

PLAN I PROGRAM PREDMETA KRATKIH PROGRAMA

Tabela T1: Informacija o predmetu

Redni broj predmeta	12	Naziv predmeta	KI301 Konstruisanje softvera						
Trajanje (dani) ¹	21	Broj dana nastave	18	Broj dana radionice	2	Ukupan broj sati aktivne nastave ²	60	Broj ESPB	6
Broj domaćih zadataka ³	18	Broj testova ³	2	Broj projekata	1	Predmet-preduslov polaganja		nema	
Predmetni nastavnici	Dr Slobodan Jovanović, red. profesor				Mail & Skype	slobodan.jovanovic@metropolitan.ac.rs slobodan.jovanovic08			
Tutori (saradnici)	MSc Nebojša Gavrilović				Mail & Skype	nebojsa.gavrilovic@metropolitan.ac.rs n.gavrilovic			
	Nikola Dimitrijević					nikola.dimitrijevic@metropolitan.ac.rs nikola.dimitrijevic			

Cilj predmeta: Cilj je ovladavanje sa tehnikama, metodama, modelima i računarskim alatima za konstruisanja softvera. Termin konstruisanje označava detaljnu izradu tj. neposrednu proizvodnju softvera i predstavlja kombinaciju detaljnog projektovanja, kodiranja, debugovanja, jediničnog testiranja, i integracionog testiranja. Posle polaganja ovog predmeta student će ovladati strategijama i tehnikama i najsavremenijim softverskim alatima za kodiranje, debugovanje, detaljno projektovanje i jedinično i integraciono testiranje softvera. U okviru ovog predmeta, studenti rade jedan praktičan projekat, da bi demonstrirali da mogu da izrađuju softverske aplikacije uz upotrebu alata i tehnika i strategija za konstruisanje softvera.

Opis predmeta: Program predmeta obuhvata sledeće teme: Upravljanje konstruisanjem, Konstrukciono planiranje, Upravljanje konfiguracijom softvera, Tehnike i pravila kodiranja, Kodiranje klasa i klasnih metoda, Debugovanje klasa i klasnih metoda, Strategije i tehnike projektovanja softvera, Projektovanje i konstruisanje klasa i klasnih metoda, Refaktorisanje klasa i klasnih metoda, Alati za refaktorisanje, Objektno orijentisana merenja softvera, Strategije i tehnike konstrukcionog testiranja, Tehnike integracije softvera i integracionog testiranja, Konstrukciona merenja, Alati za testiranje i debugovanje softvera, Strategije i tehnike poboljšanje performansi softvera, Poboljšanje kvaliteta softvera, Kolaborativno konstruisanje, Statička analiza softvera, Alati za projektovanje softvera, Alati za statičku analizu, Tehnike defanzivnog programiranja, Softverski standardi, Interna i eksterna dokumentacija softvera, itd.

Ishodi učenja predmeta:

- Student razume i zna da primenjuje tehnike i softverske alate za kodiranje softvera
- Student razume i zna da primenjuje strategije i tehnike i softverske alate za razvojno testiranje softvera
- Student razume i zna da primenjuje tehnike i softverske alate za dijagnosticiranje i popravke greški u softveru.
- Student razume i zna da primenjuje strategije i tehnike i softverske alate za detaljni dizajn softvera
- Student je stekao veštinu razumevanja i primene tehnika i softverskih alata za integraciju i integraciono testiranje softvera
- Student je stekao veštinu razumevanja i primene tehnika i softverskih alata za upravljanje konstruisanjem softvera
- Student je stekao veštinu razumevanja i primene tehnika i softverskih alata za obezbeđenja kvaliteta konstruisanja softvera

Deo korpusa znanja koji se izučava na predmetu -

- **SWEBOK V3 Guide (Software Engineering Body of Knowledge) – Oblast znanja: Software Construction:**

PLAN NASTAVE

Tablela T2: Informacija o lekciji predmeta

Lekcija	1	Nastavna jedinica:	Uvod u konstruisanje softvera
Ishod učenja:		<ul style="list-style-type: none"> Primena procesnog modela softvera kod konstruisanja softvera. 	
Korak	Tip	Tematska jedinice	Ishod učenja
1	K	Aktivnosti kod životnog ciklusa softvera. Opis aktivnosti kod životnog ciklusa softvera.	<ul style="list-style-type: none"> Razumevanje liste aktivnosti kod životnog ciklusa softvera.
	P	Primer: Softverski razvojni životni ciklus.	
	Z	Q&A pitanja	
2	K	Poređenje modela životnog ciklusa. Osobine agilnih procesa.	<ul style="list-style-type: none"> Razumevanje osobina pojedinih modela životnog ciklusa.
	P	Primer: Poređenje agilnih i vodopadnog modela.	
	Z	Q&A pitanja	
3	K	Konstruisanje softvera- definicija. Konstruisanje softvera - konkretne aktivnosti. Važnost konstruisanja softvera. Kontekst konstruisanja softvera. Metode i alati za konstruisanje.	<ul style="list-style-type: none"> Razumevanje procesa konstruisanja softvera.
	P	Primer: Konstruisanje softvera u Javi.	
	Z	Q&A pitanje	
4	K	Procesni model softvera – definicija i notacija.	<ul style="list-style-type: none"> Primena procesnog modela softvera kod konstruisanja softvera. Primena softverskog alata za kreiranje procesnog modela.
	P	Primeri: Procesni model za implementaciju jednog modula softvera. .Kreiranje procesnog modela pomoću alata.	
	Z	Zadaci za individualni rad.	
5	K	Pripreme za konstruisanje. Izbor life-cycle modela. Provera zahteva. Provera arhitektonskog dizajna. Cena popravke defekta.	<ul style="list-style-type: none"> Primena priprema za konstruisanje.
	P	Primeri: Modeli životnog ciklusa. Softverska arhitektura. Izbor modela životnog ciklusa.	
	Z	Q&A pitanja, Zadaci za vežbu	
6	DZ	Domaći zadatak DZ01 (dobija se od tutora e-mailom)	Verifikacija stečenog znanja i ishoda učenja
7	F	Forum – pitanje za diskusiju	

Tablela T2: Informacija o lekciji predmeta

Lekcija	2	Nastavna jedinica:	Upravljanje konstruisanjem softvera
Ishod učenja:		<ul style="list-style-type: none"> Primena tehnika upravljanja konstruisanjem softvera. 	
Korak	Tip	Tematska jedinice	Ishod učenja
1	K	Ključne odluke kod konstruisanja. Bitni fakti za donošenje ključnih odluka. Izbor programskog jezika.	<ul style="list-style-type: none"> Primena ključnih odluka kod konstruisanja.
	P	Primeri: Lista programskih jezika i njihovih upotreba. Kako Java program funkcioniše. Tipovi Java aplikacija.	
	Z	Q&A pitanja, Zadaci za vežbu	
2	K	Konstruisanje kod velikih i malih projekata. Proporcija razvojnih aktivnosti kod velikih i malih projekata. Vrste projekata. Produktivnost i defekti kod velikih i malih	<ul style="list-style-type: none"> Razumevanje razlika u konstruisanju kod

		projekata.	velikih i malih projekata.
	P	Primer: Upravljanje velikim projektom.	
	Z	Q&A pitanja	
3	K	Upravljanje konstruisanjem. Tehnike motivacije. Procene veličine projekta. Lista softverskih merenja.	<ul style="list-style-type: none"> Razumevanje procesa upravljanja konstruisanjem.
	P	Primer: Uvod u upravljanje projektom.	
	Z	Q&A pitanja	
4	K	Upravljanje konfiguracijom. Tehnika upravljanja konfiguracijom.	<ul style="list-style-type: none"> Razumevanje tehnike upravljanja konfiguracijom.
	P	Primer: Opis upravljanja konfiguracijom.	
	Z	Q&A pitanja	
5	K	Vežba – Pokazni primeri : Upravljanje konfiguracijom	<ul style="list-style-type: none"> Primena alata Bitbucket za upravljanje konfiguracijom.
	P	Primeri: Upoznavanje version-control alata Bitbucket. Rad sa BitBucket. Video tutorijal za Bitbucket. Integracija Bitbucket i Eclipse	
	Z	Zadaci za samostalan rad.	
6	DZ	Domaći zadatak DZ02 (dobija se od tutora e-mailom)	Verifikacija stečenog znanja i ishoda učenja
7	F	Forum – pitanje za diskusiju	

Tabela T2: Informacija o lekciji predmeta

Lekcija	3	Nastavna jedinica:	Softverski standardi i softverska dokumentacija kod konstruisanja softvera
Ishod učenja:		<ul style="list-style-type: none"> Primena tehnika dokumentovanja kod konstruisanja softvera. Primena softverskih standarda 	
Korak	Tip	Tematska jedinice	Ishod učenja
1	K	Softverski standardi. IEEE Standardi. Standardi relevantni za konstruisanje softvera.	<ul style="list-style-type: none"> Primena standarda relevantnih za konstruisanje softvera.
	P	Primer: IEEE standardi u softverskom inženjerstvu.	
	Z	Q&A pitanja	
2	K	Dokumentacija programa. Tipični softverski dokumenti. Rečnik podataka projekta.	<ul style="list-style-type: none"> Razumevanje dokumentacije softvera.
	P	Primeri: Dokument rečnika podataka. Dokument arhitekture. Dokumentacija softvera.	
	Z	Q&A pitanja.	
3	K	Vizuelna struktura programskog koda. Teorema formatiranja, i stilovi formatiranja. Alternativni stilovi formatiranja.	<ul style="list-style-type: none"> Primena teoreme formatiranja programa.
	P	Primeri: Primeri formatiranja if-iskaza, for-petlje, goto-instrukcije, deklaracije podataka, podprograma, klase, komentara.	
	Z	Zadaci za samostalan rad	
4	K	Interna dokumentacija softvera. Samo-dokumentujući programski kod. Tehnika komentarisanja u programu.	<ul style="list-style-type: none"> Primena tehnike komentarisanja programa.
	P	Primeri: Komentarisanje klase. Komentarisanje podprograma. Primer komentarisanja Java programa.	
	Z	Zadaci za samostalan rad	
5	DZ	Domaći zadatak DZ03 (dobija se od tutora e-mailom)	Verifikacija stečenog znanja i ishoda učenja
6	F	Forum – pitanje za diskusiju	

Tablela T2: Informacija o lekciji predmeta

Lekcija	4	Nastavna jedinica:	Pravila kodiranja
Ishod učenja:		<ul style="list-style-type: none"> Primena pravila kodiranja. 	
Korak	Tip	Tematska jedinice	Ishod učenja
1	K	Upotreba varijabli kod programiranja. Preporuke za inicijalizaciju varijabli. Konvencije za imenovanje varijabli.	<ul style="list-style-type: none"> Primena pravila upotrebe varijabli.
	P	Primer: Imenovanje i inicijalizacija varijabli.	
	Z	Zadaci za individualnu vežbu.	
2	K	Upotreba tipova podataka. Upotreba fundamentalnih tipova. Upotreba neobičnih tipova podataka.	<ul style="list-style-type: none"> Primena pravila upotrebe tipova podataka.
	P	Primeri: Upotreba globalnih podataka.	
	Z	Zadaci za individualnu vežbu.	
3	K	Upotreba sekvencijalnih instrukcija. Mere za otkrivanje i pojašnjavanje međuzavisnosti pojedinih podprograma. Koncept grupisanja povezanih instrukcija	<ul style="list-style-type: none"> Primena pravila upotrebe sekvencijalnih instrukcija.
	P	Primer: Sekvencijalno grupisanje u programu.	
	Z	Zadaci za individualnu vežbu.	
4	K	Upotreba If-instrukcije.	<ul style="list-style-type: none"> Primena pravila upotrebe kondicionalnih instrukcija.
	P	Primeri: Upotreba if-instrukcije. lančaneif-then-else-if instrukcije.	
	Z	Zadaci za individualnu vežbu.	
5	K	Upotreba petlji. Greške kod petlji. Preporuke za petlje.	<ul style="list-style-type: none"> Primena pravila upotrebe petljastih instrukcija.
	P	Primer: For-petlje u Javi.	
	Z	Zadaci za individualnu vežbu.	
6	K	Vežba – Pokazni primeri :Pravila kodiranja	<ul style="list-style-type: none"> Primena pravila kodiranja.
	P	Primeri: Konvencije imenovanja varijabli. Globalni podaci. Grupisanje instrukcija. Kondicionalne instrukcije. Do-while petlje.	
	Z	Zadaci za individualnu vežbu.	
7	DZ	Domaći zadatak DZ04 (dobija se od tutora e-mailom).	Verifikacija stečenog znanja i ishoda učenja
8	F	Forum – pitanje za diskusiju	

Tablela T2: Informacija o lekciji predmeta

Lekcija	5	Nastavna jedinica:	Tehnike kodiranja
Ishod učenja:		<ul style="list-style-type: none"> Primena tehnike strukturnog programiranja. Primena tehnike proceduralnog programiranja. 	
Korak	Tip	Tematska jedinice	Ishod učenja
1	K	Kompleksnost podprograma. Mera kompleksnosti podprograma.	<ul style="list-style-type: none"> Primena tehnike merenja kompleksnosti podprograma.
	P	Primer: Nivo kompleksnosti podprograma.	
	Z	Zadaci za individualnu vežbu.	
2	K	Neobične kontrolne strukture u programskom kodu. Višestruki izlaz iz programa. Instrukcija go-to. Izlazak iz	<ul style="list-style-type: none"> Primena višestrukih

		petlje.	izlaza iz programa i iz petlji.
	P	Primeri: Višestruki izlaz iz programa i iz petlji. Instrukcija go-to.	
	Z	Q&A pitanja, Zadaci za individualnu vežbu.	
3	K	Tehnika strukturnog programiranja. Preporuke kod strukturnog programiranja.	<ul style="list-style-type: none"> Primena tehnike strukturnog programiranja.
	P	Primer: Upotreba if-instrukcija. Primer višestrukog izlaza.	
	Z	Q&A pitanja, Zadaci za samostalni rad.	
4	K	Tehnika proceduralnog programiranja. Upotreba kompjuterske procedure. Upotreba pseudokoda.	<ul style="list-style-type: none"> Primena tehnike proceduralnog programiranja.
	P	Primer: Upotreba pseudokoda.	
	Z	Zadaci za samostalni rad.	
5	K	Primena flow-chart dijagrama . Poređenje sa UML AD.	<ul style="list-style-type: none"> Primena dijagrama kontrole protoka.
	P	Primeri: Primeri primene FC (flow-chart) dijagrama.	
	Z	Zadaci za samostalni rad.	
6	DZ	Domaći zadatak DZ05 (dobija se od tutora e-mailom).	Verifikacija stečenog znanja i ishoda učenja
7	F	Forum – pitanje za diskusiju	

Tablela T2: Informacija o lekciji predmeta

Lekcija	6	Nastavna jedinica:	Konstrukcioni alati
Ishod učenja:		<ul style="list-style-type: none"> Primena konstrukcionih alata. 	
Korak	Tip	Tematska jedinice	Ishod učenja
1	K	Konstrukcioni alati – uvod. Lista konstrukcionih alata. Dizajnerski alati. Editori izvornog koda u okviru IDE. Refaktoreri i metrički reporteri. Biblioteke programa. Version-control alati i rečnici podataka.	<ul style="list-style-type: none"> Razumevanje konstrukcionih alata.
	P	Primer: Razlika između okvira i biblioteke. Biblioteke otvorenog koda.	
	Z	Q&A pitanja.	
2	K	Vežba - Editori za kodiranje	<ul style="list-style-type: none"> Primena kodirajućeg editora.
	P	Primeri: Upotreba kodirajućeg editora. Opcije u editoru za kodiranje.	
	Z	Individualna vežba.	
3	K	CASE alati.	<ul style="list-style-type: none"> Razumevanje CASE alata.
	P	Primer: Alati za razvoj softvera.	
	Z	Q&A pitanja	
4	K	UML alati.	<ul style="list-style-type: none"> Primena UML alata.
	P	Primer: UML alat.	
	Z	Q&A pitanja., Zadaci za vežbu	
5	K	Alati za testiranje i za debugging. Softverske skele. Test-data Generators, i baze podataka greški. Kompajleri, linker i Build Tools.	<ul style="list-style-type: none"> Primena alata za testiranje i debugovanje.
	P	Primer: Softverske skele.	
	Z	Q&A pitanja, Individualna vežba.	
			<ul style="list-style-type: none">
7	K	Vežba –Pokazni primer: Alat FindBugs	<ul style="list-style-type: none"> Primena alata FindBugs.
	P	Primer: Alat FindBugs.	
	Z	Individualna vežba.	

8	DZ	Domaći zadatak DZ06 (dobija se od tutora e-mailom).	Verifikacija stečenog znanja i ishoda učenja
9	T	Test T01 (Usmena odbrana domaćih zadataka)	
10	F	Forum – pitanje za diskusiju	

Tablela T2: Informacija o lekciji predmeta

Lekcija	7	Nastavna jedinica:	Metode testiranja softvera
Ishod učenja:		<ul style="list-style-type: none"> Primena metoda testiranja softvera. 	
Korak	Tip	Tematska jedinice	Ishod učenja
1	K	Uvod u konstrukciono testiranje. Sistematski prilaz razvojnom testiranju. Standardi relevantni za testiranje.	<ul style="list-style-type: none"> Razumevanje sistematskom prilazu testiranju.
	P	Primeri: Ciklusi i faze testiranja softvera.	
	Z	Q&A pitanja.	
2	K	Metode testiranja softvera. Funkcionalno testiranje. Struktorno testiranje.	<ul style="list-style-type: none"> Primena metode funkcionalnog testiranja. Primena metode strukturnog testiranja.
	P	Primeri: Tipovi testiranja softvera. Tehnika testiranja „crne kutije“.	
	Z	Q&A pitanja. Zadaci za vežbu.	
3	K	Test-first metoda	<ul style="list-style-type: none"> Primena metode TDD.
	P	Primer: Test Driven Development (TDD)?	
	Z	Q&A pitanja. Zadaci za vežbu.	
4	K	Struktorno osnovno testiranje.	<ul style="list-style-type: none"> Primena metode strukturnog testiranja.
	P	Primeri: Primer osnovnog strukturnog testiranja. Tehnika testiranja "bele kutije".	
	Z	Q&A pitanja. Zadaci za vežbu.	
5	K	Vežbe – Pokazni primeri : Metode testiranja	<ul style="list-style-type: none"> Primena metoda testiranja softvera.
	P	Primeri: Primer testiranja toka podataka. Primer strukturnog testiranja.	
	Z	Zadaci za samostalni rad.	
6	DZ	Domaći zadatak DZ07	Verifikacija stečenog znanja i ishoda učenja
7	F	Forum – pitanje za diskusiju	

Tablela T2: Informacija o lekciji predmeta

Lekcija	8	Nastavna jedinica:	Praksa (alati i tehnike) testiranja softvera
Ishod učenja:		<ul style="list-style-type: none"> Primena alata i tehnika testiranja softvera 	
Korak	Tip	Tematska jedinice	Ishod učenja
1	K	Alati za podršku testiranja. Izgradnja „Softverske skele“. Test-data Generators, i baze podataka greški.	<ul style="list-style-type: none"> Razumevanje alata za testiranje softvera.
	P	Primer: Vrste alata za testiranje.	
	Z	Q&A pitanje	
2	K	Vežba - Pokazni primeri : Upotreba JUNIT alata	<ul style="list-style-type: none"> Primena alata junit.
	P	Primeri: Jedinično testiranje (Unit-Testing) sa alatom JUnit – Primer. Pokazna vežba - program za rezervaciju karata	

	Z	Zadaci za individualan rad.	
3	K	Vežba – Pokazni primeri : Java alati za testiranje	<ul style="list-style-type: none"> • Primena Java alata za testiranje.
	P	Primeri: Alat Junit. Alat FindBugs.	
	Z	Individualna vežba.	
4	K	Vežba - Pokazni primeri : Konstrukciono testiranje aplikacija	<ul style="list-style-type: none"> • Primena tehnike testiranja aplikacije korišćenjem definisanih testova. • Primena AD HOC testiranja.
	P	Primeri: Testiranje aplikacije korišćenjem definisanih testova. AD HOC testiranje	
	Z	Zadaci za individualan rad.	
5	K	Vežba - Pokazni primeri : Zadaci i pitanja iz testiranja softvera	<ul style="list-style-type: none"> • Primena tehnike testiranja „bele kutije“
	P	Primeri: Primeri testiranja tehnikom bele kutije. Tehnika testiranja bele kutije koristeći dijagrame.	
	Z	Zadaci za individualan rad.	
6	DZ	Domaći zadatak DZ08	Verifikacija stečenog znanja i ishoda učenja
7	F	Forum – pitanje za diskusiju	

Tablela T2: Informacija o lekciji predmeta

Lekcija	9	Nastavna jedinica:	Debugovanje softvera
Ishod učenja:		<ul style="list-style-type: none"> • Primena tehnika i alata za debugovanja softvera. 	
Korak	Tip	Tematska jedinice	Ishod učenja
1	K	Debugging-Uvod. Anatomija greške. Koraćanje kroz program i interaktivni debager.	<ul style="list-style-type: none"> • Primena tehnike koraćanja kroz program i alata za debugovanje.
	P	Primer: Debugovanje Java programa pomoću Eclipse alata.	
	Z	Q&A pitanja. Zadaci za individualan rad.	
2	K	Debugging tehnike. Neefikasni prilazi dijagnozi greške. Naučni prilaz debugging-u. Stabilizacija greške. Popravka greške. Tehnike za nalaženje greški.	<ul style="list-style-type: none"> • Primena tehnika za popravku greški.
	P	Primeri: Upotreba Eclipse debagera.	
	Z	Q&A pitanja. Zadaci za individualan rad.	
3	K	Alati za debugging. Kompajleri, linker i Build Tools. Osobine kompajlera. Dijagnostički alati.	<ul style="list-style-type: none"> • Primena alata za debugging.
	P	Primer: Vežba iz debugovanja (Tutorial : How to Debug a java program in eclipse and other tips or shortcuts).	
	Z	Q&A pitanja. Zadaci za individualan rad.	
4	K	Vežba – Pokazni primeri : Upotreba Java alata za debugovanje	<ul style="list-style-type: none"> • Primena tehnika i alata za debugging.
	P	Primeri: Java debugger-Eclipse Debugging. Kontrolisanje egzekucije programa. Evaluacija varijabli u debugger-u. Upotreba debbuging alata u NetBeans razvojnom okruženju.	
	Z	Zadaci za individualan rad.	
5	DZ	Domaći zadatak DZ09	Verifikacija stečenog znanja i ishoda učenja
6	F	Forum – pitanje za diskusiju	

Tablela T2: Informacija o lekciji predmeta

Lekcija	10	Nastavna jedinica:	Integracija softvera
Ishod učenja:		<ul style="list-style-type: none"> Primena tehnike i alata za integraciju i integraciono testiranje softvera. 	
Korak	Tip	Tematska jedinice	Ishod učenja
1	K	Integracija softvera-Uvod. Prednosti inkrementalne integracije. Poređenje fazne i inkrementalne integracije.	<ul style="list-style-type: none"> Razumevanje metode inkrementalne integracije.
	P	Primer: Šta je to integraciono testiranje?	
	Z	Q&A pitanja.	
2	K	Strategije inkrementalne integracije. Integracija odozgo-nadole. Integracija odozdo-nagore.	<ul style="list-style-type: none"> Primena tehnike za integraciju softvera.
	P	Primer: Jedinično i integraciono testiranje sa Maven i JUnit alatima.	
	Z	Q&A pitanja. Zadaci za vežbu.	
3	K	Procedura integracionog testiranja. Pristupi integracionom testiranju. Integracioni testovi.	<ul style="list-style-type: none"> Primena procedure integracionog testiranja.
	P	Primer: Poređenje integracionog i jediničnog testiranja.	
	Z	Q&A pitanja	
4	K	Alati za integraciono testiranje. Popularni alati za integraciono testiranje (za pisanje integracionih testova).	<ul style="list-style-type: none"> Primena alata za integraciono testiranje.
	P	Primer: Jedinično integraciono testiranje (Unit Integration Testing).	
	Z	Q&A pitanja.	
5	K	Vežbe- Pokazni primeri :Integraciono testiranje	<ul style="list-style-type: none"> Primena tehnike integracionog testiranja.
	P	Primeri: Primer integracionog testiranja. Tutorijal za integraciono testiranje.	
	Z	Zadaci za samostalan rad.	
6	DZ	Domaći zadatak DZ10	Verifikacija stečenog znanja i ishoda učenja
7	F	Forum – pitanje za diskusiju	

Tablela T2: Informacija o lekciji predmeta

Lekcija	11	Nastavna jedinica:	Strategije modelovanja
Ishod učenja:		<ul style="list-style-type: none"> Primena strategija modelovanja. 	
Korak	Tip	Tematska jedinice	Ishod učenja
1	K	Nivoi modelovanja. Hijerarhijska struktura softvera. Podela na podsisteme i restriktivna komunikacija medju podsistemima.	<ul style="list-style-type: none"> Primena nivoa modelovanja.
	P	Primer: Kako uraditi modelovanje visokog nivoa i modelovanje niskog nivoa	
	Z	Q&A pitanja, Zadaci za samostalni rad.	
2	K	Detaljno modelovanje – uvod. Detaljno modelovanje –	

		vrste.	<ul style="list-style-type: none"> Primena detaljnog modelovanja.
	P	Primer: Uvod u detaljno modelovanje (Introduction to Detailed Design).	
	Z	Q&A pitanja., Zadaci za samostalni rad.	
3	K	Faze modelovanja. Modelovanje podataka. Modelovanje interfejsa (Interface design).	<ul style="list-style-type: none"> Primena faza modelovanja.
	P	Primeri: Modelovanje podataka- primer. Modelovanje interfejsa- primer.	
	Z	Zadaci za samostalni rad.	
4	K	Objektno-orijentisano modelovanje – uvod. Veza izmedju OO analize i OO modelovanja. Artifakti o.o. analize i o.o. modelovanja.	<ul style="list-style-type: none"> Primena objektno-orijentisanog modelovanja.
	P	Primer : OO modelovanje (Object Oriented Design)	
	Z	Q&A pitanja. Zadaci za vežbu.	
6	DZ	Domaći zadatak DZ11	Verifikacija stečenog znanja i ishoda učenja
7	F	Forum – pitanje za diskusiju	

Tablela T2: Informacija o lekciji predmeta

Lekcija	12	Nastavna jedinica:	Tehnike modelovanja
Ishod učenja:		<ul style="list-style-type: none"> Primena tehnika za modelovanje softvera. 	
Korak	Tip	Tematska jedinice	Ishod učenja
1	K	Tehnike modelovanja. Tehnika rafiniranja . Modularna tehnika.	<ul style="list-style-type: none"> Primena tehnike rafiniranja. Primena modularne tehnike.
	P	Primer: Modularno modelovanje (Modular Design).	
	Z	Q&A pitanja., Zadaci za samostalni rad.	
2	K	Top-down i bottom-up modelovanje. Top-down i bottom-up programiranje.	<ul style="list-style-type: none"> Primena tehnike modelovanja 'odozgo-nadole' i 'odozdo-nagore'.
	P	Primeri: Modelovanje gore-dole i dole-gore (Top Down and Bottom Up Design).	
	Z	Q&A pitanja, Zadaci za samostalni rad.	
3	K	Tehnika identifikacije naslednosti.	<ul style="list-style-type: none"> Primena tehnike identifikacije naslednosti.
	P	Primeri: Primeri identifikacije naslednosti. Video tutorijal : Identifikacija naslednosti (Object-oriented design: Identifying an inheritance situation).	
	Z	Zadaci za samostalan rad.	
4	K	Detaljno objektno-orijentisano modelovanje. Detaljno modelovanje klasnog dijagrama. Detaljno modelovanje asocijacija. Detaljno modelovanje atributa. Detaljno modelovanje potpisa operacija.	<ul style="list-style-type: none"> Primena detaljnog o.o. modelovanja.
	P	Primer: Modelovanje Java klase.	
	Z	Zadaci za samostalni rad.	
5	DZ	Domaći zadatak DZ12	Verifikacija stečenog znanja i ishoda učenja
6	T	Test T02 (Usmena odbrana domaćih zadataka).	
7	F	Forum – pitanje za diskusiju	

Tablela T2: Informacija o lekciji predmeta

Lekcija	13	Nastavna jedinica:	Konstruisanje klase
Ishod učenja:		<ul style="list-style-type: none"> Primena tehnike konstruisanja klase. 	
Korak	Tip	Tematska jedinice	Ishod učenja
1	K	Modelovanje klase.	<ul style="list-style-type: none"> Primena modelovanja klase.
	P	Primer: Dizajn Java klase (Java Class Design). Uputstva za dizajn Java klase (Java Class Design Guidelines).	
	Z	Q&A pitanja. Zadaci za vežbu.	
2	K	Konstruisanje klase. Procedura konstruisanja klase.	<ul style="list-style-type: none"> Primena postupka konstruisanja klase.
	P	Primeri: Kreiranje Java klase (Creating your own Java Classes).	
	Z	Q&A pitanja. Zadaci za vežbu.	
3	K	Modelovanje visokokvalitetnih klase. Razlozi za kreiranje klase. Razlozi za eliminaciju klase	<ul style="list-style-type: none"> Primena modelovanja kvalitetnih klase.
	P	Primeri: . Vežba iz kohezije klase (Cohesion Tutorial).	
	Z	Q&A pitanja. Zadaci za vežbu.	
4	K	Kvalitetno oklopnjavanje (engl. encapsulation) klase. Pravila oklopnjavanja.	<ul style="list-style-type: none"> Primena kvalitetnog oklopnjavanja klase.
	P	Primer: Oklopnjavanje u Javi (encapsulation in java).	
	Z	Q&A pitanja. Zadaci za vežbu.	
5	K	Kvalitetni klasni interfejs.	<ul style="list-style-type: none"> Primena kvalitetnog interfejsa klase.
	p	Primeri: Primer dobrog interfejsa klase. Primer lošeg klasnog interfejsa. Abstrakcija i kapsulizacija.	
	Z	Q&A pitanja. Zadaci za vežbu.	
6	K	Vežba – Pokazni primeri: Konstruisanje klase.	<ul style="list-style-type: none"> Primena tehnike konstruisanja i dizajna klase.
	P	Primeri: Koraci u konstruisanju klase. Video vežba - Izgradnja Java OOP aplikacije (Build your first OOP application in Java).	
	Z	Zadaci za individualan rad.	
7	K	Transformacija klasnog dijagrama u programski kod (kodiranje pomoću klasnog dijagrama)	<ul style="list-style-type: none"> Primena klasnog dijagrama kod kodiranja
	P	Primeri: Veza između implementacionog klasnog dijagrama i programa, Implementacioni detalji klase, Mapiranje-primer	
	Z	Zadaci za individualan rad.	
8	K	Transformacija sekvencijalnih dijagrama u programski kod (kodiranje pomoću sekvencijalnog dijagrama)	<ul style="list-style-type: none"> Primena sekvencijalnog dijagrama kod kodiranja
	P	Primeri: Implementacioni sekvencijalni dijagram, Tabela mapiranja	
	Z	Zadaci za individualan rad.	
9	DZ	Domaći zadatak DZ13	Verifikacija stečenog znanja i ishoda učenja
10	F	Forum – pitanje za diskusiju	

Tablela T2: Informacija o lekciji predmeta

Lekcija	14	Nastavna jedinica:	Konstruisanje podprograma
Ishod učenja:		<ul style="list-style-type: none"> Primena tehnike konstruisanja podprograma. 	
Korak	Tip	Tematska jedinice	Ishod učenja
1	K	Konstruisanje podprograma. Proces konstruisanja podprograma. Konstruisanje podprograma pomoću PPP.	<ul style="list-style-type: none"> Primena postupka konstruisanja podprograma.
	P	Primer: Primer visokog modelovanja podprograma. Od algoritma i pseudokoda do koda (algorithm to pseudocode to code).	
	Z	Q&A pitanja. Zadaci za vežbu.	
2	K	Kodiranje podprograma. Postupak konstruisanja podprograma. Pisanje programa. Provera programa. Doterivanje programskog koda. Razlozi za formiranje podprograma. Kvalitetni podprogrami.	<ul style="list-style-type: none"> Primena postupka kodiranja podprograma.
	P	Primeri: Primer pisanja programa. Primer upotrebe pseudokoda. Video tutorijal - Komentari, argumenti i parametri u Javi (Java Comments, Arguments and Parameters).	
	Z	Q&A pitanja. Zadaci za vežbu.	
3	K	PDL tehnika. Modelovanje „odozgo-nadole“ pomoću PDL.	<ul style="list-style-type: none"> Primena PDL tehnike .
	P	Primeri: PDL-primer. Vežba iz upotrebe pseudokoda (Algorithm using Flowchart and Pseudo code).	
	Z	Q&A pitanja, Zadaci za vežbu	
4	K	Vežba–Pokazni primeri: Konstruisanje podprograma.	<ul style="list-style-type: none"> Primena tehnike konstruisanja podprograma.
	P	Primeri: Pravila za pisanje pseudokoda. Primer pseudokoda.	
	Z	Zadaci za samostalan rad	
5	K	Primena UML dijagrama aktivnosti . Modelovanje podprograma pomoću DA. Kodiranje podprograma pomoću DA.	<ul style="list-style-type: none"> Primena UML DA kod modelovanja i kodiranja podprograma
	P	Primeri: Primeri primene DA	
	Z	Zadaci za samostalni rad.	
6	DZ	Domaći zadatak DZ14	Verifikacija stečenog znanja i ishoda učenja
7	F	Forum – pitanje za diskusiju	

Tablela T2: Informacija o lekciji predmeta

Lekcija	15	Nastavna jedinica:	Refaktorisanje softvera
Ishod učenja:		<ul style="list-style-type: none"> Primena tehnike refaktorisanja softvera. 	
Korak	Tip	Tematska jedinice	Ishod učenja
1	K	Atributi dizajna. Kriterijum kvaliteta dizajna. Klasifikacija povezanosti.	<ul style="list-style-type: none"> Primena atributa dizajna.
	P	Primeri: Primer analize osobina povezanosti, kohezije, i abstrakcije. Video tutorijal -Kohezija i povezanost softvera (Cohesion and Coupling Tutorial).	

	Z	Q&A pitanja. Zadaci za vežbu.	
2	K	Refaktorisanje programa. Tehnike refaktorisanja programa. Automatizovano refaktorisanje programa. Razlozi za refaktorisanje. Refaktorisanje podprograma. Refaktorisanje klase. Bezbednost refaktorisanja.	<ul style="list-style-type: none"> Primena tehnike refaktorisanja softvera.
	P	Primer: Video vežba - Refaktorisanje koda (Code Refactoring).	
	Z	Q&A pitanja. Zadaci za vežbu.	
3	K	Alati za refaktorisanje.	<ul style="list-style-type: none"> Primena alata za refaktorisanje.
	P	Primer: Video vežba - Praktično refaktorisanje (Practical Refactoring - How to clean code in many small steps).	
	Z	Q&A pitanja, Zadaci za vežbu	
4	K	Pokazna vežba - Refaktorisanje klasa i klasnih metoda.	<ul style="list-style-type: none"> Primena tehnike refaktorisanja softvera.
	P	Primeri: Primer refaktorisanja klasne metode. Primer refaktorisanja klase. Video tutorijal - Refaktorisanje u malim koracima.	
	Z	Zadaci za samostalan rad.	
5	DZ	Domaći zadatak DZ15	Verifikacija stečenog znanja i ishoda učenja
6	F	Forum – pitanje za diskusiju	

Tablela T2: Informacija o lekciji predmeta

Lekcija	16	Nastavna jedinica:	Defanzivno programiranje
Ishod učenja:		<ul style="list-style-type: none"> Primena defanzivnog programiranja. 	
Korak	Tip	Tematska jedinice	Ishod učenja
1	K	Samoprovere programa. Assert funkcije. Upotreba assert-funkcija.	<ul style="list-style-type: none"> Primena assert-funkcija.
	P	Primer: Video vežba - Assert-funkcije u Javi (Assertions in Java).	
	Z	Q&A pitanja. Zadaci za vežbu.	
2	K	Izuzeci. Rukovanje izuzecima.	<ul style="list-style-type: none"> Primena tehnike izuzetaka u Javi.
	P	Primeri: Video vežba -Izuzeci u Javi (Exception Handling in Java).	
	Z	Q&A pitanja. Zadaci za vežbu.	
3	K	Defanzivno programiranje. Tehnike tretiranja greške. Barikade u programu.	<ul style="list-style-type: none"> Razumevanje tehnike tretiranja greške.
	P	Primer: Video vežba - Defanzivni Java kod (Defensive Java Code). Video vežba - Defanzivno programiranje u praksi (Defensive Programming : Best Practices).	
	Z	Q&A pitanja.	
4	K	Pokazna vežba – Primena defanzivnog programiranja.	<ul style="list-style-type: none"> Primena defanzivnog programiranja.
	P	Primeri: Primer defanzivnog programiranja - Java program. Primer pravilnog i nepravilnog unosa u program. Dizajn po ugovoru- primer u Javi.	
	Z	Zadaci za individualan rad.	
5	DZ	Domaći zadatak DZ16	<ul style="list-style-type: none"> Verifikacija stečenog znanja i ishoda učenja.
6	F	Forum – pitanje za diskusiju	

Tablela T2: Informacija o lekciji predmeta

Lekcija	17	Nastavna jedinica:	Kvalitet konstruisanja softvera
Ishod učenja:		<ul style="list-style-type: none"> Primena tehnike obezbeđivanja kvaliteta konstruisanja softvera. 	
Korak	Tip	Tematska jedinice	Ishod učenja
1	K	Kvalitet konstruisanja softvera. Tehnike za povećanje kvaliteta softvera. IEEE standardi za obezbeđenje kvaliteta softvera.	<ul style="list-style-type: none"> Razumevanje tehnika za povećavanje kvaliteta softvera.
	P	Primer: Video vežba - Vežba iz obezbeđenja kvaliteta (Quality assurance tutorial: How to think about quality).	
	Z	Q&A pitanja.	
2	K	Konstrukciona verifikacija softvera. Tehnika provere kodiranja.	<ul style="list-style-type: none"> Primena konstrukcione verifikacije softvera.
	P	Primeri: Video vežba- Pregled programa u praksi (Code Review Best Practices).	
	Z	Q&A pitanja. Zadaci za samostalan rad.	
3	K	Obezbedjenje kvaliteta konstruisanja. Tehnike obezbedjenja kvaliteta konstruisanja. IEEE Standards for SQA Plan.	<ul style="list-style-type: none"> Primena tehnike obezbedjenja kvaliteta konstruisanja.
	P	Primer: Video vežba- Plan obezbeđenja kvaliteta (Quality Assurance Plan).	
	Z	Q&A pitanja. Zadaci za samostalan rad.	
4	K	Formalna inspekcija softvera. Tehnika formalne inspekcije.	<ul style="list-style-type: none"> Primena tehnike formalne inspekcije.
	P	Primer: Video vežba - Šta je to formalna inspekcija ?(What is SOFTWARE INSPECTION?).	
	Z	Q&A pitanja. Zadaci za samostalan rad.	
5	K	Kolaborativno konstruisanje. „Parno programiranje“ (Pair programming). Poredjenje tehnika kolaborativnog konstruisanja.	<ul style="list-style-type: none"> Primena tehnike kolaborativnog konstruisanja.
	P	Primer: Video vežba- Parno programiranje (Pair Programming).	
	Z	Q&A pitanja. Zadaci za samostalan rad.	
6	K	Statička analiza programa. Poredjenje statičke i dinamičke analize.	<ul style="list-style-type: none"> Razumevanje tehnike statičke analize softvera.
	P	Primeri: Video vežba - Statička analiza u praksi. Video vežba - Upotreba alata za statičku analizu .	
	Z	Q&A pitanja.	
7	K	Statička analiza - Pokazna vežba.	<ul style="list-style-type: none"> Primena tehnike statičke analize softvera.
	P	Primeri: Alati za statičku analizu. Procedura statičke analize. Java primer statičke analize. Video tutorijal sa primerima statičke analize. Video tutorijal o code-review tehnikama. Primer sa FindBugs u Eclipse okruženju.	
	Z	Zadaci za samostalan rad.	
8	DZ	Domaći zadatak DZ17	<ul style="list-style-type: none"> Verifikacija stečenog znanja i ishoda učenja
9	F	Forum – pitanje za diskusiju	

Tablela T2: Informacija o lekciji predmeta

Lekcija	18	Nastavna jedinica:	Konstrukcione tehnologije
---------	----	--------------------	---------------------------

Ishod učenja:		<ul style="list-style-type: none"> • Primena table-driven metode. • Primena strategija i tehnika za podešavanje (code-tuning) programa. 	
Korak	Tip	Tematska jedinice	Ishod učenja
1	K	Pregled konstrukcionih tehnologija. "Izvršni modeli" (Executable models).	<ul style="list-style-type: none"> • Razumevanje oblasti konstrukcionih tehnologija.
	P	Primer: Kreiranje xtUML modela (Creating an xtUML Model).	
	Z	Q&A pitanja.	
2	K	Strategije podešavanja softvera. Kompjuterske operacije koje utiču na brzinu softvera.	<ul style="list-style-type: none"> • Razumevanje strategija za podešavanje (code-tuning) programa.
	P	Primeri: Optimizacija koda (Code optimization and performance in Java 8). Performanse Java programa (Java performance techniques).	
	Z	Q&A pitanja.	
3	K	Optimizacija kodiranja. Izbor prave strukture podataka. Izbor jezika, izbor tipova podataka.	<ul style="list-style-type: none"> • Primena tehnika za podešavanje programa.
	P	Primeri: Primer optimizacije. Važnost struktura podataka.	
	Z	Q&A pitanja. Zadaci za vežbu.	
6	DZ	Domaći zadatak DZ18	Verifikacija stečenog znanja i ishoda učenja
7	F	Forum – pitanje za diskusiju	

Tabela T3: Informacija o radionici predmeta

RADIONICA	Izrada projekta	Broj poena	30
Projektni zadatak	Proces konstruisanja softvera		
Objašnjenje zadatka	<p>Kroz projektni zadatak potrebno je obuhvatiti:</p> <ul style="list-style-type: none"> • Naslov i kratak opis problema • Kratka specifikacija korisničkih zahteva (ukratko da bi se stekao uvid u aplikaciju o kojoj se govori) • Kratak opis arhitekture aplikacije (razviti aplikaciju tako da ima minimum pet komponenti) • Modelovati klasni dijagram aplikacije • Modelovanje klasnih metoda pomoću pseudokoda • Izvršiti podelu aplikacije na module (predstaviti svaki modul odvojenim klasnim dijagramom) • Modelovati sekvencijalni dijagram aplikacije • Na osnovu klasnog i sekvencijalnog dijagrama izvršiti implementaciju (minimum dva modula u Java programskom jeziku) • Osmisliti pet test slučajeva i predstaviti ih tabelom (odabrati ad hoc ili deterministički način) • Izvršiti testiranje, debugovanje i integraciju aplikacije • Osmisliti pet test slučajeva integracionog testiranja i predstaviti ih tabelom (korsteći top down ili bottom up tehniku) <p>Za odabrani problem potrebno je prvo ukratko opisati problem a zatim izvršiti specifikaciju korisničkih zahteva. Na osnovu korisničkih zahteva potrebno je</p>		

	<p>izvršiti modelovanje dijagrama arhitekture kao i klasnog dijagrama koji prikazuje sve klase unutar aplikacije. Potrebno je modelovati klasne metode pomoću pseudokoda. Sledeći korak je podela sistema na module gde je prvobitno modelovani klasni dijagram potrebno podeliti na module (odvojene klasne dijagrame koji prikazuju klase tog modula i veze između klasa modula). Modelovati i sekvencijalni dijagram aplikacije. Izvršiti implementaciju minimum dva modula u Java programskom jeziku. Nakon toga u okviru projektnog zadatka potrebno je izvršiti definisanje test slučajeva. Prvo je potrebno osmisliti pet test slučajeva (ad hoc ili determinističkim načinom testiranja) i pet test slučajeva integracionog testiranja (korišćenjem top down ili bottom up tehnike). Izvršiti debugovanje i prikazati rezultate debugovanja.</p>
Izveštaj o realizaciji projekta	<p>Izveštaj treba da sadrži sledeće:</p> <ul style="list-style-type: none"> • Naslov i kratak opis aplikacije na jednoj strani Word dokumenta • Opis korisničkih zahteva na jednoj strani Word dokumenta • Prikaz klasnog dijagrama i klasnih dijagrama za svaki identifikovani modul u vidu slike dijagrama i kratkog opisa • Sekvencijalni dijagram kroz sliku dijagrama i kratak opis • Prikaz implementacije i Java programski kod (prikazati isečke programskog koda sa tekstualnim objašnjenjem) • Test slučajeve predstavljene kroz odgovarajuće tabele • Rezultate debugovanja aplikacije kroz isečke ekrana i tekstualni opis gde se vidi primena alata za otklanjanje grešaka u programskom kodu