

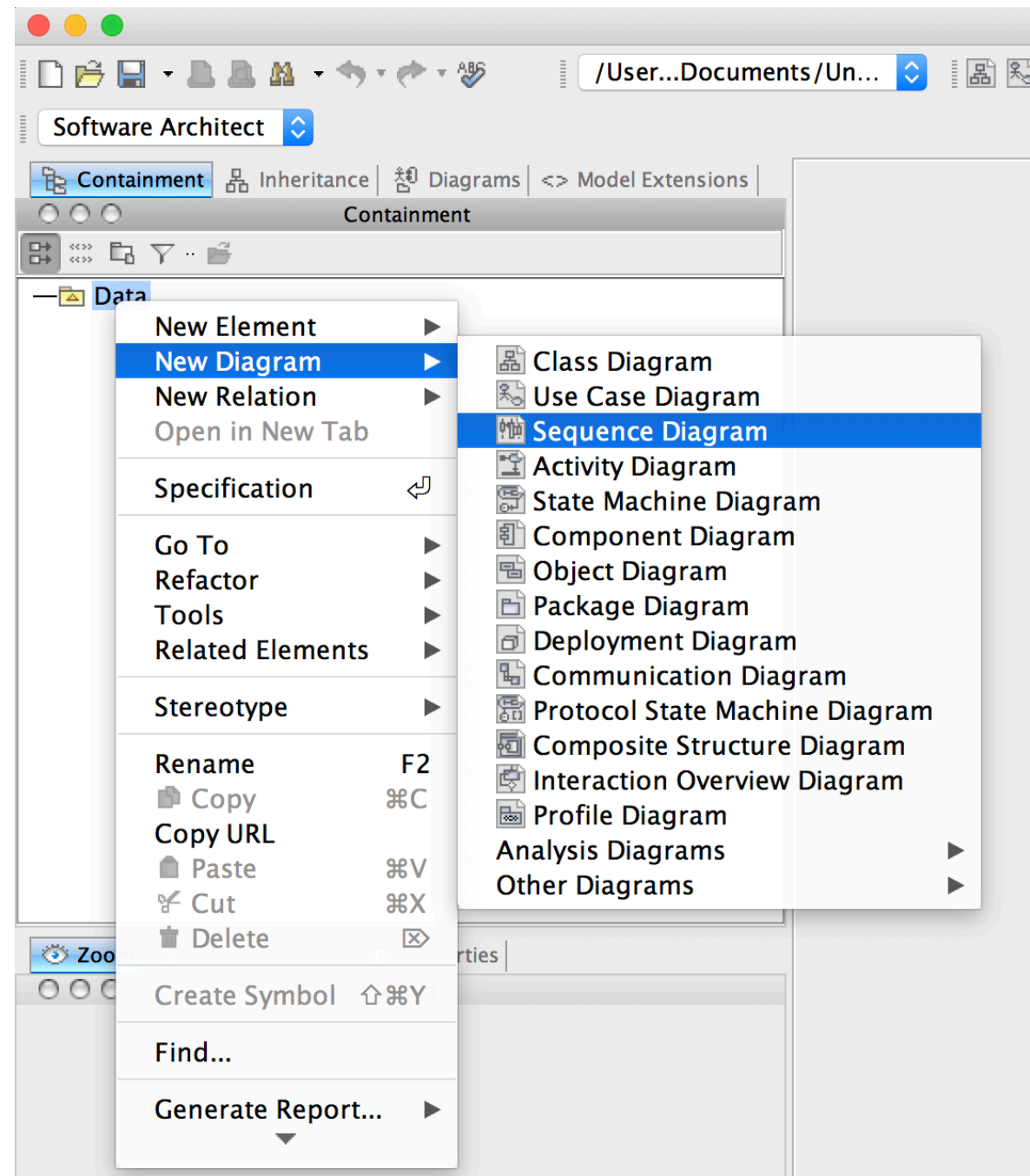
SIMS - VEŽBE 06

DIJAGRAM SEKVENCE

OSNOVNI KONCEPTI

- ▶ Dijagram sekvenci je UML dijagram interakcije
- ▶ Prikazuje hronologiju prenošenja poruka između objekata u sistemu i učesnika. Mogu se koristiti da ilustruju mogući scenario načina upotrebe, izvršenje operacija, ili jednostavno interakciju između klasa sistema
- ▶ Može se koristiti jedan ili više sekvencijalnih dijagrama da bi se odredili načini upotrebe ili da bi se identifikovale sve mogućnosti složenog ponašanja sistema
- ▶ Jedna od glavnih prednosti dijagrama sekvenci predstavlja to što se mogu prikazati najčešće interakcije, a takođe se lako mogu dodati alternativni ili paralelni scenariji korišćenjem interakcionih fragmenata. Zbog toga, u jednom dijagramu sekvence se može opisati više povezanih interakcija
- ▶ [Link](#) ka delu dokumentacije Magic Draw-a koji se tiče dijagrama sekvenci

KREIRANJE NOVOG DIJARAMA SEKVENCE



OSNOVNI ELEMENTI DIJAGRAMA SEKVENCI

▶ Dijagrami sekvenci obično sadrže:

1. Uloge

2. Poruke

▶ Sinhronne poruke

▶ Asinhronne poruke

▶ Poruke za kreiranje objekata

▶ Poruke za oslobađanje projekata

▶ "Izgubljenje" poruke

▶ "Nađene" poruke

3. Fragmenti interakcije

4. Vremenske odrednice



ULOGE

- ▶ Uloge u dijagramu sekvence mogu biti:
 - ▶ Korisnici - učesnici izolovani u okviru dijagrama slučajeva korišćenja
 - ▶ Delovi sistema
 - ▶ Sistem u celini
- ▶ Životnom linijom se naziva vertikalna, obično isprekidana linija koja je povezana sa ulogom i označava vreme u kojem je data uloga aktivna
- ▶ Vrh životne linije definisan je pravougaonikom čiji je naziv predstavljen u formi *roleName* : *Class*, pri čemu je moguće izostaviti jednu od ove dve vrednosti

patient

Tip uloge se najčešće izostavlja u ranim fazama razvoja

Objekat klase ili deo sistema

: Patient

Ukoliko se izostavi naziv uloge, ispred tipa se obavezno navode dve tačke

Neimenovani objekat klase

patient : Patient

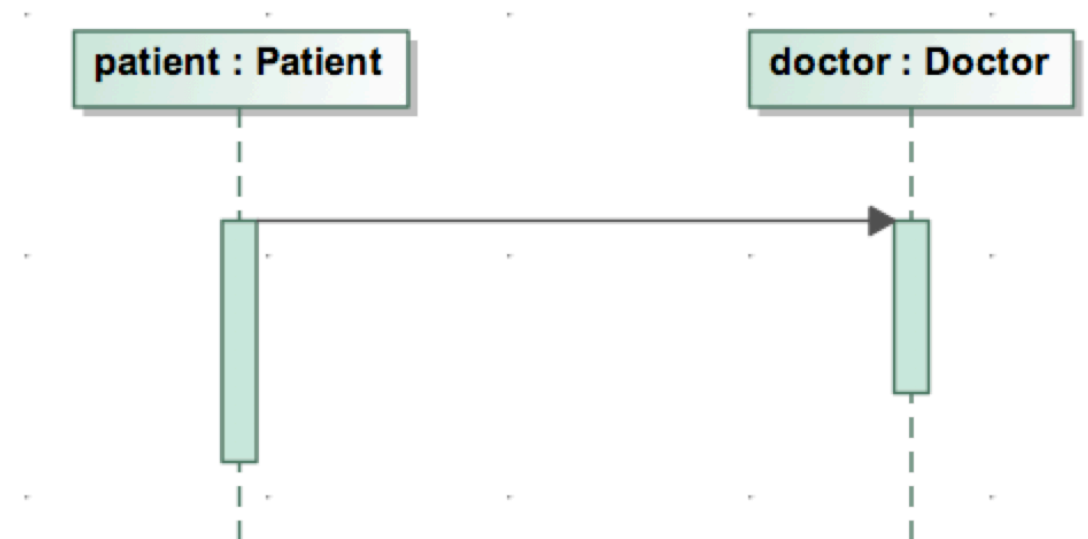
Tip uloge može biti klasa, apstraktna klasa ili interfejs

Imenovani objekat klase



ULOGE

- ▶ Aktivacije predstavljaju opcione simbole koji prikazuju vreme potrebno da se neka akcija izvrši
- ▶ Aktivacija je predstavljena uskim pravougaonikom na liniji života objekta, a predstavlja izvršavanje metode, slanje ili primanje poruke
- ▶ Aktivaciji se mogu pridružiti ili se sa nje mogu ukloniti poruke. Takođe je moguće promeniti veličinu aktivacije, pomerati je ili prouzrokovati da aktivacija preklopi druge aktivacije. Aktivacija se automatski kreira kada se kreira poruka poziva procedure.



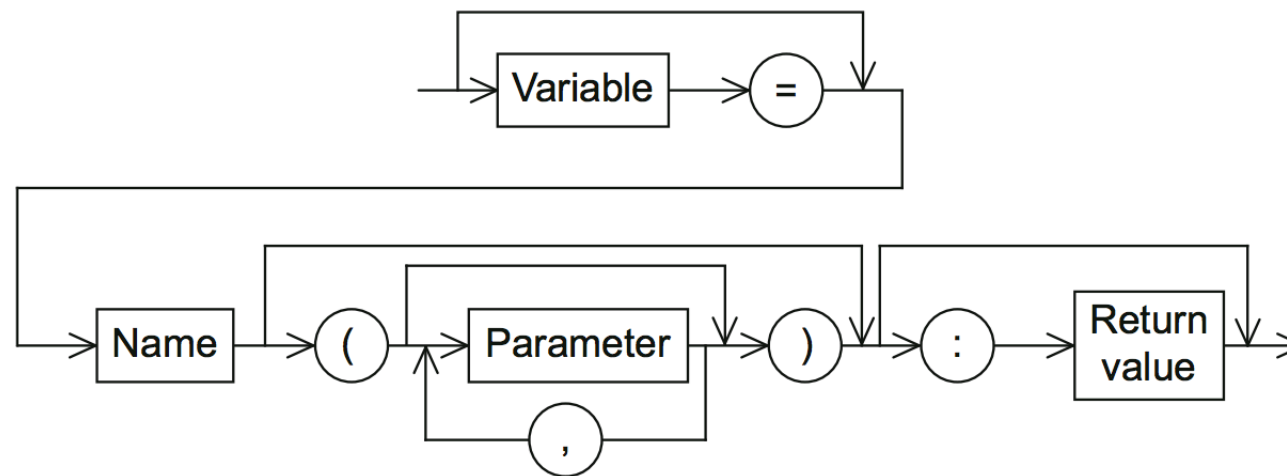
Prikaz simbola za aktivaciju na liniji života

PORUKE

- ▶ Dijagram sekvenci je dvodimenzionalni dijagram
- ▶ Uloge koje su uključene u interakciju predstavljene su na horizontalnoj osi i treba da budu definisani u jasno određenom redosledu
- ▶ Vertikalna osa modeluje hronološki redosled interakcije. Ako je hronološki poredak nije eksplicitno naveden, događaji pri vrhu vertikalne ose se odvijaju pre događaja koji se nalaze niže na vertikali (pod uslovom da se oba događaja odnose na istu liniju života)
- ▶ Dijagram sekvence poruke prikazuju kao horizontalne strelice usmerene između linija života dva učesnika
- ▶ Tip strelice definiše vrstu poruke koja se razmenjuje

PORUKE

- ▶ Prilikom specifikacije poruke naziv je jedino obavezno polje
- ▶ Primer:
 - ▶ `var = m1 : value`
 - ▶ `var` je naziv promenljive u koju se smešta povratna vrednost
 - ▶ `m1` je naziv poruke
 - ▶ `value` je povratna vrednost



Naziv poruke odgovara nazivu metode klase, ukoliko su uloge objekti klase. U suprotnom nazivi poruka mogu biti slobodan tekst

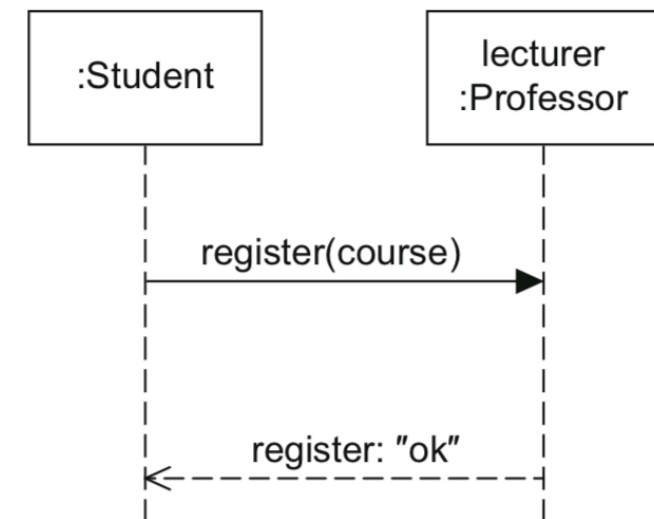
Argumenti poruke mogu biti promenljive kojima je predhodno dodeljena vrednost ili nazivi uloga koji učestvuju u komunikaciji

Format za specifikaciju poruke

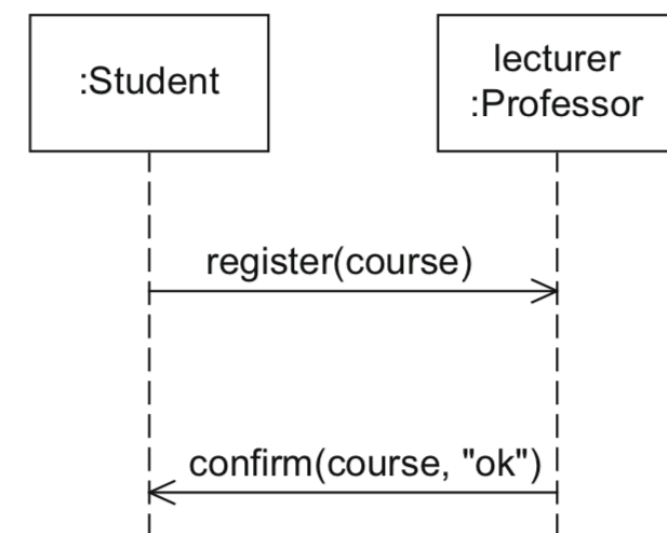


SINHRONE I ASINHRONE PORUKE

- ▶ *Sinhrona poruka* se označavaju punom strelicom sa popunjenim vrhom
 - ▶ Kod sinhronih poruka pošiljac čeka da primalac obradi poruku i ukoliko je potrebno da odgovori. Odgovor može biti predstavljen isprekidanom strelicom sa otvorenim vrhom.
- ▶ Asinhrona poruka su reprezentovane punom strelicom sa otvorenim vrhom
 - ▶ U asinhronoj komunikaciji pošiljalac nastavlja sa izvršavanjem nakon što je poslao poruku (ne čeka da je primalac obradi)



Sinhrona poruka



Asinhrona poruka

SINHRONE I ASINHRONE PORUKE

Message - login (username, password)

Specification of Message properties

Specify properties of the selected Message in the properties specification table. Choose the Expert or All options from the Properties drop-down list to see more properties.

synchCall login (username, passw

Documentation/Hyperlinks

Usage in Diagrams

Arguments

Conveyed Information

Inner Elements

Relations

Tags

Constraints

Traceability

History : synchCall login (username, password) [StudentLogi...

login (username, password)

Properties: All Customize

Message	
Name	login (username, password)
Qualified Name	StudentLogin::SequenceDiagram::Login::Logi...
Signature (operation)	
Owner	Login [StudentLogin::SequenceDiagram::...
Applied Stereotype	
Send Event	++ [StudentLogin::SequenceDiagram::Login...
Receive Event	++ [StudentLogin::SequenceDiagram::Login...
Message Sort	synchCall
Message Kind	asynchCall
Visibility	asynchSignal
Active Hyperlink	createMessage
Applied Stereotype Instance	deleteMessage
Owned Comment	reply
Owned Element	
Name Expression	
Client Dependency	
Supplier Dependency	
Namespace	Login [StudentLogin::SequenceDiagram::...

Message Sort

The sort of communication reflected by the Message

Type here to filter properties

Close

Back

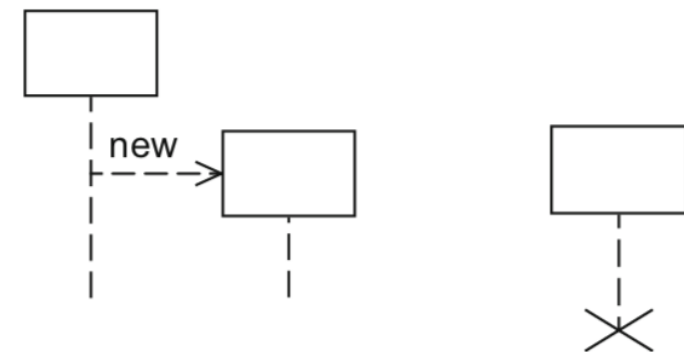
Forward

Help

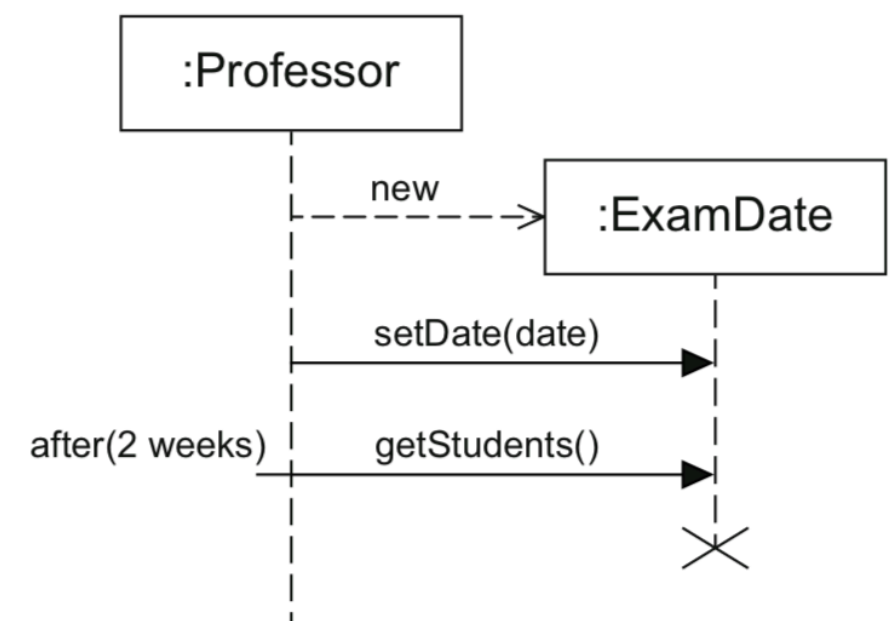


PORUKA ZA KREIRANJE I OSLOBAĐANJE OBJEKATA

- ▶ *Poruka za kreiranje objekata se označava isprekidanom strelicom sa otvorenim vrhom na čijem kraju se nalazi pravougaonik koji reprezentuje početak linije života novokreiranog objekta*
 - ▶ Novokreirani objekat se uvek crta u nastavku poruke, a ne na vrhu dijagrama
- ▶ *Poruka za oslobađanje objekata se označava isprekidanom strelicom sa otvorenim vrhom na čijem kraju se nalazi znak **X***
 - ▶ Prilikom prijema ove poruke vrši se oslobađanje tj. Dealokacija

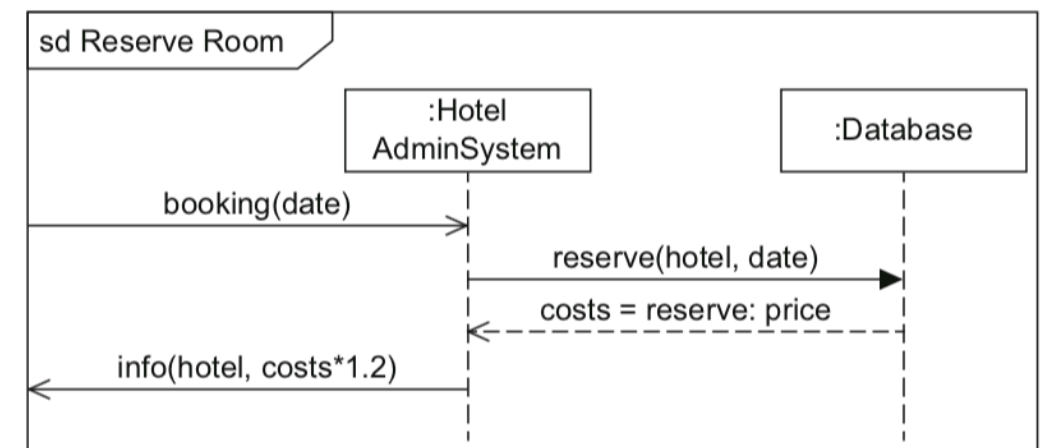
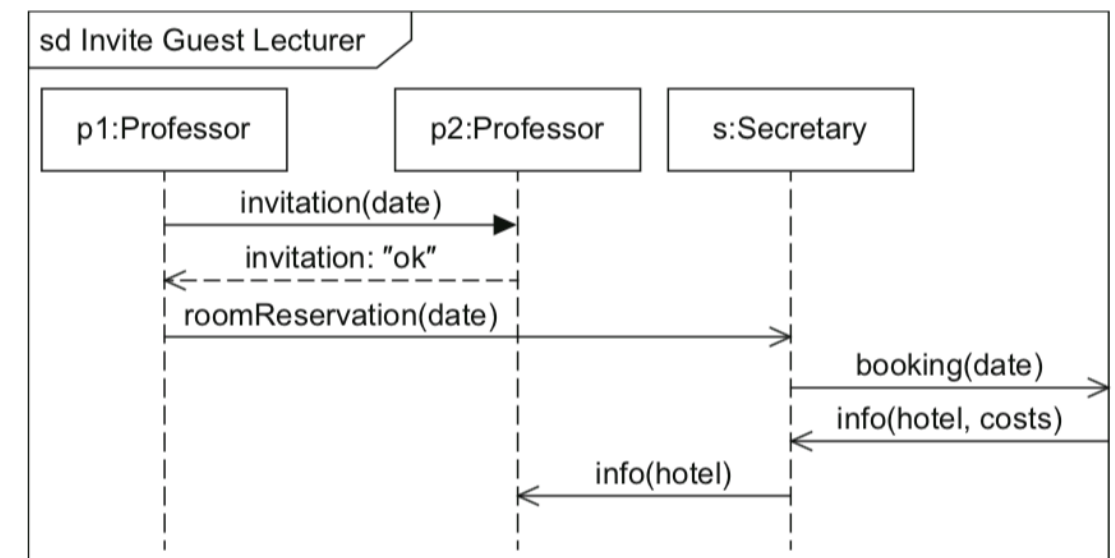


Simboli za kreiranje i oslobađanje objekata



GRANICE INTERAKCIJE I SPOLJNE PORUKE

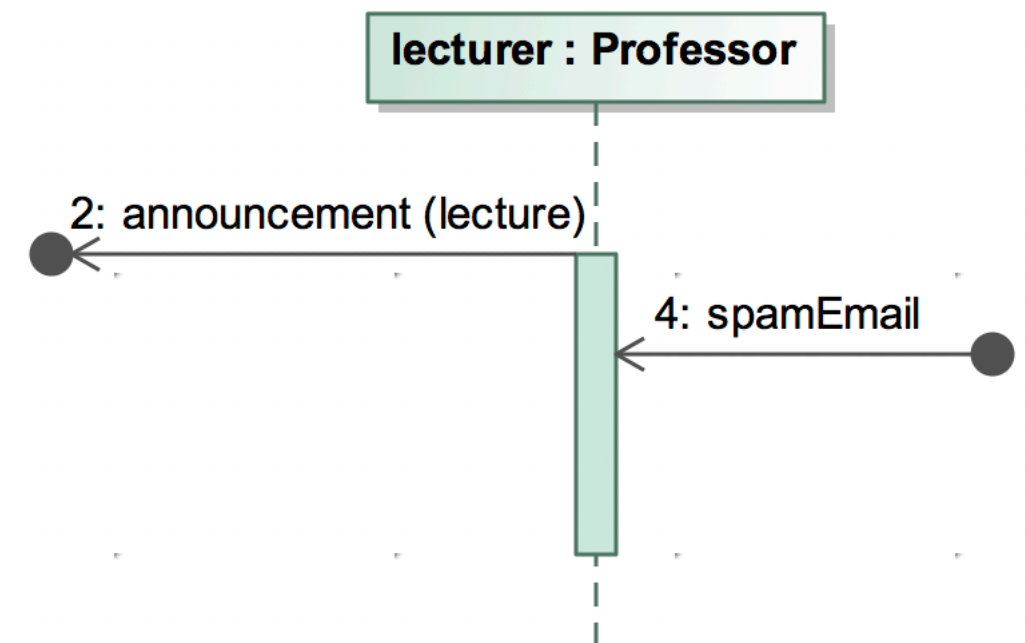
- Po pravilu, poruke ne bi smele da izlaze van okvira dijagrama sekvenci, međutim, ponekad želimo da prikazemo slanje i primanje poruka koje dolaze spolja, tj. van granica našeg sistema





“IZGUBLJENE” I “NAĐENE” PORUKE

- ▶ Ukoliko poruka nema eksplicitno definisanog pošiljaoca, može se modelovati kao *nađena poruka*
 - ▶ U tom slučaju se početak poruke modeluje crnim kružićem
- ▶ Poruku modelujemo kao *izgubljenu* ukoliko nema je primalac nepoznat ili nerelevantan
 - ▶ Primaoca modelujemo crnim kružićem

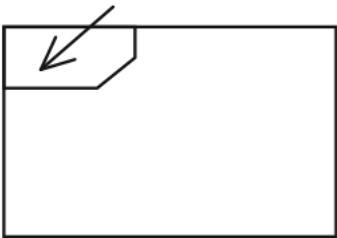


Modelovanje izgubljenih i nađenih poruka

FRAGMENTI INTERAKCIJE

- ▶ Fragmenti interakcije imaju ulogu iskaza za kontrolu toka u programskim jezicima
- ▶ UML podržava 12 tipova fragmeenata interakcije, a neki od najčešće korišćenih su:
 1. Loop - petlja
 2. Opt - opcioni fragment
 3. Break - fragment za prekid
 4. Alt - alternativni fragment
 5. Ref - referencirajući fragment
 6. Par - fragment za paralelno izvršavanje
- ▶ Fragmenti se modeluju kao pravougaonici u čijem gornjem levom uglu se nalazi petougao u kojem je upisan tip fragmenta

Operator



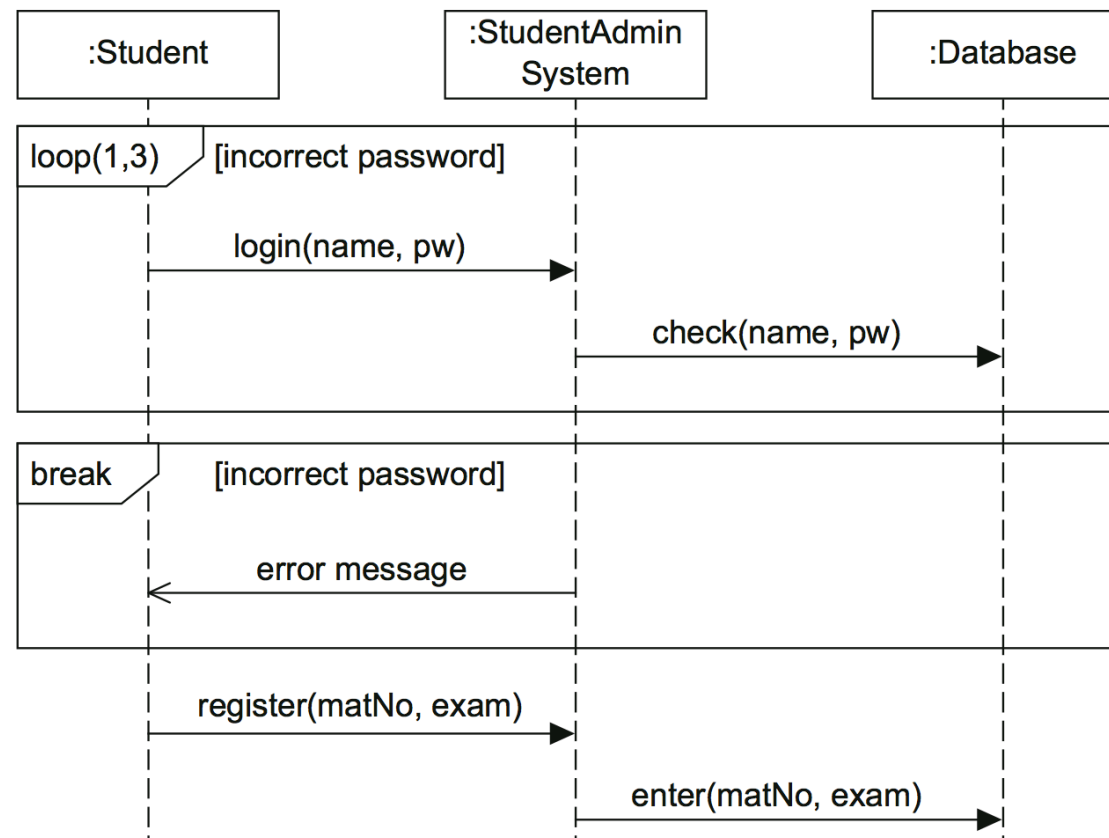
	Operator	Purpose
Branches and loops	alt	Alternative interaction
	opt	Optional interaction
	loop	Iterative interaction
	break	Exception interaction
Concurrency and order	seq	Weak order
	strict	Strict order
	par	Concurrent interaction
	critical	Atomic interaction
Filters and assertions	ignore	Irrelevant interaction parts
	consider	Relevant interaction parts
	assert	Asserted interaction
	neg	Invalid interaction

12 tipova fragmenata interakcije

PETLJA

- ▶ *Loop fragmentom* modelujemo sekvencijalno izvršavanje
- ▶ Unutar zagrada potrebno je navesti broj za izvršavanje petlje
- ▶ Unutar uglastih zagrada navodi se uslov u obliku slobodnog teksta, programskog ili pseudo-jezika

Studentu su omogućena 3 pokušaja prijavljivanja na sistem



Ukoliko je neispravna lozinka uneta 3 puta, uslov [incorrect password] biva **true**, break fragment će biti izvršen i sistem šalje poruku studentu poruku o grešci. Ostatak dijagrama će u tom slučaju biti prekočen

PREKIDNI FRAGMENT

- ▶ *Break fragmentom* modelujemo izlazak iz petlje ili fragmenta koji ga okružuje
- ▶ Ukoliko je uslov, naveden između uglastih zagrada, zadovoljen, izvršavaju se akciju unutar fragmenta, a ostatak dijagrama se preskače

OPCIONI FRAGMENT

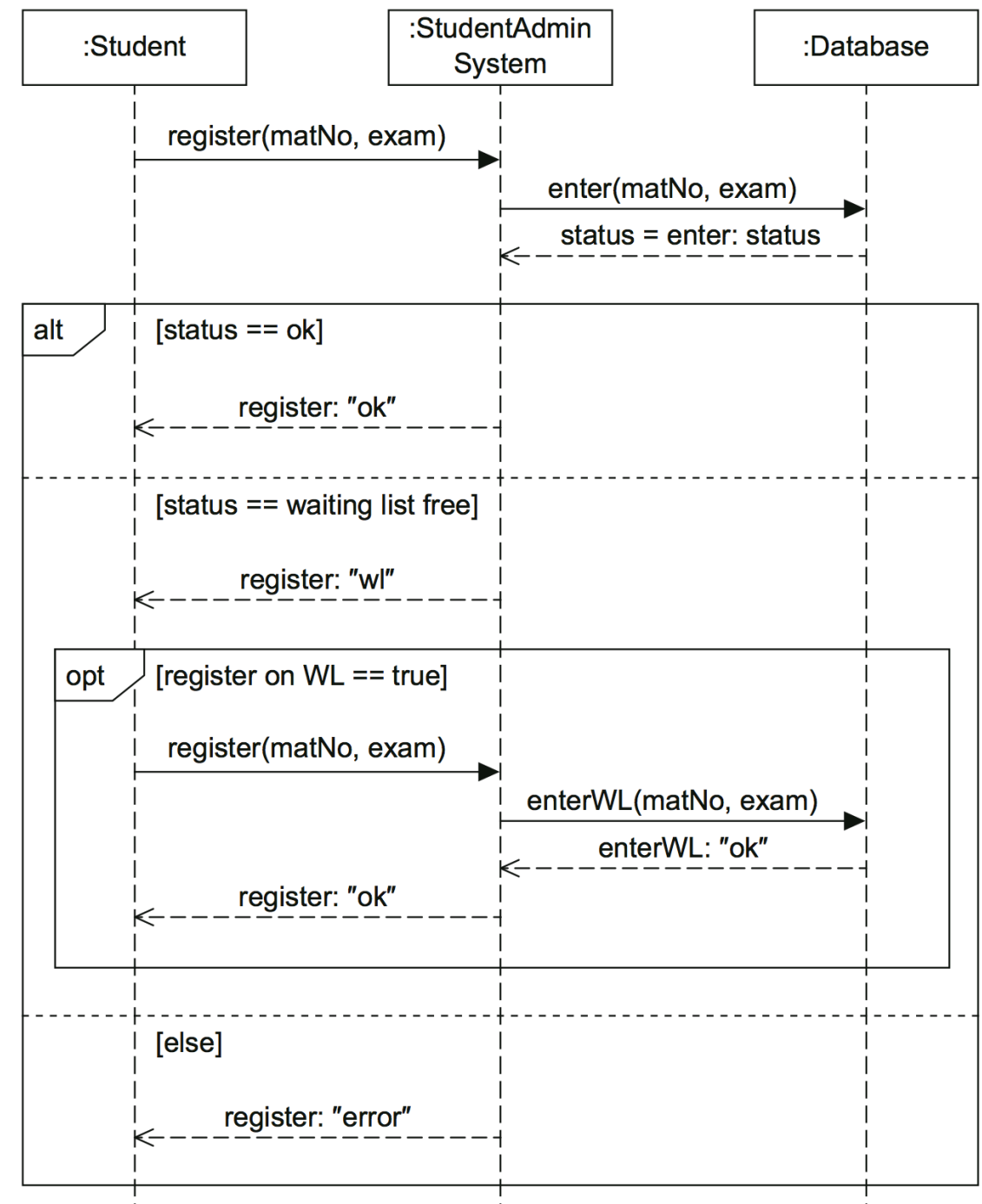
- ▶ *Opt fragment* se koristi za modelovanje uslovnog izvršavanja poruka koje se nalaze unutar njega
- ▶ Opcioni fragment odgovara if iskazu, kod kog nije definisana else grana

ALTERNATIVNI FRAGMENT

- ▶ *Alt fragmentom* modelujemo uslovno izvršavanje
- ▶ Alternativni fragment odgovara if iskazu, koji ima pridodate else if grane. Svaki uslov je predstavljen jednim operandom, pri čemu samo jedan može biti zadovoljen

Ukoliko student želi da se prijavi na ispit moguća su 3 scenarija:

1. Ostalo je mesta i student se prijavljuje
2. Ima mesta na listu čekanja, u tom slučaju odlučuje da li želi da se prijavi na listu čekanja
3. Nema mesta ni na listi čekanja, student dobija poruku obaveštenja

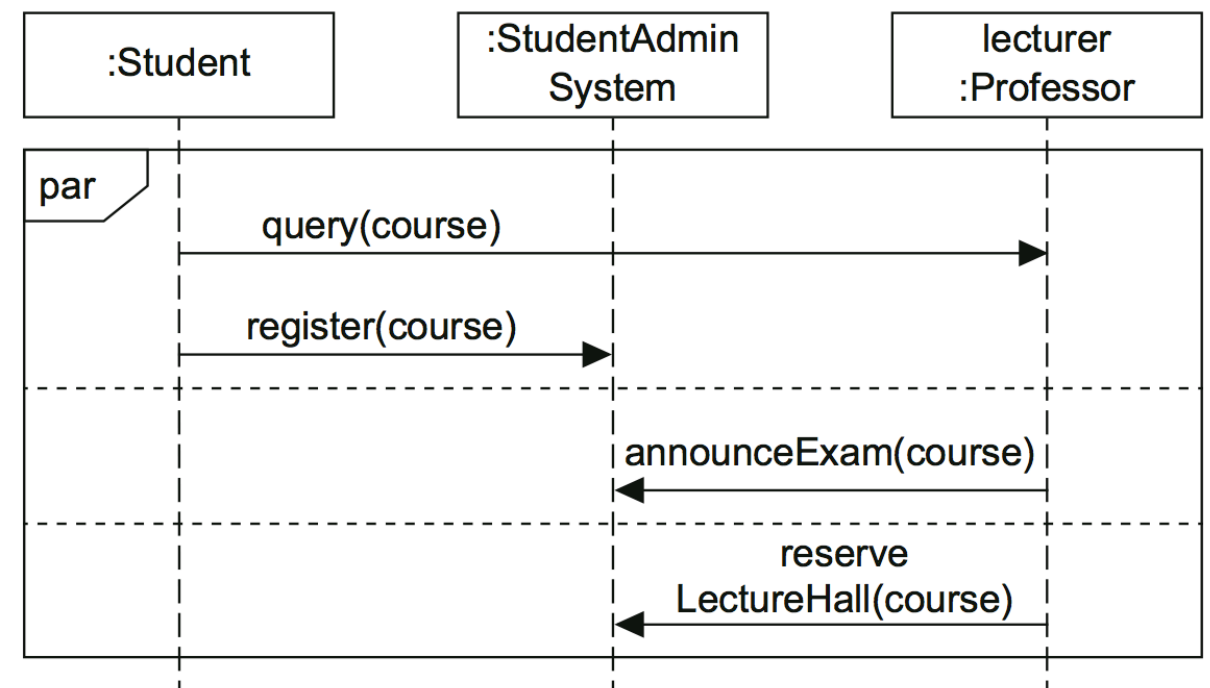


PARALELNI FRAGMENT

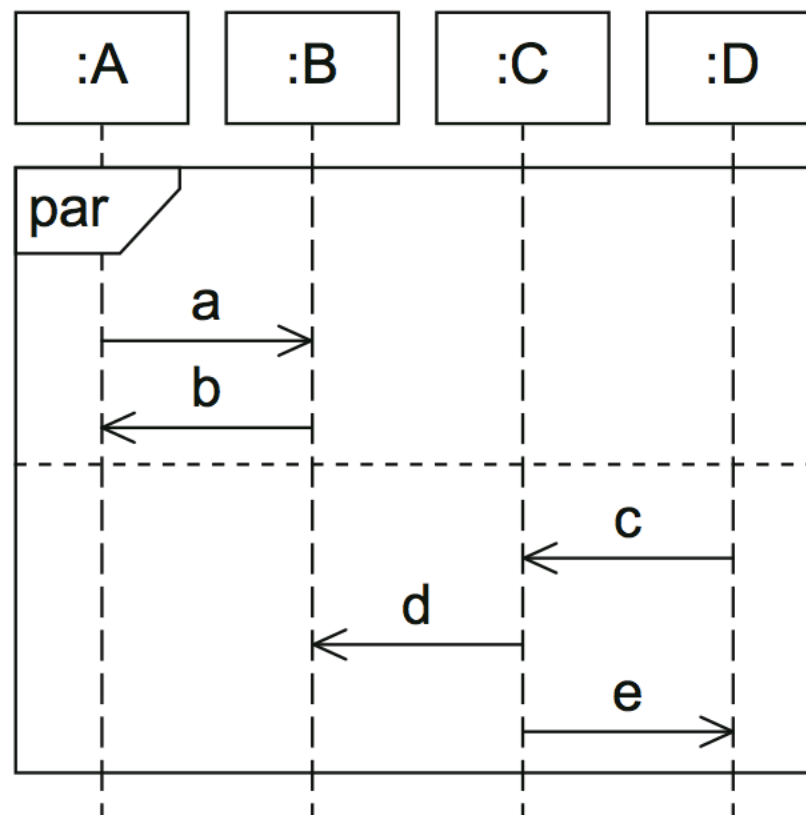
- ▶ *Par fragment* se koristi za modelovanje paralelnog izvršavanja poruka koje se nalaze unutar njega
- ▶ Paralelni fragment omogućava da zanemarimo bilo kakvu vremensku zavisnost između poruka u različitim operandima, dok unutar jedno operandu redosled ostaje ispunjen
- ▶ Paralelni fragment ima najmanje dva operandu

Na početku svakog kursa, profesor mora da obavi nekoliko aktivnosti, kao što su definisanje termina ispita ili rezervisanje sala za predavanja. Paralelni fragment je iskorišćen kako bi naznačio da nije važan redosled u kom će ove aktivnosti biti izvršene.

VAŽNO: redosled unutar operandu mora ostati ispunjen, što znači da će student uvek prvo morati poslati upit za kurs, a tek nakon toga se i prijaviti na isti



PARALELNI FRAGMENT



Traces:

T01: $a \rightarrow b \rightarrow c \rightarrow d \rightarrow e$

T02: $a \rightarrow c \rightarrow b \rightarrow d \rightarrow e$

T03: $a \rightarrow c \rightarrow d \rightarrow b \rightarrow e$

T04: $a \rightarrow c \rightarrow d \rightarrow e \rightarrow b$

T05: $c \rightarrow a \rightarrow b \rightarrow d \rightarrow e$

T06: $c \rightarrow a \rightarrow d \rightarrow b \rightarrow e$

T07: $c \rightarrow a \rightarrow d \rightarrow e \rightarrow b$

T08: $c \rightarrow d \rightarrow a \rightarrow b \rightarrow e$

T09: $c \rightarrow d \rightarrow a \rightarrow e \rightarrow b$

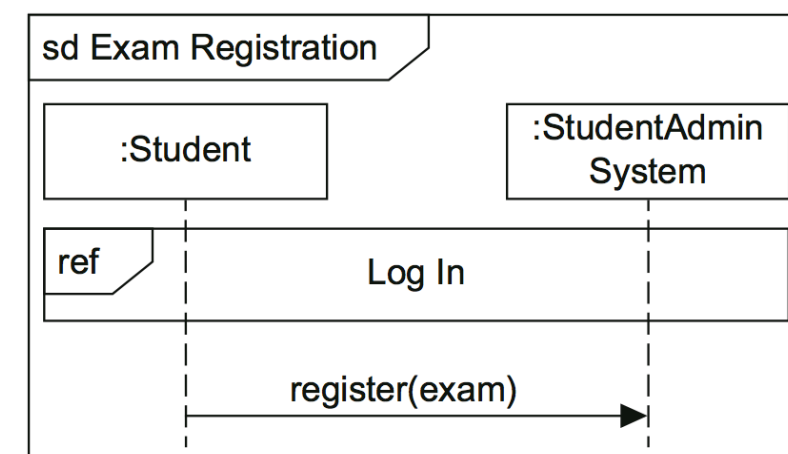
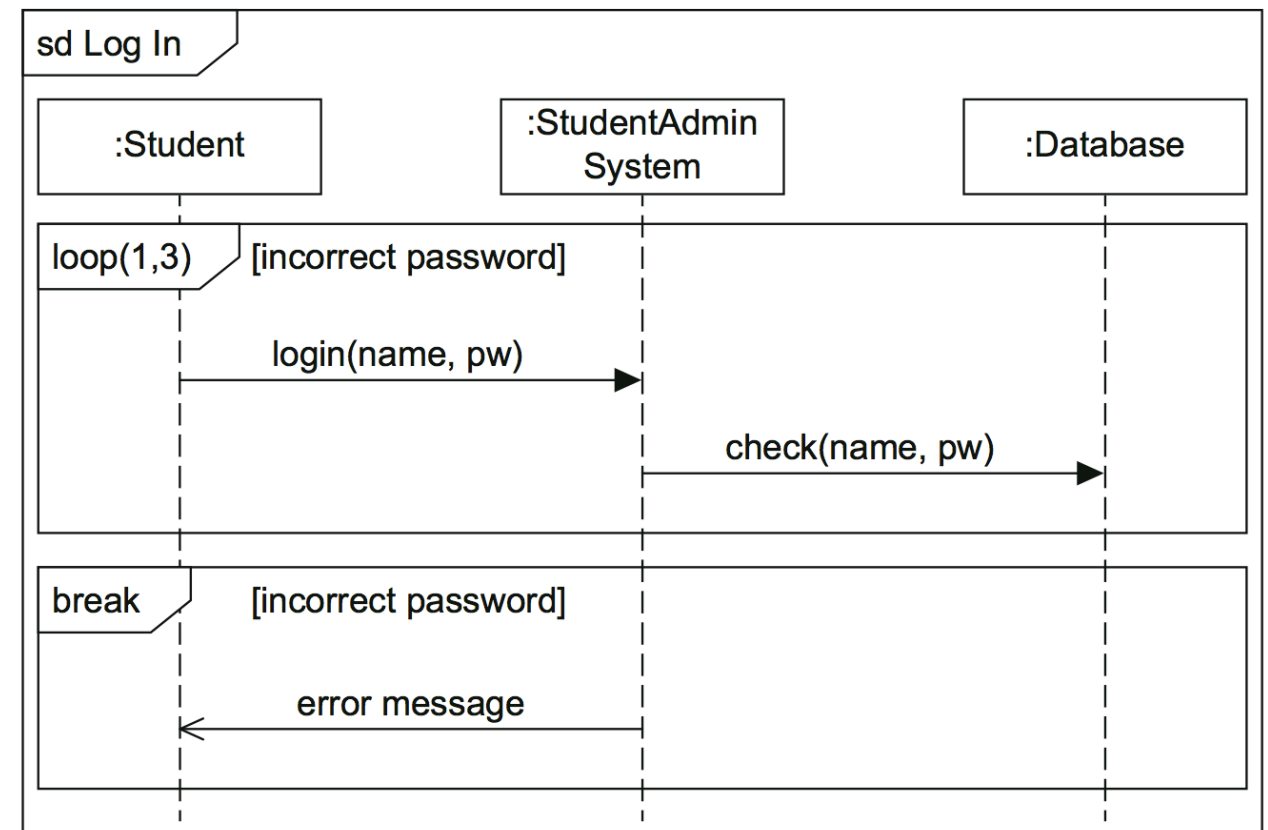
T10: $c \rightarrow d \rightarrow e \rightarrow a \rightarrow b$

Mogući redosledi izvršavanja poruka



REFERENCIRAJUĆI FRAGMENT

- ▶ Ref *fragment* se koristi kako bismo povezali dva ili više dijagrama sekvenci
- ▶ Na ovaj način možemo da ponovo iskoristimo interakcije koje su već modelovane ili da razvijemo kompleksne dijagrame u više jednostavnijih



REFERENCIRAJUĆI FRAGMENT

The screenshot displays a UML modeling environment with a sequence diagram titled 'ExamRegistration' containing a 'ref' fragment. The 'Interaction Use' property specification dialog is open, showing the 'Refers To' property set to '<UNSPECIFIED>'. A search dropdown is open, listing four matches: '<UNSPECIFIED>', 'ExamRegistration [StudentLogin::SequenceDiagram::ExamRegis...', 'Login [StudentLogin::SequenceDiagram::Login]', and 'NewRegistration [StudentAdministrationSystem::ClassDiagram::NewRegistration]'. The 'Login' option is selected. The dialog also shows a list of properties on the left and a 'Refers To' description at the bottom.

Specification of Interaction Use properties
Specify properties of the selected Interaction Use in the properties specification table. Choose the Expert or All options from the Properties drop-down list to see more properties.

History: [StudentLogin::SequenceDiagram::ExamRegistration...]

Properties: All Customize

Interaction Use	
Name	
Qualified Name	StudentLogin::SequenceDiagram::ExamRegis...
Owner	ExamRegistration [StudentLogin::Sequen...
Applied Stereotype	
Refers To	<UNSPECIFIED>
Active Hyperlink	
Applied Stereotype Instance	
Owned Comment	
Owned Element	
Name Expression	
Client Dependency	
Supplier Dependency	
Namespace	
Visibility	
Enclosing Interaction	
General Ordering	
Argument	
Covered	StudentAdministrationSystem::ClassDiagram

4 matches found

- <UNSPECIFIED>
- ExamRegistration [StudentLogin::Se...
- Login [StudentLogin::SequenceDiagram::Login]**
- NewRegistration [StudentAdministrationSystem::ClassDiagram::NewRegistration]
- OrderSystem [OrderSystem::SequenceDiagram::OrderSystem]

Refers To
Refers to the Interaction that defines its meaning

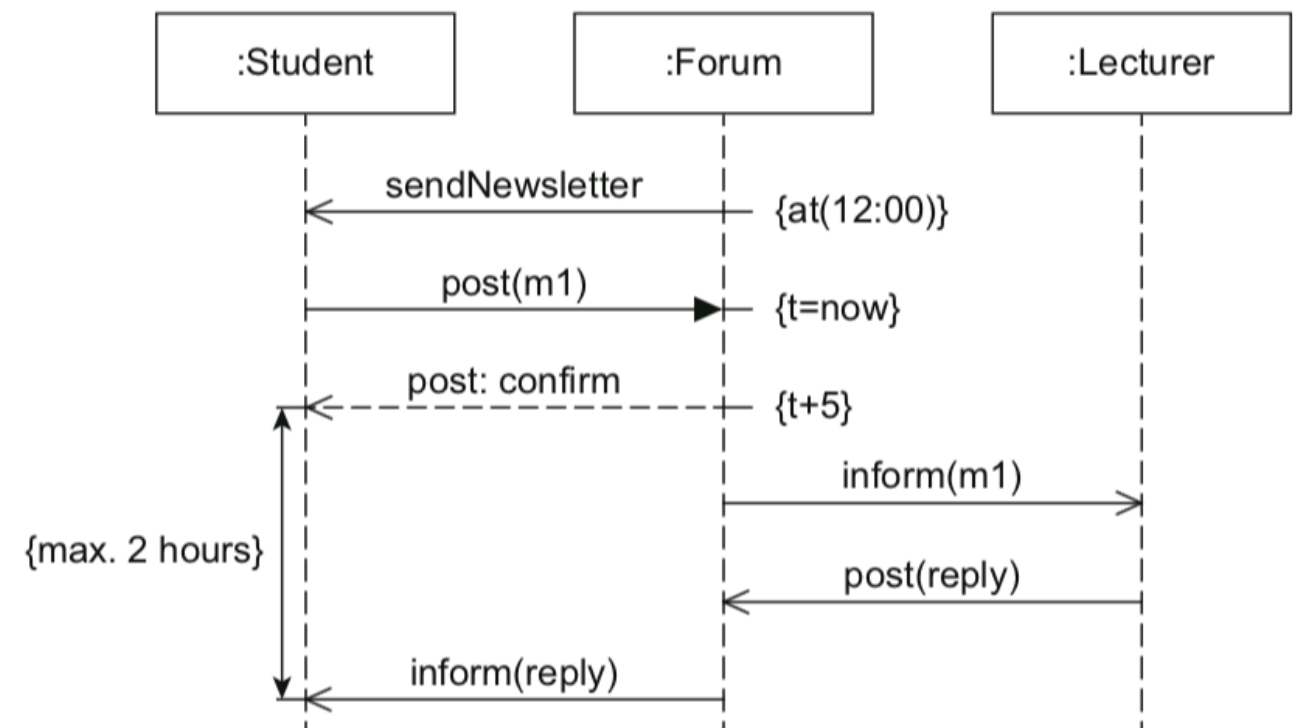
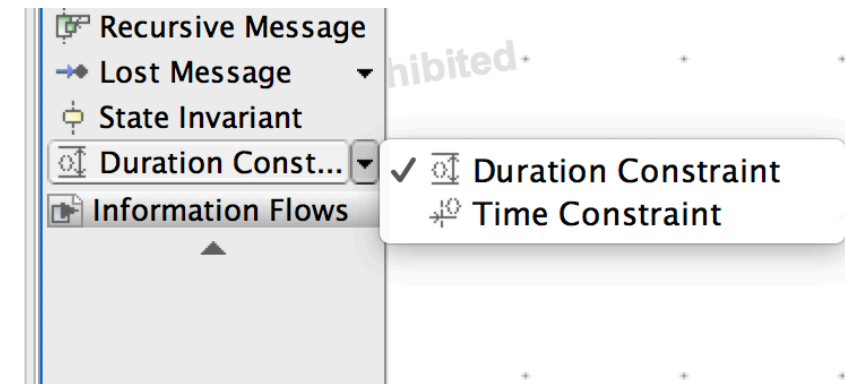
Type here to filter properties

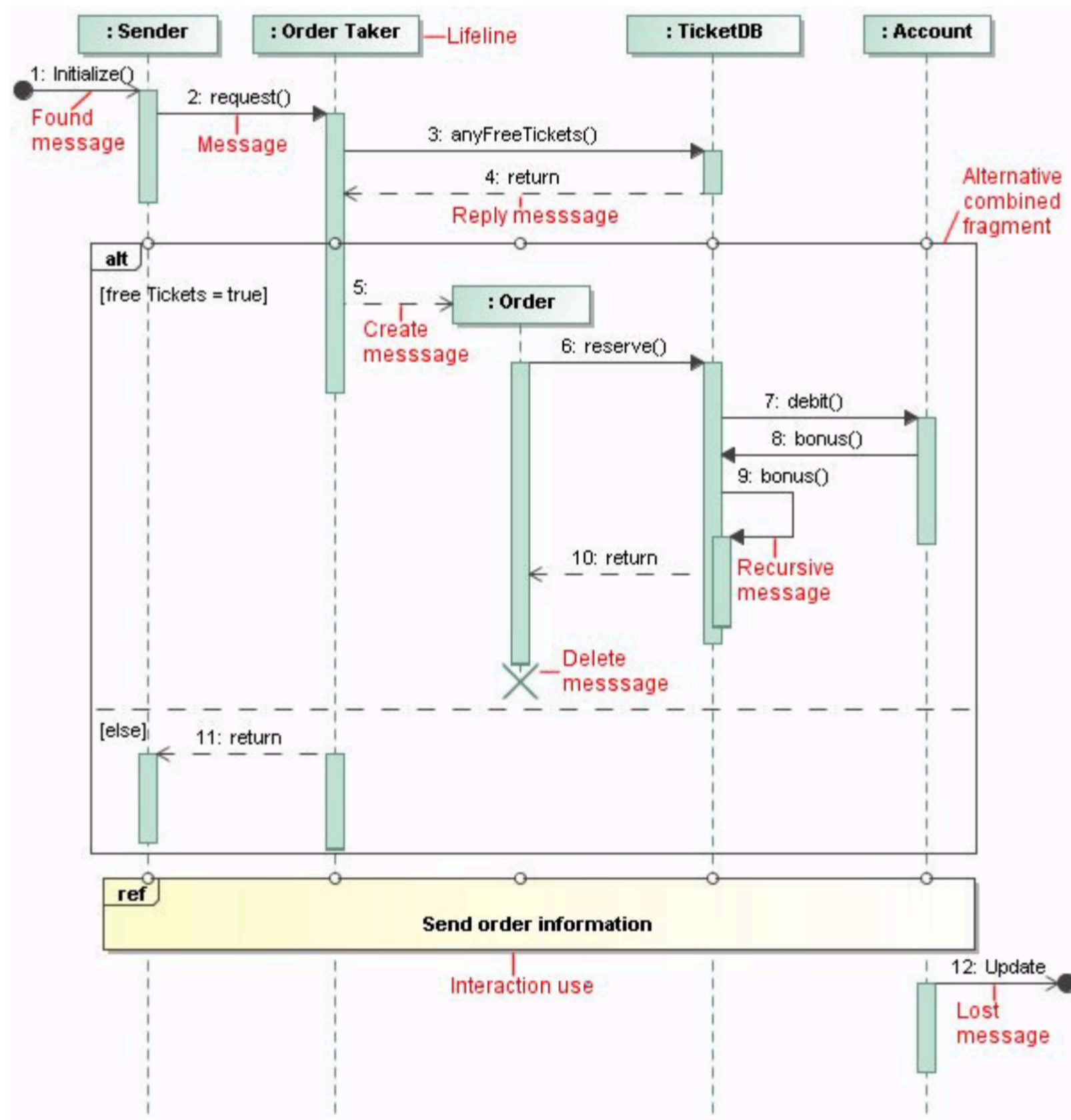
Close Back Forward Help



VREMENSKE ODREDNICE

- ▶ Vremenskim odrednicama iskazujemo:
 - ▶ Vreme u kom će doći do aktiviranja neke poruke
 - ▶ Vreme između dve poruke
- ▶ Vreme se navodi unutar vitičastih zagrada
- ▶ Moguće je specificirati:
 - ▶ Apsolutno vreme { at (12:00) }
 - ▶ Relativno vreme { after (5sek) }
 - ▶ Vremenski interval { 12:00 .. 13:00 }
 - ▶ Trenutno vreme { t = now }
 - ▶ Timing marker - horizontalna linija kojom definišemo vreme između dve poruke





ZADATAK 1

- ▶ Modelovati dijagram sekvence za scenario zasnivanja radnog odnosa.
 - Referent kadrovske službe obavlja sledeće poslove:
 - Unos i izmenu podataka o novim zaposlenima (ime, prezime, adresa, pol, datum rođenja)
 - Raspored zaposlenog na radno mesto (naziv radnog mesta i broj bodova). Prilikom zasnivanja radnog odnosa, referent prvo unosi podatke o novom radniku i zatim ga raspoređuje na neko od postojećih radnih mesta.
 - Premeštanje zaposlenog na novo radno mesto.
 - Prekid radnog odnosa iz bilo kog razloga (odlazak u penziju, sporazumni raskid, otkaz).
 - Prikaz izveštaja o aktivnim radnicima.
 - Prikaz izveštaja o penzionerima.
 - Prikaz izveštaja o svim radnim mestima radnika – tekućem i nekadašnjim, sa datumom raspoređivanja na dato radno mesto.
 - Administrator obavlja kreiranje korisničkih naloga za zaposlene. Promenu svoje lozinke mogu da obavljaju svi koji imaju korisnički nalog.
 - Na početku svakog meseca aplikacija treba da proveri sve aktivne zaposlene i šefu kadrovske evidencije pošalje spisak onih koji u narednih mesec dana pune 65 godina (spisak kandidata za penziju).
 - Šef kadrovske službe treba da ima mogućnost da radi sve poslove kao i referent, a dodatno i da održava podatke o raspoloživim radnim mestima.

ZADATAK 2

- ▶ Modelovati dijagram sekvence za Restoran
 - ▶ Scenario: predlaganje novog jela od strane kuvara, kao i proces selekcije istog od strane menadžera (isti scenario je bio opisan i dijagramom aktivnosti)