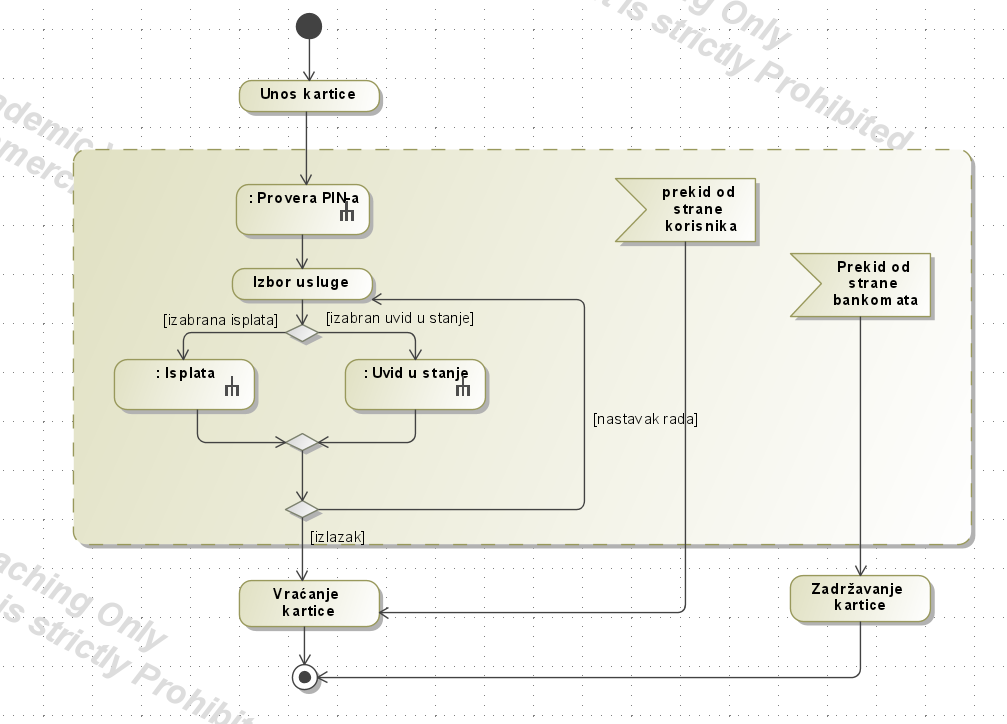
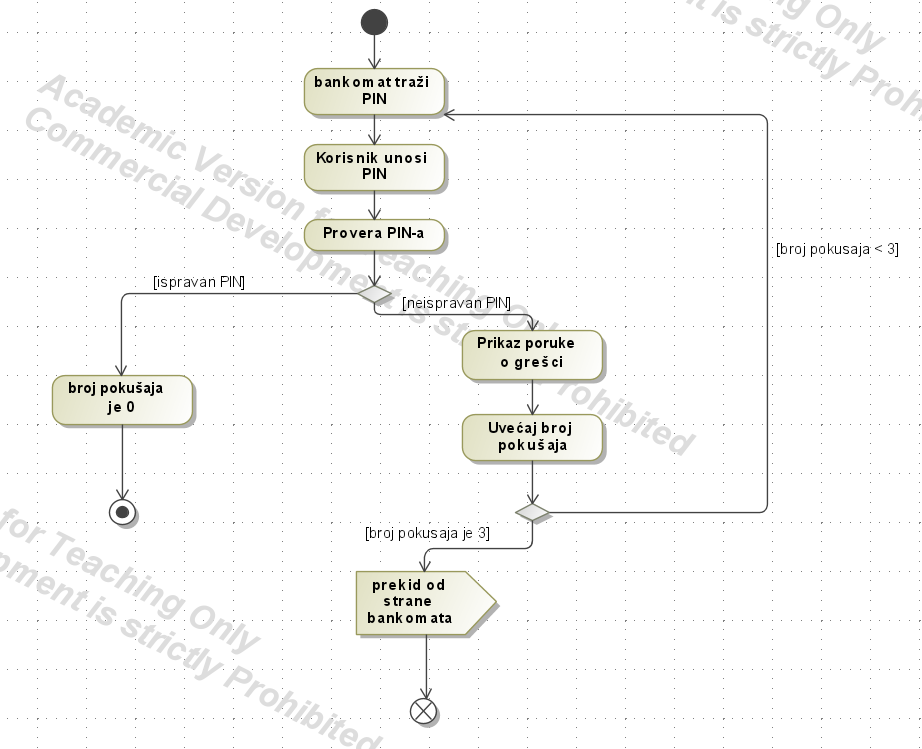
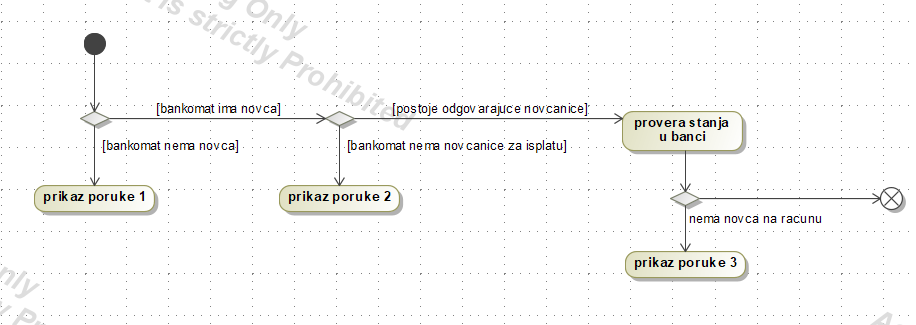
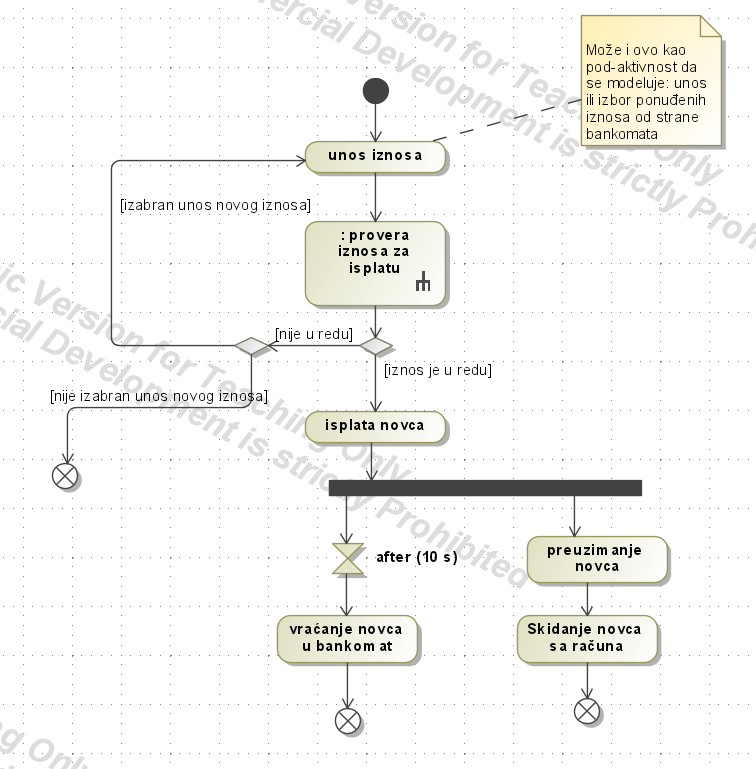
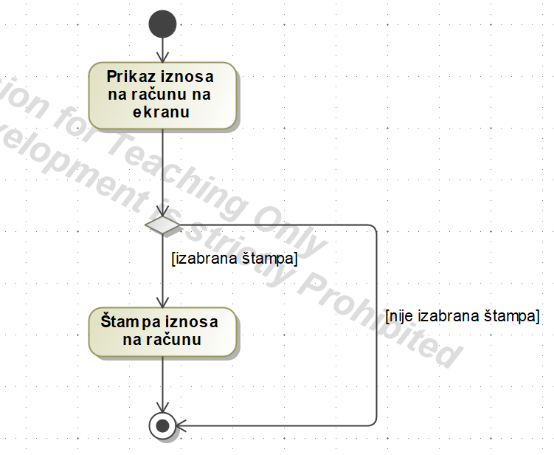
**KOLOKVIJUM 1**

**1. zadatak – BANKOMAT *(dijagram aktivnosti)***

****Korisnik ubacuje karticu u bankomat, unosi PIN i bira vrstu usluge. Ako je izabrana usluga uvid u stanje na računu, bankomat mu daje traženu informaciju (koliko ima novca na računu u banci). Ako je izabrana isplata gotovine, korisnik unosi željeni iznos i bankomat isplaćuje novac u navedenom iznosu. Na kraju, bankomat vraća korisniku karticu.

Unos pina

Dijagram aktivnosti

****

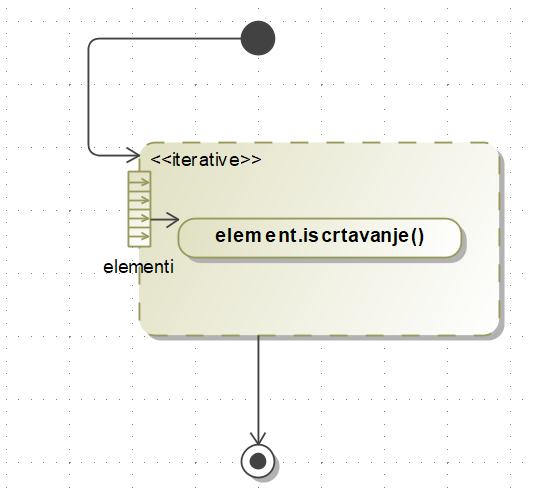
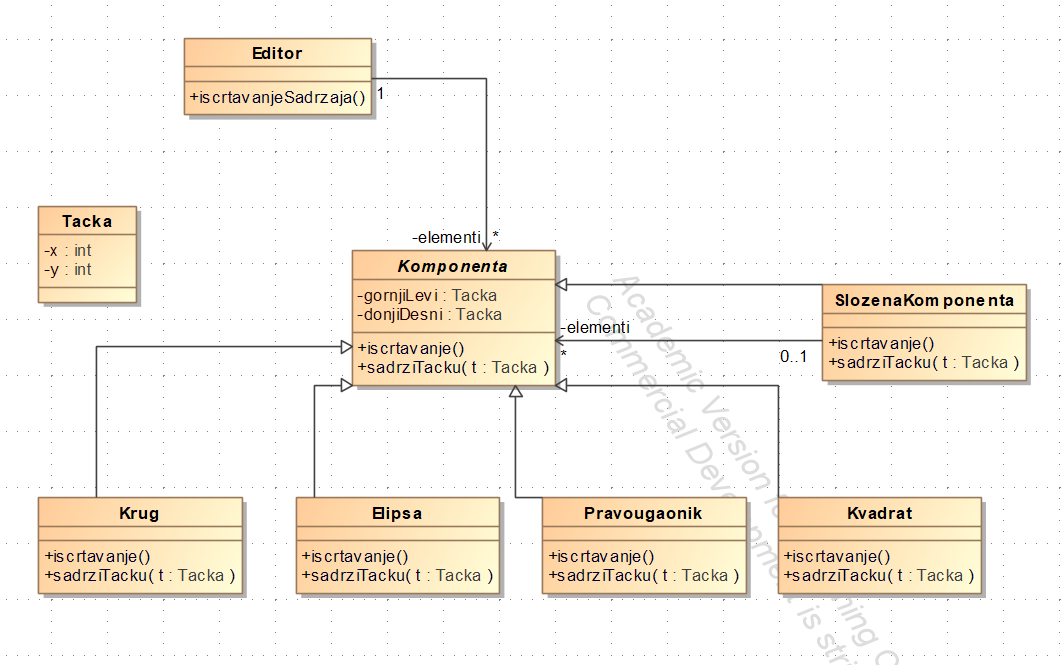
Uvid u stanje

Isplata

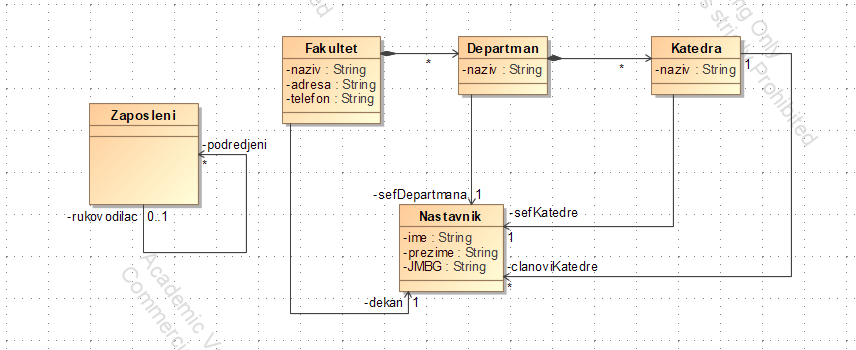
Provjera stanja

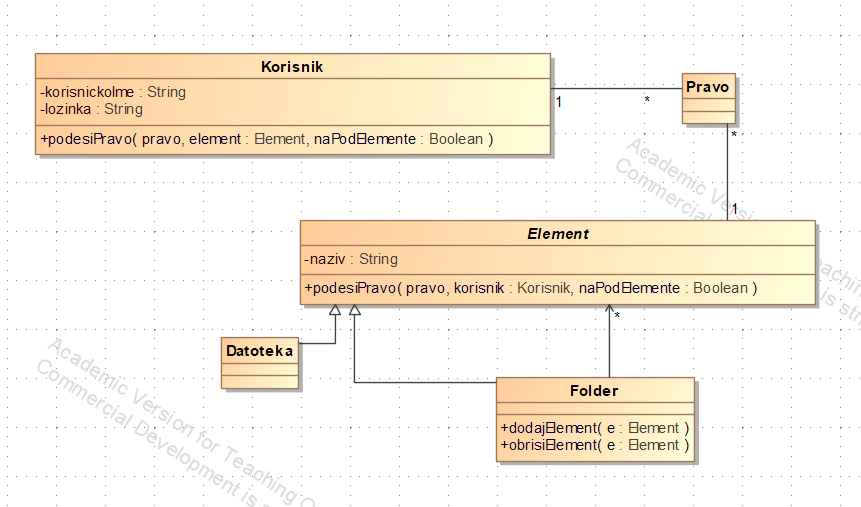
**2. zadatak – GRAFIČKI EDITOR *(potpuni klasni dijagram i dijagram aktivnosti za iscrtavanje elemenata)***

Grafički editor ima podršku za iscrtavanje jednostavnih i složenih grafičkih komponenti. Jednostavne komponente su: krug, kvadrat, pravougaonik i elipsa. Složene grafičke komponente se sastoje od jednostavnih grafičkih komponenti i prethodno kreiranih složenih grafičkih komponenti. Svaka komponenta (jednostavna ili složena) treba da poseduje: x i y koordinate gornjeg levog i donjeg desnog ugla pravougaonika u okviru kojeg se iscrtava, mogućnost da se iscrta i mogućnost da odgovori da li se zadate koordinate (x, y) nalaze u okviru nje.

****

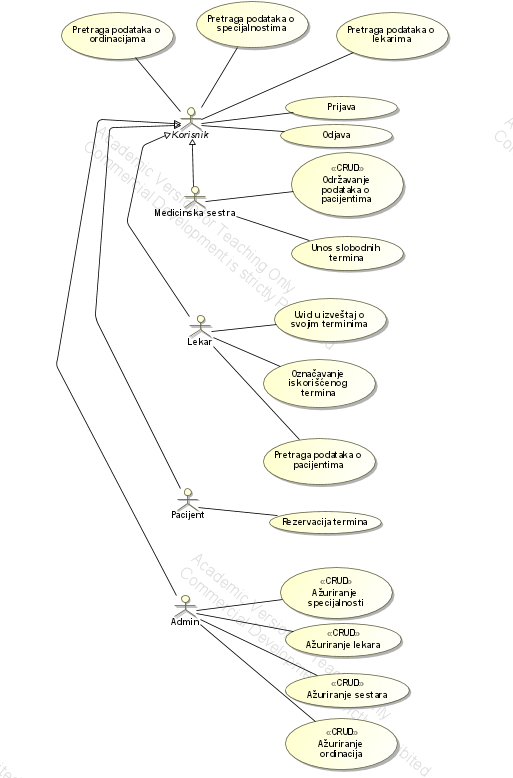
**3. zadatak – FAKULTET *(konceptualni klasni dijagram)***

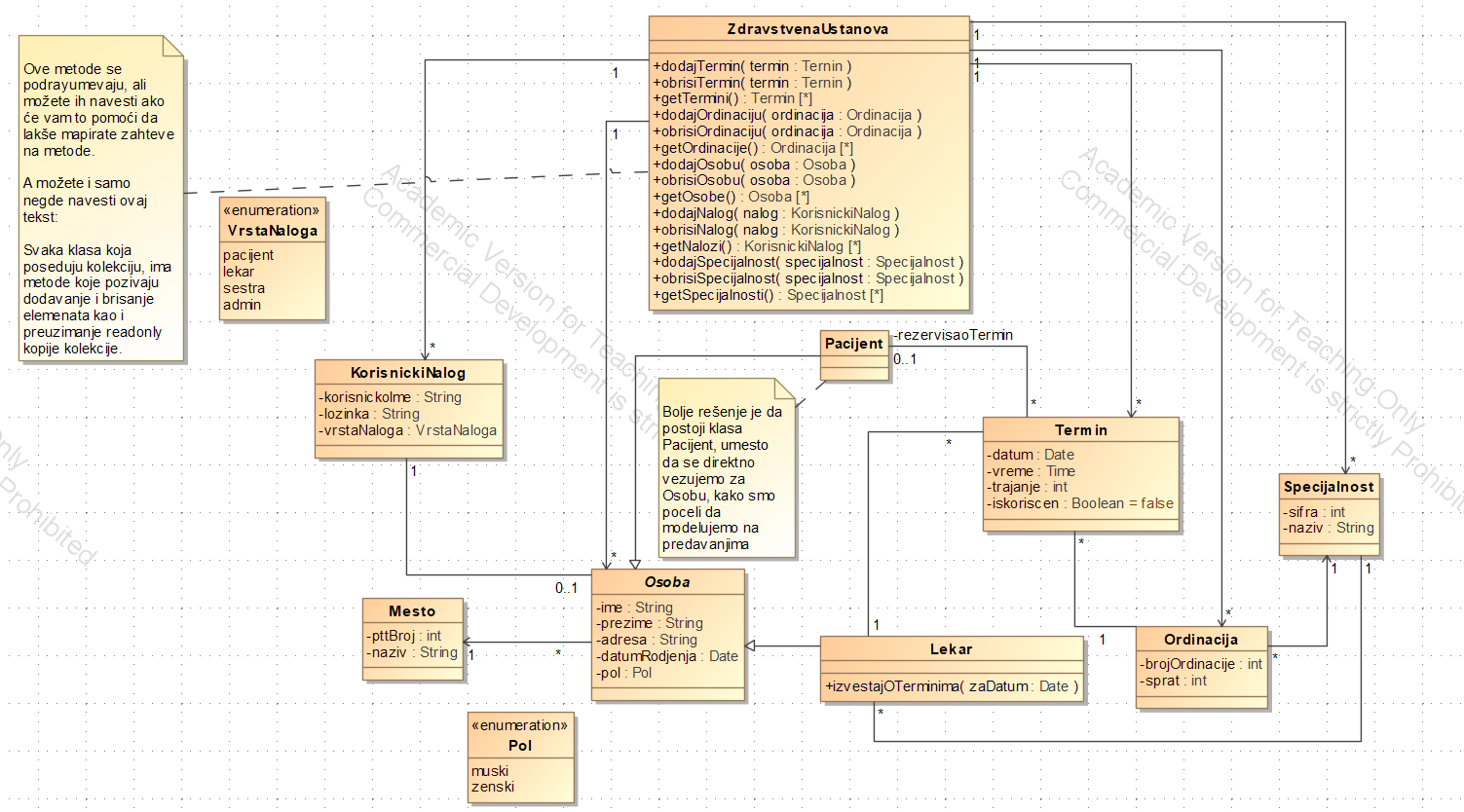
Fakultet se sastoji od departmana, a departmani od katedri. Nastavnici su raspoređeni u okviru katedri. Podaci koje je potrebno voditi za  fakultet su: naziv, adresa, telefon, dekan; za departman: naziv, rukovodilac departmana; za katedru: naziv katedre, šef katedre; za nastavnika: ime, prezime, JMBG. Dekan, rukovodilac departmana i šef katedre su nastavnici.

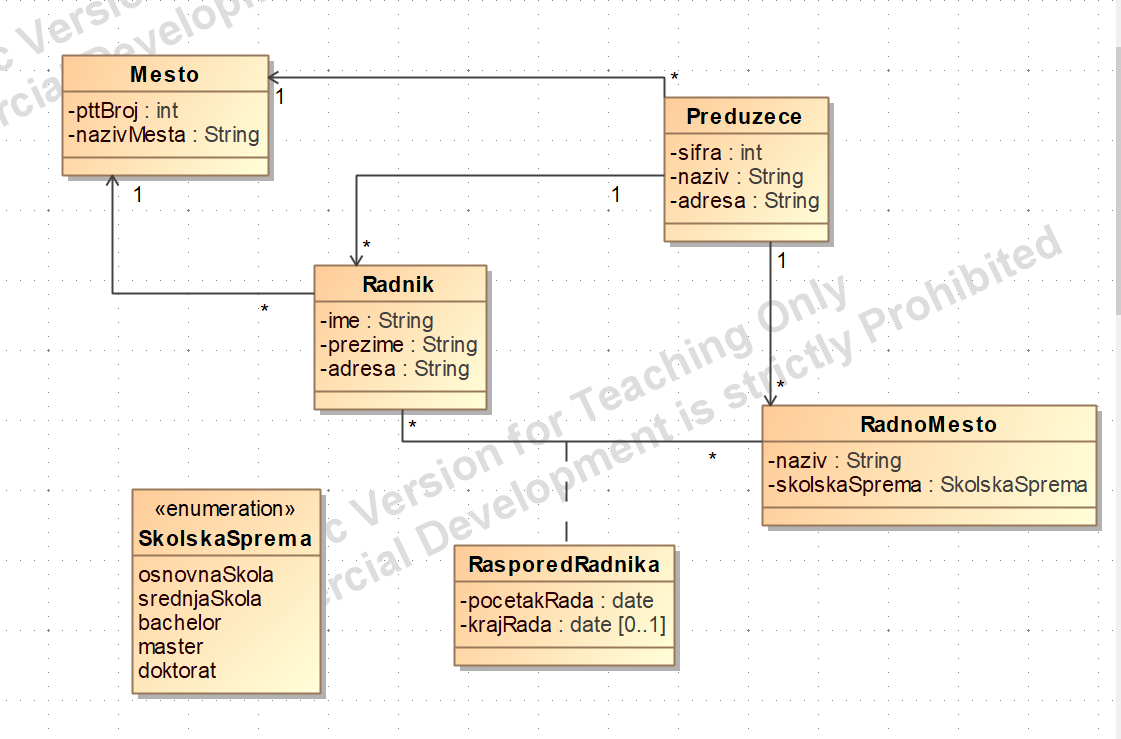


**4. zadatak – FILE SYSTEM *(potpun klasni dijagram)***

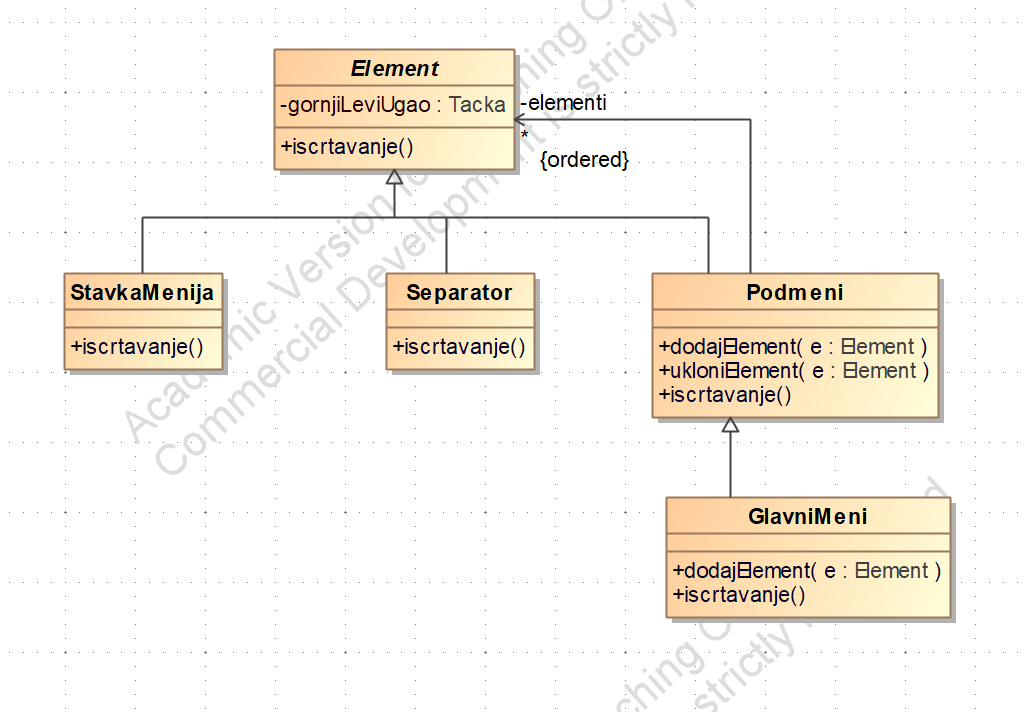
Projektovati strukturu podataka za podršku rada sistema datoteka - file system. U okviru sistema datoteka se mogu naći folderi i datoteke. U folderu se mogu naći datoteke i potfolderi. Svaki element sistema datoteka ima naziv. U folder se mogu dodavati novi elementi i brisati postojeći. Ime datoteke ili foldera se može menjati, pod uslovom da korisnik ima pravo izmene.Nad svakim elementom sistema datoteka, za svakog korisnika, može se definisati pravo pregleda, izmene i brisanja. Korisnik od podataka ima korisničko ime i lozinku.

**5. zadatak – ZDRAVSTVENA USTANOVA *(dijagram slučajeva korišćenja i klasni dijagram)***

Modelovati aplikaciju za podršku rada zdravstvene ustanove koja treba da pacijentima olakša zakazivanje pregleda. Medicinske sestre za svaki sledeći radni dan unose slobodne termine u kojima lekari mogu da prime pacijente. Termin za zakazivanje ima sledeće podatke: datum, vreme početka, trajanje, ordinacija, lekar koji radi u tom terminu. Ordinacija ima: broj ordinacije, broj sprata na kojem se nalazi i specijalnost za koju je namenjena. Svaka specijalnost ima šifru i naziv. Pacijenti iz liste termina koje su sestre unele biraju onaj termin koji njima odgovara. Po obavljenom pregledu, ako se pacijent pojavio, lekar označava da je termin iskorišćen. Lekari, medicinske sestre i pacijenti imaju sledeće podatke: ime, prezime, adresa, datum rođenja, pol (muški ili ženski). Lekar ima i specijalnost. Pored lekara, medicinskih sestara i pacijenata, korisnik ove aplikacije je i administrator, koji ima pravo da unosi i menja podatke o lekarima, ordinacijama i specijalnostima. Svi korisnici imaju pravo da pretražuju podatke o lekarima, ordinacijama i specijalnostima. Svaki korisnik ima korisničko ime i lozinku. Lekar ima pravo da pretražuje podatke o pacijentima. Unos i ažuriranje podataka o pacijentima obavljaju medicinske sestre prilikom prvog javljanja pacijenta zdravstvenoj ustanovi. Svaki lekar treba da ima mogućnost da pogleda izveštaj o svojim terminima koje su pacijenti izabrali, za zadati dan. Izveštaj treba da sadrži sledeće podatke: vreme početka, trajanje, pacijent koji je izabrao termin.

**6. zadatak – PREDUZEĆE *(konceptualni klasni dijagram)***

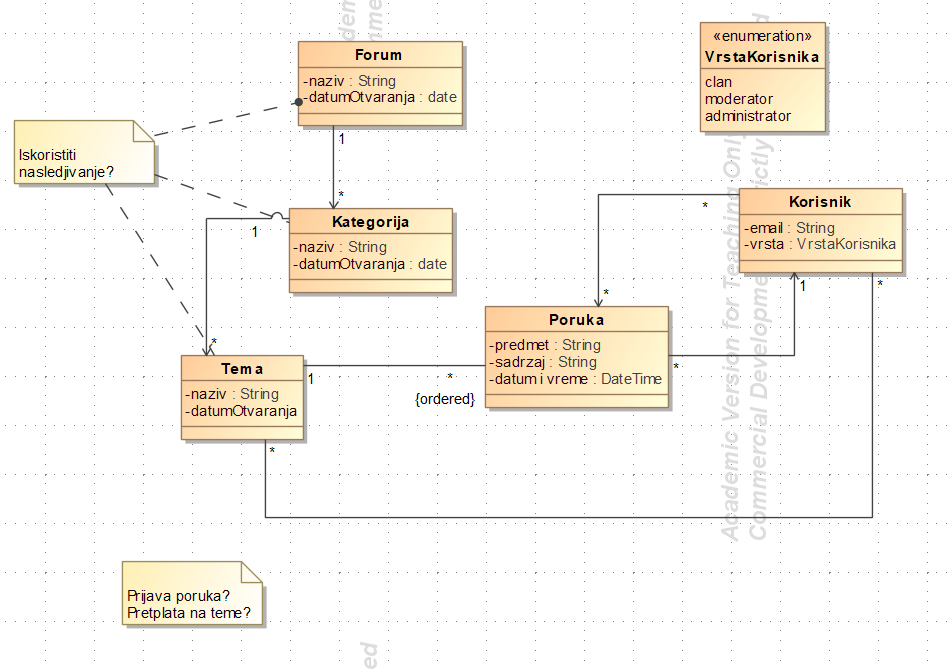
Svaki radnik u okviru preduzeća se raspoređuje na jedno radno mesto. Za radno mesto je potrebno znati naziv i minimalno potrebnu školsku spremu (osnovna škola, srednja škola, fakultet, master, doktorat). Radnik ima sledeće podatke: ime, prezime, adresa. Preduzeće ima šifru, naziv i adresu. Za svakog radnika i svako radno mesto na kojem radi/je radio u tom preduzeću je potrebno znati kada je raspoređen i kada je završio rad na tom radnom mestu (ako je završio).



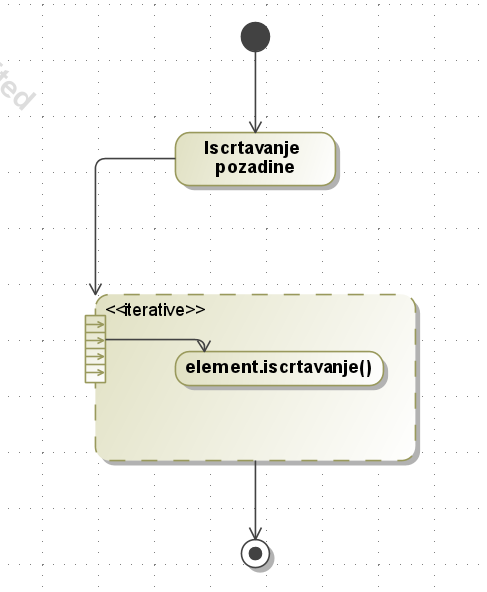
**7. zadatak – GUI BIBLIOTEKA MENI *(klasni dijagram)***

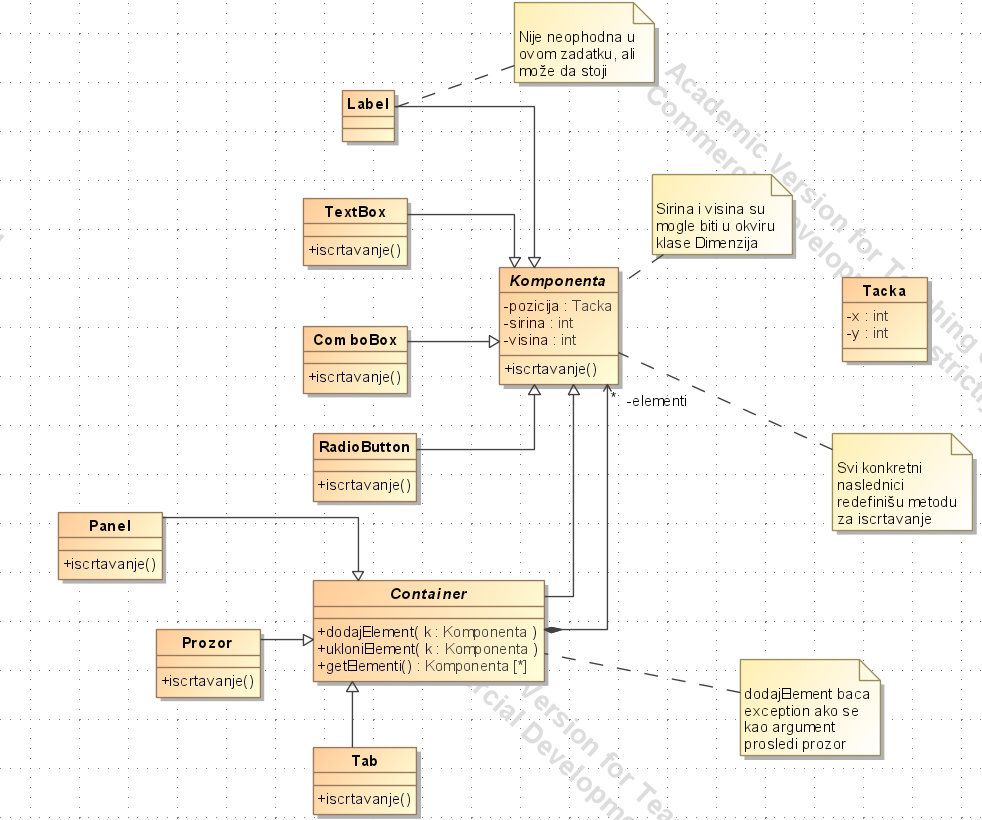
Potrebno je projektovati biblioteku grafičkih komponenti za realizaciju menija aplikacije. Glavni meni aplikacije se može sastojati od podmenija. Podmeni se može sastojati od drugih podmenija, stavki menija i separatora. Glavni meni, podmeni, stavka menija i separatori imaju poziciju na kojoj se iscrtavaju (x i y koordinata). Podmeni i stavka menija imaju i natpis. Sve navedene komponente se mogu iscrtati.

**8. zadatak – WEB FORUM *(klasni dijagram)***

****Implementirati softver koji omogućava komunikaciju na internetu u vidu web foruma. Forum ima naziv i datum otvaranja. Može da sadrži proizvoljan broj kategorija i registrovanih korisnika. Svaka kategorija ima proizvoljan broj tema. Kategorija i tema imaju naziv i datum otvaranja. Jedna tema može da budu samo u jednoj kategoriji. Teme sadrže poruke koje čitaju korisnici foruma, poređane hronološki, po datumu prispeća. Poruka ima predmet (subject), sadržaj, datum i vreme slanja, pošiljaoca (korisnik foruma) i temu kojoj pripada. Sadržaj poruke može da uključuje tekst, slike i multimedijalne sadržaje. Forum ne bi trebalo da dozvoli slanje praznih poruka, kao ni poruka koje su veće od 512KB. Da bi neko postao korisnik foruma, potrebno je da se registruje (unese e-mail adresu, nadimak, lozinku za pristup i avatar). Korisnik može da čita i piše poruke na forumu, da se pretplati na teme za koje želi da dobije obaveštenje, kao i da prijavljuje poruke koje vređaju druge učesnike. Moderator je korisnik foruma koji može da briše poruke u okviru tema i da premešta teme iz jedne kategorije u drugu. Administrator može da kreira, briše i premešta kategorije i teme. Administrator ima pravo i da blokira druge korisnike, kao i da ih promoviše u moderatore.

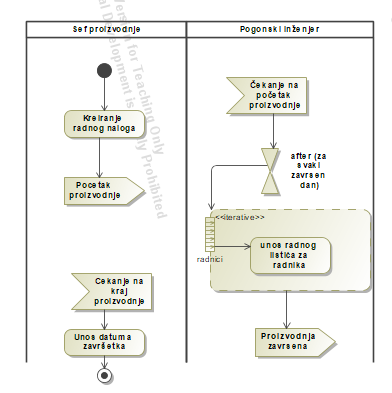
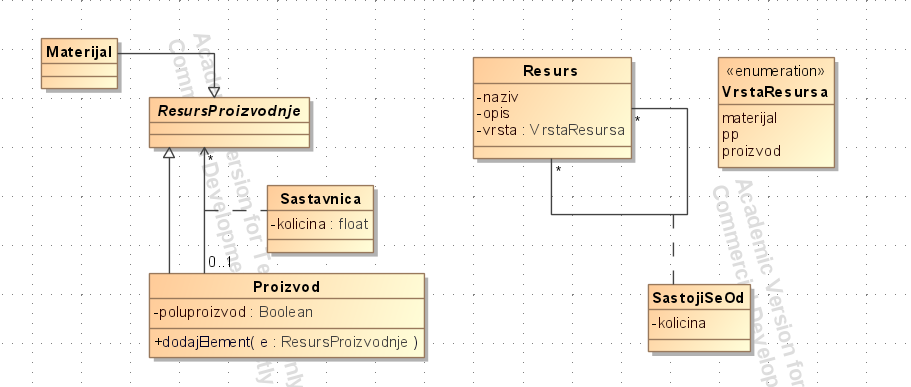
**9. zadatak – GUI BIBLIOTEKA PROZOR *(klasni dijagram i dijagram aktivnosti za iscrtavanje komponenti)***

****Potrebno je projektovati biblioteku grafičkih komponenti za realizaciju prozora grafičkih aplikacija. U okviru prozora se mogu naći komponente kao na slici. Komponente se mogu grupisati u okviru jednog ili više panela u okviru prozora. Komponente i paneli imaju x i y koordinatu gornjeg levog ugla, širinu i visinu. Potrebno je da svaka komponenta ume sebe da iscrta. Komponente koje služe za smeštanje drugih komponenti (paneli, tab-ovi, prozor) se iscrtavaju tako što iscrtaju pozadinu i zatim pozovu iscrtavanje komponenata koje se na njima nalaze.

****

**10. zadatak – PROIZVODNJA *(dijagram klasa i dijagram aktivnosti)***

Modelovati aplikaciju za podršku proizvodnje. Proizvod se sastoji od materijala i poluproizvoda. Poluproizvod se sastoji od materijala i drugih poluproizvoda. Osnovni podaci koje je potrebno voditi o materijalu, poluproizvodu i proizvodu su naziv i opis. Razlika između poluproizvoda i proizvoda je sledeća: (1) proizvod se ne može ugrađivati u druge proizvode i poluproizvode i (2) poluproizvod se ne može prodavati. Unos podataka o materijalima, proizvodima i poluproizvodima obavlja tehnolog proizvodnje. Kada je potrebno da se u okviru preduzeća počne proizvodnja nekog proizvoda ili poluproizvoda, šef proizvodnje kreira radni nalog. Radni nalog ima sledeće podatke: broj radnog naloga, datum početka proizvodnje, datum završetka proizvodnje, odgovorni radnik, proizvod ili poluproizvod koji treba da se proizvode, potrebna količina proizvoda ili poluproizvoda, radnici koji će učestvovati u proizvodnji. Kada otpočne proizvodnja po datom radnom nalogu, svakog dana pogonski inženjer unosi radne listiće koji imaju sledeće podatke: datum, radnik, broj komada koje je radnik proizveo tog dana, smena u kojoj je radio (prva, druga ili treća). Kada se završi proizvodnja po datom radnom nalogu, šef proizvodnje unosi datum završetka u radni nalog. Aplikaciju za podršku proizvodnje koristi šef proizvodnje, tehnolog proizvodnje i pogonski inženjeri. Radnici koji rade u proizvodnji ne koriste aplikaciju. Za korišćenje aplikacije potrebno je imati korisničko ime i lozinku. Radnik ima sledeće podatke: ime, prezime, adresa.

****

**11. zadatak – LOGIČKA KOLA *(dijagram klasa i dijagram aktivnosti)***

Diagram

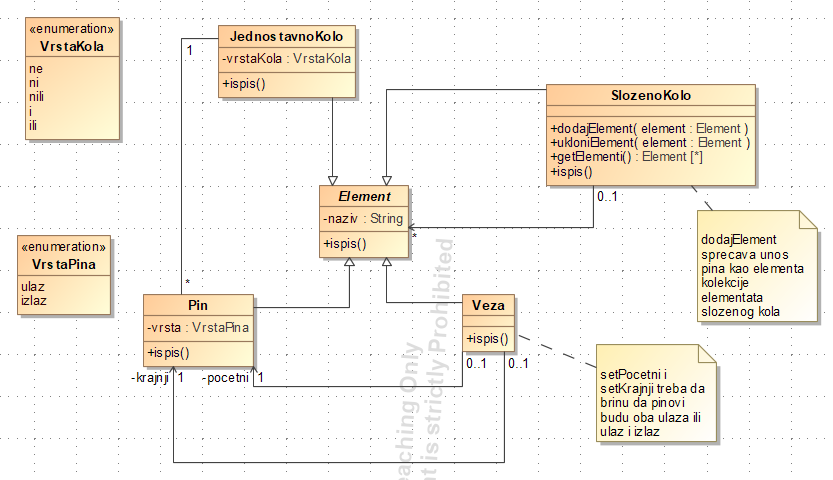
Description automatically generatedModelovati biblioteku logičkih kola. Logička kola mogu biti jednostavna i složena. Jednostavna logička kola mogu biti: NE, I, ILI, NI, NILI.

Chart, box and whisker chart

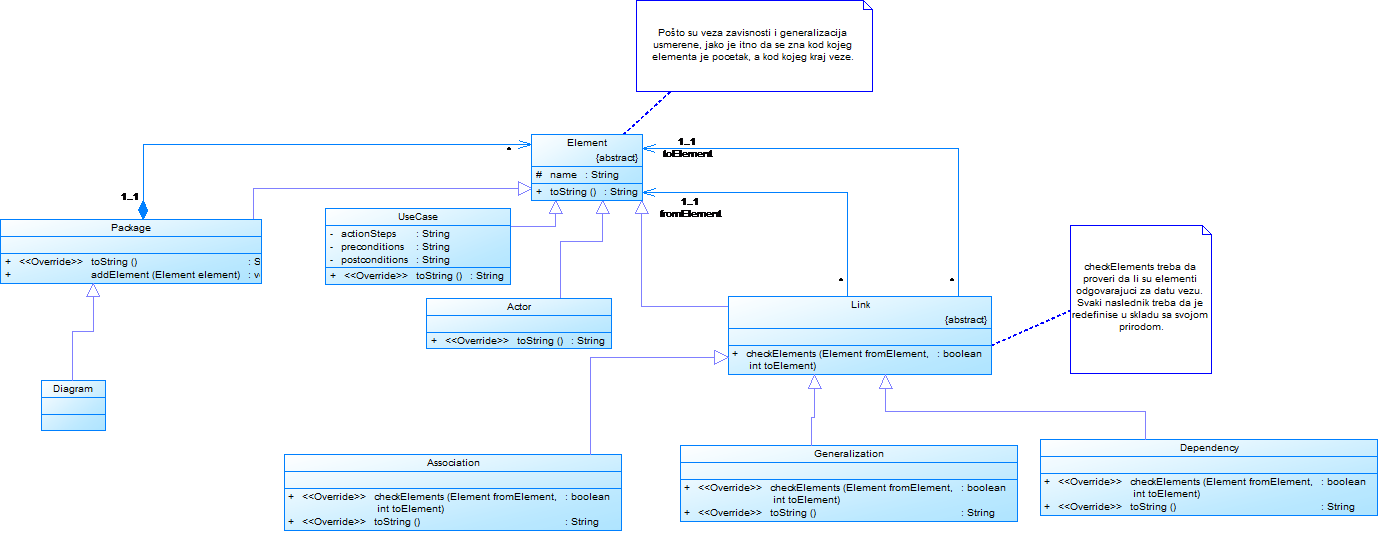
Description automatically generatedSložena kola se sastoje od jednostavnih kola i drugih složenih kola spojenih vezama. Veza može da spaja izlaz jednog kola sa ulazom drugog kola ili ulaz jednog kola sa ulazom drugog kola. Spajanje dva izlaza nije dozvoljeno. Zajednički naziv za ulaze i izlaze kola je pin.

Diagram

Description automatically generatedPodaci koji su bitni da se znaju za svako jednostavno kolo su: naziv i pinovi koje poseduje. Podaci koji su bitni za složeno kolo su: naziv i elementi od kojih se sastoji. Podaci koji su bitni za vezu su: naziv i pinovi sa kojima je povezana. Podaci koji su bitni za pin: naziv, da li je u pitanju ulaz ili izlaz, kolo kojem pripada. Svaki pomenuti element treba da ima mogućnost da ispiše podatke o sebi. Pin ispisuje svoj naziv i da li je ulazni ili izlazni. Jednostavno kolo ispisuje svoj naziv i podatke o svojim pinovima. Veza ispisuje svoj naziv i pinove sa kojima je povezana. Složena kola ispisuju svoj naziv i podatke o drugim kolima i vezama od kojih se sastoje. Ako je element složenog kola drugo složeno kolo, treba ispisati i njegovu strukturu.

****

**12. zadatak – BIBLIOTEKA USE CASE KOMPONENTI *(dijagram klasa i dijagram aktivnosti za prikaz sadržaja paketa)***

Potrebno je projektovati biblioteku komponenti za modelovanje pojednostavljenih slucajeva korišcenja Komponente mogu biti: akteri, slucajevi korišcenja, veze i paketi. Veze su: asocijacija (izmedu aktera i slucaja korišcenja), generalizacija (izmedu dva aktera ili izmedu dva slucaja korišcenja) i veza zavisnosti (izmedu dva slucaja korišcenja). Elementi paketa mogu biti: akteri slucajevi korišcenja, veze i drugi paketi. Svi navedene komponente imaju ime. Slucaj korišcenja ima i korake, preconditions i postconditions (stringovi). Svaki navedeni element treba da ima mogucnost da se iscrta na sebi svojstven nacin. Sve veze imaju koordinate pocetne tacke i krajnje tacke. Akteri, paketi i slucajevi korišcenja imaju koordinatu gornjeg levog ugla, širinu i visinu.

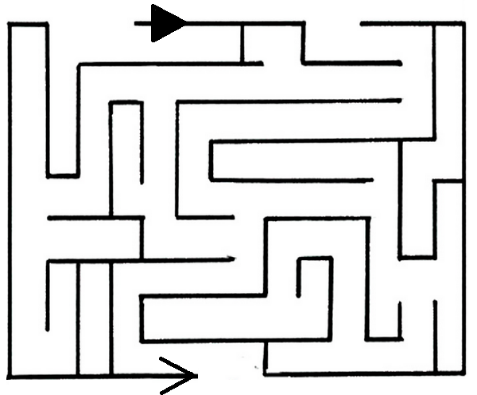
**13. zadatak – ONLINE TESTIRANJE *(dijagram slučajeva korišćenja, dijagram klasa)***

Realizovati softver za on-line testiranje studenata. Potrebno je da softver omogući sledeće: unos predmeta i nastavnika koji predaju te predmete – posao koji obavlja administrator, unos studenata koji slušaju predmete – administrator, dodela korisničkih imena i lozinki za nastavnike i studente – administrator, definisanje testova - nastavnici, rešavanje testova i uvid u rezultate (bodovi i ocene) – studenti, uvid u spisak studenata i njihovih bodova i ocena za izabrani test- nastavnici.

Testovi se formiraju tako što nastavnik bira predmet za koji želi da unese test (može da izabere samo predmet koji predaje), definiše redni broj i naziv testa i zatim unese pitanja. Svako pitanje može da ima proizvoljan broj ponuđenih odgovora, od kojih je samo jedan tačan. Formirani test je inicijalno neaktivan, dok ne dođe vreme za polaganje. Tek kad se test aktivira od strane nastavnika, studenti koji slušaju taj predmet mogu da mu pristupe i počnu rešavanje. Kada se test završi, sistem šalje rezultate studentima (samo njihovu ocenu i bodove) i nastavniku (spisak sa svim studentima koji su polagali i njihovim rezultatima).

Podaci koje je potrebno čuvati su: predmet: šifra, naziv, semestar; nastavnik i student: jmbg, prezime, ime, adresa stanovanja; test: redni broj testa, naziv testa, predmet za koji se kreira, procenat učešća tog testa u okviru ukupne ocene i vreme trajanja testa.

**14. zadatak – LAVIRINT *(dijagram klasa i dijagram aktivnosti za iscrtavanje)***

Potrebno je projektovati biblioteku grafičkih komponenti za crtanje lavirinta. Lavirint se sastoji od sledećih elemenata: zidova, ulaza i izlaza. Svaki element lavirinta ima početnu tačku (x i y koordinata), dužinu u smeru rastuće odgovarajuće koordinate i orijentaciju (horizontalna ili vertikalna). Svaki element može da se iscrta, pomeri i promeni mu se orijentacija. Prost lavirint sadrži proizvoljan broj elemenata od kojih je samo jedan ulaz i jedan izlaz. Pokušaj dodavanja drugog ulaza ili izlaza je greška. Složen lavirint poseduje druge lavirinte (proste ili složene). Ulaz složenog lavirinta je ulaz prvog sadržanog lavirinta. Izlaz složenog lavirinta je izlaz poslednjeg sadržanog lavirinta. Sadržanim lavirintima se spajaju izlaz prethodnog sa ulazom narednog lavirinta.

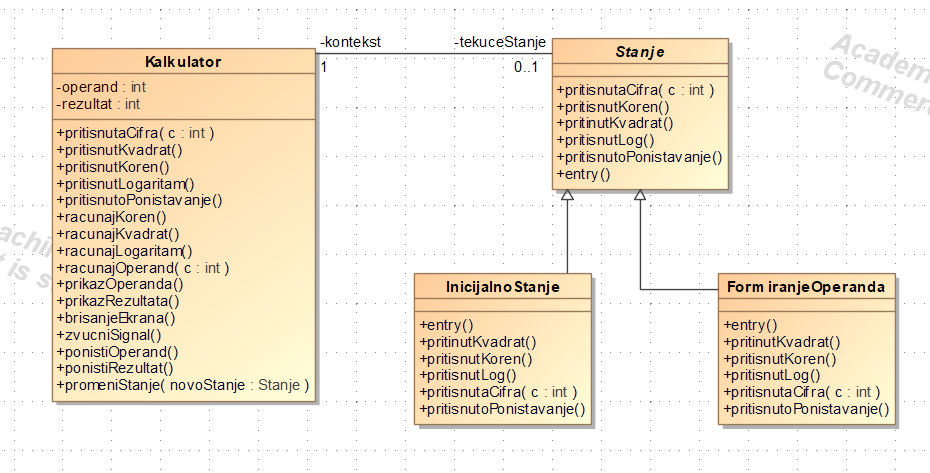
**KOLOKVIJUM 2**

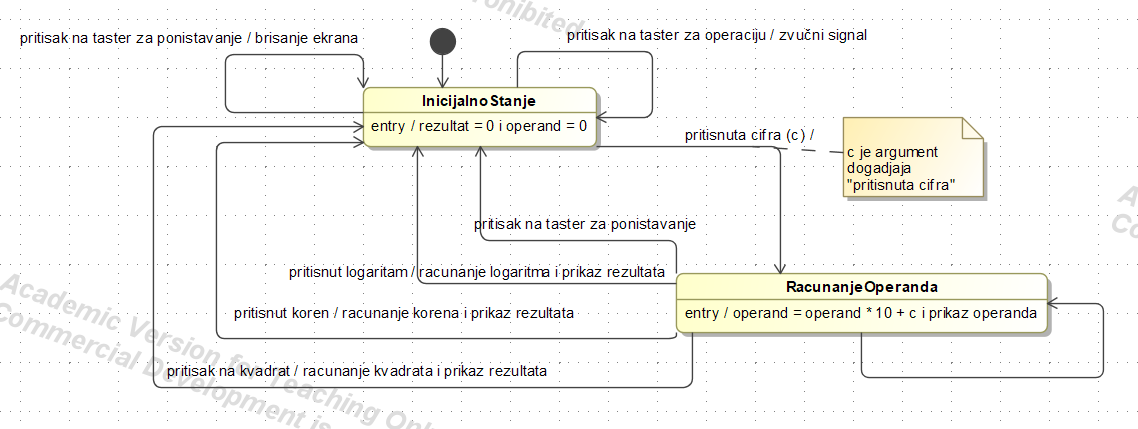
**1. zadatak – ŠTAMPAČ *(dijagram prelaza stanja)***

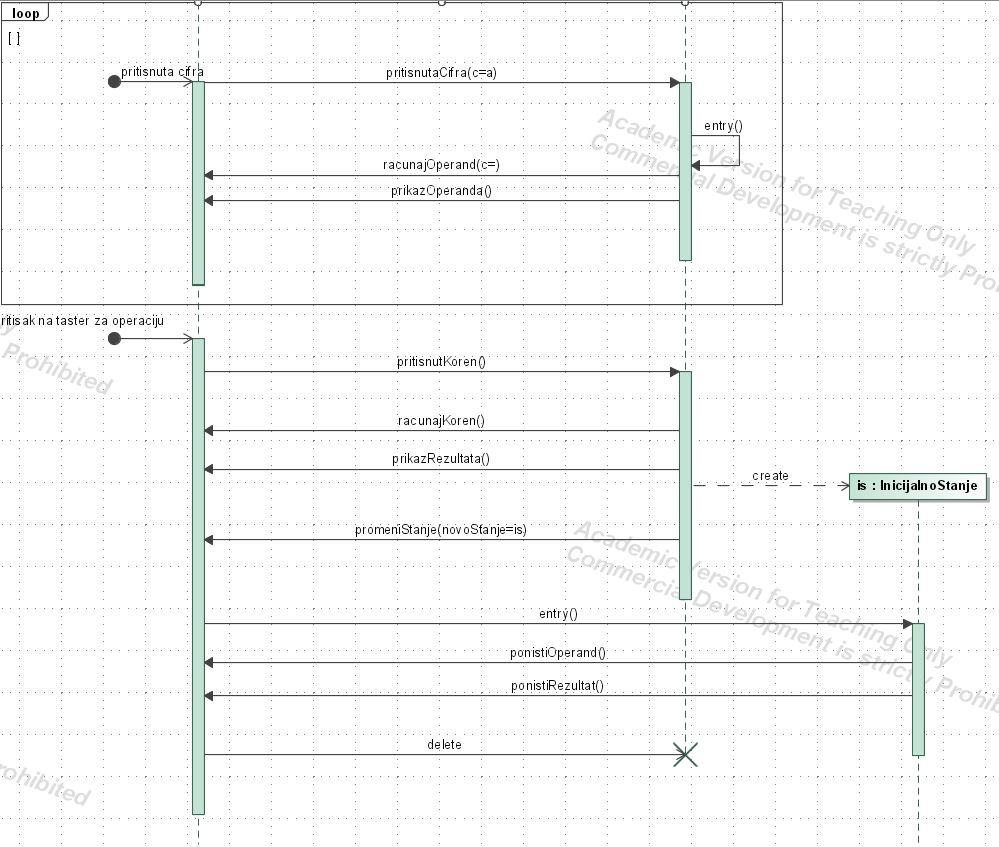
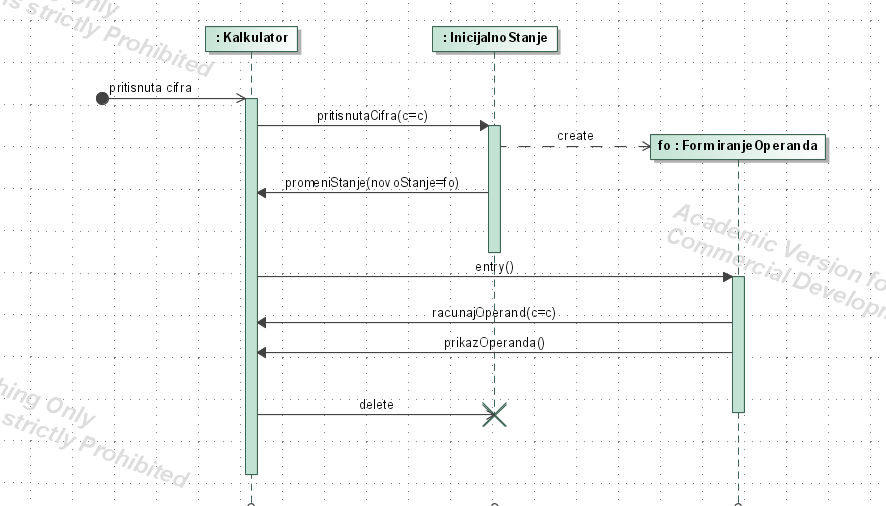
Kada se uključi, štampač se nalazi u stanju čekanja na datoteku koju treba da štampa. U tom trenutku štampač treba da prikaže status koji označava da je spreman. Kada datoteka stigne, počinje sa štampom, a status se menja u „zauzet“. Po završetku štampe, štampač se vraća u stanje čekanja na novi doku­ment i ponovo prikazuje status „spreman“. Ako se prilikom štampe zaglavi papir, štampač treba da prikaže odgovarajući status greške i da čeka da neko ukloni papir. Po uklanjanju papira, nastavlja sa štampom, a status menja na „zauzet“. Ako prilikom štampe nestane papira, štampač prikazuje status greške i čeka na dopunu papira. Kada se dopuna obavi, nastavlja sa štampom i prikazuje status „zauzet“. Radi jednostavnosti, pretpostavimo da štampač može da prima nove datoteke samo dok je u stanju čekanja na datoteku.

**2. zadatak – KALKULATOR *(state šablon, dijagram stanja, dijagram sekvence)***

Jednostavan kalkulator omogućava računanje kvadrata, korena i logaritma za osnovu 2 za višecifrene brojeve. Njegova tastatura se sastoji od cifara (0..9), znakova za operacije (x2 -kvadrat, sqrt - koren i log2 - logaritam za osnovu 2), kao i tastera za poništavanje rezultata prethodnog računanja. Posle uključivanja kalkulatora ili posle poništavanja rezultata, kalkulator se nalazi u inicijalnom stanju koje postavlja rezultat na nulu. Kalkulator u tom stanju čeka sledeći pritisak tastera. Ako se pritisne znak za operaciju (x2, sqrt ili log2) , kalkulator javlja grešku (zvučni signal) i vraća se u inicijalno stanje. Ako se pritisne cifra, kalkulator ulazi u stanje formiranja operanda (višecifrenog broja). Posle svakog pritiska cifre, operand se formira na sledeći način: operand = prethodna vrednost operanda \* 10 + uneta cifra, a na ekranu se prikazuje vrednost tog operanda. Pritiskom na taster za operaciju, kalkulator prikazuje rezultat na ekranu i prelazi u inicijalno stanje. Pritiskom na taster za poništavanje, kalkulator prelazi u inicijalno stanje i prikazuje nulu na ekranu.







**KOD PROFESORKE RJESENJA:**

3. zadatak. Jednostavan kalkulator omogućava računanje kvadrata, korena i logaritma za osnovu 2 za višecifrene brojeve. Njegova tastatura se sastoji od cifara (0..9), znakova za operacije (x2 -kvadrat, sqrt - koren i log2 - logaritam za osnovu 2), kao i tastera za poništavanje rezultata prethodnog računanja. Posle uključivanja kalkulatora ili posle poništavanja rezultata, kalkulator se nalazi u inicijalnom stanju koje postavlja rezultat na nulu. Kalkulator u tom stanju čeka sledeći pritisak tastera.

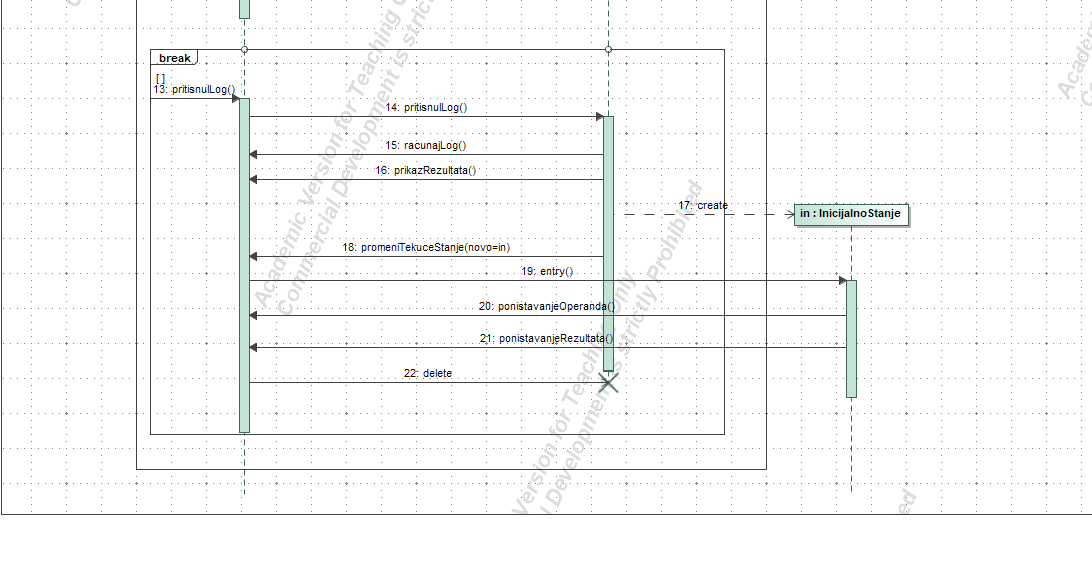
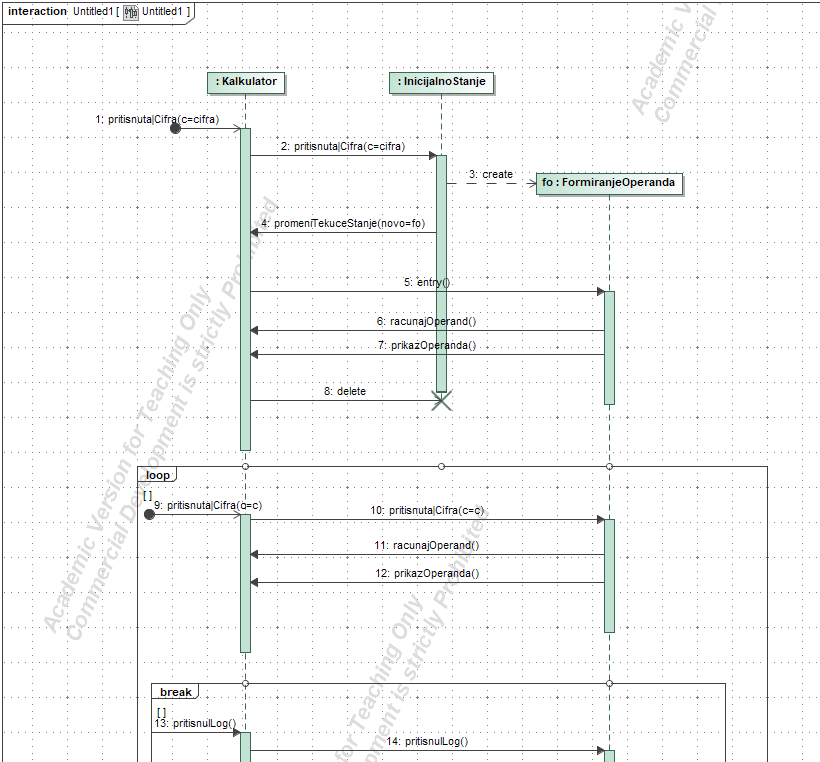
Ako se pritisne znak za operaciju (x2, sqrt ili log2), kalkulator javlja grešku (zvučni signal) i vraća se u inicijalno stanje. Ako se pritisne cifra, kalkulator ulazi u stanje formiranja operanda (višecifrenog broja). Posle svakog pritiska cifre, operand se formira na sledeći način:

operand = prethodna vrednost operanda \* 10 + uneta cifra

a na ekranu se prikazuje vrednost tog operanda. Pritiskom na taster za operaciju, kalkulator prikazuje rezultat na ekranu i prelazi u inicijalno stanje. Pritiskom na taster za poništavanje, kalkulator prelazi u inicijalno stanje i prikazuje nulu na ekranu.

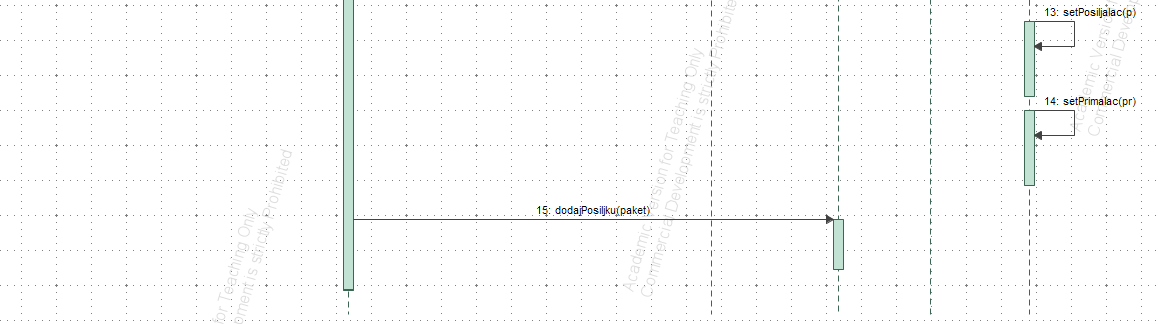
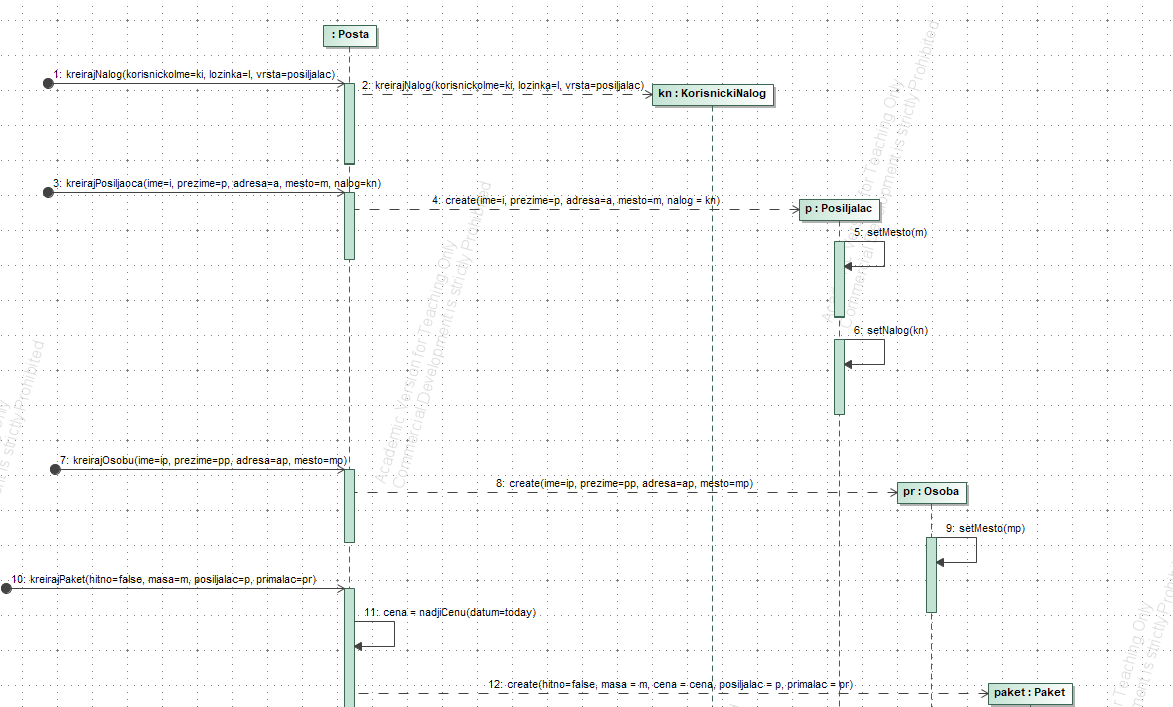
Modelovati:  
a) dijagram prelaza stanja,  
b) odgovarajući dijagram klasa za implementaciju rešenja pod a) (koristiti state projektni šablon)  
c) dijagram sekvence koji prikazuje proces unosa operanda i računanja rezultata.

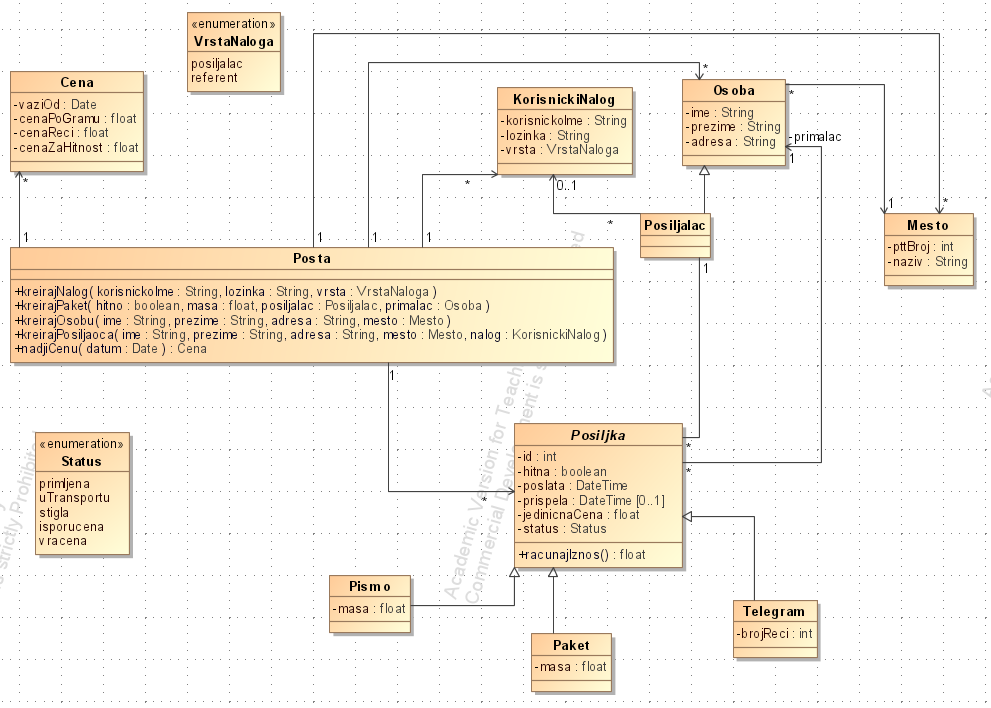
****

****

**3. zadatak – POŠILJKE *(dijagram klasa i dijagram sekvence za slanje pošiljke)***

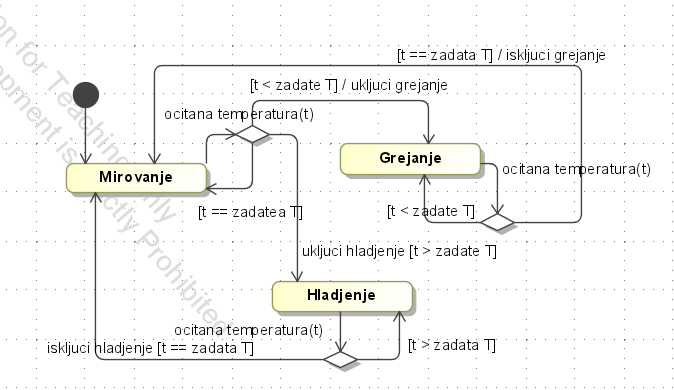
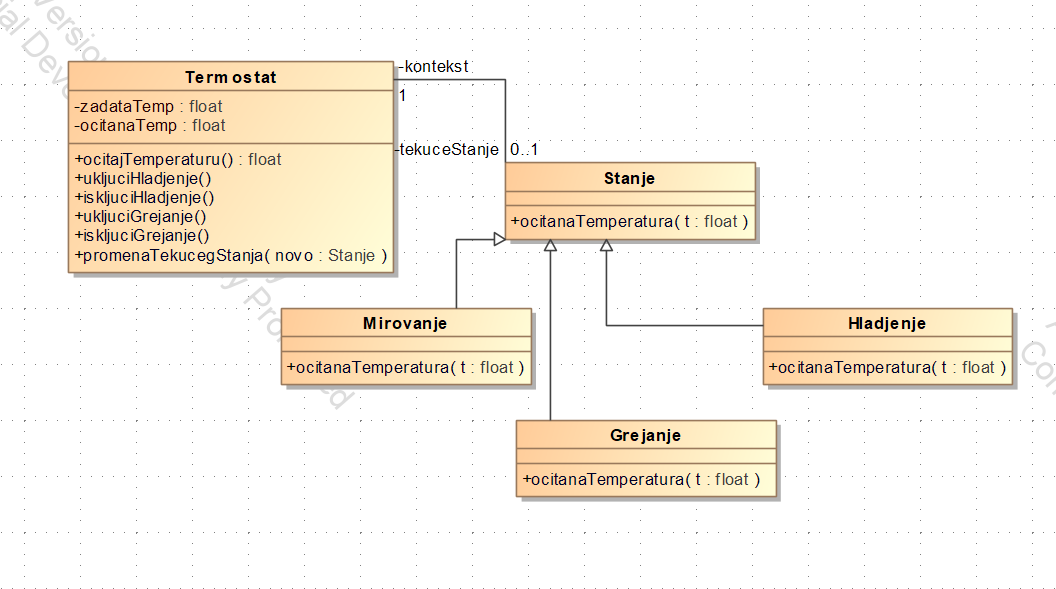
Modelovati softver za podršku sistema za slanje i prijem pošiljki. Pošiljke mogu biti: Pasma, paketi i telegrami. Cena otpremanja pisma i paketa se računa kao: masa pisma ili paketa u gramima \* cena po gramu, a telegrama na osnovu broja reči (cena reči \* broj reči). Ako se prilikom slanja naznači da je u pitanju hitna pošiljka, zaračunava se dodatni iznos koji ne zavisi od vrste pošiljke. Prilikom slanja, predaje referentu pošiljku, koji joj dodeljuje jedinstveni ID, beleži datum i vreme prijema, računa cenu i preuzima podatke pošiljaoca i primaoca. Podaci primaoca i pošiljaoca su: ime, prezime, poštanski broj, mesto i adresa stanovanja. Pošiljalac dobija i korisničko ime i lozinku, kao i ID dodeljen pošljki, da bi mogao da prati njen status (primljena, u transportu, stigla u mesto isporuke, isporučena, vraćena). Pošiljka se vraća pošiljaocu ukoliko, po prispeću u mesto isporuke, ne uspeju kontaktirati primaoca u roku od 7 dana od prijema pošiljke. Pošiljalac treba da ima mogućnost uvida u istoriju slanja svih svojih pošiljki: koja pošiljka je u pitanju, cena u trenutku slanja, datum i vreme slanja, datum i vreme prispeća. Softver treba da ima sve cene po kojima je računat ili se trenutno računa iznos pošiljki i datume kada su te cene važile.

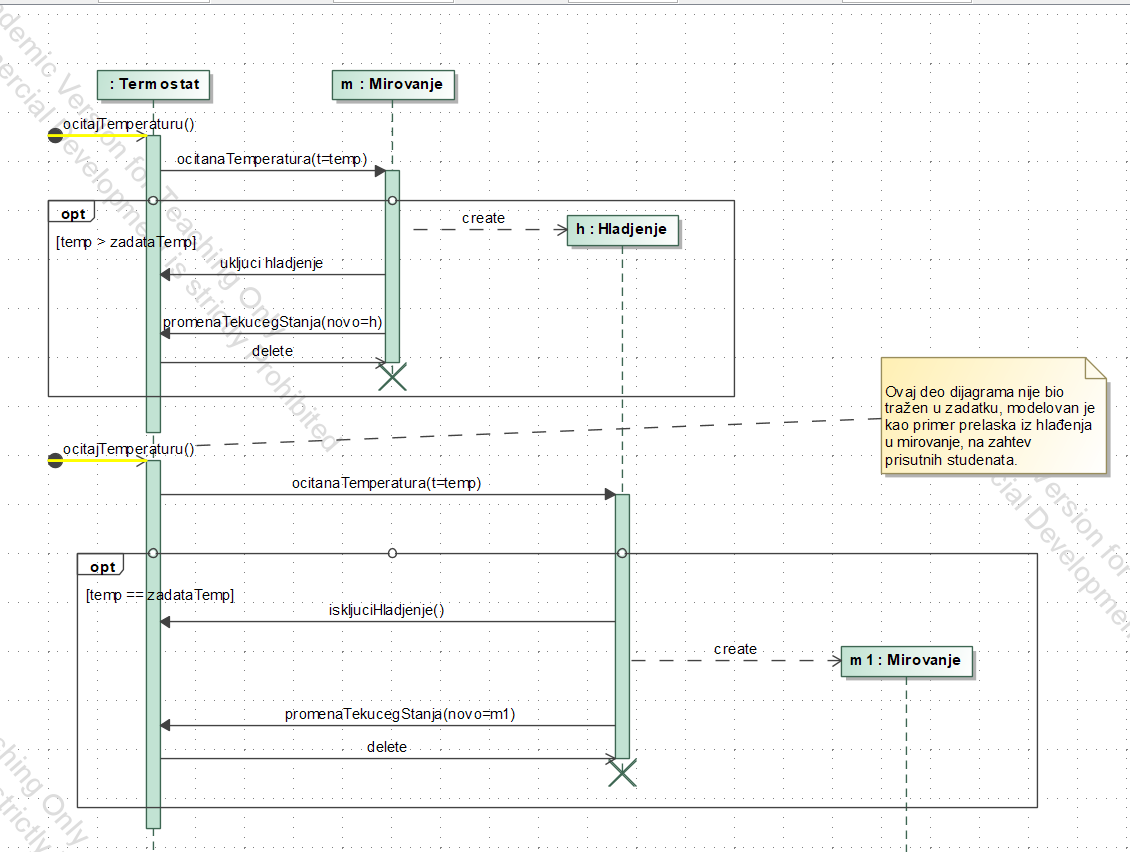




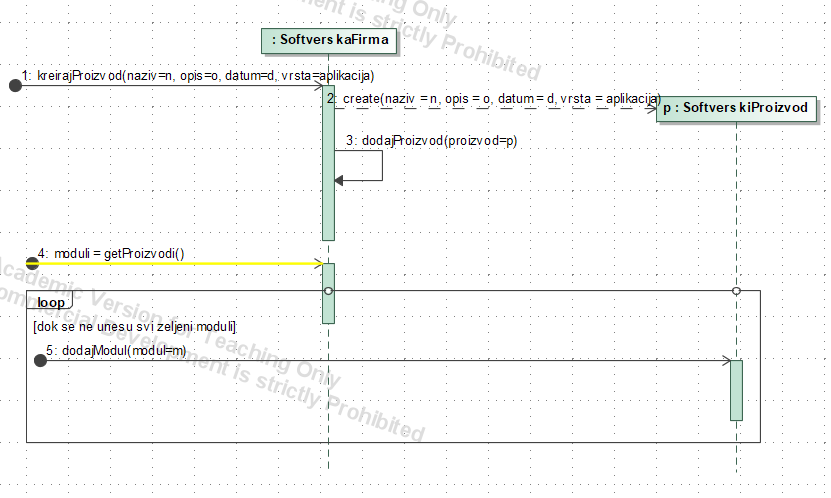
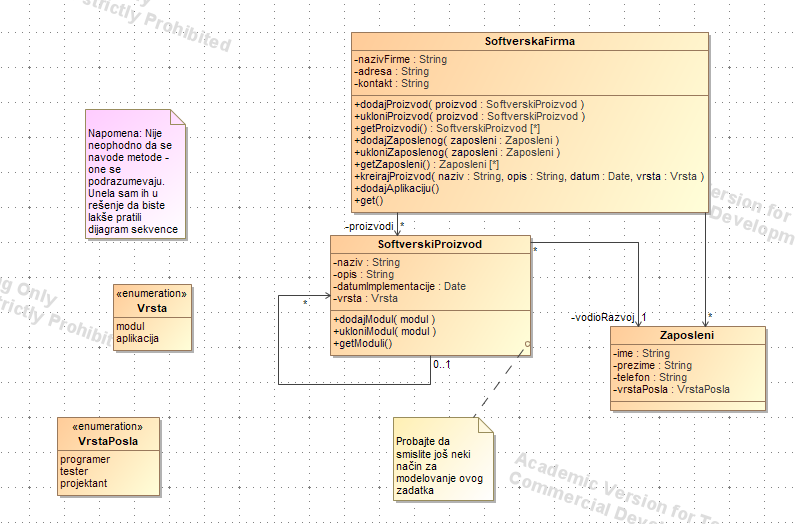
**4. zadatak – TERMOSTAT *(state šablon, dijagram stanja, dijagram sekvence za hlađenje)***

Modelovati termostat u okviru klima uređaja. Kada korisnik podesi željenu temperaturu, klima uređaj radi sledeće: Ako je spoljna temperatura veća od zadate, termostat aktivira hlađenje. Kada se spoljna temperatura izjednači sa zadatom, termostat isključuje hlađenje. Ako je spoljna temperatura manja od zadate, termostat aktivira grejanje. Kada se spoljna temperatura izjednači sa zadatom, termostat isključuje grejanje. Termostat proverava spoljnu temperaturu svakih pet sekundi.

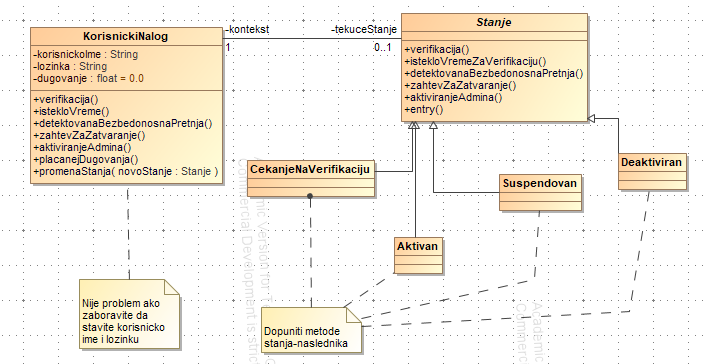
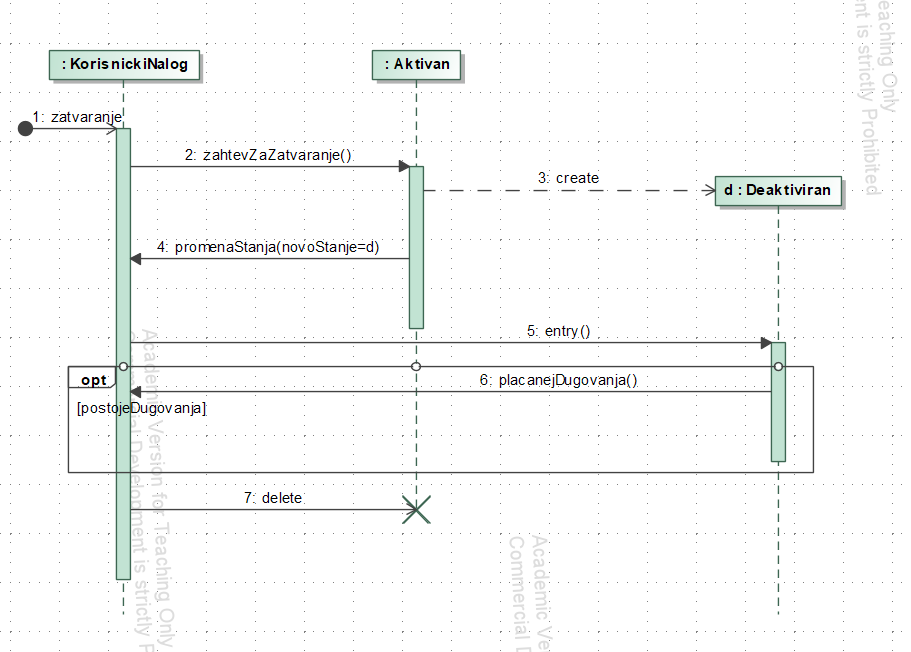
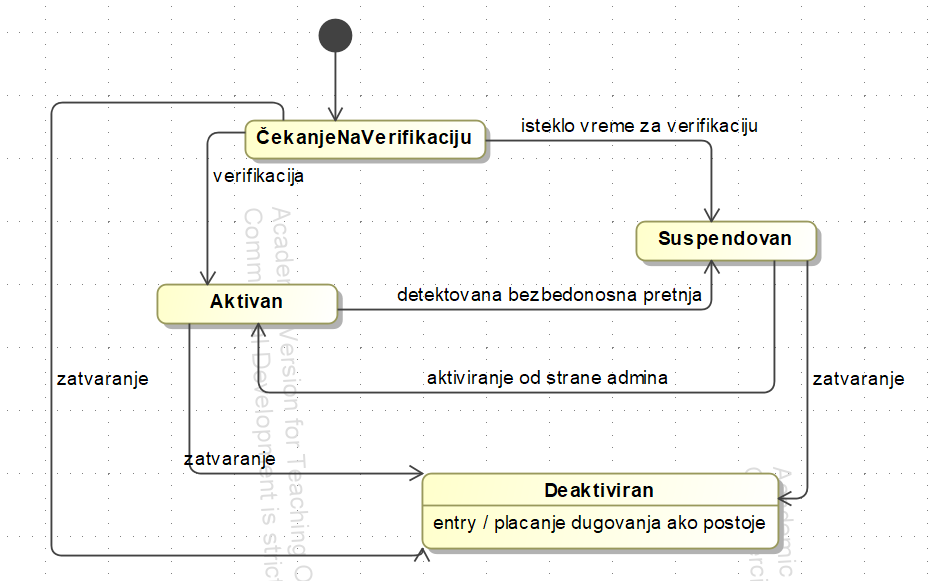
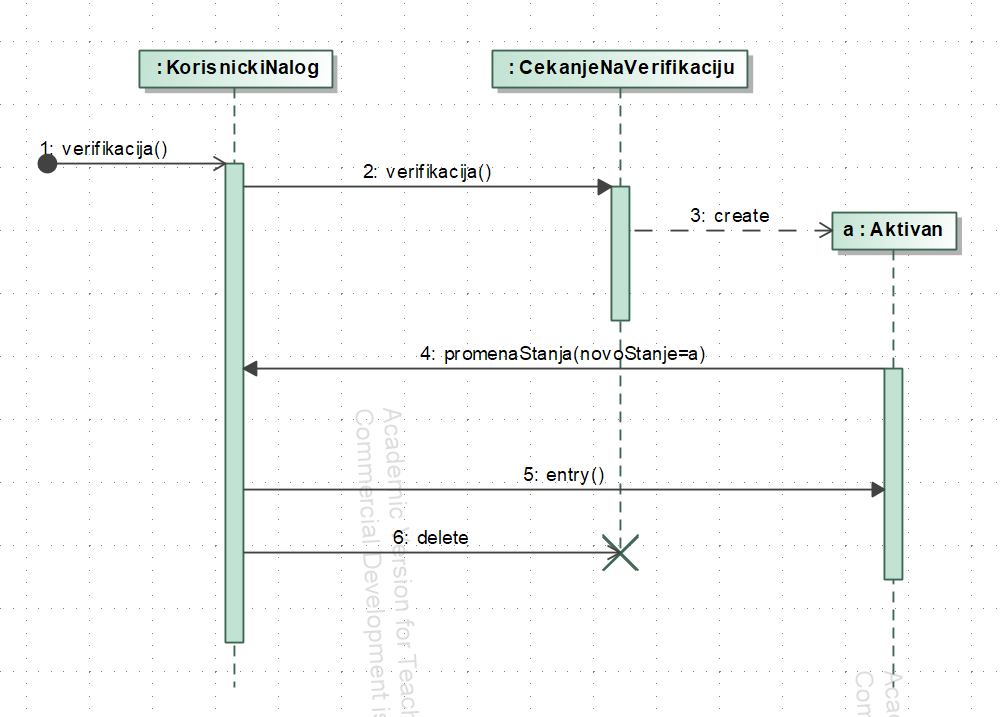


****

**5. zadatak – SOFTVERSKA FIRMA *(klasni dijagram i dijagram sekvence za unos podataka o gotovim aplikacijama)***

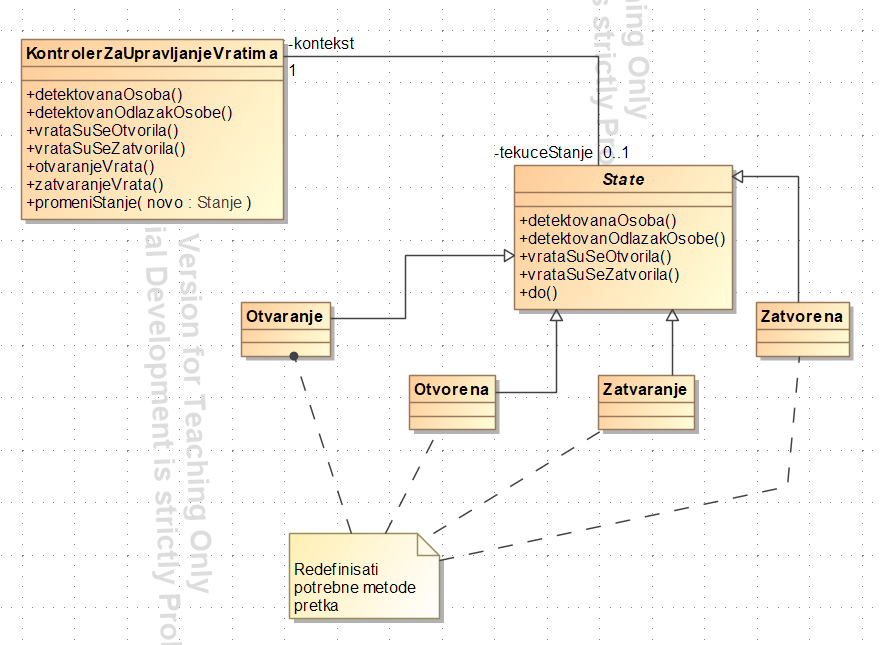
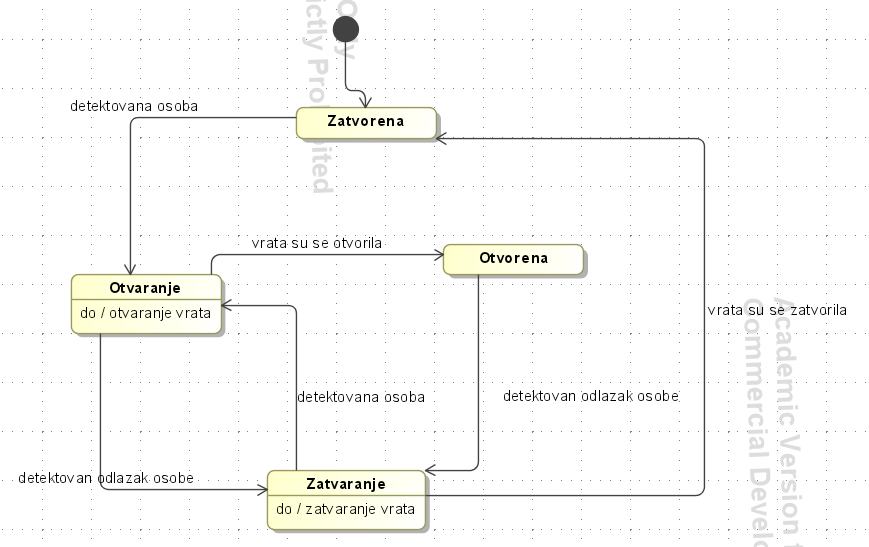
****Modelovati dijagram klasa za aplikaciju za podršku rada softverske firme koja treba da omogući ažuriranje i pretragu podataka o softverskim proizvodima koje data firma implementira kao i podataka o njenim zaposlenima. Softverski proizvodi mogu biti programski moduli ili gotove aplikacije. Gotove aplikacije se sastoje od programskih modula. Programski moduli se mogu sastojati od drugih programskih modula. Svaki softverski proizvod ima svoj naziv, opis, datum kada je implementiran i zaposlenog koji je vodio njegov razvoj. Zaposleni ima atribute: ime, prezime, broj telefona i vrstu posla koju obavlja. Moguće vrste posla su: projektant, programer i tester.

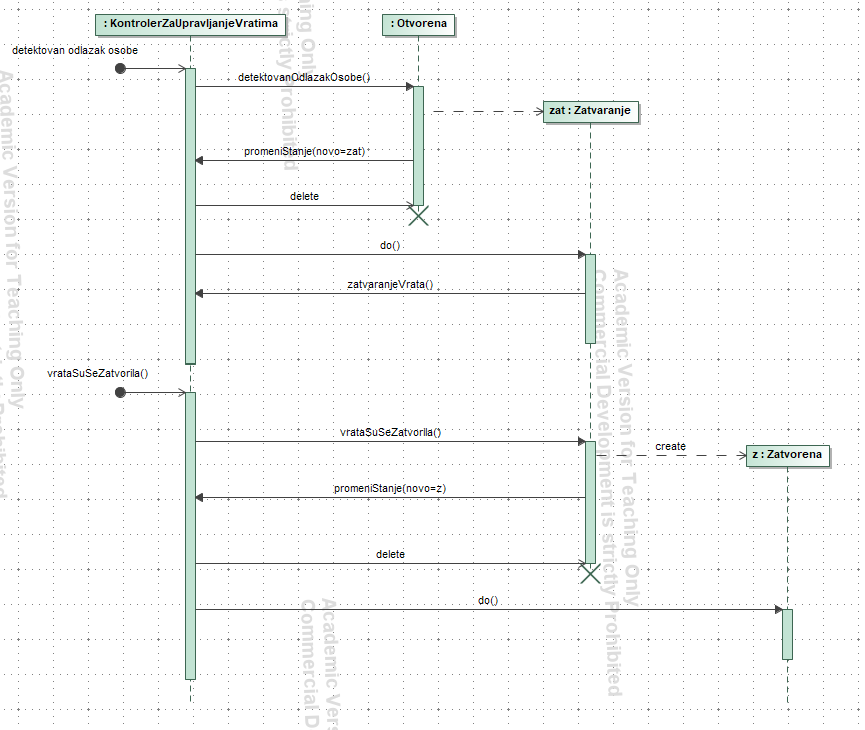
**6. zadatak – KORISNIČKI NALOG U ONLINE PRODAJI *(state šablon, dijagram stanja, dijagram sekvence za verifikaciju naloga i dijagram sekvence za deaktiviranje naloga)***

******Modelovati životni ciklus korisničkog naloga u okviru sajta za online kupovinu. Po kreiranju novog naloga, on se nalazi u stanju čekanja na verifikaciju, putem linka koji korisniku stiže na e-mail. Posle verifikacije, nalog postaje aktivan. Ukoliko se nalog ne verifikuje u vremenu od tri dana, postaje suspendovan. Nalog može postati suspendovan i ako sistem detektuje neke bezbedonosne pretnje. Iz suspendovanog režima nalog može ponovo preći u aktivni režim aktiviranjem od strane administratora. Ako korisnik pošalje zahtev za zatvaranjem naloga, nalog postaje deaktiviran. Zatvaranje se može zahtevati u svakoj životnoj fazi naloga. Prilikom zatvaranja, neophodno je platiti sva zaostala dugovanja, ako postoje.

**7. zadatak – AUTOMATSKA VRATA *(dijagram prelaza stanja, state šablon, dijagram sekvence koji prikazuje situaciju kada su vrata otvorena i senzor aktivira zatvaranje)***

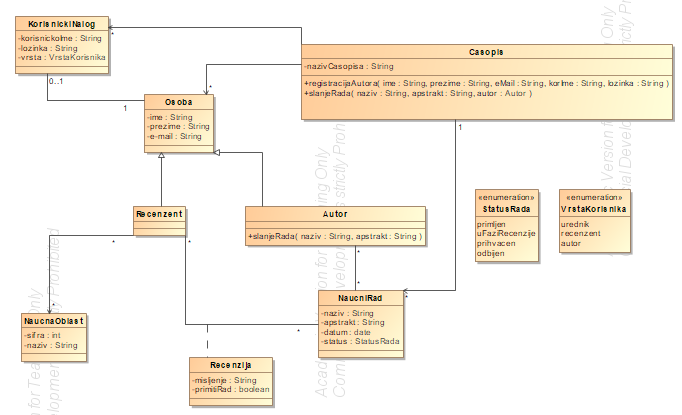
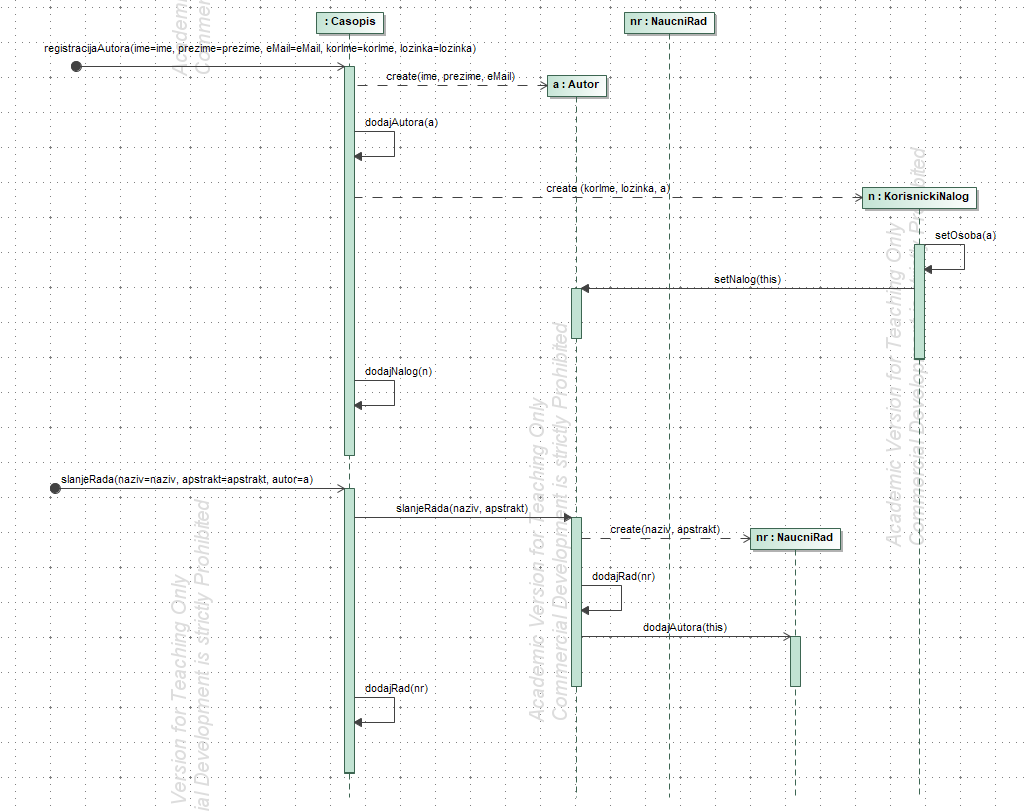
Projektovati automat za upravljanje automatskim vratima. Potrebno je da se vrata automatski otvore kada senzor detektuje da se vratima približila osoba na udaljenosti od 70 cm. Ako su vrata otvorena, a senzor detektuje da nema nikoga na udaljenosti od 70 cm, potrebno je da aktivira zatvaranje vrata. Ako u bilo kom trenutku u toku zatvaranja senzor detektuje da se neko približio, potrebno je da se istog trenutka prekine zatvaranje i počne sa otvaranjem vrata. Ako u bilo kom trenutku u toku otvaranja senzor detektuje da nema nikog u blizini vrata (osoba zbog koje se aktiviralo otvaranje se udaljila), potrebno je da se istog trenutka prekine sa  otvaranjem i pređe na zatvaranje vrata.

****

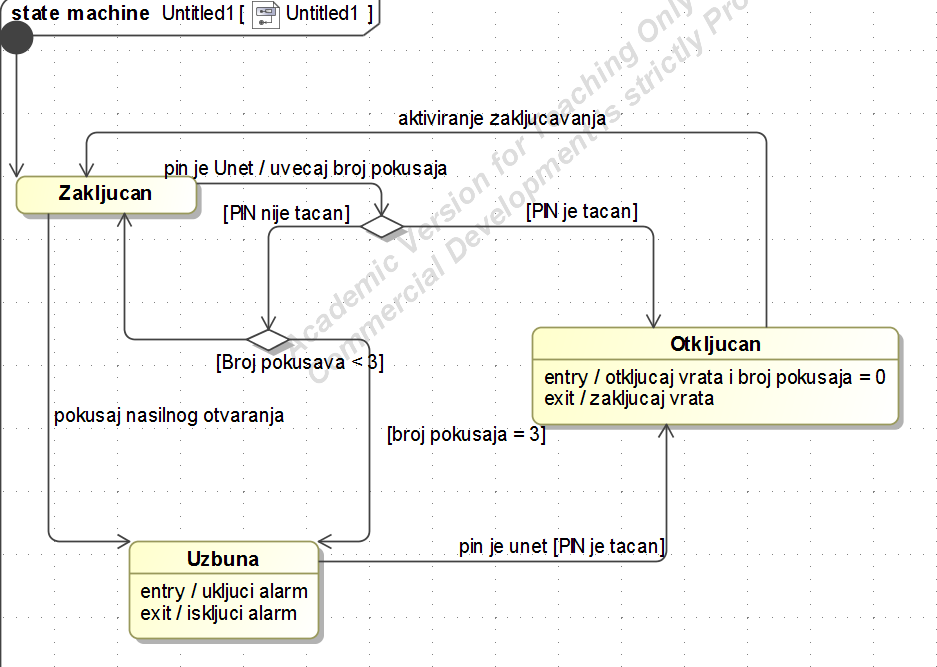
****

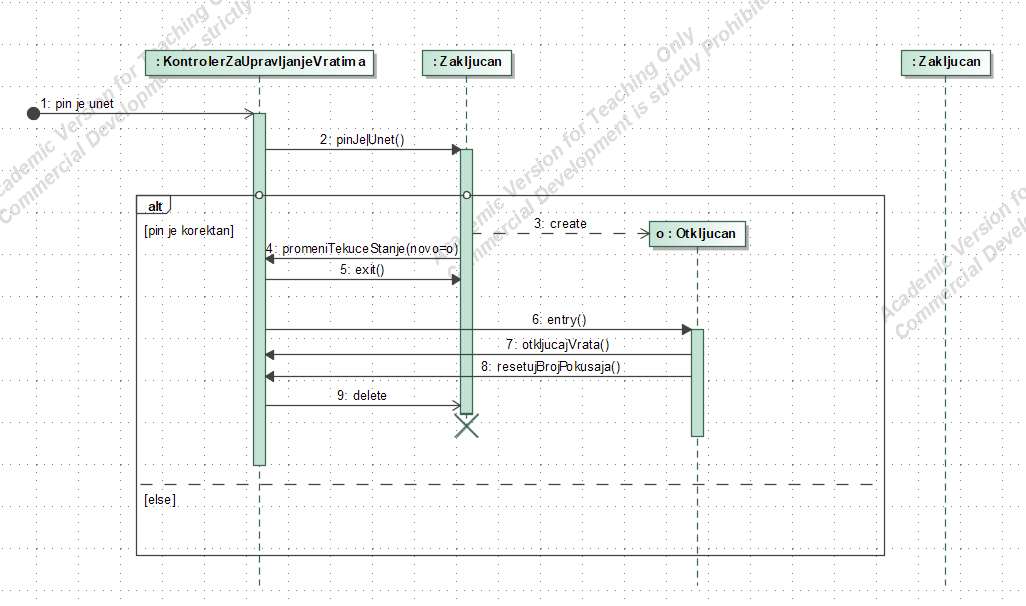
**8. zadatak – NAUČNI ČASOPIS *(konceptualni dijagram klasa, dijagram sekvence za prvu registraciju i slanje rada)***

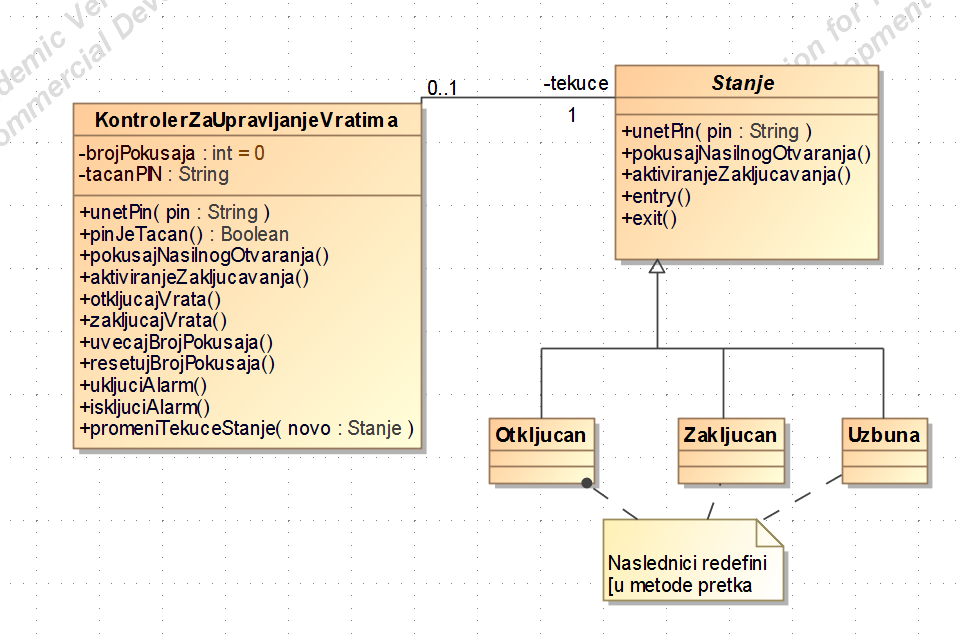
Modelovati aplikaciju za podršku uređivanja naučnih časopisa. Svaki časopis ima urednika (ime, prezime, e-mail) i recenzente (ime, prezime, e-mail, spisak naučnih oblasti za koje je recenzent specijalizovan). Naučna oblast ima sledeće podatke: šifra i naziv. Urednik i recenzenti su korisnici aplikacije. Korisnici aplikacije su i autori radova (ime, prezime, e-mail). Svaki korisnik ima korisničko ime i lozinku. Svaki rad ima podatke: naziv, apstrakt (sažetak), datum kada je poslat. Jedan rad može da ima više autora. Jedan autor može da ima više radova. Rad može biti: primljen, u fazi recenzije, prihvaćen ili odbijen. Recenzija rada ima sledeće podatke: mišljenje o radu i preporuku recenzenta da li rad treba da se prihvati ili odbije.



**9. zadatak – ALARM *(dijagram stanja, state šablon, dijagram sekvence za oktljučavanje alarma)***

Modelovati ponašanje zaštitnog alarma na vratima kuće. Alarm se ponaša na sledeći način. Ako je u stanju “zaključan”, a neko želi da uđe, potrebno je ukucati šifru. Ako je šifra korektna, vrata se otključavaju, a alarm prelazi u stanje “otključan“. Ako šifra nije korektna, moguće je pokušati njen unos još 2 puta. Ako je sva tri puta šifra pogrešna, alarm prelazi u stanje „uzbuna“. Ukucavanjem korektne šifre alarm prelazi iz stanja uzbune u stanje „otključan“. Ako detektuje pokušaj nasilnog ulaska, alarm takođe prelazi u stanje uzbune. Iz stanja „otključan“ u stanje „zaključan“ alarm prelazi pritiskom na odgovarajuće  dugme.





**10. zadatak – BIBLIOTEKA *(konceptualni dijagram klasa, dijagram sekvence za vraćanje knjige)***

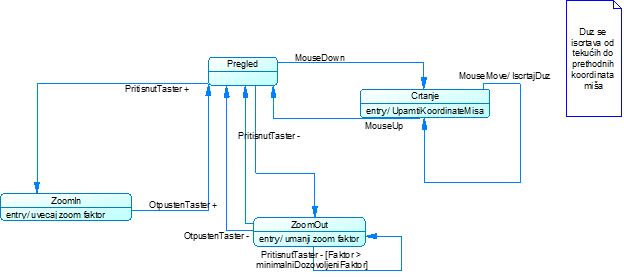
Projektovati podsistem cirkulacije u okviru informacionog sistema biblioteke koji treba da obezbedi: evidenciju članova (registracija novog člana, produžavanje članarine), evidenciju naslova i primeraka knjiga, zaduživanje knjiga (pozajmica primerka knjige članu biblioteke), vraćanje knjiga, rezervaciju knjiga, štampanje opomena o prekoračenju roka za vraćanje knjige i slanje na kućnu adresu, naplatu kazne u slučaju prekoračenja roka za vraćanje ili u slučaju gubljenja/oštećenja knjige, finansijski izveštaj za određeni radni dan (izveštaj koji prikazuje sva naplaćena sredstva u jednom danu (članarine i kazne) i daje njihov zbir), finansijski izveštaj za smenu u okviru određenog radnog dana, izveštaj o 10 najčitanijih naslova u okviru zadatog meseca, izveštaj o knjigama kojima je istekao rok za povratak (naslov knjige, inventarni broj primerka, broj članske karte, prezime, ime i adresa člana).

Prilikom evidencije knjiga, podaci o knjizi (naziv, pisac, izdavač itd) se registruju kao naslovi. Svaki naslov ima jedinstven ključ koji se naziva signatura. Svaki naslov može da ima nijedan, jedan ili više fizičkih primeraka. Ključ primerka je inventarni broj. Prilkom izdavanja knjige i vraćanja knjige, pamti se inventarni broj konkretnog primerka koji je član pozajmio. Iznos kazne u slučaju gubljenja ili oštećenja knjige odgovara ceni knjige u trenutku njene nabavke.

Prilikom produžavanja članarine, iznos članarine se određuje na osnovu kategorije članstva (npr. đak, student, penzioner, zaposleni...). Kategorija članstva, pored iznosa članarine, definiše i koliko dana član može da zadrži knjige, koliko maksimalno knjiga može da drži kod sebe u jednom trenutku i cenu kazne za dan prekoračenja.

**11. zadatak – GRAFIČKI EDITOR *(dijagram prelaza stanja, state šablon)***

Primitivni grafički editor se inicijalno nalazi u stanju za pregled. Pritiskom levog tastera miša na dole, editor ulazi u stanje crtanja. Ostaje u tom stanju sve dok je taster miša pritisnut (iscrtava se kriva linija koja prati kretnju miša). Otpuštanjem tastera miša, editor prelazi u stanje za pregled. Iz stanja za pregled može da pređe u stanje zoom-iranja slike. Stanje zoom-iranja je aktivno sve dok je pritisnut taster na tastaturi: “+” za povećanje ili “-” za smanjenje veličine slike. Iz stanja zoom-iranja se prelazi u stanje za pregled otpuštanjem tastera na tastaturi ili ukoliko zoom faktor zapreti da spadne ispod 10% ili da se poveća iznad 1000% veličine slike.



**12. zadatak – FAKULTET *(konceptualni dijagram klasa, dijagram sekvence za studenta koji bira predmet)***

Projektovati programski paket za podršku upravljanju nastavnim procesom u sklopu fakulteta. Fakultet poseduje departmane a departmani katedre. Nastavnici su raspoređeni u okviru katedri. Nastavnici mogu biti: asistent, saradnik u nastavi, docent, vanredni ili redovni profesor. U okviru departmana, nastava se izvodi na smerovima (npr. automatika, računarske nauke, softversko inženjerstvo i sl). Jedan departman može imati više smerova. Smer propisuje skup nastavnih predmeta koji mu pripadaju. Jedan predmet može biti u okviru jednog smera. Nastavnicima se dodeljuju predmeti koje izvode, pri čemu jedan nastavnik može izvoditi više predmeta i jedan predmet može imati više pridruženih nastavnika. Student bira smer koji će pohađati. U okviru smera, student sluša nastavu iz nastavnih predmeta i izvršava dodatne obaveze (projekti, seminarski, praksa i sl.), pri čemu svaki predmet i dodatna obaveza nosi propisani broj poena. Da bi student stekao diplomu smera koji pohađa, mora osvojiti minimalno definisani broj poena za taj smer. Nastavni predmeti u sklopu smera mogu biti izborni ili obavezni, pri čemu neki predmeti mogu biti slušani samo ako su ispunjeni preduslovi (položeni neki drugi predmeti koji im prethode ili odrađene neke dodatne obaveze). Navedeno definiše nastavnik za svaki svoj predmet. Nastava na izbornom predmetu se može izvoditi jedino ako je predmet odabrao minimalno definisani broj studenata (minimalni broj studenata se propisuje za svaki smer). Potrebno je da student ima uvid u predmete koje može da bira (na osnovu pripadnosti smeru i zadovoljenih uslova) a da nastavnik može da vidi listu studenata koji su izabrali svaki njegov predmet. Takođe, nastavnik treba da ima uvid u spisak svojih predmeta koji nemaju dovoljan broj studenata da bi se izvodili. Najosnovniji podaci koje je potrebno voditi o fakultetu, departmanu, katedri, predmetu i smeru su šifra i naziv, a o studentu i nastavniku su: ime, prezime, datum rođenja i broj telefona. Navedene podatke održava administrator.

**13. zadatak – BANKOMAT *(dijagram stanja, dijagrami sekvence za isplatu novca)***

Jednostavni bankomat je inicijalno u stanju čekanja. Kada korisnik ubaci karticu, bankomat traži da se unese PIN kod. Ako PIN kod nije ispravan, bankomat obaveštava korisnika o tome i ponovo traži da se unese PIN kod. Korisnik ima najviše tri pokušaja. Ako treći put pogreši, bankomat mu uzima karticu i vraća se u stanje čekanja na sledećeg korisnika. Ako je PIN kod ispravan, bankomat prelazi u stanje isplate, gde čeka da korisnik unese iznos koji želi da mu se isplati. Ako je iznos veći od raspoloživog iznosa novca koji ima na računu, bankomat ga obaveštava o tome i vraća se na čekanje novog iznosa za isplatu. Kada korisnik unese iznos za koji ima dovoljno novca na računu, bankomat isplaćuje novac, vraća karticu i prelazi u stanje čekanja na novog korisnika. Pritiskom na dugme za prekid se korisniku vraća kartica i prelazi u stanje čekanja na novog korisnika.

**14. zadatak – GRAĐEVINA *(dijagram slučajeva korišćenja, dijagram klasa, dijagram prelaza stanja i dijagram sekvence za projektovanje kuće)***

Potrebno je realizovati softver za podršku projektovanja gradjevinskih objekata. Kuca je gradevinski objekat koji sadrži niz spratova i jedan krov. Sprat je oblika kvadra opisan dužinom, širinom i visinom. Krov je oblika cetvorostrane piramide opisan dužinom i širinom osnovice i visinom. Sprat i krov mogu se spojiti sa prethodnim spratom. Potrebno je da se može izracunati zapremina i površina osnove za sve gradjevinske objekte i njihove elemente. Kuca se gradi u tri faze: 1. Projektovanje (zadaje se broj spratova), 2. Izgradnja spratova (Dodaju se ranije projektovani spratovi jedan po jedan. Spratovi ne moraju da budu istog izgleda, bitno je samo da imaju istu površinu osnove). 3. Izgradnja krova (dodaje se ranije projektovan krov). Greška je ako se pokuša izgraditi više od predvidenog broja spratova ili ako se pokuša dodati krov pre nego što su svi spratovi izgradeni ili kada vec postoji krov. Potrebno je da softver svaku od opisanih grešaka spreci uz odgovarajuce obaveštenje korisniku. Takode je potrebno da softver vodi korisnika (da mu daje informaciju šta je sledece potrebno da uradi po završetku svake faze). Projektanti projektuju gradjevinske objekte. Postoje projektanti specijalizovani za spratove, za krovove i za kuce (koji sklapaju kuce od unapred isprojektovanih krovova i spratova). Za krovove i spratove postoje tipski parametri koji definišu dimenzije, broj prozora (za spratove i krovove) i broj terasa (za spratova). Potrebno je dozvoliti da svaki od projektanata može da menja te parametre za svoju vrstu gradevinskog objekta.