

## PLAN I PROGRAM PREDMETA

**IT355 Veb sistemi 2**

**Napomena: Plan i program predmeta važi 12 meseci, od 15. februara 2021. do 15. februara 2022. godine**

Napomena: Ovaj program važi 12 meseci, tj. do početka jesenjeg semestra u školskoj 2021/22.godini, a poslednji ispit se po njemu može polagati u aprilskom roku 2022.godine. Posle ovih rokova, važi novi Plan za predmet koji se objavljuje februara 2022. godine. Studenti koji ne polože ispit do aprila 2022.godine, rade sve predispitne obaveze ponovo, u skladu sa novim Planom i u skladu sa njim polažu ispit u junskom roku 2022. godine ili nekom od kasnijih rokova.

PODACI O NASTAVNOM OSOBLJU	
<b>Klasična nastava - Beograd</b>	
Predavanja	Milićević L. Vladimir
Vežbanja	Nikola Dimitrijević (Beograd i Internet studenti),
e-mail adresa nastavnika	<a href="mailto:vladimir.milicevic@metropolitan.ac.rs">vladimir.milicevic@metropolitan.ac.rs</a>
Skype adresa nastavnika	vladimir.milicevic.kg
Termini za konsultacije nastavnika preko Skype	Ponedeljak od 17 do 19h (ili u nekom drugom terminu u dogovoru sa nastavnikom)
e-mail adresa saradnika	<a href="mailto:nikola.dimitrijevic@metropolitan.ac.rs">nikola.dimitrijevic@metropolitan.ac.rs</a>
Skype adresa saradnika	nikdimmet
Termini za konsultacije saradnika preko Skype	17h-18h ponedeljak, 18h-19h utorak (ili u nekom drugom terminu u dogovoru sa asistentom)
<b>Onlajn nastava (preko Interneta)</b>	
Nikola Dimitrijević	Nikola Dimitrijević
e-mail adresa saradnika za konsultacije sa studentima	<a href="mailto:nikola.dimitrijevic@metropolitan.ac.rs">nikola.dimitrijevic@metropolitan.ac.rs</a>
Skype adresa saradnik za konsultacije sa studentima	nikdimmet
Termini za konsultacije preko Skype	17h-18h ponedeljak, 18h-19h utorak (ili u nekom drugom terminu u dogovoru sa asistentom)
<b>Hibridna nastava u Centru u Nišu</b>	

Vežbe i konsultacije u vezi predavanja	Nikola Dimitrijević
e-mail adresa saradnika	<a href="mailto:nikola.dimitrijevic@metropolitan.ac.rs">nikola.dimitrijevic@metropolitan.ac.rs</a>
Skype adresa saradnika	nikdimmet
Termini za konsultacije saradnika preko Skype	17h-18h ponedjeljak, 18h-19h utorak (ili u nekom drugom terminu u dogovoru sa asistentom)
<b>PODACI O PREDMETU</b>	
Semestar	5
Preduslovi	Veb Sistemi 1
Broj ECTS	8
Broj časova predavanja nedeljno	3
Broj časova grupnih (pokaznih) vežbi nedeljno	1
Broj časova individualnih vežbanja nedeljno	2
Broj časova samostalnog istraživačkog rada nedeljno	/
<b>PODACI O PREDISBITNIM OBAVEZAMA I ISPITU</b>	
Broj domaćih zadataka tokom semestra	15
Maksimalan broj poena za jedan domaći zadatak	1,5
Broj testova tokom semestra	5
Maksimalan broj poena za jedan test	1,5
Broj projekata tokom semestra	1
Maksimalan broj poena za jedan projekat	30
A) Maksimalan broj poena za zalaganje studenta tokom semestra	10
B) Maksimalan broj poena za predispitne obaveze	60
Testovi ( 5 x 1,5)	7,5
Domaći zadaci (15 x 1,5)	22,5
C) Maksimalan broj poena za ispit	30
<b>UKUPAN BROJ POENA (A+B+C):</b>	<b>100</b>
Vreme trajanja ispita u minutima	135 minuta

Forma ispita	Ispit se sastoji iz praktičnog i teorijskog dela. Na teorijskom delu student odgovara na 5 pitanja na papiru. Praktičan deo se sastoji od izrade zadatka na računaru koje je student dobio od asistenta ili predmetnog nastavnika.
Računarski alati ili pribor koji se koriste na ispitu	Računar sa instaliranom podrškom: JDK 8, NetBeans ili Eclipse IDE sa podrškom za Spring, Spring Boot, Hibernate, JPA i Maven. Strogo je zabranjena upotreba mobilnih telefona, kao i Interneta (na teorijskom delu ispita). Obavezno poneti indeks. Papir nije potrebno donositi – dobija se u učionici pred ispit.

### Osnovna Literatura:

1. Vladimir Milićević, IT355 Web sistemi 2, autorizovana predavanja u elektronskom obliku, Fakultet informacionih tehnologija, Beograd, 2019/2020

### Dopunska literatura:

2. Marten Deinum, Josh Long, and Daniel Rubio, Spring 5 Recipes Fourth Edition, Apress
3. Spring Framework Reference Documentation - <https://docs.spring.io/spring-framework/docs/current/spring-framework-reference/html/index.html>
4. Craig Walls, Spring in Action, Manning
5. Craig Walls, Spring Boot in Action, Manning

### Veb linkovi:

1. <http://www.javacodegeeks.com/tutorials/java-tutorials/enterprise-java-tutorials/spring-tutorials/>
2. <http://www.tutorialspoint.com/spring/>
3. <http://www.javatpoint.com/spring-tutorial>

### Cilj predmeta

Predmet uvodi studente u oblast projektovanja i programiranja serverske strane (Back-end). Izučavaju se veb klijent-server i višeslojne arhitekture, servisno orijentisana arhitektura, aspekt-orijentisano programiranje (AOP), inverzija kontrole (IoC) sa ubacivanjem zavisnih komponenti (DI), J2EE tehnologija i druge savremene tehnike. Kroz ovaj predmet student će biti upoznat sa Java baziranim okvirom za projektovanje i programiranje složenih veb aplikacija - Spring framework, tehnikama osiguranja bezbednosti veb sistema, tehnikama i alatima testiranja veb aplikacija. Posebno, cilj predmeta je savladavanje sledećih Spring tema: savladavanje osnovnih i naprednih Spring IoC koncepata, savladavanje tehnika aspektno-orijentisanog programiranja u Springu, rešavanje problema bezbednosti web aplikacija u Springu, upravljanje web tokovima u Springu, rad u Spring MVC okviru, savladavanje Spring REST servisa, rad sa bazama podataka (data access) u Springu kroz okvire Spring JDBC, Spring ORM

i Hibernate, testiranje web aplikacija kroz Spring podršku za JUnit4 i TestNG, rad sa sistemskim porukama u Springu, kreiranje portlet Spring aplikacija, primena Spring Boot – a za olakšavanje podešavanja i razvoja Spring aplikacija, integracija okvira Angular sa Springom.

## Opis predmeta

Predmet obrađuje napredni Spring okvir za razvoj JAVA web aplikacija. Teme su veoma pažljivo birane da bi studenti stekli predstavu o svim ključnim Spring konceptima i principima – odgovarajućim podokvirima ugrađenim u glavni Spring paket, i da bi mogli efikasno i efektivno da kreiraju Spring web aplikacije srednjeg nivoa složenosti i da steknu odličnu osnovu za upoznavanje naprednih Spring koncepata i principa. Tokom izučavanja predmeta student će savladati osnovne i napredne koncepte IoC (Inversion of Control) u Spring okviru. Za podršku implementaciji najvećeg stepena modularnosti i efikasnosti objektno – orijentisanih sistema, student će savladati primenu tehnika aspektno-orijentisanog programiranja u Springu. Posebno bitan akcenat je izučavanje najnovijeg bezbednosnog Spring podokvira – Spring Security, posebno nad back – end servisima. Studenti, takođe, detaljno izučavaju upravljanje veb tokovima u Java veb aplikacijama primenom Spring Flow podokvira. Posebno je značajno izučavanje **back – end** servisa koji se grade u Java aplikacijama primenom najnovije generacije Spring 4 okvira. Studenti će detaljno, u vezi sa navedenim, učiti da programiraju nad Spring MVC okvirom, kreiraju i koriste RESTful servise, kreirane pomoću Spring okvira, razvijaju baze podataka i servise nad njima primenom ORM pristupa koji uključuje primenu JPA i Hibernate okvira u sinergiji sa Springom. Posebno, kao najnapredniji način komuniciranja sistemskih komponenata, studenti će izučavati JMS (Java Messaging System) u Springu, koji predstavlja pouzdaniju i napredniju alternativu za specifični RMI (Remote Method Invocation) Java API. Posebno, studenti će da uče kako da razvijaju portlet aplikacije u Springu i mehanizme njihovog angažovanja na brojnim veb portalima. Kao poseban dodatak, a u svrhu implementacije najnovijih trendova iz Java i framework programiranja, izučavaće se specifični okvir Spring Boot koji vlastitim alatima nudi značajno olakšavanje podešavanja i razvoja Spring aplikacija. Kao podršku uspešnom razvoju Java veb aplikacija primenom, Spring i Spring Boot okvira, student će učiti da piše i izvršava testove JUnit 4 i TestNG, primenom Spring Test.

## Ishodi učenja predmeta

- Student razume arhitekturu Spring okvira sa svim podokvirima i modulima;
- Student razume i koristi Spring IoC kontejner;
- Student koristi konfiguracije zrna u IoC kontejneru;
- Student razume i koristi napredne elemente Spring IoC kontejnera;
- Student razume i koristi okvire Spring AOP i Aspectj;
- Student koristi i upravlja web tokovima u Springu;
- Student piše testove za Spring aplikacije;
- Student upravlja perzistencijom, koristi Spring JDBC i ORM;
- Student koristi RESTful servise u Spring aplikacijama;
- Student je osposobljen da koristi bezbednosne koncepte u Spring aplikacijama;
- Student je osposobljen da kreira širok spektar web aplikacija u Spring okviru.

- Student će biti osposobljen za rad sa porukama u Spring sistemima i kreiranje Portlet aplikacija;
- Student koristi Spring Boot za podešavanje i razvoj jednostavne Spring aplikacije, kao i za testiranje
- Student koristi Grails uSpring Boot – u, Aktuatore i razvija složenije aplikacije primenom Spring Boot.
- Student koristi integraciju okvira Angular i Spring za kreiranje MVC aplikacija

### Deo korpusa znanja koji se izučava na predmetu :

Predmet primenjuje korpus znanja (the Body of Knowledge): **Information Technology 2008** - Curriculum Guidelines for Undergraduate Degree Programs in Information Technology, Nov. 2008, Association for Computing Machinery (ACM), IEEE Computer Society.

Knowledge area	Knowledge unit	Topic	Lekcija	Naziv objekta učenja
Platform Technologies (PT)	PT. Enterprise Deployment Software	Middleware frameworks	L01	Umetanje zavisnosti i inverzija kontrole
			L01	Moduli spring okvira
			L01	Scenariji upotrebe okvira spring 4
			L01	Upravljanje zavisnošću i konvencija naziva
			L01	Logovanje
			L01	Spring 4 noviteti
			L01	Spring 4.1 noviteti
			L01	Spring 4.2 noviteti
			L01	Spring 4.3 noviteti
			L01	Spring 5 - noviteti
			L01	Pokazni primer - maven konfiguracioni fajl

		Configuration, definition and management	L01	Vežba 1- maven projekat
			L01	Domaći zadatak 1
			L15	Generalni opis sistema
			L15	Specifični zahtevi
			L15	Funkcionalni zahtevi
			L15	Dijagrami slučajeva koriscenja
			L15	Nefunkcionalni zahtevi
			L15	Zahtevi baze podataka
			L15	Dijagrami aktivnosti
			L15	Modeli analize sistema
			L15	Dijagrami sekvenci
			L15	Dijagrami komunikacije
			L15	Implementacioni dijagrami
			L15	Vežba 15
			L15	Domaći zadatak 15
<b>Web Systems &amp; Technologies (WS)</b>	WS. Web Technologies	Server-side programming	L02	Inverzija kontrole
			L02	Umetanje zavisnosti
			L02	Di i ioc - pokazni primer
			L02	Spring ioc container
			L02	Umetanje zavisnosti u springu
			L02	Umetanje zavisnosti u springu - pokazni primer
			L02	Kompatibilnost unazad - xml konfigurisanje
			L02	Lekcija 2 - vežbe
			L02	Domaći zadatak 2
			L05	Dispatcher servlet
			L05	Mapiranje zahteva sa @requestmapping

			L05	Presretanje zahteva
			L05	Rešavanje pogleda pomoću naziva
			L05	Pridruživanje vrednosti u kontroleru
			L05	Upravljanje formama pomoću kontrolera
			L05	Spring mvc - pokazna vežba
			L05	Spring mvc - individualna vežba 5
			L05	Domaći zadatak 5
			L06	Spring boot uvodna razmatranja
			L06	Razvoj prve spring boot aplikacije
			L06	Spring boot mvc - pokazni primer i početna analiza
			L06	Primena automatskih konfiguracija
			L06	Prilagođavanje spring podešavanja
			L06	Redefinisanje spring boot podešavanja
			L06	Proširivanje podešavanja pomoću osobina
			L06	Prilagođavanje prikazivanja grešaka veb aplikacije
			L06	Pokazni primer - gradle upravljanje zavisnostima
			L06	Vežba 6
			L06	Domaći zadatak 6
			L13	Razvoj spring serverske komponente
			L13	Kreiranje spring maven projekta
			L13	Kreiranje i podešavanje baze podataka
			L13	Kreiranje java konfiguracija
			L13	Kreiranje modela
			L13	Kreiranje dao i servisnog nivoa
			L14	Razvoj jednostavnog portleta
			L14	Mapiranje portlet zahteva u rukovaoce

			L14	Rukovanje formama portleta
			L14	Pokazni primer - angažovanje portleta
			L14	Pokazna i individualna vežba 14
			L14	Domaći zadatak 14
			L15	Implementacija i struktura
			L15	Api
			L15	Podaci
			L15	Repozitorijumi
			L15	Vežba 15
			L15	Domaći zadatak 15
			L13	Lekcija 13 - pokazna i individualna vežba
			L13	Domaći zadatak 13
		Emerging technologies	L03	Podrška aspectj anotacijama u springu
			L03	Deklarisanje aspekata pomoću aspectj anotacija
			L03	Informacija tačke dodira
			L03	Određivanje prednosti među aspektima
			L03	Ponovna upotreba definicije tačke prekida
			L03	Izrazi presečnih tačaka
			L03	"predstavljanje" ponašanja zrnju
			L03	Predstavljanje stanja zrnima
			L03	Deklarisanje aspekata xml konfiguracijama
			L03	Upredanje aspectj aspekata u springu
			L03	Podešavanje aspectj aspekata u springu
			L03	Umetanje spring zrna u objekte domena
			L03	Vežba 3
			L03	Domaći zadatak 3



			L07	Groovy i spring boot cli
			L07	Razvoj spring boot cli aplikacija
			L07	Testiranje spring boot cli aplikacija
			L07	Grails u spring boot
			L07	Objektno - relaciono mapiranje u grails
			L07	Groovy server pages
			L07	Aktuator
			L07	Upotreba krajnjih tačaka aktuatora
			L07	Vežba 7
			L07	Domaći zadatak 7
		Standards & standards bodies	L04	Jsp arhitektura
			L04	Jsp životni ciklus
			L04	Jsp sintaksa
			L04	Jsp direktive
			L04	Jsp akcije
			L04	Jsp - implicitni objekti
			L04	Procesiranje forme
			L04	Upravljanje kolačićima u jsp stranicama
			L04	Preusmeravanje jsp stranica
			L04	Jstl - javax server pages standard tag library
			L04	Jsp - javabeen
			L04	Jsp - jezik izraza
			L04	Vežbe 4 - jsp - osnove
			L04	Domaći zadatak 4
		Web services	L09	Objavljivanje xml baziranih rest servisa springom
			L09	Objavljivanje json baziranih rest servisa springom

			L09	Pristup rest servisima u springu 4 i 5
			L09	Pokazni primer 1 - jednostavni restful servis
			L09	Pokazni primer - rest servis crud
			L09	Vežbe 9 - pokazna i individualna vežba
			L09	Domaći zadatak 9
			L13	Rest kontoler
			L13	Lekcija 13 - pokazna i individualna vežba
			L13	Domaći zadatak 13
		Client-side programming	L13	Razvoj angular klijentske komponente
			L13	Kreiranje angular projekta
			L13	Programske datoteka angular projekta
			L13	Servisi klijenta
			L13	Pozivi metoda servisnog nivoa klijenta
			L13	Lekcija 13 - pokazna i individualna vežba
			L13	Domaći zadatak 13
	WS. Architecture	Information	L05	Mapiranje izuzetaka u pogledu
			L05	Spring mvc - pokazna vežba
			L05	Spring mvc - individualna vežba 5
			L05	Domaći zadatak 5
		Usability	L07	Angažovanje spring boot aplikacije
			L07	Pokazni primer - spring boot actuator
			L15	Testiranje sistema
			L15	Demonstracija rada sistema
			L15	Vežba 15
			L15	Domaći zadatak 15
			L07	Vežba 7

	WS. Web Development	Database integration	L07	Domaći zadatak 7
			L08	Problem direktnog korišćenja orm okvira
			L08	Jpa i hibernate
			L08	Podešavanje produkcije orm resursa u springu
			L08	Hibernate kontekstualne sesije
			L08	Jpa umetanje konteksta
			L08	Pokazna vežba 1 - spring 4/5, hibernate 5 i jpa 2
			L08	Pokazna vežba 2 - spring 4/5, hibernate 5 bez xml
			L08	Individualna vežba 8 - spring mvc i hibernate
			L08	Domaći zadatak 8
	WS. Vulnerabilities	Client Security	L10	Bezbednost url pristupa
			L10	Spring security csrf zaštita
			L10	Prijavljivanje u spring veb aplikaciji
			L10	Provera korisnika
			L10	Upravljanje bezbednošću u pogledima
			L10	Domaći zadatak 10
			L10	Vežba 10 - pokazna i individualn vežba
		Server security	L10	Upravljanje kontrolom pristupa
			L10	Zaštita poziva metoda
			L10	Vežba 10 - pokazna i individualn vežba
<b>Human Computer Interaction (HCI)</b>	HCI. Developing Effective Interfaces	Localization, globalization	L05	Podrška za internacionalizaciju
			L05	Spring mvc - pokazna vežba
			L05	Spring mvc - individualna vežba 5
			L05	Domaći zadatak 5
		Usability testing	L07	Testiranje pomoću spring boot

	HCI. Human-Centered Evaluation		L07	Integracioni testovi u spring boot - u
			L07	Testiranje veb aplikacije
			L07	Testiranje veb bezbednosti
			L07	Testiranje izvršavanja aplikacije
			L07	Vežba 7
			L07	Domaći zadatak 7
<b>Integrative Programming and Technologies (IPT)</b>	IPT. Intersystem Communications	Message and queuing services	L11	Slanje i prijem jms poruka u springu
			L11	Rad sa porukama bez jmstemplate spring podrške
			L11	Rad sa porukama sa jmstemplate spring podrškom
			L11	Rad sa porukama sa podrazumevanim destinacijama
			L11	Primena klase jmsgatewaysupport
			L11	Konverzija jms poruka
			L11	Upravljanje jms transakcijama
			L11	Kreiranje pojo-a baziranih na porukama u springu
			L11	Kreiranje konekcije
			L11	Pokazni primer - spring 4 / 5 + jms + activemq
			L11	Pokazna i individualna vežba 11
			L11	Domaći zadatak 11
<b>System Integration and Architecture</b>	SIA. Integration and Deployment	System release: pilot and acceptance testing and defect repair	L12	Kreiranje testova pomoću okvira junit4 i testng6
			L12	JUnit 4 testiranje
			L12	Testng 6 testiranje
			L12	Kreiranje jediničnih i integracionih testova
			L12	Jedinično testiranje spring mvc kontrolera
			L12	Kontekst aplikacije i integracioni test
			L12	Umetanje test alata u integracioni test
			L12	Upravljanje transakcijama u integracionom testu

			L12	Pristup bazi podataka u integracionom testu
			L12	Primena opštih test anotacija u springu
			L12	Pokazna vežba - spring 4 junit4 test
			L12	Individualna vežba 12
			L12	Domaći zadatak 12

**Način ocenjivanja:** Student se ocenjuje u toku celog semestra. Ocenjuju se njegovi domaći zadaci, testovi i aktivnost u nastavi. Na kraju u ispitnom roku, ocenjuje se i pisani ispit. Ocene se daju u poenima. Maksimalni broj poena je 100 (uključujući i pisani ispit). Na pisanom ispitu student može dobiti do 30 poena, a aktivnosti u toku semestra (predispitne obaveze) mogu mu doneti do 70 poena, po sledećoj strukturi:

- **Domaći zadaci – 22.5 poena:** Posle izučavanja određene nastavne jedinice, odnosno, posle vežbi u okviru jednog predavanja (jedna nedelja) predviđeno je da studenti dobiju zadatak koji treba samostalno da reše. Svaki student dobija različit zadatak i kao preduslov za izlazak na ispit u obavezi je da uradi i pošalje sve zadatke. Predviđeno je ukupno **15 zadataka**, a svaki uspešno rešen zadatak predat u zadatom roku obezbeđuje studentu **maksimalno 1,5 poena**, pod uslovom uspešne odbrane urađenog zadatka. Za studente tradicionalne (u Beogradu) ili hibridne nastave (Centar u Nišu) rok za predaju zadataka je **7 dana nakon izdavanja**, a **posle tog roka umanjuje ostvaren broj poena za 50%**. Krajnji rok za predaju domaćeg zadatka i za studente tradicionalne nastave i za studente onlajn nastave je 10 (deset) dana pre ispitnog roka u kome student polaže ispit.

Projektovano vreme potrebno za izradu zadataka iznosi:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
2,5h	2,5h	3,0h	2,5h	2,5h	2,5h	3,0h	2,5h	2,5h	2,5h	3,0h	3,0h	2,5h	3h	2,5h

**Upustvo za dostavu domaćih zadataka:** Po urađenom domaćem zadatku, student ga dostavlja na email asistentu. Naziv fajla u kome student dostavlja domaći zadatak glasi **IT355\_Ime\_Prezime\_BrojIndeksa\_DZXX**. Na primer, ako želim da predam prvi domaći zadatak tada naziv fajla mora biti IT355\_ Vladimir\_Milićević\_1976\_DZ01, pod pretpostavkom da mi je broj indeksa 1976. U fajlu mora stojati postavka zadatka, čije rešenje šalžete. U slučaju da imate više fajlova u okviru jednog domaćeg zadatka treba ih numerisati redosledom kako treba da ih pregleda asistent. Na primer, ako za prethodni primer imam dva fajla koja dostavljam asistentu tada ih treba označiti sa IT355\_ Vladimir\_Milićević\_1976\_DZ01\_deo01 i IT355\_ Vladimir\_Milićević\_1976\_DZ01\_deo02. Svi fajlovi poslani u drugačijoj formi od propisane, biće vraćani bez pregledanja, sve dok se ne dovedu u propisanu formu.

- **Projekat - 30 poena:** Svaki student dobija jedan projekt od asistenta kojim treba da pokaže svoju spremnost da primeni stečena znanja u konkretnom primeru. Procenjeno vreme rada na projektu je 20 sati. Ako student prilikom ocenjivanja projekta ne dobije najmanje 50% predviđenih poena (15 poena), on mora da ga doradi. U suprotnom, dobija 0 poena, tj. ne može dobiti manje od 50% predviđenih poena. **Student koji ne dobije više od 50% predviđenih poena ne može izaći na ispit.** Za studente tradicionalne nastave rok za predaju projektnog zadatka je **15 dana od kraja semestra**, a **posle tog roka umanjuje se broj ostvarenih poena za 30%**. Krajnji rok za predaju projekta i za studente tradicionalne nastave i za studente onlajn nastave je 10 (deset) dana pre ispitnog roka u kome student polaže ispit.

- **Testovi – 7.5 poena** – U skladu sa Planom nastave (videti donju tabelu) predviđeno je da student u naznačenim nedeljama treba da uradi ukupno **5** testova, čime može da obezbedi najviše **7,5** poena (svi testovi se boduju maksimalno sa po 1,5 bodova). Student mora test da uradi u nedelji datoj u Planu nastave (donja tabela), tj. u roku od 7 dana po predavanju. **Za svaki test urađen van datog termina, poeni dobijeni testom se umanjuju za 50%**, s tim što je krajnji rok za polaganje testa – 10 dana pre ispita. Studenti onlajn nastave testove moraju uraditi najkasnije 10 dana pre ispita ispita (bez umanjenja broja poena). Predmetni profesor ili nadležni asistent poslaće dodatno obaveštenje o testovima (u nedelji kada je predviđena izrada testova).
- **Testovi za samotestiranje u okviru lekcija:** Svaka lekcija ima dva testa za samotestiranje studenta. Student je položio test ako je tačno odgovorio na 5 od 6 postavljenih pitanja (koji se slučajno programski generišu iz baze pitanja). Ukoliko u prvom pokušaju student to ne ostvari, može preko mape uma koja je priložena uz test, da pretražuje nastavni materijal da bi našao tačne odgovore, i da nastavi rešavanje otvorenog testa. To može da ponavlja bez ograničenja, sve dok ne dobije bar 5 od 6 tačnih odgovora, tj. sve dok test bude uspešno urađen. Ove godine, neuspešan test ne ograničava da student nastavi čitanje preostalog nastavnog materijala lekcije, ali time gubi mogućnost da dobije 10 poena na zalaganje, koji se dodeljuju studentu koji uspešno uradi testove u svim lekcijama, i koji je učestvovao na forumima svih lekcija, ali i koji nema više od 5 izostanka sa predavanja i vežbi tokom semestra (ako je student klasične nastave). Da bi to ostvario, student mora da uspešno položi ove testove u lekcijama u roku od 7 dana od dana održavanja predavanja po rasporedu nastave. Ostvarivanjem 10 poena na zalaganje (videti dole), student dobija jednu ocenu veću posle završnog ispita. Cilj je da se stimuliše stalni rad studenta na učenju tokom semestra (a ne kampanjsko učenje pred ispit) i njegova bolja priprema za nastavu u okviru sledeće lekcije.
- **Forum:** Cilj foruma je da podstiče proaktivni rad studenata i njihovu saradnju tokom nastave na predmetu. Svaki forum omogućava da student postavi problem, vezan za lekciju, i da potraži pomoć svojih kolega. Bilo da se problem odnosi na deo predavanja, bilo na deo vežbi. Za postavljanje problema na forum, studentu se priznaje aktivnost na forumu. Isto se priznaje aktivnost na forumu i studentu koji prvi reaguje na ovaj poziv studenta za pomoć, pod uslovom da je njegov savet uspešno rešio problem koji je student postavio. Ovo znači, da forum može imati i onoliko postavljenih problema, koliko ima studenata. Student koji ima evidentiranu aktivnost na opisan način na forumu svake lekcije, i ako ima uspešno urađene testove u okviru svih lekcija, dobija 10 poena na zalaganje, pod uslovom da nema više od 5 izostanka sa predavanja i vežbi tokom semestra (ako je student klasičnih studija).
- **Zalaganje studenta u nastavi u toku semestra – 10 poena:** Aktivnost u nastavi (određivanje poena za zalaganje) se ocenjuje:
  - Kod studenata tradicionalnog oblika nastave (Beograd i Niš) primenom sledećih kriterijuma:
    1. Dolazi pripremljen za nastavu, čitanjem nastavnih materijala pre predavanja i vežbi
    2. Redovnost u pohađanju nastave. **Student tradicionalne koji nije, iz bilo kog razloga, pohađao nastavu na više od 30% časova predavanja i vežbanja, tj. koji ima više od 5 izostanaka (bez obzira na razlog izostanka), dobija automatski 0 poena na zalaganje.** Ovo ograničenje ne važi samo u specijalnim slučajevima kada je student zbog duže bolesti sa ili bez bolničkog tretmana, ali uz potvrdu nadležnog lekara, bio prinuđen da odsustvuje iz nastave više od 5 nedelja.
    3. Redovnost i kvalitet ispunjenja predispitnih obaveza. **Student koji ne uradi sve svoje predispitne obaveze najkasnije 15 dana po završetku nastave na predmetu, dobija nula (0) poena na zalaganje**

4. **Student koji uradi sa uspehom sve testove u okviru lekcija i uspešno reši problem koji je tema foruma u svim lekcijama dobija maksimalnih 10 poena za zalaganje**, pod uslovom da je ispunio uslove pod 2. i 3. Student mora da reši testove u lekcijama i problem u forumu lekcije u roku od 7 dana od dana održavanja predavanja.
  5. Učestvovanja u diskusijama na vežbama i predavanjima, kao i učestalost konsultacija tokom izrade projekta,
    - Kod onlajn studenata (studije preko Interneta):
      1. Ranije (u odnosu na ispit) ispunjenje predispitnih obaveza
      2. Konsultacije tokom semestra u vezi rada na predispitnim obavezama
      3. Učešće na forumu predmeta (LAMS)
- **Pismeni ispit – 30 poena:** Pravo polaganja ispita ima student koji je stekao najmanje 35 poena realizacijom svojih predispitnih obaveza, koji je pokušao da uradi sve predviđene testove. Student prijavljuje ispit na kraju semestra, odnosno pre svakog ispitnog roka, u roku koji objavi uprava univerziteta. **Prijavljeni ispit se ne može otkazati.** Taksa za polaganje ispita se naplaćuje i studentu koji je prijavio polaganje ispita, a nije se pojavio na ispitu..

### Pravila vezana za predmet:

**Nastava:** Student na tradicionalnoj ili hibridnoj nastavi mora da dođe pripremljen na svako predavanje, odnosno vežbu time što je unapred pročitao pripremljeno gradivo za dato predavanje preko LAMS-a i pripremio se za diskusiju na predavanju, odn. za rešavanje zadataka na vežbama, kao i postavljanje pitanja u vezi pojedinih nejasnoća sa kojima se suočio tokom proučavanja gradiva.

**Konsultacije:** Svim studentima nastavnik i asistent su na raspolaganju tokom cele školske godine za konsultacije u prostorijama Univerziteta, na Skype-u u terminu prema dogovoru sa studentima, primenom pričaonice (čata) ili foruma predmeta koje nudi LAMS, ili preko mejlova.

**Mentorski rad:** Svaki student dobija svog mentora koji mu pomaže tokom studiranja na fakultetu. Svaki student dobija obaveštenje ko mu je mentor i ima mogućnost da kontaktira mentora po bilo kom pitanju koje se tiče njegovog uspešnog rada na fakultetu. Svaki mentor je u obavezi da kontaktira studente za koje je odgovoran kako bi utvrdio eventualne potrebe ili probleme sa kojima se student suočava i kako bi mu pomogao u rešavanju istih.

**Domaći zadaci:** U toku semestra se rade domaći zadaci. Svaki student dobija od asistenta poseban domaći zadatak. Po urađenom zadatku, student ga dostavlja na email asistentu u skladu sa oznakom fajla koja je data u postavci zadatka. Asistent može vratiti studentu predat domaći zadatak na doradu, ako smatra da ga nije dobro uradio, sa napomenom razloga zbog kojih mu vraća zadatak. Svi zadaci moraju biti rešeni, predati i prihvaćeni od strane asistenta do kraja semestra. **Asistent zakazuje odbranu predatih domaćih zadataka studentima tradicionalne i hibridne nastave.** Student na odbrani treba da dokaže samostalnost u rešavanju domaćeg zadatka, pokaže da razume način rešavanja zadatka, da ga izloži i odgovori na dodatna pitanja asistenta. Odbrana domaćeg zadatka onlajn studenata se može obaviti preko Skype-a, ili preko mejla. O načinu odbrane, asistent obaveštava studenta mejlom. Zadaci se ocenjuju poenima na osnovu prikazanog

razumevanja problema/teme pri odbrani zadatka, urednosti, sistematičnosti i nivoa detaljnosti prikaza i obrade zadatka. Ukoliko se pri odbrani utvrdi da student nije samostalno radio domaći zadatak, student dobija 0 poena za zadatak.

**Testovi:** Student mora da samostalno uradi svaki od predviđenih testova, nezavisno od broja ostvarenih poena.

**Projekat:** Projekat je najvažnija i najznačajnija predispitna obaveza studenta. Na početku semestra objavljuje se spisak tema projektnih zadataka. Student može da izabere jednu od ponuđenih tema i da u dogovoru sa asistentom, dobije sve neophodne dodatne informacije i instrukcije za rešavanje izabranog projektnog zadatka. *Svaki student dobija jedinstven i specifični projektni zadatak* najkasnije do 3. nedelje nastave. Uz poseban zahtev i obrazloženje, asistent može studentu odobriti i projektni zadatak na temu koju predloži student, ali asistent je dužan da definiše ostale neophodne podatke i informacije za rad, tj. određuje kontekst u slučaju u kome se rešava prihvaćen projektni zadatak. Projektni zadatak definiše sve aktivnosti koje student treba da uradi da bi došao da krajnjeg rezultata koji se traži. Student je dužan da u Izveštaju u urađenom projektnom zadatku, prvo navede (citira) projektni zadatak, a onda da u svakom poglavlju koje se odnosi na jednu aktivnost projekta, svoj rad i rezultata rada na toj aktivnosti. Pri tome, poštuje redosled aktivnosti definisane projektnom zadatkom. Na kraju projekta, izlaže konačan rezultat i daje zaključak izveštaja. Tekst Izveštaja se piše u trećem licu jednine i mora da bude tehnički, jezički i profesionalno dobro urađen, usklađen za uzorkom dokumenta koji definiše asistent.

- **Rad na projektu:** Student, po pravilu, kontinualno radi svoj projekat tokom semestra, u skladu sa stečenim novim i potrebnim znanjima. Preporučuje se studentu da u toku rada, konsultuje asistenta, tj. da mu da na uvid urađeni deo projekta, kako bi bio siguran da je dobro shvatio svoj zadatak i primenio pravilan način rešavanja projektnog zadatka. Projekat se ne radi na kraju semestra ili kasnije. On se radi paralelno sa savlađivanjem gradiva, kako bi olakšao studentu savlađivanje novog gradiva i time pomogao u pripremi za savladavanje narednog gradiva na predmetu.
- **Predaja projekta:** Zahtevan rok za predaju *Izveštaja o urađenom projektnom zadatku* je do kraja 14. nedelje nastave za studente tradicionalne nastave (Beograd) ili hibridne nastave (Centar u Nišu), a za studente onlajn nastave – 10 dana pre ispita. Studentima tradicionalne i hibridne nastave koji predaju Izveštaj o urađenom projektnom zadatku 15 i više dana po završenoj nastavi na predmetu, **umanjuje se broj ostvarenih poena za 30%**. Krajnji rok za predaju projekta, za sve studente, je 10 dana pre ispitnog roka u kome žele da polažu ispit. Studentima onlajn nastave (koji ne prate nastavu u Centru u Nišu) se ne umanjuje broj poena, jer su oni, po pravilu zaposleni, ili sprečeni da redovno studiraju (primenom tradicionalnog i hibridnog oblika nastave).
- **Odbrana i ocenjivanje projekta:** Ako student prilikom ocenjivanja projekta ne dobije najmanje 50% predviđenih poena (15 poena), on mora da ga doradi. U suprotnom, dobija 0 poena. Student koji ne dobije više od 50% predviđenih poena ne može izaći na ispit. Student je dužan da projekat odbrani kod asistenta. Odbrana projekata studenata tradicionalne nastave i hibridne nastave (u Centru u Nišu) vrši se u 15. nedelji jesenjeg semestra za vreme vežbi, a ako je potrebno, i van vežbi – na dodatnim časovima. Van ovog termina, student može da brani projekat u posebnom terminu koji odredi asistent pred svaki ispitni rok. Cilj odbrane projekta je da asistent ustanovi da li je student samostalno radio projekat i u tom cilju odbrana projekta podrazumeva da student demonstrira znanje koje je iskoristio za izradu projekta. Odbrana projekta za studente internet nastave obavlja se preko Skype-a, a ako je za studenta prihvatljivo, u prostorijama univerziteta.

Na LAMS sistemu se nalazi uputstvo za izradu projekta.



**Ispit:** Ispit se polaže u okviru ispitnih rokova u računarskoj učionici fakulteta. Za vreme ispita nije dozvoljeno korišćenje mobilnih telefona ili drugih komunikacionih uređaja, kao ni Interneta. Ispit ima teorijski i praktični deo. Teorijski deo čine pitanja na koje student treba da pruži odgovore u pisanom obliku. Praktični deo čine zadaci koje student treba da reši uz primenu softverskog alata navedenog u ispitnom zadatku. Teorijski deo obezbeđuje 10 poena, a praktični deo 20 poena. *Student je položio ispit ako položi oba dela, tj. da ima najmanje 5 poena na teorijskom delu, i najmanje 10 poena na praktičnom delu ispita.* Student koji prepisuje na ispitu, udaljava se sa ispita, i **kažnjava se sa 10 predispitnih poena**. Protiv studenta za koga se utvrdi da je drugom kolegi pomagao u toku ispita, biće pokrenut disciplinski postupak i biće određena adekvatna mera. Ukoliko student pokaže znatno slabije rezultate na pismenom ispitu nego što je pokazao radom na predispitnim obavezama, asistent/nastavnik može da zatraži od studenta da **naknadno brani svoj rad na domaćim zadacima i projektu**. To se može primeniti ako je student na ispitu dobio manje od 20 poena, a na predispitnim obavezama je dobio više od 50 poena. Ukoliko nastavnik posumja da je student uradio pismeni ispit zadatak prepisujući, može da sa njim održi i **usmeni deo ispita**, kako bi proverio njegovo znanje. Ukoliko student pokaže vidno slabije znanje nego što to pokazuje njegov pismeni deo ispita, nastavnik daje ocenu na ispitu na osnovu ocene njegovog usmenog dela ispita.

**Za studente koji ne polože ispit najkasnije do početka prolećnog semestra školske 2021/22.godine, važe sledeća pravila:**

- Studenti tradicionalne koji nisu ostvarili uslov za upis naredne godine studija u školskoj 2020/21. godini, imaju obavezu ponovnog slušanja nastave i ponovnog rada na svim predispitnim obavezama, jer im se poništavaju do tada stečeni poeni, tj. poeni stečeni u prethodnoj školskoj godini. U tom slučaju, predispitne obaveze i ispit rade po Planu i programu koji važi od 15 februara 2022. godine. Ispit mogu da polažu u junskom ispitnom roku 2022. godine ili kasnije, ako su ostvarili uslov za izlazak na ispit, tj. uradili nove predispitne obaveze i stekli najmanje 35 poena.
- Studenti tradicionalne koji su ostvarili uslov za upis naredne godine studija u školskoj 2020/21. godini, mogu preneti predmet u narednu godinu, bez obaveze ponovnog rada na predispitnim obavezama, ukoliko su prethodno ostvarili uslov za izlazak na ispit. Ako student nije ostvario dovoljan broj poena za izlazak na ispit, radiće predispitne obaveze kao onlajn student (odnosno moći će da svoje predispitne obaveze pošalje najkasnije 10 dana pre ispitnog roka u kojem student želi da polaže ispit, ali ispit moraju položiti najkasnije u aprilskom roku 2022. godine. Ako i *posle tog ispitnog roka ne polože ispit, poništavaju im se poeni na svim do tada urađenim predispitnim obavezama, i moraju da rade nove predispitne obaveze i pripremaju ispit po Planu i programu koji važi od 15. februara 2022. godine.*
- Studenti onlajn nastave, rade predispitne obaveze i polažu ispit po ovom Planu i programu sve do aprilskog roka 2022. godine. Ako i *posle tog ispitnog roka ne polože ispit, poništavaju im se poeni na svim do tada urađenim predispitnim obavezama, i moraju da rade nove predispitne obaveze po Planu i programu koji važi od 15. februara 2022. godine.*
- Studenti koji su odslušali četvrtu godinu i stekli status apsoluta rade predispitne obaveze i polažu ispit po ovom Planu i programu sve do aprilskog roka 2022. godine. Predispitne obaveze mogu poslati najkasnije 10 dana pre ispitnog roka u kojem student želi da prijavi ispit. Ako i *posle tog ispitnog roka ne polože ispit, poništavaju im se poeni na svim do tada urađenim predispitnim obavezama, i moraju da rade nove predispitne obaveze po Planu i programu koji važi od 15. februara 2022. godine.*

**Ukupna ocena na predmetu** se dobija sabiranjem poena dobijenih radom na predispitnim obavezama (maksimalno do 70) i poena sa ispita (maksimalno do 30) i to na sledeći način (definisano Zakonom o visokom obrazovanju):

- do 50 poena, ocena 5
- od 51 do 60 poena, ocena 6

- od 61 do 70 poena, ocena 7
- od 71 do 80 poena, ocena 8
- od 81 do 90 poena, ocena 9
- od 91 do 100 poena, ocena 10.

#### PRAVILA U VEZI SA OSTVARENJEM USLOVA ZA UPIS U NAREDNU GODINU STUDIJA

Za studente koji ne polože ispit do kraja školske 2021/22. godine, tj. do 30.4.2022. godine, a nisu položili ispit is IT355, važe sledeća pravila:

- Studenti tradicionalne i hibridne nastave koji nisu ostvarili uslov za upis naredne godine studija, imaju obavezu ponovnog slušanja nastave i ponovnog rada na svim predispitnim obavezama, jer im se poništavaju svi poeni stečeni u prethodnoj školskoj godini. Ispit mogu da polažu u januarsko-februarskom ispitnom roku ili kasnije, ako su ostvarili uslov za izlazak na ispit, tj. uradili nove predispitne obaveze i stekli najmanje 35 poena.
- Studenti tradicionalne i hibridne nastave koji su ostvarili uslov za upis naredne godine studija mogu preneti predmet u narednu godinu, bez obaveze ponovnog rada na predispitnim obavezama, ukoliko su prethodno ostvarili uslov za izlazak na ispit. Ako student nije ostvario dovoljan broj poena za izlazak na ispit, radiće predispitne obaveze kao onlajn student (odnosno moćiće da svoje predispitne obaveze pošalje najkasnije 10 dana pre ispitnog roka u kojem student želi da polaže ispit, ali ispit moraju položiti najkasnije u oktobarskom roku 2022. godine. Ako i *posle tog ispitnog roka ne polože ispit, poništavaju im se poeni na svim do tada urađenim predispitnim obavezama, i moraju da rade nove predispitne obaveze i pripremaju ispit po Planu i programu koji važi od 01. oktobra 2022. godine.*
- Studenti onlajn nastave, rade predispitne obaveze i polažu ispit po ovom Planu i programu sve do oktobarskog roka 2022. godine. Predispitne obaveze mogu poslati najkasnije 10 dana pre ispitnog roka u kojem student želi da prijavi ispit. Ako i *posle tog ispitnog roka ne polože ispit, poništavaju im se poeni na svim do tada urađenim predispitnim obavezama, i moraju da rade nove predispitne obaveze po Planu i programu koji važi od 01. oktobra 2022. godine.*

Studenti koji su odslušali četvrtu godinu i stekli status apsoluta rade predispitne obaveze i polažu ispit po ovom Planu i programu sve do oktobarskog roka 2022. godine. Predispitne obaveze mogu poslati najkasnije 10 dana pre ispitnog roka u kojem student želi da prijavi ispit. Ako i *posle tog ispitnog roka ne polože ispit, poništavaju im se poeni na svim do tada urađenim predispitnim obavezama, i moraju da rade nove predispitne obaveze po Planu i programu koji važi od 01. oktobra 2022. godine.*

**PLAN NASTAVE**

Nedelja	Čas	Nastavna jedinica	Tematske jedinice (objekti i podobjekti učenja su boldirani)	Ishodi učenja – znanja ili veštine koje student treba da dobije	Vežbe (1 čas pokazne vežbe i 3 časa individualnih vežbi)
1		<b>L01 - Uvod u Spring 4</b>	<ol style="list-style-type: none"> <li><b>Umetanje zavisnosti i inverzija kontrole:</b> Šta su umetanje zavisnosti i inverzija kontrole?.</li> <li><b>Moduli Spring okvira:</b> Organizacija Spring okvira. Core Container. AOP, instrumenti i rad sa porukama. Pristup podacima i integracija. Web i Test moduli.</li> <li><b>Scenariji upotrebe okvira Spring 4:</b> Upravljanje zavisnošću i konvencija naziva. Logovanje.</li> <li><b>Upravljanje zavisnošću i konvencija naziva:</b> Upravljanje zavisnošću i umetanje zavisnosti. Pregled Spring artifakt datoteka. Upravljanje zavisnostima primenom Maven - a. Maven "Bill Of Materials" zavisnosti - BOM. Gradle upravljanje zavisnostima. Distribucija zip datoteka.</li> <li><b>Logovanje:</b> Logging zavisnost. Log4j 1.2 ili 2.x. Log4j 1.2 ili 2.x - dodatna razmatranja. Izbegavanje commons-logging. Upotreba SLF4J sa Log4j ili Logback. Šta je Spring? (video materijal).</li> <li><b>Spring 4 noviteti:</b> Nove stavke i unapređenja u Spring 4.0 okviru. Pregled noviteta i unapređenja okvira Spring 4.0. Napredni noviteti i unapređenja okvira Spring 4.0. Napredni noviteti i unapređenja okvira Spring 4.0 - nastavak. Šta je novo u okviru Spring 4.0 video materijali..</li> <li><b>Spring 4.1 noviteti:</b> Pregled noviteta i unapređenja okvira Spring 4.1. JMS (Java Messaging System) unapređenja. Unapređenja keširanja. Spring 4.1 veb unapređenja. Spring 4.1 veb unapređenja - dopunska razmatranja. Unapređenja za WebSocket razmenu poruka. Unapređenja iz oblasti testiranja Spring aplikacija.</li> <li><b>Spring 4.2 noviteti:</b> Unapređenja osnovnog (core) kontejnera. Unapređenja osnovnog (core) kontejnera - dodatna razmatranja. Unapređenja pristupa podacima i JMS unapređenja. WebSocket unapređenja. Test</li> </ol>	<p>Student razume organizaciju Spring okvira i strukturu Spring aplikacije</p>	<p>Kreiranje Maven konfiguracionog fajla i inicijalne structure Spring projekta.</p> <p>Domaći zadatak 1</p>

			<p>unapređenja sa okvirom Spring 4.2. Test unapređenja sa okvirom Spring 4.2 - dopunska razmatranja.</p> <p>9. <b>Spring 4.3 noviteti:</b> Unapređenja osnovnog kontejnera. Unapređenja pristupa podacima, keširanja i JMS sistema. Unapređenja testiranja. Podrška za nove biblioteke i generacije servera.</p> <p>10. <b>Spring 5 - noviteti:</b> Nove funkcije u radnom okviru Spring 5.0. Osnovne nadogradnje. Kompatibilnost pokretanja JDK 9. JDK 8 i Spring 5.0. Podrška za reaktivno programiranje. Funkcionalni veb radni okvir. Java modularnost sa Jigsaw-om. Uklonjene funkcije. Nove funkcije Spring Boota 2.0.</p> <p>11. <b>Pokazni primer - Maven konfiguracioni fajl:</b> Datoteka pom.xml. Zadatak za samostalni rad.</p>		
2		<b>L02 - Spring IOC kontejner</b>	<p>1. <b>Inverzija kontrole:</b> Pozicioniranje IoC kontejnera. Koncept inverzije kontrole.</p> <p>2. <b>Umetanje zavisnosti:</b> Šablon umetanja zavisnosti.</p> <p>3. <b>DI i IoC - pokazni primer:</b> Kreiranje interfejsa i implementacionih klasa. Kreiranje servisne klase i analiza.</p> <p>4. <b>Spring IoC Container:</b> Instanciranje IoC kontejnera.</p> <p>5. <b>Umetanje zavisnosti u Springu:</b> Tipovi umetanja zavisnosti u Spring okviru. Umetanje zavisnosti preko konstruktora. Umetanje zavisnosti preko setter metode. Umetanje zavisnosti preko polja klase. Automatsko povezivanje zavisnosti. Automatsko povezivanje zavisnosti - XML konfiguracije. Primena lenjih zrna.</p> <p>6. <b>Umetanje zavisnosti u Springu - pokazni primer:</b> Kreiranje i podešavanje Maven projekta. Prikaz strukture Spring Maven projekta. Zahtevi projekta. Kreiranje interfejsa. Implementacione klase interfejsa. Primena specifičnih Spring anotacija. Kreiranje servisne klase. Repozitorijum aplikacije. Kreiranje i pretraga komponenata Spring aplikacije. Spring (Java) konfigurisanje konteksta aplikacije. Učitavanje konteksta aplikacije. Prevođenje i pokretanje kreiranog Spring projekta.</p> <p>7. <b>Kompatibilnost unazad - XML konfigurisanje:</b> Instanciranje Spring IoC kontejnera. Instanciranje konteksta aplikacije i preuzimanje zrna. Struktura</p>	<p>Student razume i vlada konceptima umetanja zavisnosti i upravljanja zavisnostima. Primenjuje Java konfiguracije Spring aplikacije.</p>	<p>Izrada prve Spring aplikacije, podešavanje konteksta aplikacije, upravljanje zavisnostima.</p> <p>Domaći zadatak 2</p>

			projekta i Maven pom.xml. Java klase projekta. Konfiguracija zrna.		
3		<b>L03 - Aspektno - orijentisano programiranje u Springu</b>	<ol style="list-style-type: none"> <li><b>Podrška AspectJ anotacijama u Springu:</b> Obezbeđivanje podrške za AspectJ anotacije. Podrška za AspectJ anotacijama - nastavak primera. Podrška za AspectJ anotacijama - Java konfiguracije.</li> <li><b>Deklarisanje aspekata pomoću AspectJ anotacija:</b> AspectJ anotacije. Savet tipa @Before. Savet tipa @Before - dopunska razmatranja. Testiranje izvršavanja saveta @Before. Savet tipa @After. Saveti tipa @AfterReturning i @AfterThrowing. Testiranje izvršavanja saveta @AfterReturning. Testiranje izvršavanja saveta @AfterThrowing. Savet tipa @Around. Testiranje izvršavanja saveta @Around.</li> <li><b>Informacija tačke dodira:</b> Pristup informaciji tačke dodira.</li> <li><b>Određivanje prednosti među aspektima:</b> Redosled primene aspekata. Zadatak za samostalni rad 1.</li> <li><b>Ponovna upotreba definicije tačke prekida:</b> Definicija presečne tačke. Ponovna upotreba presečne tačke - demo.</li> <li><b>Izrazi presečnih tačaka:</b> Pisanje izraza presečnih tačaka u AspectJ. Šabloni potpisivanja metoda. Dodatna razmatranja. Šabloni potpisivanja tipova. Kombinovanje izraza presečnih tačaka. Deklarisanje parametara presečnih tačaka. Zadatak za samostalni rad 2.</li> <li><b>"Predstavljanje" ponašanja zrnju:</b> Savet tipa Introduction. Uvođenje saveta Introduction. "Predstavljanje" ponašanja zrnju - demo.</li> <li><b>Predstavljanje stanja zrnima:</b> Upravljanje stanjima. Upravljanje stanjima - demo.</li> <li><b>Deklarisanje aspekata XML konfiguracijama:</b> XML umesto anotacija. Deklarisanje aspekata. Deklarisanje presečnih tačaka. Deklarisanje saveta. Deklarisanje predstavljanja.</li> <li><b>Upredanje AspectJ aspekata u Springu:</b> Upredanje tokom učitavanja. Upredanje tokom učitavanja - primer. Upredanje tokom učitavanja - nastavak primera. Load-Time upredanje primenom AspectJ Weaver. Spring Load-Time Weaver. Zadatak za samostalni rad 3.</li> </ol>	Student koristi Aspektno – orijenisane koncepte i principe u Spring aplikacijama.	Primena presretnača u Spring aplikacijama.  Domaći zadatak 3 TEST 1.

			<p>11. <b>Podešavanje AspectJ aspekata u Springu:</b> Definisanje problema. Zadatak za samostalni rad 4.</p> <p>12. <b>Umetanje Spring zrna u objekte domena:</b> Problem Spring zrna i objekata domena. AnnotationBeanConfigurerAspect. Povezivanje pomoću ID zrna. Zadatak za samostalni rad 5.</p>		
4		<b>L04 - JSP i JSTL</b>	<p>1. <b>JSP arhitektura:</b> JSP mehanizam. JSP procesiranje.</p> <p>2. <b>JSP životni ciklus:</b> Prikaz životnog ciklusa JSP. Faze životnog ciklusa JSP.</p> <p>3. <b>JSP sintaksa:</b> JSP direktive. JSP akcije. JSP - Implicitni objekti.</p> <p>4. <b>JSP direktive:</b> JSP direktive - pregled. Page direktiva. Direktiva include. Direktiva taglib.</p> <p>5. <b>JSP akcije:</b> Uvod u JSP akcije. Akcija tipa jsp:include. Akcija tipa jsp:useBean. Akcija tipa jsp:setProperty. Akcija tipa jsp:getProperty. Primer korišćenja JSP akcija - zrna u JSP. Akcija tipa jsp:forward. Akcija tipa jsp:plugin. Akcije tipa jsp:element, jsp:attribute i jsp:body. Akcija tipa jsp:text.</p> <p>6. <b>JSP - Implicitni objekti:</b> Pregled implicitnih objekata. Objekat request. Objekat response. Primer HTTP zaglavlja odgovora. Objekat out. Objekat session. Praćenje sesije u JSP stranici - primer. Objekat application. Objekti config i pageContext. Objekti page i Exception. Još malo o izuzecima u JSP stranicama.</p> <p>7. <b>Procesiranje forme:</b> GET i POST metoda. Primer GET metode primenom URL. Primer GET metode primenom forme. Primer POST metode primenom forme. Prosleđivanje check - box podataka u JSP program. Čitanje svih parametara forme.</p> <p>8. <b>Upravljanje kolačićima u JSP stranicama:</b> Kolačići (Cookies). Lista Cookie metoda. Upotreba JSP za podešavanje kolačića. Podešavanje kolačića u JSP - primer. Čitanje kolačića u JSP - primer. Brisanje kolačića u JSP - primer.</p> <p>9. <b>Preusmeravanje JSP stranica:</b> Preusmeravanje stranica. Preusmeravanje stranica - demonstracija.</p> <p>10. <b>JSTL - JavaServer Pages Standard Tag Library:</b> JavaServer Pages Standard Tag Library. JSTL osnovni</p>	<p>Student koristi JSP i JSTL za kreiranje pogleda u Spring veb aplikacijama</p>	<p>Kreiranje različitih JSP pogleda primenom JSTL</p> <p>Domaći zadatak 4</p>

			<p>tagovi. JSTL tagovi formatiranja. JSTL SQL i XML tagovi. JSTL funkcije.</p> <p>11. <b>JSP - JavaBean:</b> Klasa JavaBean. Konstruisanje JavaBean klase i pristup zrnima. JSP datoteka za pristup Java zrnju.</p> <p>12. <b>JSP - jezik izraza:</b> EL - Expression Language (jezik izraza). JSP EL - elementi sintakse. JSP EL - elementi sintakse - nastavak.</p>		
5		<b>L05 - Spring MVC</b>	<ol style="list-style-type: none"> <li><b>Dispatcher Servlet:</b> Predstavljanje MVC komponenata. Definisanje klase kontrolera. Definisanje klase kontrolera - dopunska razmatranja. Spring MVC aplikacija - domenske klase. Razvoj servisnog nivoa Spring MVC aplikacije. Podešavanje Spring MVC aplikacije. Kreiranje konfiguracionih Java klasa za Spring MVC aplikaciju. Kreiranje konfiguracionih klasa - dopunska razmatranja.. Kreiranje Spring MVC kontrolera - osnovne anotacije. Kreiranje Spring MVC kontrolera - dodatne anotacije. Kreiranje Spring MVC kontrolera - upotreba servisa. Kreiranje Spring MVC kontrolera - POST metoda. Kreiranje JSP pogleda.. Angažovanje web aplikacije. Spring MVC konfiguracije i DispatcherServlet - video.</li> <li><b>Mapiranje zahteva sa @RequestMapping:</b> Definisanje problema. Mapiranje zahteva metodom. Mapiranje zahteva klasom. Mapiranje zahteva pomoću tipa HTTP zahteva. Dodavanje korisnika - model i servis. JSP stranica i demo.</li> <li><b>Presretanje zahteva:</b> Koncept Handler Interceptor. Kreiranje presretača. Metode presretača. Kreiranje presretača - primena adapter klase. Konfiguracija presretača i kreiranje pogleda. Handler Interceptor - video materijal.</li> <li><b>Podrška za internacionalizaciju:</b> Lokalizacija. Rešavanje lokalizacije atributom sesije. Rešavanje lokalizacije kolačićima. Promena lokalnih podešavanja korisnika. Rad sa lokalno osetljivim porukama - registracija zrna. Kreiranje eksternih resursa za poruke. Modifikacija JSP. Zadatak za samostalni rad 1.</li> <li><b>Rešavanje pogleda pomoću naziva:</b> Definisanje problema. Rešavanje pogleda na osnovu naziva šablona i lokacije. Rešavanje pogleda iz XML konfiguracione</li> </ol>	<p>Student kreira Spring MVC aplikaciju, rukuje http zahtevima preko kontrolera, čuva podatke u modelu, kreira JSP pogleda.</p>	<p>Kreiranje prve Spring MVC aplikacije</p> <p>Domaći zadatak 5</p>

			<p>datoteke. Rešavanje pogleda iz Resource Bundle. Rešavanje pogleda sa više rešavača. Prefiks preusmeravanja. Zadatak za samostalni rad 2.</p> <p>6. <b>Mapiranje izuzetaka u pogledu:</b> Definisanje problema. JSP stranica izuzetka. Zadatak za samostalni rad 4.</p> <p>7. <b>Pridruživanje vrednosti u kontroleru:</b> Anotacija @Value. Anotacija @Value - kreiranje odgovarajućeg kontrolera. Zadatak za samostalni rad 5.</p> <p>8. <b>Upravljanje formama pomoću kontrolera:</b> Definisanje problema. Kreiranje pogleda forme. Kreiranje pogleda forme - obrada grešaka. Obrada servisa forme. Kreiranje kontrolera forme. Inicijalizacija objekta atributa modela. Obezbeđivanje podataka referenci forme. Obezbeđivanje podataka referenci forme - nastavak. Povezivanje osobina kreiranih tipova. Povezivanje osobina kreiranih tipova - nastavak. Demonstracija upravljanja formama. Provera podataka forme. Provera podataka forme - datoteka sa porukama. Provera podataka forme - kontroler. Primena validatora - demonstracija. Napomene u vezi sa trajanjem sesije. Rad sa formama u Spring MVC - video materijal.</p> <p>9. <b>Spring MVC - pokazna vežba:</b> Stranice sa formom za pretragu i rezultatima pretrage. Izbor iz rezultata pretrage i forma za rezervaciju.</p>		
6		<b>L06 - Uvod u Spring Boot</b>	<p>1. <b>Spring Boot uvodna razmatranja:</b> Nova Spring definicija. Nov pogled na Spring. Nov pogled na Spring - dodatna razmatranja. Spring Boot osnove. Spring Boot - automatsko konfigurisanje aplikacije. Spring Boot - početne zavisnosti. COMMAND-LINE INTERFACE (CLI). Akuator.</p> <p>2. <b>Razvoj prve Spring Boot aplikacije:</b> Spring Boot MVC - pokazni primer i početna analiza. Primena automatskih konfiguracija.</p> <p>3. <b>Spring Boot MVC - pokazni primer i početna analiza:</b> Definisanje Spring MVC aplikacije u Spring Initializr. Analiza kreiranog Spring Boot projekta. Kreirana klasa ReadingListApplication. Kreirana test klasa. Podešavanje osobina aplikacije. Pregled zavisnosti i dodataka kreiranog projekta. Redefinisanje početnih zavisnosti.</p>	<p>Student koristi Spring Initializr za kreiranje početne strukture Spring Boot aplikacije, koristi početna podešavanja i automatske konfiguracije, koristi datoteke osobina za fino podešavanje aplikacije i upravlja greškama.</p>	<p>Prva Spring Boot aplikacija</p> <p>Domaći zadatak 6</p> <p>TEST 2</p>



			<ol style="list-style-type: none"> <li><b>Primena automatskih konfiguracija:</b> Uvodno razmatranje. Funkcionalnosti Spring Boot aplikacije - domen. Funkcionalnosti Spring Boot aplikacije - interfejs repozitorijuma. Funkcionalnosti Spring Boot aplikacije - veb interfejs - kontroler. Funkcionalnosti Spring Boot aplikacije - veb interfejs - pogledi. Pokretanje aplikacije.</li> <li><b>Prilagođavanje Spring podešavanja:</b> Redefinisanje Spring Boot podešavanja. Proširivanje podešavanja pomoću osobina. Prilagođavanje prikazivanja grešaka veb aplikacije.</li> <li><b>Redefinisanje Spring Boot podešavanja:</b> Problem automatskog konfigurisanja. Obezbeđivanje aplikacije. Kreiranje vlastitih bezbednosnih podešavanja. Servis za čuvanje detalja korisnika..</li> <li><b>Proširivanje podešavanja pomoću osobina:</b> Upotreba PROPERTIES datoteka. Fino podešavanje izvora podataka.</li> <li><b>Prilagođavanje prikazivanja grešaka veb aplikacije:</b> Podešavanje stranica grešaka.</li> <li><b>Pokazni primer - Gradle upravljanje zavisnostima:</b> Kreiranje Spring Boot Gradle projekta. Učitavanje Gradle projekta. Gradle konfiguraciona datoteka. Kreiranje korisnika u bazi podataka i datoteka osobina. Pokretanje aplikacije - prijavljivanje na aplikaciju. Dodavanje i pregled dodatnih knjiga. Dodatna razmatranja.</li> </ol>		
7		<b>L07 - Spring Boot - novi alati i tehnologije</b>	<ol style="list-style-type: none"> <li><b>Testiranje pomoću Spring Boot:</b> Integracioni testovi u Spring Boot - u. Testiranje veb aplikacije. Testiranje veb bezbednosti. Testiranje izvršavanja aplikacije.</li> <li><b>Integracioni testovi u Spring Boot - u:</b> Automatsko podešavanje integracionog testa. SpringJUnit4ClassRunner u Spring Boot integracionom testu. Analiza kreiranog integracionog testa.</li> <li><b>Testiranje veb aplikacije:</b> Testiranje veb aplikacije - uvodno razmatranje. Spring Mock MVC. Mock MVC test - početni test kod i deprekacije. Mock MVC test - dodatna razmatranja i test metoda za GET zahtev. Mock MVC test - test metoda za POST zahtev.</li> <li><b>Testiranje veb bezbednosti:</b> Podešavanje testiranja bezbednosti veb aplikacije. Kreiranje test primera.</li> </ol>	Student koristi aktuator, integralne Spring Boot testove, usvojio je primenu jezika Groovy I okvira Grails u integraciji sa Spring Boot, koristi CLI I GSP poglede	<p>Kreiranje Spring Boot Grails veb aplikacije</p> <p>Domaći zadatak 7</p>

			<ol style="list-style-type: none"> <li>5. <b>Testiranje izvršavanja aplikacije:</b> Testiranje na realnom serveru.</li> <li>6. <b>Groovy i Spring Boot CLI:</b> Razvoj Spring Boot CLI aplikacija. Testiranje Spring Boot CLI aplikacija.</li> <li>7. <b>Razvoj Spring Boot CLI aplikacija:</b> CLI - Interfejs komandne linije. Podešavanje CLI projekta. Kreiranje Groovy klase. Perzistencija u Spring Boot CLI (Groovy) aplikaciji. Kreiranje Groovy kontrolera. Dodatna razmatranja i kompletiranje Spring Boot CLI projekta.</li> <li>8. <b>Testiranje Spring Boot CLI aplikacija:</b> Kreiranje i izvršavanje test klase.</li> <li>9. <b>Grails u Spring Boot:</b> Objektno - relaciono mapiranje u Grails. Groovy Server Pages.</li> <li>10. <b>Objektno - relaciono mapiranje u Grails:</b> GORM. GORM entiteti. GORM kontroler. GORM - podešavanje bezbednosti. Definisane pogleda pomoću Groovy Server Pages.</li> <li>11. <b>Groovy Server Pages:</b> Analiza GSP listinga i demonstracija.</li> <li>12. <b>Aktuator:</b> Upotreba krajnjih tačaka aktuatora.</li> <li>13. <b>Upotreba krajnjih tačaka aktuatora:</b> Pregled krajnjih tačaka aktuatora. Pregled konfiguracionih detalja - informacije o povezivanju zrna. Pregled konfiguracionih detalja - automatsko konfigurisanje. Pregled konfiguracionih detalja - provera konfiguracionih osobina. Pregled konfiguracionih detalja - primena /configprops. Pregled konfiguracionih detalja - informacije kontrolera. Pregled metrika aplikacije. Pregled metrika aplikacije - nastavak. Praćenje veb zahteva. Praćenje zdravlja aplikacije. Video materijal.</li> <li>14. <b>Angažovanje Spring Boot aplikacije:</b> Angažovanje war fajla na servlet kontejneru - uvodno razmatranje. Proširivanje klase SpringBootServletInitializer. Obeležavanje ugrađenog servlet kontejnera i promena tipa arhive. Kontroler, fina podešavanja i demonstracija.</li> <li>15. <b>Pokazni primer - Spring Boot Actuator:</b> Podešavanje krajnjih tačaka. Podešavanje krajnjih tačaka - nastavak.</li> </ol>		
8		L08 - Spring ORM	<ol style="list-style-type: none"> <li>1. <b>Problem direktnog korišćenja ORM okvira:</b> Prelazak na viši nivo upravljanja bazom podataka. Hibernate i JPA. Čuvanje objekata pomoću Hibernate API i XML</li> </ol>	Student primenjuje ORM kroz JPA i	Vežba 5 se dograđuje uvođenjem baze

		<p>mapiranja. Alternativa za *.cfg.xml fajl. Hibernate API i XML mapiranje - dodatno izlaganje.</p> <p>2. <b>JPA i Hibernate:</b> Čuvanje objekata pomoću Hibernate API i JPA anotacija. Hibernate API i JPA anotacije - dodatno izlaganje. Čuvanje objekata pomoću JPA. Čuvanje objekata pomoću JPA - dodatna razmatranja. Čuvanje objekata pomoću JPA i Hibernate - demonstracija.</p> <p>3. <b>Podešavanje produkcije ORM resursa u Springu:</b> Problem podešavanja ORM resursa. Podešavanje produkcije Hibernate sesije u Springu. Podešavanje produkcije Hibernate sesije u Springu - drugi način. Džoker za resurse. Kreiranje i upotreba izvora podataka. Podešavanje produkcije JPA menadžera entiteta u Springu. Podešavanje produkcije JPA menadžera entiteta - testiranje. Produkcija JPA menadžera entiteta - Java EE. HibernateJpaVendorAdapter - dopunsko razmatranje. Zadatak za samostalni rad 1. Video materijali.</p> <p>4. <b>Hibernate kontekstualne sesije:</b> Kreiranje DAO klase. Registrovanje menadžera transakcija. Hibernate kontekstualne sesije i izuzeci. Zadatak za samostalni rad 2.</p> <p>5. <b>JPA umetanje konteksta:</b> Čuvanje objekata pomoću JPA umetanja konteksta. JPA umetanja konteksta - podešavanja. Primena anotacije PersistenceUnit. JPA umetanja konteksta - izuzeci. Zadatak za samostalni rad 3.</p> <p>6. <b>Pokazna vežba 1 - Spring 4/5, Hibernate 5 i JPA 2:</b> Zahtevi zadatka. Entitetska klasa za ORM mapiranje. Dodavanje zavisnosti u projekat. Kreiranje DAO (Data Access Object) klase. Jedinica perzistencije. Spring konfiguracija. Klasa za demonstraciju. Kreiranje baze podataka. Demonstracija kreirane aplikacije. Video materijal.</p> <p>7. <b>Pokazna vežba 2 - Spring 4/5, Hibernate 5 bez XML:</b> Zahtevi zadatka. Dodavanje zavisnosti u projekat. Entitetska klasa za ORM mapiranje. DAO klasa. Servisni nivo aplikacije. Datoteka osobina i konfigurisanje aplikacije. Main klasa i demonstracija.</p>	<p>Hibernate u Spring MVC aplikacijama</p>	<p>podataka sa Hibernate ORM podrškom</p> <p>Domaći zadatak 8</p>
--	--	---	--	---

9	L09 - Spring REST	<ol style="list-style-type: none"> <li><b>Objavljivanje XML baziranih REST servisa Springom:</b> Problem XML baziranih REST servisa. Registrovanje zrna rešavača. Inicijalizator REST veb aplikacije. Domen za REST servis. Servisni nivo REST aplikacije. Prevođenje objekta u XML - demo. Zadatak za samostalni rad 1. Primena anotacije @ResponseBody za kreiranje XML. Modifikovanje konfiguracija - demo rezultata. Zadatak za samostalni rad 2. Ograničeno prikazivanje podataka. Vraćanje odgovora klijentu pomoću ResponseEntity. Zadatak za samostalni rad 3.</li> <li><b>Objavljivanje JSON baziranih REST servisa Springom:</b> Primena MappingJackson2JsonView za kreiranje izlaza. Mapiranje i konfigurisanje za JSON podršku. JSON odgovor na URL zahtev. Zadatak za samostalni rad 4. Primena anotacije @ResponseBody za kreiranje JSON izlaza. Objavljivanje JSON baziranih REST servisa Springom - demo. Zadatak za samostalni rad 5.</li> <li><b>Pristup REST servisima u Springu 4 i 5:</b> Problem pristupa REST servisima. Životni ciklus REST servisa. Klasa RestTemplate. Pristup REST servisu - demo. Prijem podataka iz URL sa parametrima. Vraćanje podataka kao objekata. Zadatak za samostalni rad 6. RestTemplate - video materijal.</li> <li><b>Pokazni primer 1 - jednostavni RESTful servis:</b> Kreiranje jednostavnog RESTful servisa. Kreiranje klase za reprezentaciju resursa. Kreiranje kontrolera resursa. Kreiranje glavne klase aplikacije i pokretanje primera. Video materijal.</li> <li><b>Pokazni primer - REST servis CRUD:</b> Struktura primera i zahtevi. Kreiranje datoteke pom.xml. Podešavanje konfiguracija baziranih na anotacijama. Servlet 3 Java konfiguracije. Kreiranje modela. Customer Data Access Object (DAO). Customer REST kontroler. Testiranje aplikacije - prikazivanje korisnika. Testiranje aplikacije - izmena podataka korisnika. Testiranje aplikacije - dodavanje novih korisnika. Testiranje aplikacije - brisanje korisnika. Video materijal.</li> </ol>	Student koristi REST servise pri kreiranju Spring beckenda	<p>Vežba 8 se nadograđuje integracijom REST servisa nad postojećom bazom podataka.</p> <p>Domaći zadatak 9</p> <p>TEST 3</p>
---	-------------------	--	--	--

10	L10 - Spring Security podokvir	<ol style="list-style-type: none"> <li><b>Bezbednost URL pristupa:</b> Spring Security CSFR zaštita.</li> <li><b>Spring Security CSFR zaštita:</b> Primena CSFR zaštite. Primena CSFR zaštite - demo. Video materijal.</li> <li><b>Prijavljivanje u Spring veb aplikaciji:</b> Definisanje problema povezivanja. Isključivanje osnovnih bezbednosnih podešavanja. Primena osnovne HTTP provere. Prijavljivanje putem forme. Prijavljivanje putem forme - kreiranje vlastite stranice. Prijavljivanje putem forme - prikazivanje greške. Primena servisa za odjavljivanje. Primena servisa za odjavljivanje - stranica za odjavljivanje. Sprečavanje keširanja stranica. Podrška za anonimno prijavljivanje na aplikaciju. Podrška za pamćenje korisnika. Prijavljivanje putem forme - konfiguraciona klasa. Prijavljivanje putem forme - demo. Prijavljivanje putem forme - demo (nastavak). Video materijal. Zadatak za samostalni rad 1.</li> <li><b>Provera korisnika:</b> Načini provere korisnika. Provera korisnika čiji se podaci čuvaju u memoriji aplikacije. Provera korisnika iz baze podataka. Deklarisanje izvora podataka. Dodatna razmatranja u vezi sa korišćenjem baza podataka. Enkripcija lozinke. Provera korisnika - demo. Video materijal. Zadatak za samostalni rad 2.</li> <li><b>Upravljanje kontrolom pristupa:</b> Provera kontrole pristupa. Princip glasanja. Podrazumevani menadžer odluke. Implementacija glasača. Definisanje korisničkog menadžera odluke. Upravljanje kontrolom pristupa - demo. Zadatak za samostalni rad 3.</li> <li><b>Zaštita poziva metoda:</b> Problem poziva metoda. Obezbeđivanje poziva metode primenom anotacija. Video materijal. Zadatak za samostalni rad 4.</li> <li><b>Upravljanje bezbednošću u pogledima:</b> Problem bezbednosti pogleda. Uslovno prikazivanje sadržaja pogleda. Zadatak za samostalni rad 5.</li> <li><b>Napredni mehanizmi zaštite Spring Web aplikacija</b></li> </ol>	Student koristi bezbednosne Spring alate za obezbeđivanje pristupa Spring MVC aplikaciji	Vežba 9 se nadograđuje integracijom Spring Security servisa nad postojećom Spring MVC aplikacijom  Domaći zadatak 10
11	L11 - Spring Messaging	<ol style="list-style-type: none"> <li><b>Slanje i prijem JMS poruka u Springu:</b> Rad sa porukama bez JmsTemplate Spring podrške. Rad sa porukama sa JmsTemplate Spring podrškom. Rad sa porukama sa podrazumevanim destinacijama. Primena klase JmsGatewaySupport.</li> </ol>	Student koristi Spring JMS za intersistemske	Kreiranje destinacije za poruke na serveru Apache ActiveMQ,

		<p>2. <b>Rad sa porukama bez JmsTemplate Spring podrške:</b> Simulacija komunikacije između dve aplikacije. Model i servisi. Slanje poruke bez JmsTemplate Spring podrške. Prijem poruke bez JmsTemplate Spring podrške. Java konfiguracije podsistema šalter i prateća služba. Glavne klase projekata. Broker JMS poruka. Prijem i slanje poruke - demo.</p> <p>3. <b>Rad sa porukama sa JmsTemplate Spring podrškom:</b> Slanje poruka pomoću Spring JMS šablona. Konfigurisanje slanja poruka. Prijem poruka pomoću Spring JMS šablona. Konfigurisanje prijema poruka. Prijem i slanje poruke pomoću Spring JMS šablona - demo.</p> <p>4. <b>Rad sa porukama sa podrazumevanim destinacijama:</b> Konfigurisanje podrazumevane JMS destinacije. Slanje i prijem poruka sa podrazumevanih destinacija.. Prijem i slanje poruke sa podrazumevanih destinacija - demo.</p> <p>5. <b>Primena klase JmsGatewaySupport:</b> Opcije primene klase JmsGatewaySupport. Nasleđivanje klase JmsGatewaySupport. Prijem i slanje poruke primenom JmsGatewaySupport - demo.</p> <p>6. <b>Konverzija JMS poruka:</b> Problem konverzije JMS poruka. Slanje i prijem mape pomoću JMS šablona. Kreiranje konvertera JMS poruka. Konfigurisanje kreiranog JMS konvertera. Implementacija kreiranog konvertera poruka. Konverzija JMS poruka - demo. Zadatak za samostalni rad 1.</p> <p>7. <b>Upravljanje JMS transakcijama:</b> JMS transakcije. Primena transakcija na slanje i prijem JMS poruka. Java konfiguracije za primenu JMS transakcija. Prijem i slanje poruke primenom JMS transakcija - demo. Zadatak za samostalni rad 2.</p> <p>8. <b>Kreiranje POJO-a baziranih na porukama u Springu:</b> Definisanje problema osluškivanja poruka. Kreiranje osluškivača JMS poruka. Podešavanje osluškivača. Osluškivanje JMS poruka - Main klasa. Zadatak za samostalni rad 3. Osluškivanje JMS poruka pomoću POJO-a. Podešavanje POJO osluškivanja. Konverzija JMS poruka. Modifikacija osluškivača JMS poruka.</p>	komunikaciju udaljene pozive	I	postavljanje i čitanje poruke sa servera.
					Domaći zadatak 11

			<p>Upravljanje JMS transakcijama. Upravljanje JMS transakcijama - demo. Zadatak za samostalni rad 4.</p> <p>9. <b>Kreiranje konekcije:</b> Problem konekcije.</p> <p>10. <b>Pokazni primer - Spring 4 / 5 + JMS + ActiveMQ:</b> Sinergija tehnologija i alata. Izgled stranica pogleda. Tehnologije i struktura projekta.. Konfiguracione datoteke. Oslušivanje poruka. Servis za slanje poruka. Maven datoteka pom.xml. Kontroler i servis za porudžbine. Interfejs - repozitorijum. Modelske klase. BasicUtil i AppInitializer datoteke. Video materijal.</p>		
12	L12 - testiranje	Spring	<ol style="list-style-type: none"> <li>1. <b>Kreiranje testova pomoću okvira JUnit4 i TestNG6:</b> JUnit 4 testiranje. TestNG 6 testiranje.</li> <li>2. <b>JUnit 4 testiranje:</b> Pisanje JUnit4 testa. JUnit 4 testiranje - demonstracija.</li> <li>3. <b>TestNG 6 testiranje:</b> TestNG testiranje. TestNG testiranje - demonstracija. TestNG testiranje upravljano podacima.</li> <li>4. <b>Kreiranje jediničnih i integracionih testova:</b> Jedinični i integracioni testovi. Priprema aplikacije za jedinično testiranje - domen. Priprema aplikacije za jedinično testiranje - DAO sloj aplikacije. Kreiranje jedinačnih testova izolovanih klasa - JUnit 4 test. Kreiranje jedinačnih testova izolovanih klasa - demo. Primena stub i mock objekata. Kreiranje stub objekta. Primena stub objekta - demo. Kreiranje mock objekta. Primena stub mock - demo. Kreiranje i primena integracionih testova.</li> <li>5. <b>Jedinično testiranje Spring MVC kontrolera:</b> Problem testiranja kontrolera. Testiranje kontrolera. Testiranje kontrolera - demo. Video materijal.</li> <li>6. <b>Kontekst aplikacije i integracioni test:</b> Spring alati za podršku testiranju. Dodatne napomene i konfiguracija konteksta aplikacije. Pristupanje kontekstu aplikacije sa TestContext u JUnit 4. Testiranje konteksta aplikacije - demo. Video materijal.</li> <li>7. <b>Umetanje test alata u integracioni test:</b> Definisanje problema. Umetanje test alata u integracioni test u JUnit 4. Umetanje test alata u integracioni test u JUnit 4 - demonstracija.</li> <li>8. <b>Upravljanje transakcijama u integracionom testu:</b> Problem upravljanja transakcijama u integracionom</li> </ol>	<p>Student koristi okvire JUnit4 i TestNg u sinergiji sa Spring Security okvirom, za testiranje Spring MVC aplikacije</p>	<p>Kreiranje JUnit testova nad postojećom Spring MVC aplikacijom sa prethodnih vežbi</p> <p>Domaći zadatak 12</p> <p>TEST 4</p>



			<p>testu. Baza podataka i JDBC. Konfiguracije i izvor podataka. Upravljanje transakcijama TestContext okvirom u JUnit 4.</p> <p>9. <b>Pristup bazi podataka u integracionom testu:</b> Problem pristupa bazi podataka u integracionom testu. Test klasa za pristup bazi podataka. Pristup bazi podataka pomoću ovira TestContext u JUnit 4 - demo.</p> <p>10. <b>Primena opštih test anotacija u Springu:</b> Opšte test anotacije. Primena opštih test anotacija u JUnit 4. Zadatak za samostalni rad.</p>		
13		<b>L13 - Integracija Spring (Boot) i Angular</b>	<ol style="list-style-type: none"> <li>1. <b>Razvoj Spring serverske komponente:</b> Kreiranje Spring Maven projekta. Kreiranje i podešavanje baze podataka. Kreiranje Java konfiguracija. Kreiranje modela. Kreiranje DAO i servisnog nivoa. REST kontroler.</li> <li>2. <b>Kreiranje Spring Maven projekta:</b> Kreiranje komponente Book API. Kreiranje Spring Maven projekta. Kreiranje Java paketa Spring Maven projekta.</li> <li>3. <b>Kreiranje i podešavanje baze podataka:</b> Datoteka za podešavanje konekcije sa bazom podataka. Kreiranje baze podataka. Kreiranje baze podataka - napomene.</li> <li>4. <b>Kreiranje Java konfiguracija:</b> AppConfig datoteka. WebConfig datoteka. Dispatcher Servlet. Zadatak za samostalni rad 1.</li> <li>5. <b>Kreiranje modela:</b> Modelska klasa Book. Preslikavanje entiteta u tabelu.</li> <li>6. <b>Kreiranje DAO i servisnog nivoa:</b> Kreiranje DAO interfejsa i implementacije. Kreiranje servisnog interfejsa i implementacije.</li> <li>7. <b>REST kontroler:</b> Kreiranje REST kontrolera. Testiranje backend funkcionalnosti - GET zahtevi. Testiranje backend funkcionalnosti - DELETE zahtev. Testiranje backend funkcionalnosti - POST zahtev. Testiranje backend funkcionalnosti - PUT zahtev. Konačna struktura backend projekta. Video materijal.</li> <li>8. <b>Razvoj Angular klijentske komponente:</b> Kreiranje Angular projekta. Programske datoteka Angular projekta. Servisi klijenta. Pozivi metoda servisnog nivoa klijenta.</li> </ol>	<p>Student povezuje znanje stečeno na predmetima IT355 i IT255 kroz integraciju okvira Angular i Spring</p>	<p>Kreiranje Angular frontenda nad postojećom Spring MVC aplikacijom sa prethodnih vežbi.</p> <p>Domaći zadatak 13</p>



			<p>9. <b>Kreiranje Angular projekta:</b> Angular frontend projekat. Kreiranje direktorijuma projekta. Inicijalna struktura kreiranog Angular projekta.</p> <p>10. <b>Programske datoteka Angular projekta:</b> Modelska klasa - book.ts. Datoteka book.component.ts. Redefinisiranje datoteka app.module.ts i app.component.html.</p> <p>11. <b>Servisi klijenta:</b> Kreiranje servisa klijenta. Servis - GET metode i upravljanje greškama. Servis - PUT i POST metoda. Servis - DELETE metoda i registrovanje servisa. Provera GET servisa. Provera GET servisa - demo. Provera GET servisa iz veb stranice - demo.</p> <p>12. <b>Pozivi metoda servisnog nivoa klijenta:</b> Provera servisnih metoda klijenta. Provera POST servisa. Provera POST servisa - kreiranje forme. Provera POST servisa - demo. Provera DELETE servisa. Provera DELETE servisa - kreiranje dugmeta za poziv servisa. Provera DELETE servisa - demo. Provera PUT servisa. Provera PUT servisa - demo. Video materijal.</p>		
14		<b>L14 - Spring Portlet MVC okvir</b>	<p>1. <b>Razvoj jednostavnog portleta:</b> Definisanje problema. Komponenta DispatcherPortlet. Razvoj portala - primer. Podešavanje portlet aplikacije. Kreiranje konfiguracionih fajlova. Kreiranje portlet kontrolera. Kreiranje portlet kontrolera - konfiguraciona datoteka. Vraćanje naziva pogleda u pogled. Kreiranje portlet pogleda. Angažovanje portlet aplikacije. Demonstracija portlet aplikacije.</p> <p>2. <b>Mapiranje portlet zahteva u rukovaoc:</b> Definisanje problema. Konfiguracije i dodavanje portleta. Mapiranje zahteva portlet režimom - režim pogleda. Mapiranje zahteva portlet režimom - režim uređivanja. Mapiranje zahteva portlet režimom - režim Help. Mapiranje zahteva po parametru.</p> <p>3. <b>Rukovanje formama portleta:</b> Primena kontrolera formi. Primena kontrolera formi - nastavak razmatranja. Kreiranje kontrolera forme. Kreiranje pogleda forme. Provera podataka forme. Provera podataka forme - dodatna razmatranja.</p> <p>4. <b>Pokazni primer - angažovanje portleta:</b> Apache Pluto preuzimanje i pokretanje. Učitavanje portleta. Izbor veb</p>	Student koristi Spring Portlet MVC okvir za kreiranje Portlet aplikacija za portale	<p>Kreiranje portleta i integrisanje u porpal Apache Pluto</p> <p>Domaći zadatak 14</p>

			archive. Testiranje kreiranog portleta. Zadatak za samostalni rad.		
15		<b>L15 - Rekapitulacija gradiva</b>	<ol style="list-style-type: none"> <li><b>Generalni opis sistema:</b> Perspektiva proizvoda. Funkcije proizvoda i zahtev korisnika. Generalna ograničenja.</li> <li><b>Specifični zahtevi:</b> Funkcionalni zahtevi. Dijagrami slučajeva korisćenja. Nefunkcionalni zahtevi. Zahtevi baze podataka. Dijagrami aktivnosti.</li> <li><b>Funkcionalni zahtevi:</b> Pregled funkcionalnih zahteva. Zahtevi: Korisnik ima pristup sistemu i kreira svoj nadimak. Zahtevi: Pristup javnom i privatnom časniku. Zahtevi: Pregled korisnika na mreži i novih korisnika. Zahtevi: Pregled izlaska sa sistema i pregled aktivnog kucanja poruke. Zahtevi: Slanje priloga i slanje priloga sa lokanog računara.</li> <li><b>Dijagrami slučajeva korisćenja:</b> Pregled dijagrama.</li> <li><b>Nefunkcionalni zahtevi:</b> Pregled nefunkcionalnih zahteva. Performanse, pouzdanost i upotrebljivost. Integritet, samostalnost i portabilnost.</li> <li><b>Zahtevi baze podataka:</b> Konceptualni model baze podataka. Fizički model baze podataka.</li> <li><b>Dijagrami aktivnosti:</b> Pregled dijagrama aktivnosti. Dijagram aktivnosti za pristup sistemu. Dijagram aktivnosti za slanje privatnih poruka. Dijagram aktivnosti za slanje javnih poruka. Dijagram aktivnosti za slanje priloga. Dijagram aktivnosti za slanje lokalnih priloga. Dijagram aktivnosti za pregled korisnika. Dijagram aktivnosti za čitanje poruka. Dijagram aktivnosti za status pisanja.</li> <li><b>Modeli analize sistema:</b> Dijagrami sekvenci. Dijagrami komunikacije.</li> <li><b>Dijagrami sekvenci:</b> Sekvencijalni dijagrami - napomene. Dijagram sekvenci za pristup. Dijagram sekvenci za slanje privatnih poruka. Dijagram sekvenci za slanje javnih poruka. Dijagram sekvenci za slanje priloga. Dijagram sekvenci za lokalno slanje priloga. Dijagram sekvenci za pregled korisnika. Dijagram sekvenci za čitanje poruka. Dijagram sekvenci za prikaz statusa pisanja.</li> </ol>	Student je u potpunosti savladao izradu Spring MVC projekta	<p>Vežba izrade Spring MVC aplikacije od početka do produkcije.</p> <p>Domaći zadatak 15</p> <p>TEST 3</p>

			<p>10. <b>Dijagrami komunikacije:</b> Dijagrami komunikacije - napomene. Dijagram komunikacije za pristup. Dijagram komunikacije za slanje privatnih poruka. Dijagram komunikacije za slanje javnih poruka. Dijagram komunikacije za slanje priloga. Dijagram komunikacije za slanje lokalnih priloga. Dijagram komunikacije za pregled korisnika. Dijagram komunikacije za čitanje poruka. Dijagram komunikacije za status pisanja.</p> <p>11. <b>Implementacioni dijagrami:</b> Dijagram realizacije. Infrastrukturni dijagram. Dijagram arhitekture. Arhitektura sistema - dodatna razmatranja. Kompozicija sistema. Dijagram mreže.</p> <p>12. <b>Implementacija i struktura:</b> API. Podaci. Repozitorijumi.</p> <p>13. <b>API:</b> CookieController i GroupController. GroupController. GroupController - nastavak. MessageController. MessageController - nastavak. UserController. UserController - nastavak.</p> <p>14. <b>Podaci:</b> MetCookie. MetGroup. MetMessage. MetUser.</p> <p>15. <b>Repozitorijumi:</b> CookieRepository i GroupRepository. MessageRepository i UserRepository.</p> <p>16. <b>Testiranje sistema:</b> Jedinično testiranje – JUnit. Integraciono testiranje. Sistemsko testiranje.</p> <p>17. <b>Demonstracija rada sistema:</b> Učitavanje početne stranice. Stranica sa funkcionalnostima sistema. Stranica sa funkcionalnostima sistema - nastavak.</p>		
--	--	--	---	--	--