

1. OPĆE INFORMACIJE																								
1.1. Naziv kolegija	Programsko inženjerstvo i informacijski sustavi	1.6. Semestar	5																					
1.2. Nositelj kolegija	Dr.sc. Mihael Kukec, v. pred.	1.7. Bodovna vrijednost (ECTS)	5																					
1.3. Suradnici		1.8. Način izvođenja nastave (broj sati P+V+S+e-učenje)	30 P + 30 V																					
1.4. Studijski program (stručni, specijalistički diplomski stručni studij)	Stručni studij	1.9. Kratica kolegija	PIIS																					
1.5. Status kolegija (O, I)	O	1.10. Šifra kolegija	5140																					
2. OPIS KOLEGIJA																								
2.1. Ciljevi kolegija	Stjecanje temeljnih znanja o programskom inženjerstvu kroz sintezu vještina i znanja stečenih u prethodnim kolegijima. Upoznaje se proces razvoja informacijskog sustava, s naglaskom na njegovu analizu i oblikovanje te implementaciju sa stanovišta projektanta. Upoznaju se agilne metode te primjena grafičkog prikaza Unificiranim jezikom za modeliranje (engl. Unified Modeling Language – UML). Upoznavanje žiotoznog ciklusa programske podrške.																							
2.2. Uvjeti za polaganje kolegija i ulazne kompetencije koje su potrebne za kolegij, korelativnost i korespondentnost s drugim kolegijima	Nije definirano, ali za polaganje je potrebno položiti - Objektno orijentirano programiranje I, Baze podataka I, Programski alati u programiranju																							
2.3. Očekivani ishodi učenja na razini kolegija (4-10 ishoda učenja)	<table border="1"> <thead> <tr> <th>Red.Br.</th> <th>MINIMALNI ISHODI UČENJA</th> <th>ŽELJENI ISHODI UČENJA</th> </tr> </thead> <tbody> <tr> <td></td> <td>Po uspješnom završetku predmeta, student će moći:</td> <td>Uspješan student bi trebao moći:</td> </tr> <tr> <td>1.</td> <td>Definirati osnovne pojmove i ciljeve upravljanja razvojem programske podrške vezane uz programsko inženjerstvo.</td> <td>Objasniti sveobuhvatne pojmove, složene modele i oblike upravljanja vezane uz programsko inženjerstvo.</td> </tr> <tr> <td>2.</td> <td>Opisati jednostavne principe objektno orijentirane analize i oblikovanja programske podrške kroz sve faze</td> <td>Primijeniti metodu objektno orijentirane analize i oblikovanja programske podrške na jednostavnim primjeru</td> </tr> <tr> <td>3.</td> <td>Opisati modele razvoja programske podrške kroz njihov razvoj.</td> <td>Razlikovati specifičnosti pojedinog modela razvoja programske podrške u odnosu na druge modele</td> </tr> <tr> <td>4.</td> <td>Objasniti osnovu podjelu dijagrama i primjenu UMLa te njegovu svrhu u okviru programskog inženjerstva</td> <td>Primijeniti dijagrame UMLa s obzirom na specifičnost problema koji se rješava</td> </tr> <tr> <td>5.</td> <td>Opisati sastavne elemente slučaja upotrebe te pripadajućeg dijagrama te dijagrama modela domene</td> <td>Razviti scenarije primjene i modele domene za konkretan problem oblikovanja programskog sustava</td> </tr> </tbody> </table>			Red.Br.	MINIMALNI ISHODI UČENJA	ŽELJENI ISHODI UČENJA		Po uspješnom završetku predmeta, student će moći:	Uspješan student bi trebao moći:	1.	Definirati osnovne pojmove i ciljeve upravljanja razvojem programske podrške vezane uz programsko inženjerstvo.	Objasniti sveobuhvatne pojmove, složene modele i oblike upravljanja vezane uz programsko inženjerstvo.	2.	Opisati jednostavne principe objektno orijentirane analize i oblikovanja programske podrške kroz sve faze	Primijeniti metodu objektno orijentirane analize i oblikovanja programske podrške na jednostavnim primjeru	3.	Opisati modele razvoja programske podrške kroz njihov razvoj.	Razlikovati specifičnosti pojedinog modela razvoja programske podrške u odnosu na druge modele	4.	Objasniti osnovu podjelu dijagrama i primjenu UMLa te njegovu svrhu u okviru programskog inženjerstva	Primijeniti dijagrame UMLa s obzirom na specifičnost problema koji se rješava	5.	Opisati sastavne elemente slučaja upotrebe te pripadajućeg dijagrama te dijagrama modela domene	Razviti scenarije primjene i modele domene za konkretan problem oblikovanja programskog sustava
Red.Br.	MINIMALNI ISHODI UČENJA	ŽELJENI ISHODI UČENJA																						
	Po uspješnom završetku predmeta, student će moći:	Uspješan student bi trebao moći:																						
1.	Definirati osnovne pojmove i ciljeve upravljanja razvojem programske podrške vezane uz programsko inženjerstvo.	Objasniti sveobuhvatne pojmove, složene modele i oblike upravljanja vezane uz programsko inženjerstvo.																						
2.	Opisati jednostavne principe objektno orijentirane analize i oblikovanja programske podrške kroz sve faze	Primijeniti metodu objektno orijentirane analize i oblikovanja programske podrške na jednostavnim primjeru																						
3.	Opisati modele razvoja programske podrške kroz njihov razvoj.	Razlikovati specifičnosti pojedinog modela razvoja programske podrške u odnosu na druge modele																						
4.	Objasniti osnovu podjelu dijagrama i primjenu UMLa te njegovu svrhu u okviru programskog inženjerstva	Primijeniti dijagrame UMLa s obzirom na specifičnost problema koji se rješava																						
5.	Opisati sastavne elemente slučaja upotrebe te pripadajućeg dijagrama te dijagrama modela domene	Razviti scenarije primjene i modele domene za konkretan problem oblikovanja programskog sustava																						

	6.	Opisati UML dijagram razreda i sekvencijski UML dijagram	Razviti UML dijagram razreda te sekvencijski UML dijagram na konkretnom slučaju problema oblikovanja programskog sustava	
2.4. Sadržaj kolegija detaljno razrađen prema satnici nastave (kalendar nastave)	Datum	Teme i ishodi		Satnica
	1.	Predavanje: Uvod u kolegij. Temeljni pojmovi, definicija programskog inženjerstva.		2
		Vježba: Upoznavanje s ciljevima i metodama rada tijekom vježbi, upoznavanje s programskim alatima.		2
	2.	Predavanje: Proces stvaranja softvera, faze stvaranja programske podrške, pojam model procesa stvaranja softvera.		2
		Vježba: UML, strukturni i ponašajni dijagrami, specificiranje zahtjeva pomoću slučajeva/scenarija korištenja (engl. Use case), elementi dijagrama slučajeva korištenja (engl. use case diagram).		2
	3.	Predavanje: Modeli procesa stvaranja softvera. Model vodopada, model evolucijskog razvoja, Model formalnog razvoja, Model usmjeren na ponovnu upotrebu, model inkrementalnog razvoja.		2
		Vježba: Izrada specifikacije sustava pomoću slučaja/scenarija korištenja, izrada dijagrama za slučajeve korištenja.		2
	4.	Predavanje: Objektno orijentirana analiza i oblikovanje, definicija, osnovne sastavnice, dijagram modela domene, dijagram razreda i dijagram interakcije.		2
		Vježba: Elementi dijagrama domene, izrada dijagrama domene za postojeće slučajeve primjene		2
	5.	Predavanje: Provjera usvojenog znanja pismeni test koji obuhvaća teoretska znanja i programske primjere		2
		Vježba: Presentacija projekta		2
	6.	Predavanje: Elementi dijagrama modela domene, konceptualni razred, pronalaženje konceptualnih razreda, njegove sastavnice. Primjeri.		2
		Vježba: Izrada dijagrama modela domene za postojeće slučajeve korištenja.		2
	7.	Predavanje: Elementi dijagrama modela domene. Tipovi poveznica, brojnost, sintaksa svojstva dijagrama razreda u svrhu izrade konceptualnog razreda.		2
		Vježba: Izrada dijagrama modela domene za postojeće slučajeve korištenja.		2
	8.	Predavanje: UML dijagram razreda: sastavnice, svojstva i njihova sintaksa, svojstva kao poveznice,		2

		operacije i njihova sintaksa. Vježba: Stvaranje dijagrama razreda.	2	
	9.	Predavanje: UML dijagram razreda, unošenje poveznica među razredima, tipovi poveznice te njihovo značenje: povezivanje, generalizacija, agregacija i kompozicija. Poveznice i brojnost, vidljivost razreda. Vježba: Stvaranje dijagrama razreda, unošenje poveznica među razredima.	2 2	
	10.	Predavanje: Provjera usvojenog znanja pismeni test koji obuhvaća teoretska znanja i programske primjere Vježba: Prezentacija projekta	2 2	
	11.	Predavanje: Interakcije među objektima, sekvencijski dijagram Vježba: Izrada sekvencijskog dijagrama za prethodno stvorene dijagrame razreda.	2 2	
	12.	Predavanje: Sekvencijski dijagram, sastavnice i njihovo značenje, Vježba: Izrada sekvencijskog dijagrama za prethodno stvorene dijagrame razreda.	2 2	
	13.	Predavanje: Obrasci u razvoju programske podrške, raspodjela odgovornosti, obrazac „creator“ Vježba: Implementiranje oblikovanog sustava, provjera i primjena korištenih obrazaca.	2 2	
	14.	Predavanje: Obrasci u razvoju programske podrške, obrazac „controller“, Vježba: Implementiranje oblikovanog sustava, provjera i primjena korištenih obrazaca.	2 2	
	15.	Predavanje: Provjera usvojenog znanja pismeni test koji obuhvaća teoretska znanja i programske primjere Vježba: Prezentacija projekta	2 2	
	2.5. Vrste izvođenja nastave:	<input checked="" type="checkbox"/> predavanja <input type="checkbox"/> seminari i radionice <input checked="" type="checkbox"/> vježbe <input type="checkbox"/> obrazovanje na daljinu <input type="checkbox"/> mješovito e-učenje <input type="checkbox"/> terenska nastava		2.6. Komentari:
		<input checked="" type="checkbox"/> samostalni zadaci <input type="checkbox"/> multimedija i mreža <input type="checkbox"/> laboratorij <input type="checkbox"/> mentorski rad <input type="checkbox"/> (ostalo upisati)		
	2.7. Obveze studenata	<ul style="list-style-type: none"> - Pohađanje predavanja i vježbi: <ul style="list-style-type: none"> - Redovni studenti trebaju prisustvovati na najmanje 80% od ukupnog broja sati predavanja i na najmanje 80% od ukupnog broja sati vježbi da bi ostvarili pravo na potpis. 		

	<ul style="list-style-type: none"> - Izvanredni studenti koji upisuju kolegij trebaju prisustvovati na barem 30% od ukupnog broja sati predavanja i na barem 30% od ukupnog broja sati vježbi da bi ostvarili pravo na potpis. - Aktivno sudjelovanje u nastavnom procesu - Izrada domaćih zadaća u okviru praktičnih programskih projekata - Izlaženje na međuispite <p>Donošenje svog potrebnog pisaćeg pribora na nastavu.</p>					
2.8. Praćenje rada studenata (<i>upisati udio u ECTS bodovima za svaku aktivnost tako da ukupni broj ECTS bodova odgovara bodovnoj vrijednosti kolegija</i>)	Pohađanje nastave	2	Pisani ispit		Projekt	2
	Eksperimentalni rad		Istraživanje		Praktični rad	
	Esej		Referat		Kontinuirana provjera znanja	
	Kolokviji	2	Seminarski rad		(ostalo upisati)	
	Aktivnost u nastavi		Usmeni ispit		(ostalo upisati)	
2.9. Radno opterećenje studenata						
2.10. Ocjenjivanje i vrjednovanje rada studenata tijekom nastave i na završnom ispitu	<p>Kolegij ima definiranih 6 ishoda učenja. Na kolegiju se može osvojiti najviše 200 bodova. Ishodi učenja budu se i provjeravaju se kroz slijedeće metode</p> <p>Prvi teoretski međuispit (M1): do 20 bodova Drugi teoretski međuispit (M2): do 30 bodova Treći teoretski međuispit (M3): do 50 bodova Vježbe L1: do 20 bodova Vježbe L2: do 30 bodova Vježbe L3: do 50 bodova</p> <p>Početni kriterij za izračunavanje ocijene izražen je ovim popisom:</p> <ul style="list-style-type: none"> - 100-125 Dovoljan (2) - 126-150 Dobar (3) - 151-175 Vrlo dobar (4) 					

- 176-200 Izvrstan (5)

Konačan kriterij za izračunavanje ocjene biti će stvoren temeljem Normalne razdiobe ukupno ostvarenih bodova svih studenata na međuispitima M1, M2 i M3 te P1, P2 i P3. Ukoliko je novostvoreni kriterij temeljem Normalne razdiobe nepovoljniji za studente primijeniti će Početni kriterij.

	M1	M2	M3	Projekt	Projekt	Projekt	MAX
				P1	P2	P3	
I1	10			10			20
I2	10			10			20
I3		15			15		30
I4		15			15		30
I5			25			25	50
I6			25			25	50
Ukupno	20	30	50	20	30	50	200

2.11. Obvezna literatura	Red.br.	Naziv
		Nema
	1.	Robert Manger, Mladen Mauher: Programsko inženjerstvo, Algebra 2010., Zagreb
2.12. Dopunska literatura	Red.br.	Naziv
	1.	R. S. Pressman: Software Engineering: A Practitioner's Approach 6/e, McGraw-Hil, 2005
	2.	Mike Cohn, Succeeding with Agile: Software Development Using Scrum, Addison-Wesley Professional; 1 edition, 2009.
	3.	Sommerville, Software Engineering, 9th edition, Addison-Wesley, 2010
3. DODATNE INFORMACIJE O KOLEGIJU		
3.1. Pohađanje nastave	Unatoč dozvoljenom malom broju izostanaka, preporuča se kontinuirani polazak na nastavu u punom obimu predviđene satnice. Pohađanje nastave se provjerava pri početku predavanja i vježbi kontinuirano, za svaki sat nastave tijekom semestra.	
3.2. Kontaktiranje s nastavnikom		
3.3. Informiranje o kolegiju		
3.4. Pisani radovi		
3.5. Ostalo (dodati po potrebi)	<p>U semestru će studenti prikupljati bodove na 3 međuispita.</p> <p>U pravilu, prvi međuispit piše se nakon prvih 4 tjedna nastave i pokriva ishode učenja obrađene u prvih 4 tjedana. Drugi međuispit piše se nakon drugih 9 tjedna nastave i pokriva ishode učenja obrađene do devetog tjedna nastave. Treći međuispit piše se nakon drugih 14 tjedana nastave i pokriva ishode učenja obrađene do 14. tjedna nastave.</p> <p>Vrstu pitanja definira nastavnik, no sva pitanja i zadaci pokrivaju gradivo kolegija odnosno ishode učenja.</p> <p>Student dodatnim radom i zalaganjem kroz izradu domaćih zadaća može ostvariti dodatan iznos bodova kojim njegov ukupan iznos bodova ne prelazi 200.</p>	

