

PLAN I PROGRAM ZA PREDMET
CS111 OBJEKTNO ORIJENTISANO PROGRAMIRANJE

PODACI O NASTAVNOM OSOBLJU	
Predavanja	Prof. dr Dragan Domazet, redovni profesor
Vežbanja	Dipl.Ing. Mateja Opačić, saradnik
e-mail adresa nastavnika	dragan.domazet@metropolitan.edu.rs
Skype adresa nastavnika	dragan.domazet1
Termini za konsultacije nastavnika preko Skype (2 termina nedeljno)	Ponedeljak od 10 do 12h i utorak od 15 do 17h
e-mail adresa saradnika	mateja.opacic@metropolitan.edu.rs
Skype adresa saradnika	mateja.opacic
Termini za konsultacije saradnika preko Skype	Sreda od 11 do 12 h, sreda od 19 do 20 h
PODACI O PREDMETU	
Semestar	1
Preduslovi	Za ovaj predmet nema preduslova
Broj ECTS	9
Broj časova predavanja nedeljno	2
Broj časova grupnih (pokaznih) vežbi nedeljno	1
Broj časova individualnih vežbanja nedeljno	3
Broj časova samostalnog istraživačkog rada nedeljno	0
PODACI O PREDISBITNIM OBAVEZAMA I ISPITU	
Broj domaćih zadataka tokom semestra	15
Maksimalan broj poena za jedan domaći zadatak	1,5
Broj testova tokom semestra	15

Maksimalan broj poena za jedan test	1
Broj kolokvijuma tokom semestra	0
Maksimalan broj poena za jedan kolokvijum	0
Broj projekata tokom semestra	1
Maksimalan broj poena za jedan projekat	25
Broj seminarских radova tokom semestra	0
Maksimalan broj poena za jedan seminarски rad	0
Maksimalan broj poena za zalaganje studenta tokom semestra	7,5
Maksimalan broj poena za predispitne obaveze	70
Maksimalan broj poena za ispit	30
Vreme trajanja ispita u minutima	180
Forma ispita (obrisati nepotrebne opcije)	Izrada programa u razvojnom okruženju NetBeans
Računarski alati ili pribor koji se koriste na ispitu	NetBeans razvojno okruženje

Literatura (nastavni materijal):

Osnovna literatura za ovaj predmet su tekstovi predavanja objavljeni na FIT e-Learning sistemu.

1. Predavanja i vežbanja objavljena na e-Learning sistemu
2. **JAVA SOFTWARE SOLUTIONS** - Foundations of program design, Fourth edition, John Lewis and William Loftus, Addison Wesley, 2005, (preporučeni udžbenik)
3. Tekst predavanja za predmet CS111 koje se daje studentima (obavezna literatura)
4. Client/Server Programming with Java and CORBA, 2nd Edition, Robert Orfali and Dean Harkey, Wiley (dopunski udžbenik)
5. Developing Java Enterprise Applications, Stephen Asbury and Scott Weiner, Wiley (dopunski udžbenik)

Web lokacije (dopunska literatura):

1. http://www.debreuil.com/docs/ch01_Objects.htm
2. <http://oopweb.com/Cpp/Documents/ObjectsFirst/VolumeFrames.html>
3. <http://oopweb.com/Java/Documents/ThinkingInJava/VolumeFrames.html>
4. <http://oopweb.com/Java/Documents/IntroToProgrammingUsingJava/VolumeFrames.html>
5. http://java.sun.com/j2ee/tutorial/1_3-fcs/index.html
6. <http://java.sun.com/docs/books/tutorial/getStarted/index.html>
7. <http://java.sun.com/docs/books/tutorial/java/concepts/>
8. http://www.javaworld.com/channel_content/jw-oop-index.shtml
9. http://www.sun.com/books/java_series.html
10. <http://www.zib.de/Visual/people/mueller/Course/Tutorial/tutorial.html>
11. <http://www.javaworld.com/javaworld/jw-04-2001/jw-0406-java101.html?>
12. <http://oopweb.com/Algorithms/Documents/PLDS210/VolumeFrames.html>
13. <http://csis.pace.edu/~bergin/patterns/ppoop.html>
14. <http://www.omg.org/>
15. <http://netbeans.org>

Cilj predmeta:

Predmet uvodi studente u osnovne koncepte programiranja iz objektno-orijentisane perspektive. Počinje se sa UML dijagramima i osnovama njihovog korišćenja pa se zatim prelazi na Java programski jezik i principe objektno orijentisanog programiranja. Student dobija osnovna znanja o osnovama objektno-orijentisanog (OO) programiranju kao što je korišćenje klasa i objekata, prenos poruka između objekata, nasleđivanje i

podklase, upravljačke strukture, algoritmi, proste strukture podataka, upotreba API i GUI. Studenti takođe dobijaju početna znanja o osnovnim konceptima i principima projektovanja OO aplikacija, o šablonima projektovanja, OO analizi i projektovanju, kao i o osnovama softverskog inženjerstva. Predmet je neophodan preduslov za predmet CS112 i druge programerski orijentisane predmete studijskog programa.

Način ocenjivanja:

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

- 15 poena – Testovi: Posle svakog predavanja student mora da položi test preko FIT eLearning sistema vezan za to predavanje. Test je položio ako je odgovorio tačno na najmanje 50% pitanja. Test se može polagati tri puta u dozvoljenom periodu. Dozvoljeni period za polaganje testova je do kraja školske godine, tj. do 1. oktobra, ali uvek najkasnije 10 dana pre ispitnog roka u kome žele da polažu ispit. Testovi će se ocenjivati na sledeći način:
 - za 51% do 60% - 0,2 poena
 - za 61% do 70% - 0,4 poena
 - za 71% do 80% - 0,6 poena
 - za 81% do 90% - 0,8 poena
 - za 91% do 100% - 1 poena
- 22,5 poena – Domaći zadaci: Posle izučavanja određene nastavne jedinice, odnosno, posle vežbi u okviru jednog predavanja (jedna nedelja) predviđeno je da svaki student dobije zadatak koji treba samostalno da reši. Predviđeno je ukupno 15 zadataka, a svaki uspešno rešen zadatak predat u zadatom roku obezbeđuje studentu do 1,5 poena. Dozvoljeni period za predaju zadataka je do kraja školske godine, tj. do 1. oktobra, ali uvek najkasnije 10 dana pre ispitnog roka u kome žele da polažu ispit. Student mora da uradi svaki od dobijenih domaćih zadataka, sa minimalnom ocenom od 0,3 poena. Bez ispunjenja tog uslova, ne može izaći na ispit, bez obzira na ukupan broj poena koje ima na predispitnim obavezama.
- 25 poena – Projekt: Svaki student dobija jedan projekt kojim treba da pokaže svoju spremnost da primeni stečena znanja u konkretnom primeru. Dozvoljeni period za predaju projekta je do kraja školske godine, tj. do 1. oktobra, ali uvek najkasnije 10 dana pre ispitnog roka u kome žele da polažu ispit. Minimalni broj poena koje student može da dobije na projektu je 12. To znači da će mu asistent vratiti na doradu projekat ako je ocenjen sa manje do 12 poena. Ako student ne uspe da popravi projekat i dobije najmanje 12 poena, gubi pravo izlaska na ispit, bez obzira na ukupan broj poena koje ima na ostalim predispitnim obavezama. Student stiče pravo izlaska na ispit tek kada ostvari najmanje 12 poena na projektu i ima najmanje 35 poena za rad na predispitnim obavezama.
- 7,5 poena – Zalaganje studenta u nastavi u toku semestra: Od studenata se očekuje da redovno rade svoje zadatke i da polažu testove, kao i da učestvuju u diskusijama na diskusionom forumu predmeta. Studenti se podstiču da pokažu samoinicijativnost, interes za

izučavanje programa na predmetu, da pomažu kolegama u razumevanju gradiva i da pokažu agilnost u radu na projektu koji dobiju da rade. Posebno se ceni da sve svoje predispitne obaveze završe u datim rokovima, a najkasnije do kraja semestra u kome se predaje ovaj predmet.

- **Student tradicionalne nastave** dobija 0,5 poena nedeljno za zalaganje ako pripremljen dolazi na nastavu i aktivno učestvuje u diskusijama (neobavezno) na temu koja je zadata za tekuću nedelju na času ili forumu i ako je uradio zadatak za datu nedelju u nedelji kada je predavano to gradivo.
- **Student koji studiraju preko Interneta** dobijaju poene ako redovno u toku semestra rade zadatke i domaće zadatke, tj. ako ne rade „kampanjski“ te na kraju semestra, pred ispit, predaju sve domaće zadatke odjednom. Slično važi i za projekat. On treba da se radi tokom semestra, a da se završi na kraju semestra, uz povremena konsultacije sa asistentom. Posebno se ceni aktivnost studenta na diskusionom forumu predmeta, na kome student može da postavlja pitanja vezana za nastavne teme, ali i da pruža odgovore na tuđa pitanja. Ceni se stručna diskusija između studenata.
- **30 poena – Pismeni ispit:** Radi se u okviru ispitnih rokova u učionici fakulteta. Za vreme ispita nije dozvoljeno korišćenje mobilnih telefona ili drugih komunikacionih uređaja, a nije dozvoljena ni poseta bilo kom sajtu koji omogućava komunikaciju (web mail, facebook ili druge socijalne mreže, mrežne igre,...). Student je položio pismeni ispit ako je dobio najmanje 15 poena.
 - Radi se u okviru ispitnih rokova u računarskoj učionici fakulteta sa mogućnošću korišćenja literature u papirnom obliku, eLearning sistema univerziteta ili internet strane sa dokumentacijom.
 - Student bira 5 od dobijena 6 zadataka.. Svaki zadatak definiše jedan specifični programerski problem koji student treba da reši u Javi korišćenjem PC računara sa razvojnim okruženjem NetBeans. Zadaci su iz gradiva koje se radi na predmetu.
 - Student koji prepisuje na ispitu, udaljava se sa ispita, i kažnjava se sa 10 poena.
 - Ispit traje 3h
- **Kazneni poeni:** Student tradicionalne nastave gubi po 0,5 poena za svaki neopravdani izostanak sa predavanja ili vežbanja nedeljno (jedan poen ako je odsustvo i sa predavanja i sa vežbanja). Negativni poeni se oduzimaju prilikom izračunavanja konačne ocene od zbiru poena dobijenih za predispitne obaveze i ispit.

Pravila vezana za predmet:

- Student na tradicionalnoj nastavi **MORA da dođe pripremljen na svako predavanje** time što je unapred pročitao pripremljeno gradivo za dato predavanje i pripremio se za diskusiju na predavanju kao i postavljanje pitanja u vezi pojedinih nejasnoća sa kojima se suočio tokom proučavanja gradiva.
- Predavanje nije klasično, već predstavlja koncept diskusije i inicijative za razmišljanje i samostalno prepoznavanje rešenja na pojedine probleme. Stoga budite pripremljeni da na **predavanju budete aktivni učesnik**.
- Sve predispitne obaveze student na tradicionalnoj nastavi mora da ispuni u rokovima koji se daju prilikom dodeljivanja obaveze, a ako u tome ne uspe, onda najkasnije deset dana pre dana polaganja ispita koji je prijavio (uz dobijanje negativnih poena za zalaganje). Student koji prati nastavu isključivo preko Interneta, svoje predispitne obaveze mora da ispuni najkasnije do 10 dana pre dana polaganja ispita koji

je prijavio.

- Student prijavljuje ispit na kraju semestra, u roku koji objavi uprava fakulteta. Student može i da otkáže polaganje ispita, ali to mora da otkáže pismenim putem (elektronskom porukom referentu za studentska pitanja) najkasnije tri dana pre dana polaganja ispita.

Napomene povodom izrada domaćih zadataka:

- U toku semestra se radi 15 domaćih zadataka u Javi 6 (JDK 1.6) ili UML modelu, a u razvojnom alatu NetBeans. Student dostavlja izvorni kod-program urađenog zadatka (projektni folder). Zadatke treba raditi u razvojnom okruženju NetBeans 7, koja će mu stajati na raspolaganju i za vreme ispita, koji se polaže za računarom u računarskoj učionici. Zadaci se daju posle svake grupne (pokazne) vežbe. Po urađenom zadatku, student ga dostavlja asistentu. Poslati zadaci moraju da budu funkcionalni/ispravni, odnosno neće biti ocenjivani zadaci koji ne mogu da se kompajliraju.
- Svaki student dobija posebni zadatak na email od svog asistenta. Sledeći zadatak može da dobije tek kada na prethodnom zadatku dobije bar 0,3 poena. Prvi zadatak dobije prve nedelje a svaki sledeći dobija ili kada dođe ta nedelja, ako je student uspešno završio zadatak za prethodnu nedelju, ili nakon što uspešno završi prethodni zadatak. Na poseban zahtev studenta koji nastavu prati preko Interneta, asistent mu šalje sve domaće zadatke odjednom, ili fazno, zavisno od zahteva studenta.
- Vreme potrebno za izvršavanje pojedinih zadataka od 15 min od 1.5h zavisno od zadatka i veštine studenta.
- Student za koga se utvrdi da nije samostalno uradio domaći zadatak, kažnjava se kaznenim poenima, tj. smanjivanjem broja poena koje je stekao radom na predispitnim obavezama. Broj kaznenih poena zavisi od stepena nesamostalnog rada studenta i broja zadataka na kojima je ustanovljen nesamostalan rad.

Napomene povodom projekta:

- Projektni zadatak studenti dobijaju na početku semestra, a mora da ga završe do kraja semestra odnosno do 10 dana pre ispita. Studenti koji prate nastavu preko Interneta sami biraju vreme kada žele da dobiju projektni zadatak i kada mogu da ga predaju asistentom. Međutim, to mora da urade najmanje 10 dana pre dana polaganja ispita. Preporučuje se studentima da što ranije predaju izveštaj o urađenom projektu, kako bi imali vremena na dorade projekta na osnovu primedbi asistenta. Na ovaj način mogu da povećaju broj poena koje dobijaju za kvalitet urađenog projekta, te i veću ocenu na ispitu. U slučaju predaje izveštaja 10 dana pre dana ispita, nije moguća nikakva naknadna dorada projekta.
- 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. Uz poseban zahtev i obrazložene, 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. Tema za projektni zadatak može da bude: neka jednostavna 2D igrice, software za evidenciju ili arhiviranje, simulacija rada (nekog fizičkog zakona, okretanja meseca oko zemlje, rada nekog mehanizma,...) ili bilo šta što se uklapa u tehnologije koje se uče na predmetu
- Svaki projekat mora da ima, UML dijagram klasa, GUI (Swing ili 2D java), Java Doc dokumentaciju u svim klasama, bar 4-5 klasa, a nije

preporučljivo da projektni zadatak sadrži više od 20 klasa.

- Projektni zadatak mora da bude urađen u skladu sa gradivom i preporukama predstavljenim na predavanjima i vežbama, uz korišćenje standardne Java Dokumentacije. Projekat se radi u Javi 6 (JDK 1.6), dostavlja se izvorni kod urađenog projekta (projektni folder). Projekat treba da se radi u razvojnom okruženju NetBeans. Sva pitanja kao i izveštaj o realizovanom projekatu se šalju asistentu na email mateja.opacic@metropolitan.edu.rs.
- Projekat mora da se brani, poeni stečeni na zadacima i projektu važe tek nakon uspešno odbranjenog projekta. Cilj odbrane projekta je da asistent ustanovi da li je student samostalno radio zadatke i projekat i u tom cilju odbrana projekta podrazumeva da student demonstrira znanje koje je iskoristio za izradu zadataka i projekta. Odbrana projekta može da bude tokom individualnih vežbi ili u zakazanom terminu pre svakog ispitnog roka. Student koji ne odbrani projekat mora da radi sve zadatke i projekat iz početka, poništiće mu se poeni stečeni na zadacima i projektu, jer će se smatrati da nije samostalno radio predispitne obaveze. U slučaju da se ustanovi da student nije samostalno radio svoj projekat, primenjuju se i mere koje su navedene u slučaju nesamostalnog rada domaćih zadataka.
- Rok za predaju projektnih zadataka je najkasnije 10 dana pre ispita.

Važna napomena:

Da bi student stekao pravo da izađe na ispit, MORA da ispuni sledeće uslove:

1. Mora da na predispitnim obavezama dobije najmanje 50% mogućih poena, tj. 35 od 70 maksimalno mogućih
2. Student mora da dobije bar 12 poena na projektnom zadatku, i da odbrani projektni zadatak.
3. Student mora da ima celovito i parcijalno urađene sve domaće zadatke, ocenjene sa ocenom od 0,3 do 1,5 poena. Ne smatra se urađenim domaći zadatak ako je ocenjen ocenom manjom od 0,3 poena.

Student je položio ako je na ispitu dobio najmanje 50% mogućih poena, tj. minimalno 15 poena i ako je na predispitnim obavezama takođe dobio najmanje 50% poena, tj. 35 poena.

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 (ovo je 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.

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 pismenom ispitu dobio manje od 20 poena, a na predispitnim obavezama je dobio više od 60 poena. Zavisno od uspešnosti odbrane predatih zadataka i projekta, asistent/nastavnik može da umanji studentu ranije odobrene poene za predispitne obaveze. Odbrana predispitnih obaveza se obavlja usmeno. U izuzetnim slučajevima, kada

asistent/nastavnik utvrdi da student ne može da odbrani urađene zadatke i projekat, tj. pokaže da ne vlada potrebnim znanjem i time pokazuje da nije samostalno uradio zadatke i projekat, asistent/nastavnik može, posle odobrenja predmetnog nastavnika, da poništi pojedine domaće zadatke ili projekat u celosti, što znači da ih student mora ponovo da radi.

Mere za sprečavanje prepisivanja

Ukoliko student prepíše zadatak ili projekat od drugog studenta ili ukoliko prepíše nečiji rad sa interneta ili iz literature, student dobija **10 (deset) kaznenih poena** iz datog predmeta. Ukoliko smatra da nema osnova za navedenu meru, student ima pravo da uloži žalbu dekanu fakulteta.

Ukoliko se utvrdi da student prepisuje na ispitu, on se odmah udaljuje sa ispita.

Protiv studenta za koga se utvrdi da je drugom kolegi dao svoj urađeni zadatak ili projekat, biće pokrenut disciplinski postupak i biće određena adekvatna mera.

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.

PLAN NASTAVE

Nedelja	Čas	Nastavna jedinica	Tematske jedinice	Rezultat – znanja ili veštine koje student treba da dobije	Vežbe
1	1	Uvod u računarske sisteme	Upoznavanje studenata sa predmetom Uvod u računarske sisteme Hardver i softver. Analogni i binarni signali. Mašinski jezik i jezici višeg nivoa. Prevođenje (kompilacija) i interpretacija programa.	Razumevanje funkcija komponenata računarskih sistema Razumevanje pojmova mašinskog jezika i jezika višeg nivoa.	Upoznavanje studenata sa savremenim primenama starijih sistema programiranja strukturalno i modularno Sličnosti i razlike među različitim programskim jezicima (C, C++, Paskal, Java, Basic)
	2		Programski jezici višeg nivoa Istorijski pregled razvoj programskih jezika Klasifikacija programskih jezika	Razumevanje šta je programski jezik višeg nivoa i njihove klasifikacije	
2	1	Objektno-orijentisano modeliranje	Pojam objekta i objektno-orijentisanog modeliranja Pojam klase Atributi i metodi Poruke	Razumevanje pojma objekt i klasa Razumevanje pojma poruka u kontekstu OO modeliranja	Definisanje UML slučajeva. Nekoliko primera. Kreiranje OO modela pojedinih problema (bez programiranja), sa upotrebom UML
	2	Objektno-orijentisano modeliranje	Veze (relacije) između klasa. Dijagrami klasa Dijagrami klasa	Razumevanje veza (relacija) između objekata i tipova ovih veza Sposobnost da se za neki problem kreira OO model	Primeri za razlikovanje klasa od objekata. Utvrđivanje potrebnih atributa i metoda, tj. određivanje potrebnog ponašanja
3	1	Objektno-orijentisano modeliranje primenom UML	UML strukturni dijagrami	Sposobnost korišćenja UML strukturnih dijagrama	Primeri izrade UML modela za različite tekstualno opisane probleme
	2	Objektno-orijentisano modeliranje primenom UML	UML dijagrami ponašanja	Sposobnost korišćenja UML dijagrama ponašanja	Upoznavanjem sa UML alatima (rational i drugi) za OO modeliranje primenom UML.

Nedelja	Čas	Nastavna jedinica	Tematske jedinice	Rezultat – znanja ili veštine koje student treba da dobije	Vežbe
4	1	Upoznavanje sa programskim jezikom Java	Java bytecode, Java kompajler i Java virtuelna mašina. Kreiranje i izvršenje Java programa Imena datoteka. Sintaktičke greške. Bagovi. Ciklus popravke grešaka. Izvršenje programa.	Sticanje osnovnih informacija o Javi Razumevanje specifičnosti i dobrih svojstava Jave Naučiti proceduru kompiliranja i izvršenja Java programa	Rad na jednostavnim primerima programiranja u Javi
	2	Upoznavanje sa programskim jezikom Java	Prvi programi u Javi Konstruktori Pozivanje metoda	Sticanje osnovnih saznanja o programiranju u Javi	
5	1	Osnovne programske konstrukcije (PF1)	Primitivni podaci i objekti. Primitivni tipovi u Javi. Numerički tipovi podataka. Wrapper klase za primitivne tipove	Upoznavanje sa konkretnim tipovima podataka koji postoje u programskom jeziku Java i Smalltalk	Prvi program i unos podatak u aplikaciju
	2	Osnovne programske konstrukcije (PF1)	Promenljive. Iskazi dodele. Osnovne metode čitanja ulaza i štampanja izlaza. Klase i objekti u programskom jeziku Smalltalk.	Upoznavanje sa promenljivim veličinama i sa njihovim deklarisanjem Upoznavanje sa metodima Java klase za učitavanje podataka i štampanje rezultata	
6	1	Osnovne programske konstrukcije (PF1)	Izrazi i aritmetički operatori.	Vrste izraza. Upoznavanje sa aritmetičkim operatorima. Konstante.	Rad sa operatorima i matematičkim funkcijama
	2	Objektno orijentisano programiranje (PL6)	Reference na objekte. Promenljive i operatori dodele u Smalltalk-u	Priimitivni tipovi podataka i objekti. Kopiranje referenci. Operator == i reference.	
7	1	Osnovne programske konstrukcije (PF1)	Relacioni i logički operatori Uslovni iskazi.	Vrste relacionih operatora. Vrste logičkih operatora. Iskazi kontrole toka (if, switch)	Objašnjavaju se naredbe kontrole toka, kako se one koriste i kako se definišu uslovi za njihovo izvršavanje.

Nedelja	Čas	Nastavna jedinica	Tematske jedinice	Rezultat – znanja ili veštine koje student treba da dobije	Vežbe
	2	Osnovne□ programske□ konstrukcije□ (PF1)	Iterativne kontrolne strukture. Donošenje odluka u programu u□ pisanim u Smalltalk-u	Upoznavanje sa petljama (while, for, do-while)	U okviru ove lekcije će biti objašnjeni tipovi petlji i sintaksa petlji u javi
8	1	Objektno orijentisano programiranje (PL6)	Određivanje atributa Učaučenje. Modifikatori vidljivosti. Deklarisanje i inicijalizacija atributa. Tipovi promenljivi u klasi. Metodi pristupa (get i set metodi). Upotreba „this“	Razumevanje početnog koraka u projektovanju klasa – određivanju atributa Razumevanje koncepta učaučenja kod OO programiranja i skrivanja informacija	Učenje o principima objektno orijentisanog programiranja. Više detalja o enkapsulaciji. Na času se uče i koji tipovi modifikatora vidljivosti postoje i kako se i kad oni koriste
	2	Objektno orijentisano programiranje (PL6)	Konstruktori. Inicijalizacije vrednosti atributa pomoću konstruktora.	Razumevanje uloge konstruktora u inicijalizaciji početnog stanja objekta	U okviru ove lekcije objašnjava se domen važenja promenljiva i njihova upotreba. Objašnjena je upotreba static modifikatora kao i ostalih modifikatora koje do sad nisu rađeni.
9	1	Objektno orijentisano programiranje (PL6) Mehanizmi apstrakcije (PL5)	Apstrakcije. Funkcije i procedure. Metodi u Javi	Pojam apstrakcije u OO programiranju. Procedure i funkcije kao mehanizmi apstrakcije.	Upoznavanje studenata sa radom metoda. Prikazani su loš primer kako se ne programiraju metode u javi i isti primer samo ispravno napisan. Prikazani su mehanizmi prosleđivanja parametara po referenci i po vrednosti. Prikazano je kako može da se promeni default način prosleđivanja parametara u metodu. Na kraju je prikazano i kako funkcioniše metod sa varijabilnim brojem parametara.
	2	Objektno orijentisano programiranje (PL6)	Parametri metoda. Metodi bez argumenata. Pravila konvefrzije. Argumenti metoda. Preklapanje metoda Učaučenje u SmallTalk-u..	Slanje promenljivih u metode. Domen parametara. Lokalne promenljive. Preklapanje metoda.	

Nedelja	Čas	Nastavna jedinica	Tematske jedinice	Rezultat – znanja ili veštine koje student treba da dobije	Vežbe
10	1	Objektno orijentisano programiranje (PL6)	Nasleđivanje.	Nasleđivanje kao jedan od osnovnih koncepata OO programiranja. Jednostruko i višestruko nasleđivanje. Hijerarhija klasa. Sintaksa nasleđivanja u Javi. Preklapanje metoda osnovne klase. Apstraktne klase. Apstraktni metodi. Modifikatori vidljivosti i nasleđivanje.	Prikazana je izgradnja hierarhije klasa u javi i objašnjavaju se mehanizmi koji stoje iza toga
	2	Tipovi podataka (PL4)	Složeni tipovi. Mapiranje. Rekurzivni tipovi Tipiziranje	Statičko i dinamičko vezivanje. Shvatanje tipiziranja i statičkog i dinamičkog tipiziranja	
11	1	Osnovne strukture podataka (PF3)	Nizovi. Implementacija nizova u Javi. Nizovi kao objekti. Inicijalizacija niza. Jednodimenzioni nizovi u Javi: Kopiranje nizova. Nizovi i petlje. Nizovi kao parametri. Nizovi objekata.	Razumevanje koncepta nizova. Sticanje znanja o operacijama sa jednodimenzionim nizovima.	Rad sa nizovima, sintaksa niza, i način korišćenja nizova.
	2	Osnovne strukture podataka (PF3)	Dvodimenzioni nizovi u Javi Nizovi u SmallTalk-u	Razumevanje dvodimenzionaih nizova i operacija nad njima	
12	1	Osnovne strukture podataka (PF3)	Uopšteno o kolekcijama, vector properties, preference... Struktura novih kolekcija. Kolekcija set.	Nakon ove lekcije, student je sposoban da koristi vector, properties i preference kolekcije kao i kolekcije tipa set	Rad sa kolekcijama. Rad sa najčešće korišćenim kolekcijama Listama, ArrayList i LinkedList, setovima, kao primer uzet je TreeSet, i mapa. Pored osnovnih metoda pristupa i popunjavanja kolekcija prikazane su i metode koje pretražuju kolekcije, sortiraju kolekcije, mešaju sadržaj kolekcije itd.
	2	Osnovne strukture podataka (PF3)	Nove kolekcije, kolekcije tipa List i kolekcije tipa Map, pomoćne klase za rad sa kolekcijama.	Nakon ove lekcije student je sposoban da koristi kolekcije tipa List i kolekcije tipa Map kao i pomoćne klase za rad sa kolekcijama.	
13	1	Osnovne tehnike u grafici (GV1)	Grafičke komponente u Javi za razvoj GUI: klase AWT i Swing paketa, menadžeri rasporeda grafičkih komponenti.	Razumevanje i ovladavanje osnovnom tehnikom kreiranja grafičkog prikaza GUI	Primer kreiranja aplikacije, sa upotrebom State design patterna

Nedelja	Čas	Nastavna jedinica	Tematske jedinice	Rezultat – znanja ili veštine koje student treba da dobije	Vežbe
	2	Osnovne tehnike u grafici (GV1)	Interakcija korisnika i aplikacije: dodeljivanje akcija, osluškivača akcija, adapteri. Rad sa mišem, prozorima, kursorom i zvukom.	Razumevanje rada i ovladavanje tehnikom povezivanja akcija sa grafičkim komponentama GUI radi ostvarenja interakcije korisnika sa aplikacijom.	
14	1	Osnovne tehnike u grafici (GV1)	Programiranje osnovnih elemenata grafike, rad sa panelima i crtanje po njima.	Nakon ovog časa student će biti sposoban da barata sa grafikom na panelima.	Primer kreiranja aplikacije, koja se zasniva na grafici. Korišćenje metoda grafika, animacija.
	2	Osnovne tehnike u grafici (GV1) Osnovni Rendering (GV5) Računarska animacija (GV8)	Rad sa textom, promena veličine i oblika fonta, efekti sa tekstom. Prikazivanje slike i neke osnovne operacije i transformacije nad slikama. Pravljenje animacija u Javi.	Nakon ovog časa student će umeti da kreira jednostavne animacije i da primenjuje efekte nad tekstom, ako i da radi sa slikom i da primeni jednostavne transformacije nad slikom.	
15	1	Algoritmi i rešavanje problema (PF2)	Algoritmi: Osnovni koncepti, predstavljanje, pseudo kod	Rešavanje problema pomoću računara. Definisanje problema, definisanje rešenja. Testiranje i kodiranje. Način predstavljanja algoritma, promena algoritma, alternativni algoritmi.	Upoznavanje sa metodima kreiranja softverskog proizvoda. Prikaz nekoliko najpoznatijih metodologija implementacije životnog ciklusa softvera. Detaljnije pokazana RUP metodologija. Jednostavan primer korišćenja RUP metodologije razvoja softvera.
	2	Rekurzija (PF4)	Rekurzija: Uvodna razmatranja, opšti pojmovi. Rekurzije u Javi.	Upoznavanje sa teoretskim osnovama vezanim za rekurziju. Razlaganje problema na manje delove. Implementacija rekurzije u Javi. Primeri problema rešenih pomoću rekurzije.	

Napomena: Nastavni program ovog predmeta zadovoljava sve preporuke i zahteve definisane od strane profesionalnih američkih udruženja IEEE/ACM za studijske programe iz oblasti informacionih tehnologija, računarskih nauka i softverskog inženjerstva. (Computing Curricula 2001). Skraćenice u zagradama nastavnih jedinica odgovaraju skraćenicama za jedinice znanja koje se obrađuju u nastavnim jedinicama, a koje su definisane i detaljno opisane u fondu znanja (Body of Knowledge) predmeta CS111o datim u izveštaju Computing Curricula IEEE/ACM koji je dat na sajtu Univerziteta Metropolitan.