computer systems and architectures; • awareness of capabilities and limitations of information technologies.

Predvideni študijski rezultati:	**Intended learning outcomes:**
Znanje in razumevanje: *Študent/Študentka:* • dobi pregled nad razvojnim okoljem za programski jezik java; • dobi pregled in obvlada uporabo javanskega programskega vmesnika (API-ja); • se spozna s postopki za pisanje učinkovitih javanskih programov; • obvlada dokumentiranje javanskih programov; • obvlada pisanje preprostih mrežnih aplikacij; • obvlada pisanje preprostih grafičnih programov in grafičnih vmesnikov; • pozna osnove vključevanja javanskih programov v operacijski sistem in aplikacijske programe; • pozna prednosti in omejitve, ki jih prinaša prenosljivost javanskih programov.	Knowledge and understanding: *Students should:* • review the development environment for the Java programming language; • review and master the use of the Java application programming interface (API); • be familiarised with procedures for writing effective Java programs; • master the documentation of Java programs; • master the writing of simple network applications; • master the writing of simple graphical programs and graphical interfaces; • know the basics of including Java programs in the operating system and application programs; • know the advantages and limitations of the portability of Java programs.

Metode poučevanja in učenja:	• Learning **and teaching methods:**
• *predavanja* z aktivno udeležbo študentov: razlaga, debata, vprašanja in odgovori, primeri, reševanje problemov; • *laboratorijske vaje* na računalniku: ilustracija snovi skozi zastavljene programerske naloge.	• *lectures* with active participation of students: explanation, debate, question and answers, examples, problem solving; • *laboratory work* on a computer: illustration of content through the set programming tasks.

Načini ocenjevanja:	Delež (v %) Weight (in %)	**Assessment:**
Način (pisni izpit, ustno spraševanje, naloge, projekt): • pisni (ustni) izpit	100	Types (written examination, oral examination, coursework, project): • written (oral) exam