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

Študijski program in stopnja **Study programme and level**	**Študijska smer** **Study field**	**Letnik** **Academic year**	**Semester** **Semester**
Računalništvo in spletne tehnologije, visokošolski strokovni študijski program prve stopnje	-	Prvi	Prvi
Computer Science and Web Technologies, first cycle Professional Study Programme	-	First	First

Vrsta predmeta / Course type	Obvezni / Obligatory
Univerzitetna koda predmeta / University course code:	2-RST-MAG-P-2019-03-05

Predavanja **Lectures**	**Seminar** **Seminar**	**Vaje** **Tutorial**	**Klinične vaje work**	**Druge oblike študija**	**Samost. delo Individ. work**	**ECTS**
30	-	45	-	-	105	6

Nosilec predmeta / Lecturer:	

Jeziki / Languages:	**Predavanja / Lectures:**	Slovenski / Slovenian, Angleški / English
	Vaje / Tutorial:	Slovenski / Slovenian, Angleški / English

Pogoji za vključitev v delo oz. za opravljanje študijskih obveznosti:	**Prerequisites:**
Pogoj za vključitev v delo je osnovno znanje vsaj enega izmed standardnih programskih jezikov in poznavanje koncepta objektno orientiranega programiranja. Pogoj za pristop k izpitu je priprava in zagovor projektne naloge.	It is required for a student to be familiar with at least one of the standard programming languages and knowing the concept of object-oriented programming. To attend the exam, a student has to prepare and present a project assignment.

Vsebina:	**Content (Syllabus outline):**

- Uvod v Visual Studio.
- Uvod v C# (sintaksa in gradniki, razhroščevanje, tipi projektov).
- Napredno objektno-orientirano programiranje (življenjski cikel objektov, polimorfizmi, abstraktni razredi in vmesniki).
- Generični tipi, zbirke in podatkovne strukture (s poudarkom na prostorski in časovni zahtevnosti).
- Anonimni tipi, metode, Lambda izrazi, Jezikovno integrirane poizvedbe (LINQ).
- Niti (uvod v delo z nitmi, delo v ozadju, prioritete, sinhronizacija, zaklepanje, varnost, signaliziranje).
- Uporaba niti (asinhroni vzorci, BackgroundWorker, prekinitve, varne ustavitve, štoparice)
- Paralelizacija (zakaj paralelizacija, uvod v ogrodje PFX, koncepti in komponente).
- Paralelni LINQ (razvrščanje, omejitve, prekinitve, optimizacija)
- Paralelizacija opravil (kreiranje in zagon opravil, čakanje, prekinitve, razporejevalci)

- Introduction to Visual Studio.
- Introduction to C# (syntax and building blocks, debugging, project types).
- Advanced object-oriented programming (objects' life cycles, polymorphisms, abstract classes and interfaces).
- Generics, collections and data structures (with emphasis on space and time complexities).
- Anonymous types, methods, Lambda expressions, Language integrated queries (LINQ).
- Threads (introduction to working with threads, working in background, priorities, synchronization, locking, safety, signaling).
- Using threads (asynchronous pattern, BackgroundWorker, interrupting, safe cancellation, timers).
- Parallelization (why parallelization, introduction to PFX, concepts and components).
- Parallel LINQ (ordering, limitations, cancellation, optimization).
- Task parallelism (creating and running tasks, waiting, cancelling, schedulers).

Temeljni literatura in viri / Readings:

- ALBAHARI, Joseph (2011) Threading in C#, O'Reilly Media Inc.
- JOHNSON, Bruce (2015) Professional Visual Studio 2015, John Wiley & Sons, Inc. , Indianapolis, Indiana.
- PERKINS, Benjamin, HAMMER VIBE, Jacob, REID, Jon D. (2018) Beginning C# 7 Programming with Visual Studio 2017, John Wiley & Sons, Inc., Indianapolis, Indiana.
- RINGLER, Rodney (2014) C# Multithreaded and Parallel Programming, Packt Publishing, Birmingham, UK.
- SKEET, Jon (2014) C# in Depth, Manning Publications Co., USA.

Cilji in kompetence:

Učna enota prispeva k razvoju naslednjih splošnih in predmetno-specifičnih kompetenc:

- Sposobnost prepoznavanja priložnosti za inoviranje in zasnovo novih spletnih storitev in aplikacij.
- Sposobnost algoritmičnega razmišljanja.
- Obvladovanje sodobnih visokozmogljivih orodij in specifične programske opreme za delo z njim.

Objectives and competences:

The instructional unit contributes to the development of the following general and subject-specific competences:

- Ability to recognize opportunities to innovate and develop new online services and applications.
- Ability for algorithmic thinking.
- Management of modern high-performance tools and specific software for working with it.

• Sposobnost samostojnega razvoja zahtevnih programskih rešitev. • Poznavanje in sposobnost uporabe naprednih orodij za razvoj programske opreme. • Razumevanje konceptov vzporednega izvajanja kode in razvoj zahtevnejše programske opreme na osnovi paralelnega procesiranja.	• Ability to independently develop advanced software solutions. • Knowledge and ability to use advanced tools for software development. • Understanding of the concepts of parallel code execution and development of advanced software solutions based on parallel processing.

Predvideni študijski rezultati:	**Intended learning outcomes:**
Znanje in razumevanje: • Študentje bodo spoznali eno izmed najbolj razširjenih orodij za razvoj programske opreme in ga bili sposobni uporabljati pri delu. • Podrobno bodo spoznali napredne koncepte programskega jezika C#, kar bo podlaga za razvoj zahtevnejše programske opreme, ki bo temeljila na modularni sestavi programskih komponent. • Razumeli bodo koncepta niti in vzporednega procesiranja ter ju bili sposobni uporabiti v praksi. Prenosljive spretnosti: • Pridobljeno znanje bo nadgradilo razumevanje razvoja algoritmov. • Podobno, bodo znanja podlaga za študij bolj specifičnih predmetov s področja visokozmogljivega računalništva ter spletnih tehnologij.	Knowledge and understanding: • Students will get acquainted with one of the most common IDEs and will be able to use it in practice. • Students will learn in detail advanced concepts of the programming language C# what will be the base for development of advanced software applications. • Students will understand the concepts of threads and parallel processing and will be able to use them in practice. Transferable skills: • Obtained knowledge will upgrade the understanding of algorithms development. • Similarly, the unit is a basis for study of more specific subjects, such as high-perfomance computing units or web technologies.

Metode poučevanja in učenja:	**Learning and teaching methods:**
• Predavanja z aktivno udeležbo študentov (razlaga, diskusija, vprašanja, primeri). • Vaje (reševanje nalog).	• Lectures with active students participation (explanations, discussion, questions, examples). • Laboratory work (solving problems).

Načini ocenjevanja:	Delež (v %) / Weight (in %)	**Assessment:**
• pisni izpit • projektna naloga	50 % 50 %	• written exam • project assignment