

<b>Naziv predmeta</b>	ANALIZA I DIZAJN SOFTVERA					
<b>Skraćeni naziv</b>	<b>Status</b>	<b>Semestar</b>	<b>ECTS</b>	<b>Fond časova (P+A+L)</b>		
RN-ADS	obavezan	6.	8	3	3	
<b>Šifra predmeta</b>	RN-ADS					
<b>Vrsta i nivo studija, studijski program:</b> Akademске studije prvog ciklusa studija; Studijski program Računarske nauke						
<b>Uslovljenost drugim predmetima:</b> Nema uslova prijavljivanja i slušanja predmeta.						
<b>Ciljevi izučavanja predmeta:</b> Osposobiti studente za proces profesionalne konstrukcije softvera zasnovane na modelovanju faze dizajna i standardizaciji procesa i proizvoda faze konstrukcije softverskih sistema. Studenti treba da usvoje osnovna znanja o principima analize i dizajna sistema iz realnog svijeta korištenjem univerzalnog jezika za OO modeliranje UML i tako se pripreme za rad na konkretnim projektima kroz metodologije softver inženjeringa.						
<b>Ime i prezime nastavnika i saradnika:</b>						
<b>Metod nastave i savladavanje gradiva:</b> Nastava se izvodi u obliku predavanja i vježbi na računaru. Učenje, testovi, domaći radovi, seminarski rad i konsultacije.						
<b>Sadržaj predmeta po sedmicama:</b>						
1.	Osnovni pojmovi konstrukcije softvera.					
2.	Arhitektura softvera, objekti, šabloni, okviri i aplikacije.					
3.	Metafore za bolje razumijevanje razvoja softvera.					
4.	Programerske konvencije.					
5.	Osnovni koncepti dizajna softvera i izrada specifikacije dizajna uz oslonac na objektno modelovanje.					
6.	Statičko i dinamičko modelovanje softvera (klase, interfejsi, dijagrami sekvenci, dijagrami aktivnosti, paketi, raspoređenost).					
7.	Osnovni pojmovi konstrukcije softvera.					
8.	Izbor programskog jezika.					
9.	Prvi test					
10.	Standardizacija vizuelnih i funkcionalnih karakteristika softvera.					
11.	Osnovne postavke kvalitetne konstrukcije softvera i izbor metoda, tehnika i alata za konstrukciju softvera.					
12.	Testiranje na nivou klasa, unit-a, modula, funkcija.					
13.	Izrada test scenarija.					
14.	Rukovanje izuzecima. Modelovanje i implementacija mehanizama zaštite i očuvanja integriteta.					
15.	Struktura programskog koda, makroi i inline rutine, template-biblioteke, primjena rekurzije, dinamičko generisanje koda.					
16.	Kooperativni razvoj softvera i timski rad.					
17.	Drugi test					
<b>Opterećenje studenta po predmetu:</b>						
Nedjeljno:			U semestru:			
Kreditni koeficijent			<b>Ukupno opterećenje za predmet:</b>			
8/30=0,26			8 kredita x 30 sati/kreditu=240 sati			
Nedjeljno opterećenje:			Aktivna nastava: 6 x15=90 sati predavanja i vježbi.			
= 0,26 x 40 sati			<b>Kontinualna provjera znanja: 10 sati</b>			
= 10 sati			<b>Završna provjera znanja: 5 sati</b>			
			<b>Samostalan rad: učenje, seminarski, konsultacije 120 sati</b>			
<b>Obaveze studenta:</b> Studenti su obavezni da: pohađaju nastavu, urade domaće radove, seminarski rad i testove, da rade kolokvije i posjećuju konsultacije.						
<b>Literatura:</b> Vujović V., (2021). Projektovanje i dizajn softvera – koncepti, principi i projektni obrasci, Dis-company, d.o.o. Pale; Phlegers, L. (2006). Software engineering Theory and Practice. CET Computer Equipment and Trade. Beograd.; Booch, G., Rumbaugh, J., Jacobson, I. (2006). Unified Modeling Language User Guide.; Bentley, Whitten, Mc Graw-Hill. (2006). Systems Analysis & Design for the Global Enterprise.						
<b>Oblici provjere znanja i ocjenjivanje:</b> Redovno prisustvo nastavi donosi do 10 bodova, kolokvijumi, testovi domaći radovi i seminarski rad donose do 40 bodova, završni ispit donosi do 50 bodova. Prolazna ocjena se dobije ako se sakupi 55 ili više bodova.						

**Posebna napomena za predmet:** Nema