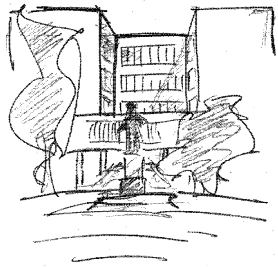


[P271]  
**Информациони системи**

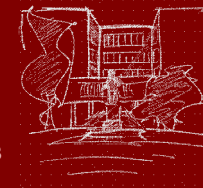
3



**Саша Малков**  
Универзитет у Београду  
Математички факултет  
2023/2024

[P271]  
**Информациони системи**

Саша Малков



Тема 4

**Технике описивања процеса  
(наставак)**

[P271] Информациони системи – Саша Малков – 2023/24 – час 3

1

Технике описивања процеса / Описивање структуре процеса

**Технике за описивање структуре процеса**



- Баве се структуром процеса
  - Декомпозицијом процеса на једноставније потпроцесе
  - Анализом саставних делова процеса
    - обухваћених функција
    - повезаних података
    - ...
  - Временским међузависностима
- Пружају информације за планирање и пројектовање

Универзитет у Београду – Математички факултет

[P271] Информациони системи – Саша Малков – 2023/24 – час 3

2

Технике описивања процеса / Описивање структуре процеса

**Дијаграм тока података – ДТП**



- Једна од основних техника структурних методологија
  - моделирање пословне структуре окружења
  - функционална декомпозиција система
  - модуларно представљање окружења
  - концизно графичко представљање
    - разумљиво свим субјектима
- Употребљава се и у другим методологијама

Универзитет у Београду – Математички факултет

[P271] Информациони системи – Саша Малков – 2023/24 – час 3

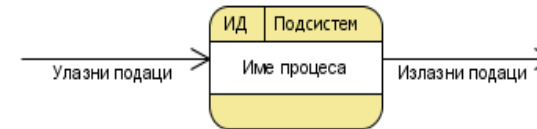
3

## Дијаграм тока података – ДТП (2)

- Основни елементи:
  - ток података
  - процеси
  - складишта података
  - извори и исходи
- Графички изглед се разликује, зависно од извора и методологије

## Дијаграм тока података – ДТП (3)

- Ток података се представља стрелицама
- Процеси се представљају као кругови (елипсе) или заобљени правоугаоници



- Сваки процес мора да има бар један улазни и бар један излазни ток
- Назив (или ознака) подсистема у коме се процес одвија може бити записан и у дну, а не изнад

## Дијаграм тока података – ДТП (4)

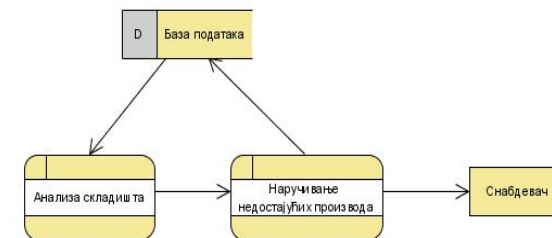
- Складишта података се представљају правоугаонцима отвореним на десној страни
- уобичајено је да се врста складишта означава у квадрату на левој страни правоугаоника



- Складишта могу бити
  - датотеке
  - база података
  - физичка складишта
    - архива докумената
    - филмова
  - и друго

## Дијаграм тока података – ДТП (5)

- Извори и исходи (спољни ентитети и субјекти) се представљају као квадрати или правоугаоници



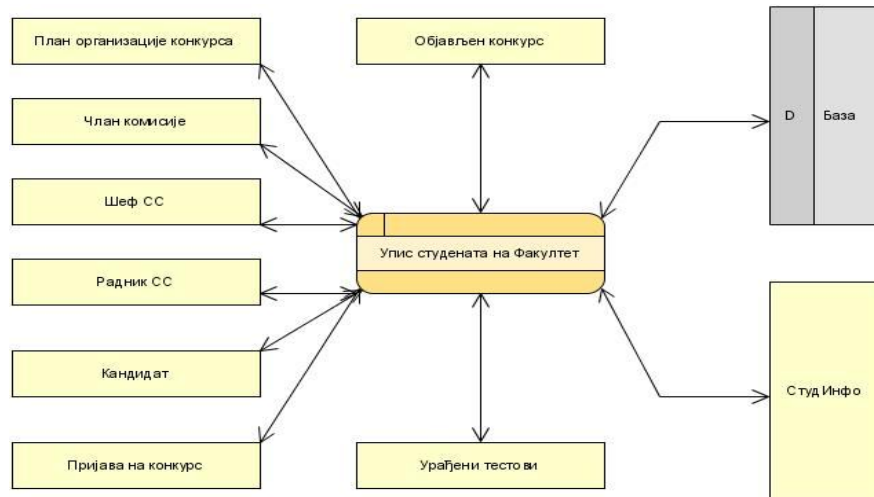
## Дијаграм тока података – ДТП (6)

- Правила обликовања (Mason, 1994)
  - детаљно упознавање са процесом
  - препознавање извора и исхода
  - препознавање складишта података
  - препознавање процеса

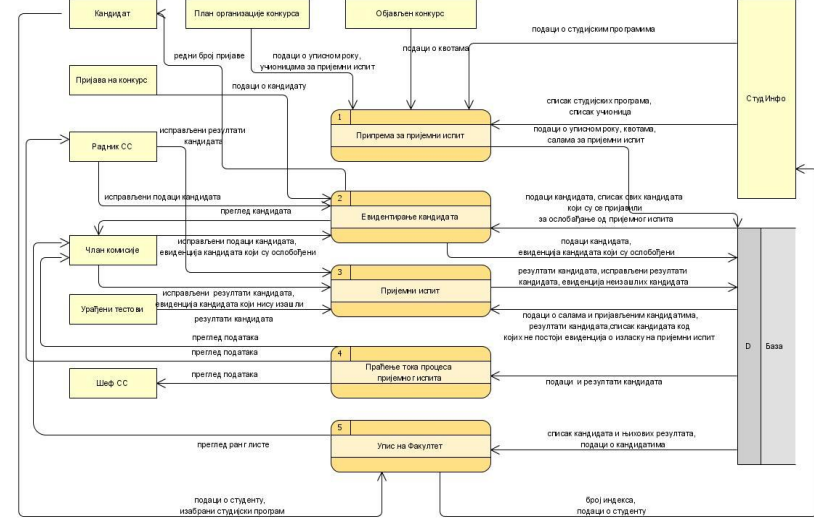
## Дијаграм тока података – ДТП (7)

- Нивои дијаграма тока података
  - дијаграм контекста – читав систем (подсистем) је један процес
    - циљ је да се сагледа положај система у окружењу у коме постоји и функционише
      - субјекти (корисници)
      - интерфејси
      - други системи
  - ниво 0 – разликују се главни процеси у читавом систему (подсистему)
    - препознавање основних компоненти система и њихових интеракција
  - ниво 1 – за сваки процес из ДТП нивоа 0 се прави посебан ДТП
    - даља функционална декомпозиција
  - по потреби се праве и дијаграми нижих нивоа

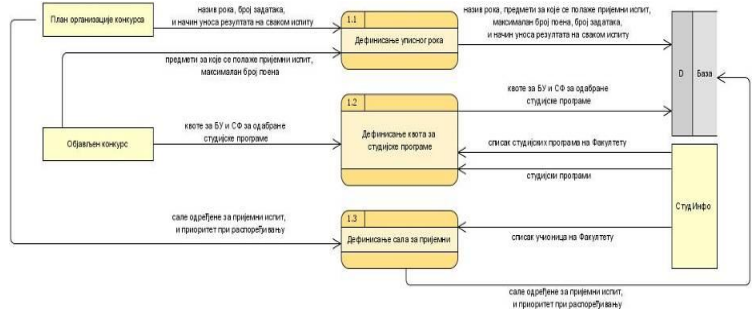
## Дијаграм тока података – ДТП (8) – Дијаграм контекста



## Дијаграм тока података – ДТП (9) – Дијаграм нивоа 0, хијерархијска декомпозиција система

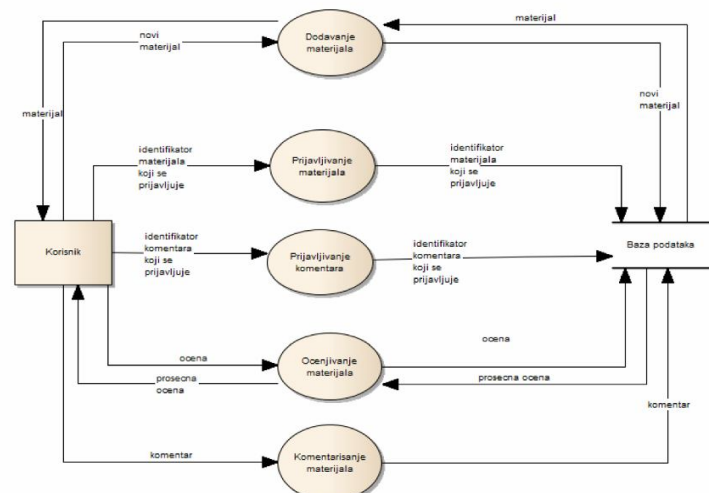


### Дијаграм тока података – ДТП (10) – Дијаграм нивоа 1, припрема за пријемни испит



Универзитет у Београду – Математички факултет

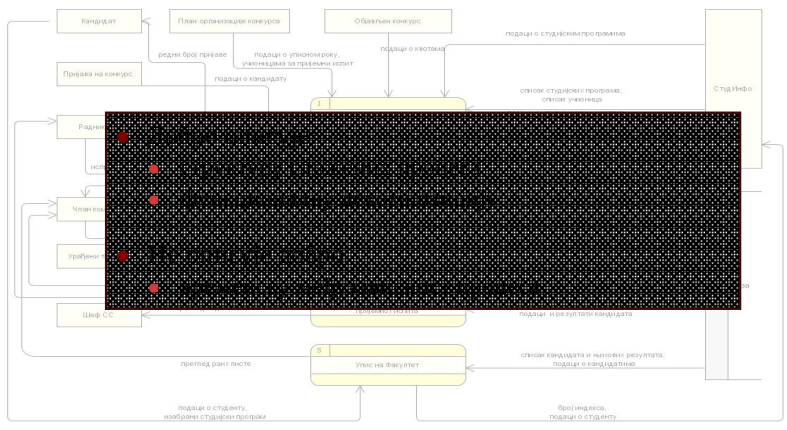
### Дијаграм тока података – ДТП (11) – Алтернативни начини представљања



Универзитет у Београду – Математички факултет

### Технике описивања процеса / Описивање структуре процеса

## Дијаграм тока података – ДТП (12) – Резиме



Универзитет у Београду – Математички факултет

### Технике описивања процеса / Описивање структуре процеса

## Матрице

- Матрице се употребљавају за моделирање и представљање односа у веома различитим случајевима
- Неке од примена обухватају представљање:
  - односа послова и функција
  - односа ентитета (података) и функција
  - односа функција и субјеката
  - односа ентитета (података) и субјеката
  - и друго

Универзитет у Београду – Математички факултет



## Матрице (2) – Послови и функције

- Намена:
  - документовање односа послова и функција
  - препознавање функција са локалним и ширим значајем
  - препознавање атомичних функција
- Колоне представљају послове а врсте функције (или обратно)
  - Послови обухватају крупније поступке или догађаје који се дешавају у оквиру домена
  - Функције представљају појединачне (потенцијално сложене) операције
    - пожељно је да функције буду атомичне (тј. да се свака функција употребљава само као целина, а никада делимично)
  - Сваком послу мора да одговара бар једна функција
  - Број и величина таблица зависи од величине и сложености система
    - идеално је да се сваки посао и свака функција појављују у тачно по једној табlici
    - у супротном су могуће грешке услед редундантности



## Матрице (3) – Послови и функције

|                        | Посао | Упис на факултет | Упис године | Исписивање са факултета |
|------------------------|-------|------------------|-------------|-------------------------|
| Функција               |       |                  |             |                         |
| Пријављивање кандидата |       | X                |             |                         |
| Уписивање на факултет  |       | X                |             |                         |
| Промена статуса        |       |                  | X           | X                       |
| Избор предмета         |       | X                | X           |                         |
| Уписивање године       |       | X                | X           |                         |
| Одређивање школарине   |       | X                | X           |                         |
| Проверавање дуговања   |       |                  | X           | X                       |
| Евидентирање уплате    |       | X                | X           | X                       |



## Матрице (4) – Ентитети и функције

- Намена:
  - документовање односа ентитета и функција
  - препознавање ентитета са локалним и ширим значајем
- Колоне представљају ентитете а врсте функције (или обратно)
  - Свака функција мора да употребљава бар један ентитет
  - Функција може употребљавати ентитет на различите начине (ПЧМБ, енгл. *CRUD*):
    - Прављење – функција прави нови ентитет
    - Читање – функција чита постојећи ентитет
    - Мењање – функција (потенцијално) мења постојећи ентитет
    - Брисање – функција брише постојећи ентитет
  - Број таблица зависи од величине и сложености система



## Матрице (5) – Ентитети и функције

|                        | Ентитет | Пријава на конкурс | Индекс | Пријава за упис године | Уписана година | Уписан курс | Фин. обавеза |
|------------------------|---------|--------------------|--------|------------------------|----------------|-------------|--------------|
| Функција               |         |                    |        |                        |                |             |              |
| Пријављивање кандидата |         | П                  |        |                        |                |             |              |
| Уписивање на факултет  |         | Ч                  | П      |                        |                |             |              |
| Промена статуса        |         |                    | ЧМ     |                        |                |             |              |
| Избор предмета         |         |                    |        | П                      |                |             |              |
| Уписивање године       |         |                    |        | ЧМ                     | П              | П           |              |
| Одређивање школарине   |         |                    | Ч      |                        | ЧМ             | Ч           |              |
| Проверавање дуговања   |         |                    | Ч      |                        |                |             | Ч            |
| Евидентирање уплате    |         |                    | Ч      |                        |                |             | ЧМ           |



## Матрице (6) – Друго

- Матрица односа процеса и субјеката
  - сагледавање које функције се употребљавају у којим организационим јединицама
- Матрица односа ентитета и субјеката
  - сагледавање који подаци се употребљавају у којим организационим јединицама
  - употребљава се претежно за
    - означавање ентитета који се ад-хок одржавају
      - на пример, неки подаци о службеницима у кадровској служби
    - означавање орг.јединица које читају податке
      - на пример управа чита податке ради прављења извештаја
- и друго



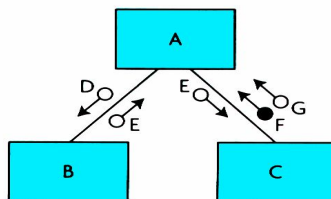
## Дијаграми структуре

- Представљају функционалну декомпозицију проблема
  - хијерархијска организација
- Графичко представљање структуре
  - правоугаоницима се представљају процеси или модули
  - линијама се повезују процеси / модули
  - уз линије се наводе подаци који се преносе



## Дијаграми структуре (2)

- Модул А може да користи (позива) модуле В и С
  - редослед позива се не види из овог дијаграма
- При позивању се прослеђују назначени подаци
  - попуњен кружић означава *застајавицу* која утиче на даљи ток операција



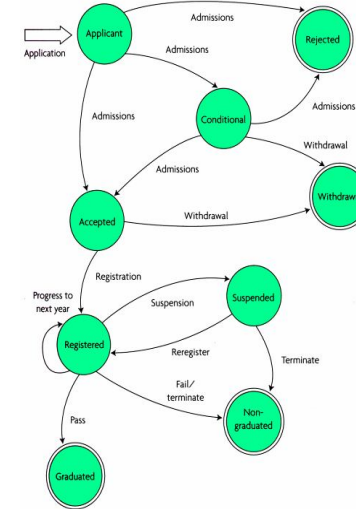
## Дијаграми структуре (3)

- Користе се у различитим методологијама
- Тежи се
  - минималној међузависности модула (спрегнутост)
  - вишеструкој употреби модула (кохезија)

## Животни циклус ентитета

- Употребљава се за описивање стања кроз која пролази неки ентитет
- Одговара дијаграму стања *UML*-а
  - биће обрађено у оквиру *OO* техника

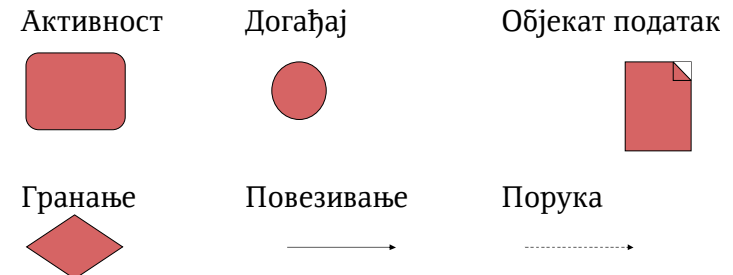
## Животни циклус ентитета - пример



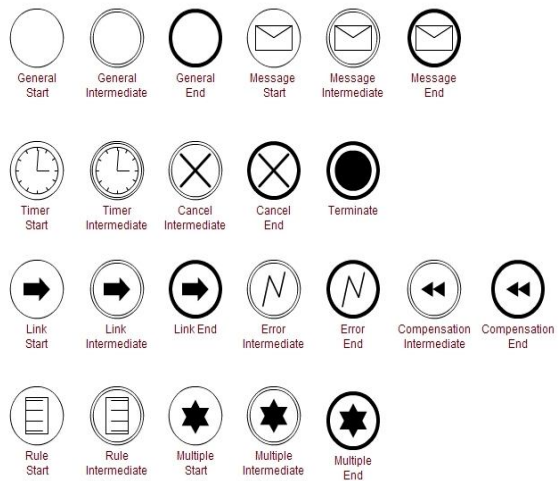
## Моделирање пословних процеса - BPMN

- Нотација за моделирање пословних процеса (енгл. *Business Process Modeling Notation - BPMN*)
  - напреднија нотација за представљање пословних процеса
  - развијена од стране *OMG* у периоду 2001-2006 (v1)
- Основне врсте дијаграма:
  - дијаграм процеса
  - дијаграм сарадње
- Нотација обухвата различите облике записивања нефункционалних захтева

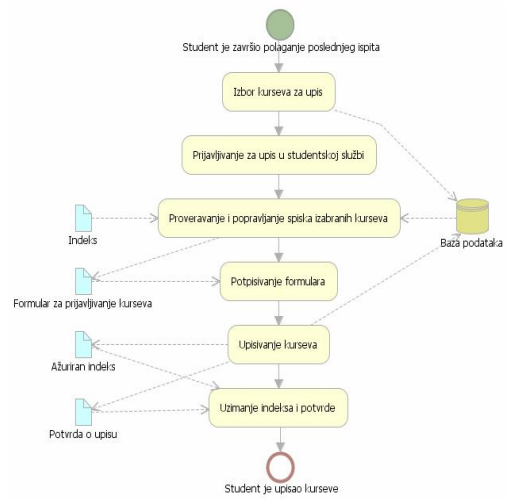
## Основни елементи



### Додатни симболи догађаја

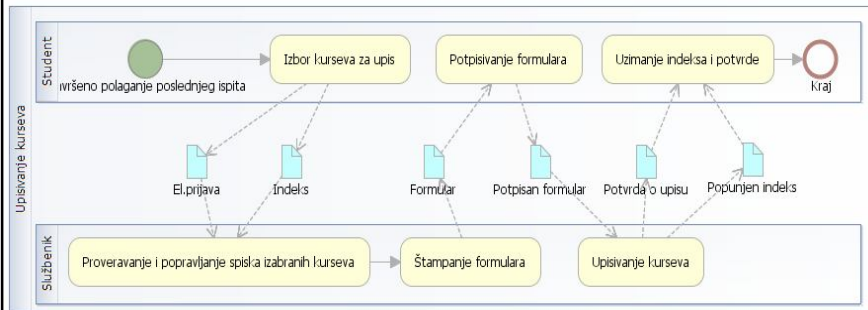


### Пример дијаграма процеса

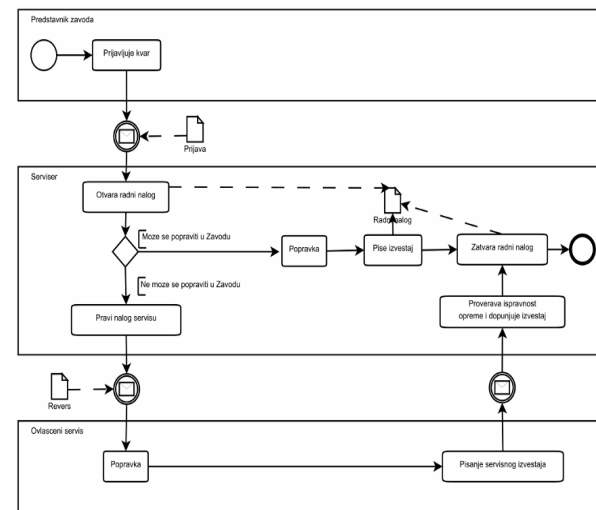


### Технике описивања процеса / Описивање структуре процеса / BPMN

## Пример дијаграма сарадње

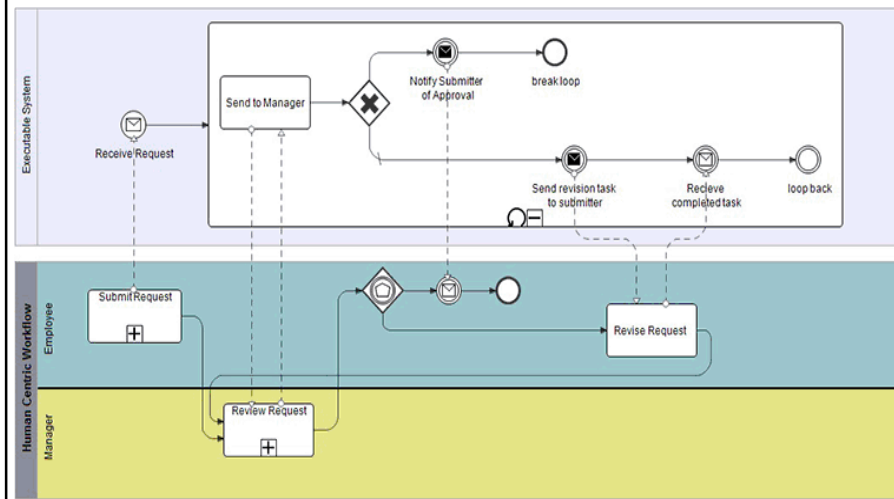


### Пример дијаграма сарадње са применом додатних симбола

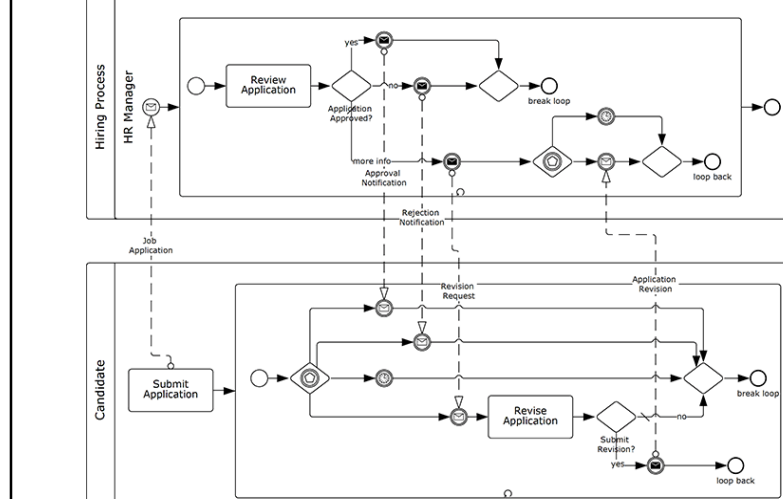




Пример дијаграма сарадње са применом додатних симбола (2)



Пример дијаграма сарадње са применом додатних симбола (3)



## Литература за тему

- *Avison, Fitzgerald, Information Systems Development (4.ed), McGraw Hill, 2005*
- [www.bpmn.org/Documents/OMG\\_BPMN\\_Tutorial.pdf](http://www.bpmn.org/Documents/OMG_BPMN_Tutorial.pdf)
- [www.bpmn.org/Documents/6AD5D16960.BPMN\\_and\\_BPM.pdf](http://www.bpmn.org/Documents/6AD5D16960.BPMN_and_BPM.pdf)

# Информациони системи

Саша Малков

[P271]

## Тема 5

# Технике моделирања података

## Технике моделирања података

- Моделирају
  - структуру података
  - међусобне односе података
- Најважније технике
  - Дијаграми ентитета и односа
  - Дијаграми табела
  - Дијаграми класа података

## Дијаграми ентитета и односа

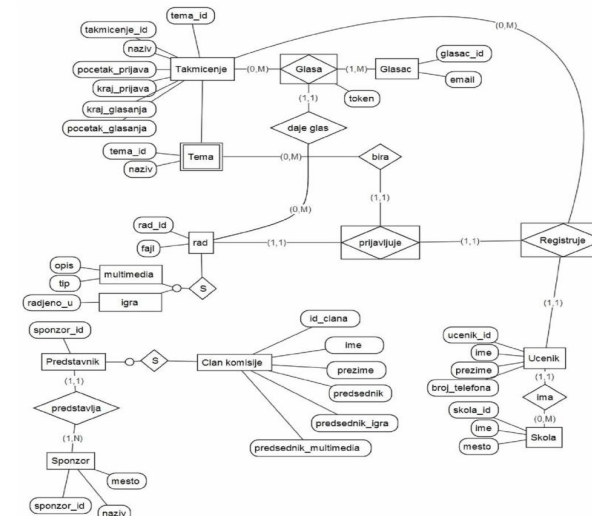
(енгл. *Entity-Relationship (ER) diagrams*)

- Веома значајна техника моделирања и описивања података
- Потекла из “Модела ентитета и односа” као модела података
- Добро описује семантику података и односа
- Обрађено детаљно у оквиру База података и Пројектовања база података
  - научити из литературе

## Дијаграми ентитета и односа (2)

- Првенствено намењена за концептуално моделирање
- Релативно лако се преводи у логички модел табела
- Скоро еквивалентно дијаграмима класа података

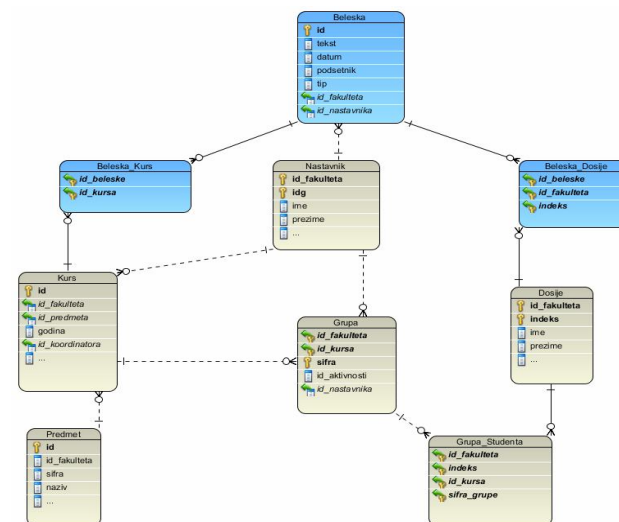
## Дијаграми ентитета и односа - пример 1



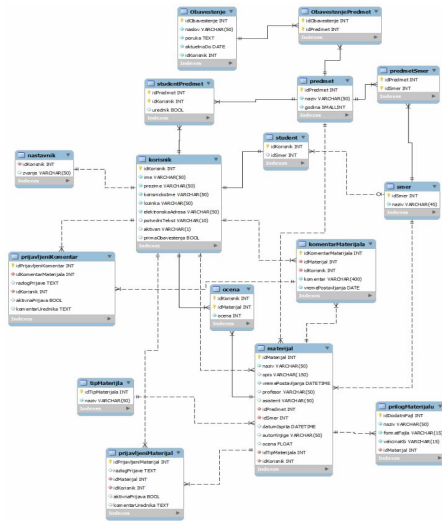
## Дијаграми табела

- Дијаграмска техника релационог модела података
  - Релативно често се погрешно називају ЕР дијаграмима, иако је потпуно другачија техника
- Не описују добро семантику односа
- Погодни су за логичко и физичко моделирање
- Обрађено детаљно у оквиру База података и Пројектовања база података
  - научити из литературе

## Дијаграми ентитета и односа – пример 2



## Дијаграми ентитета и односа – пример 3



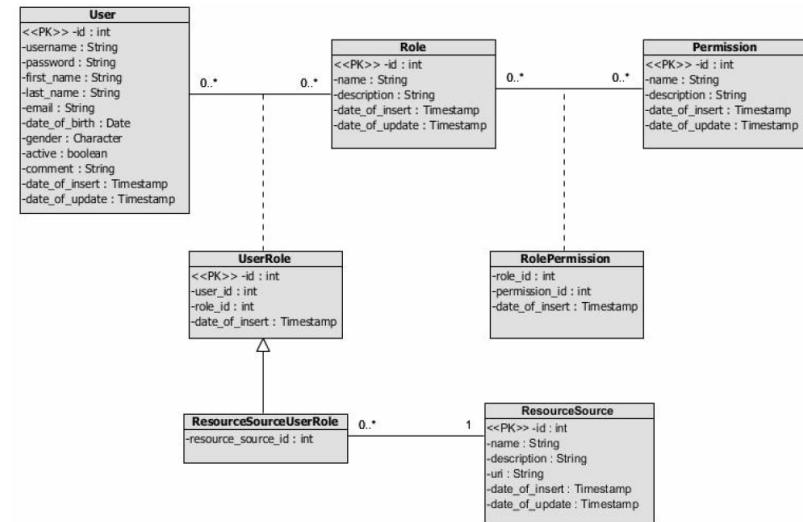
## Дијаграми класа података

- Данас се често употребљава уместо дијаграма ентитета и односа
  - шира техника од дијаграма ентитета и односа
  - моделирање неких аспеката понашања
  - моделирање хијерархија ентитета
  - моделирање података ближе апликацији него бази података
- Специфичан вид дијаграма класа, тзв. **дијаграми класа њодашака** или **дијаграми класа ентитетна**
  - само подкуп који се односи на структуру и односе
  - изоставља се понашање класа

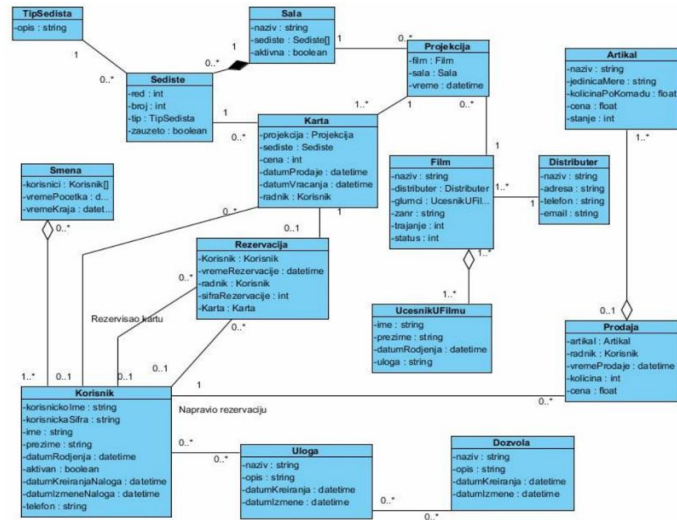
## Дијаграми класа података

- Ова техника је добра за све нивое моделирања, али је најбоља за концептуално моделирање података
  - Добро описује семантику података и односа
- Биће детаљније обрађени као ОО техника
- Обрађено детаљно у оквиру База података и Пројектовања база података
  - научити из литературе

## Дијаграма класа ентитета - пример 1



## Дијаграма класа ентитета - пример 2



## Избор технике

- Дијаграми ентитета и односа и дијаграми класа су апстрактнији од дијаграма табела
  - добро описују семантику података и односа
  - описују неке аспекте понашања
  - погодни за концептуално моделирање
- Дијаграми табела су прилагођени логичком и физичком моделу, када се прави релациона база података
  - ближи су физичкој имплементацији
  - практично се баве само структуром података
- Дијаграми класа података су нешто ближи имплементацији (и пројекту) апликације
  - добро се уклапају у контекст општих ОО методологија и техника
  - лакши и прегледнији за употребу од дијаграма ентитета и односа



## Избор технике (2)

- Из дијаграма класа се скоро једнозначно може извести дијаграм табела
  - може да се бира између више начина имплементирања наслеђивања
  - обрнуто не важи, зато што се различити односи дијаграма класа преводе у исте односе на дијаграму ентитета и односа
- У фази моделирања система је обично боље да се употребљава дијаграм класа
- У фази физичког пројектовања базе података је згоднији дијаграм табела

## Информациони системи

Саша Малков



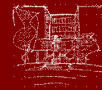
### Тема 6

## Објектно-оријентисане технике



## Објектно оријентисане технике

- Основни концепти ОО техника:
  - у жижу стављају објекте, а не процесе
  - различите дијаграмске технике за моделирање понашања, структуре, односа и животног циклуса објеката



## Појам објекта

- Објекат је апстракција нечега у домену проблема, која описује способност система да о томе чува информацију, интерагује са тиме или обоје.
  - (Coad, Yourdon, 1990)
- Објекат је концепт, апстракција или нешто са јасним границама и смислом у односу на конкретан проблем. Објекат има две сврхе: да помогне размевању стварног света и пружи практичну основу за рачунарску имплементацију.
  - (Rumbaugh at al., 1991)
- Објекти се описују одговорима које могу да дају на питања:
  - Ко сам ја?
  - Шта могу да урадим?
  - Шта знам?
  - (Wirf-Brock, 1990)
- Објекти имају стање, понашање и идентитет
  - (Booch, 1994)



## Појам објекта (2)

- Резиме:
  - Објекат је апстракција нечег конкретног у домену проблема
  - Објекат мора да има стање, понашање или знање
    - (не обавезно све)
  - Објекат има животни циклус
    - настајање
    - постојање
    - нестајање



## Појам класе

- Класа је апстракција скупа објеката који имају “довољно сличности”
  - Као критеријум сличности се преваходно узима у обзир понашање
  - Стање и знање објеката не утиче значајно на њихову класификацију
  - Објекат представља конкретан примерак класе



## Дефиниција ОМГ

- “Класа је опис скупа објеката који деле исте атрибуте, операције, методе, односе и семантику. Сврха класе је да декларише колекцију метода, операција и атрибута која у потпуности описује структуру и понашање тих објеката.”
- “Објекат је примерак који потиче из класе, структуриран је и понаша се у складу са својом класом.”



## Атрибути и методи

- Атрибути су средство за описивање
  - стања
  - знања
- Методи су средство за описивање
  - понашања



## Објекти и класе данас

- Класе се формално дефинишу као скупови свих објеката који:
  - Имају прописане атрибуте
    - одговарајућег имена и типа
  - Могу да примене прописане методе
    - одговарајућег имена и типа
  - Задовољавају одговарајуће формалне услове
    - семантика атрибута и метода



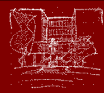
## У програмским језицима

- У већини савремених програмских језика се распознавање да ли неки објекат припада некој класи одвија прагматично, без примене ад-хок правила
- Припадност објекта класи
  - по правилу се одређује у тренутку прављења објекта
  - не може се мењати током живота објекта
- Напомена:
  - Постоје језици код којих се припадност класи установљава динамички, проверавањем семантичких правила



## Основни концепти ООМ

- Енкапсулација
- Интерфејс
- Полиморфизам
- Наслеђивање
  - Специјализација и генерализација
  - Хијерархије класа



## Енкапсулација

- Структура објеката је њихова интерна ствар
  - не сме се излагати спољашњем свету
  - атрибутима се сме приступати само посредно
- Постоје случајеви када неки атрибути представљају основу понашања објеката или класа
  - тада им се може омогућити јавни приступ
  - и тада је ипак препоручљиво енкапсулирање
- Сврха
  - Апстраховање структуре методима
  - Виши ниво међусобне независности класе од модула у којима се употребљава



## Интерфејс

- Објекат (класа) пружа спољашњим корисницима само скуп метода путем којих могу комуницирати са њим
- Такав скуп метода се назива *јавни интерфејс* објекта (класе)
- Интерфејс се обликује тако да омогући обављање једног целовитог посла
- Сврха
  - Суштина објекта је у њиховом понашању. Интерфејс омогућава то понашање.
  - Сужавањем интерфејса на суштину функционисања објекта прикрива се сва сложеност имплементације и пружа виши ниво независности објекта од окружења
- Напомена
  - Ако објекат има више интерфејса, значи да има више функција, па је потребно размотрити његово разлагање на више објеката



## Полиморфизам

- Полиморфизам подразумева да се једном написан код може употребљавати за различите врсте објеката
- Постоје три основне врсте полиморфизма: хијерархијски, параметарски и имплицитни
  - Сви ОО програмски језици омогућавају хијерархијски полиморфизам
  - Неки савремени ООПЈ омогућавају параметарски полиморфизам
  - Само ретки језици подржавају имплицитни полиморфизам
- Сврха
  - Полиморфизам омогућава писање апстрактнијег кода, који има високу употребљивост



## Наслеђивање

- Наслеђивање класе је еквивалентно увођењу једносмерне парцијално уређене релације “јесте” између класа
  - Класа А “јесте” класа Б ако сваки објекат класе А има све особине које имају и објекти класе Б
- Ако класа А “јесте” класа Б каже се и да је
  - А “изведена” класа из Б или А “је потомак” класе Б
  - Б “основна” класа за А или Б “је предак” класе А
- Релација “јесте” је парцијална релација поретка (рефлексивна и транзитивна)
  - Представља основу за грађење хијерархија класа
- Сврха:
  - Наслеђивање се користи за експлицитно означавање сличности међу класама (објектима)
  - Представља основу за хијерархијски полиморфизам: ако се нешто може урадити са објектом класе Б, онда се то може урадити и са сваким објектом класе А која је изведена из Б



## Специјализација и генерализација

- Наслеђивање се посматра у два смера, као
  - специјализација или
  - генерализација
- Ако класа А “јесте” Б, а класа Б “није” А онда:
  - класа А је посебан (специјалан) случај класе Б
  - класа Б је општији (генералан) случај класе А





## УМЛ

- Обједињени језик за моделирање (*Unified Modeling Language - UML*)
- Настао обједињавањем више различитих приступа ОО пројектовању и одговарајућих језика моделирања



## УМЛ (2)

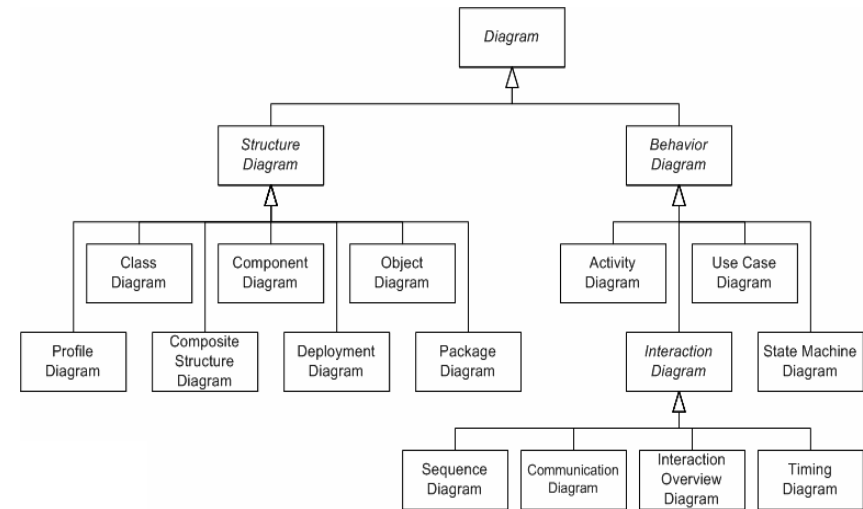
- Сам језик не представља методологију, али је обликован према паралелно развијаној методологији – “Обједињени приступ” (*Unified Approach*)
- Сам језик је развијен, публикован и шире прихваћен пре одговарајуће методологије
- Данас се користи у практично свим ОО развојним методологијама, па и шире



## УМЛ – Врсте дијаграма

- Сви се деле у три групе:
  - дијаграми понашања
  - дијаграми интеракције
  - структурни дијаграми

## УМЛ – Структурни дијаграми



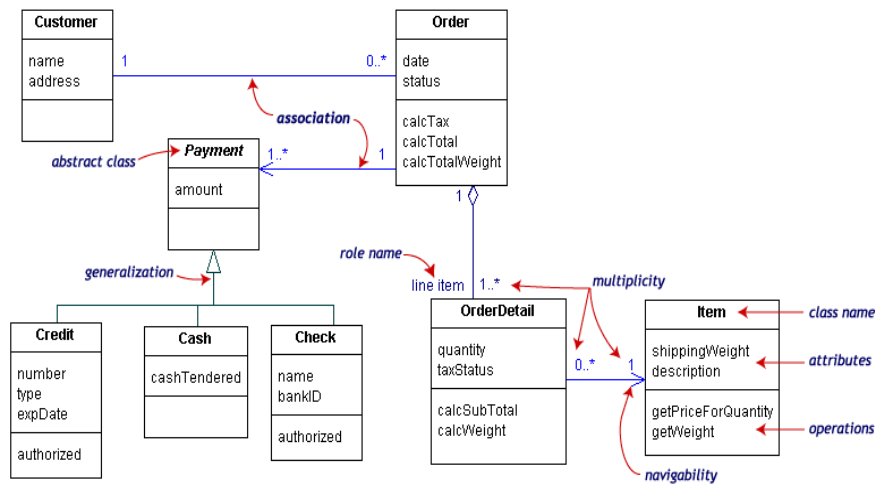
## УМЛ – Структурни дијаграми

- Дијаграм класа (*class diagram*)
- Дијаграм компоненти (*component diagram*)
- Дијаграм објеката (*object diagram*)
- Дијаграм профила (*profile diagram*)
- Дијаграм сложене структуре (*composite structure diagram*)
- Дијаграм испоручивања (*deployment diagram*)
- Дијаграм пакета (*package diagram*)

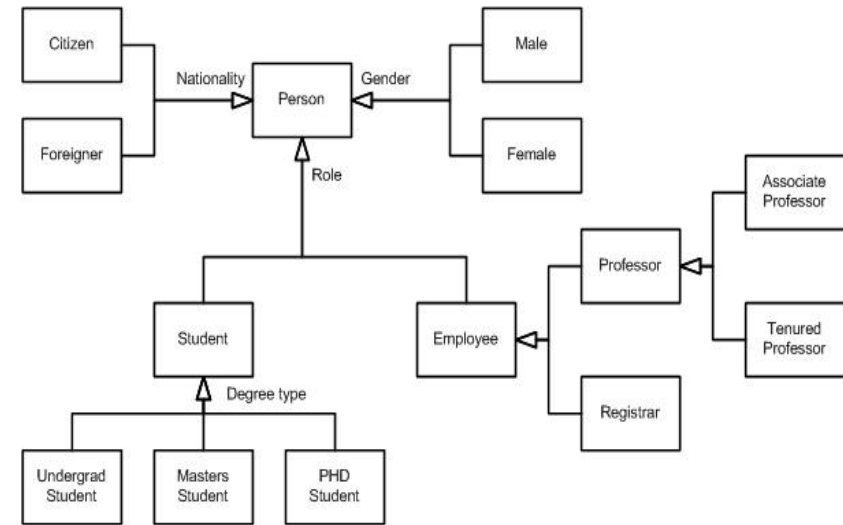
## УМЛ – Дијаграм класа

- Илуструје елементе статичног модела, као што су класе, њихов садржај и међусобне односе
- Садржи
  - називе класа
  - атрибуте класа
  - методе класа
  - специјализацију и генерализацију
  - односе
    - асоцијацију
    - агрегацију
    - композицију

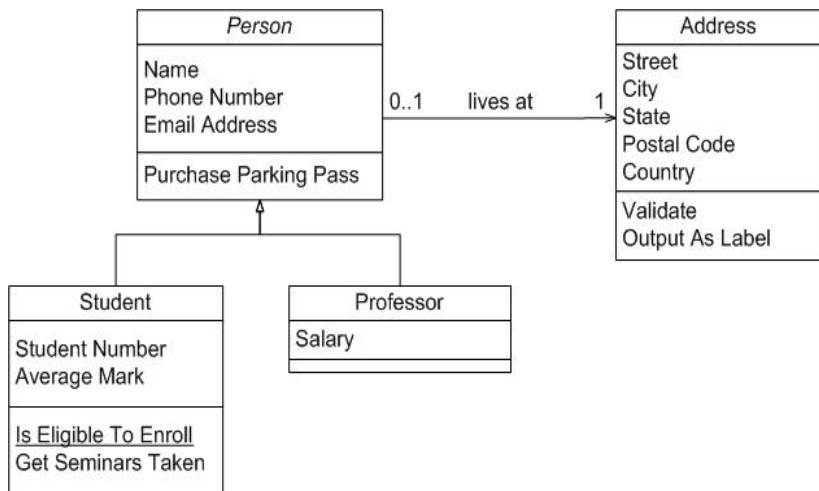
### УМЛ – Дијаграм класа – пример 1



### УМЛ – Дијаграм класа – пример 2



UML – Дијаграм класа – пример 3



Универзитет у Београду - Математички факултет

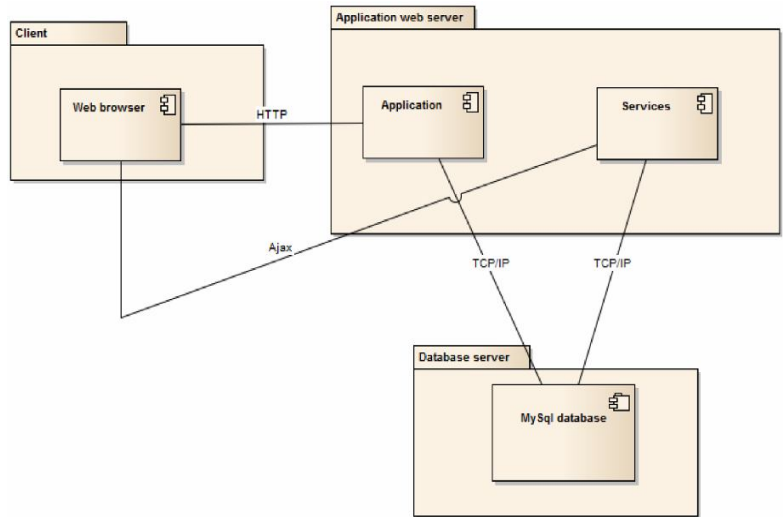
UML – Дијаграм компоненти



- Илуструје компоненте које чине апликацију, систем или организацију.
- Садржи
  - називе компоненти
  - њихове међусобне односе
  - јавне интерфејсе

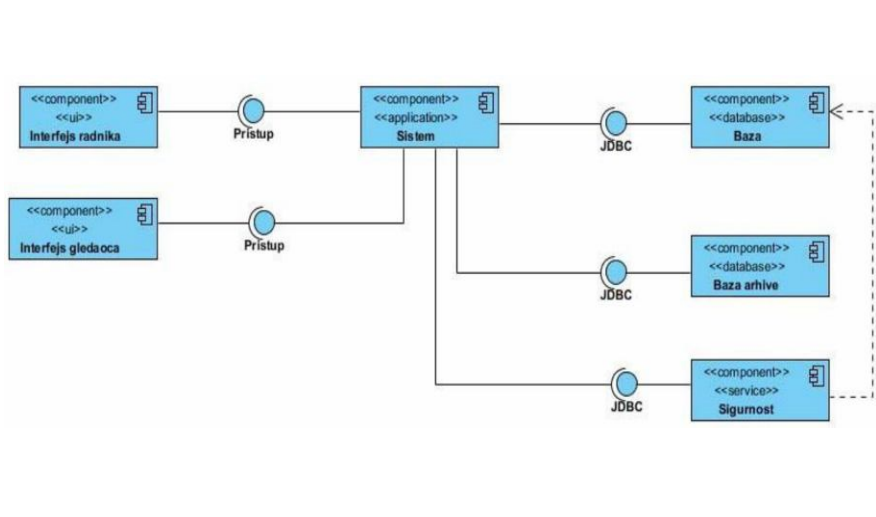
Универзитет у Београду - Математички факултет

UML – Дијаграм компоненти – пример 1



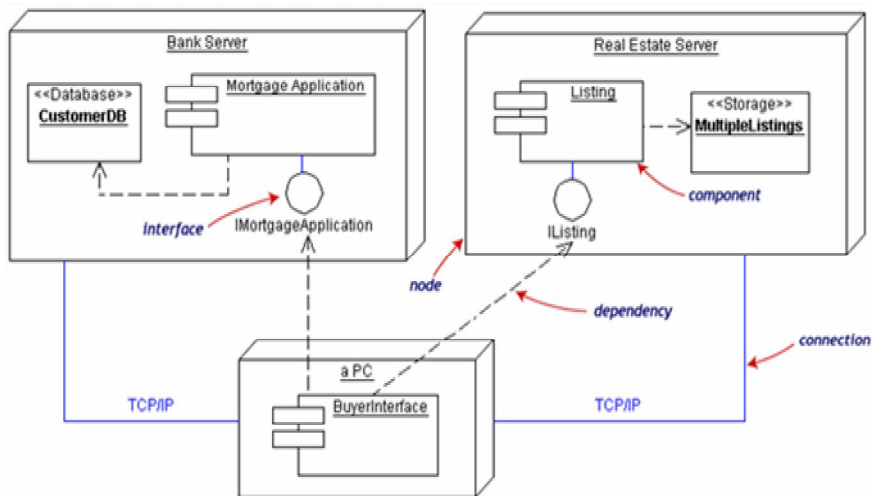
Универзитет у Београду - Математички факултет

UML – Дијаграм компоненти – пример 2



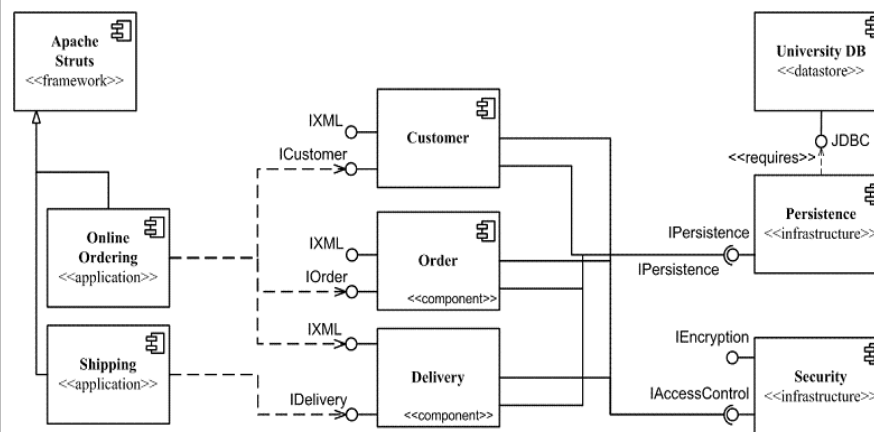
Универзитет у Београду - Математички факултет

### UML – Дијаграм компоненти – пример 3



Универзитет у Београду - Математички факултет

### UML – Дијаграм компоненти – пример 4



Copyright 2005 Scott W. Ambler

Универзитет у Београду - Математички факултет

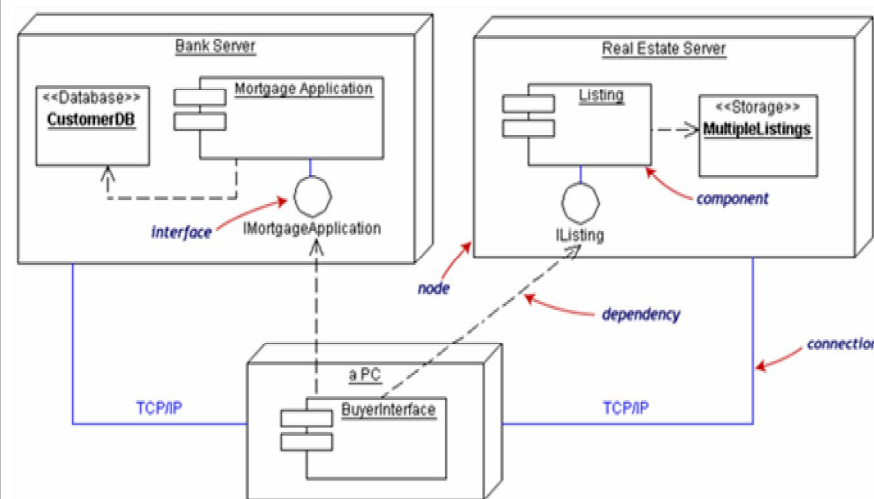
## UML – Дијаграм испоручивања



- Представља елементе физичке архитектуре система.
- Садржи
  - чворове (сервере)
  - софтверске или хардверске подсистеме
  - међусобне везе подсистема
  - може да илуструје и заступљеност компоненти у подсистемима

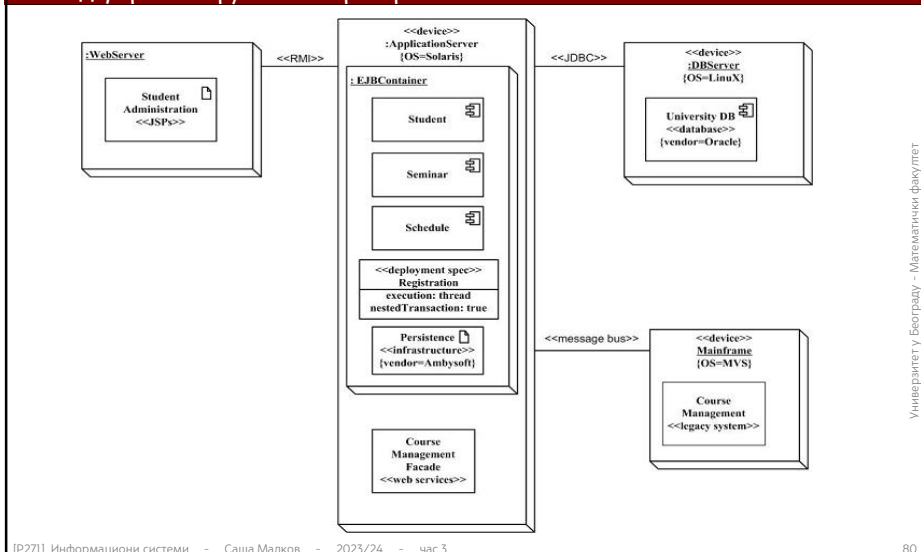
Универзитет у Београду - Математички факултет

### UML – Дијаграм испоручивања – пример 1

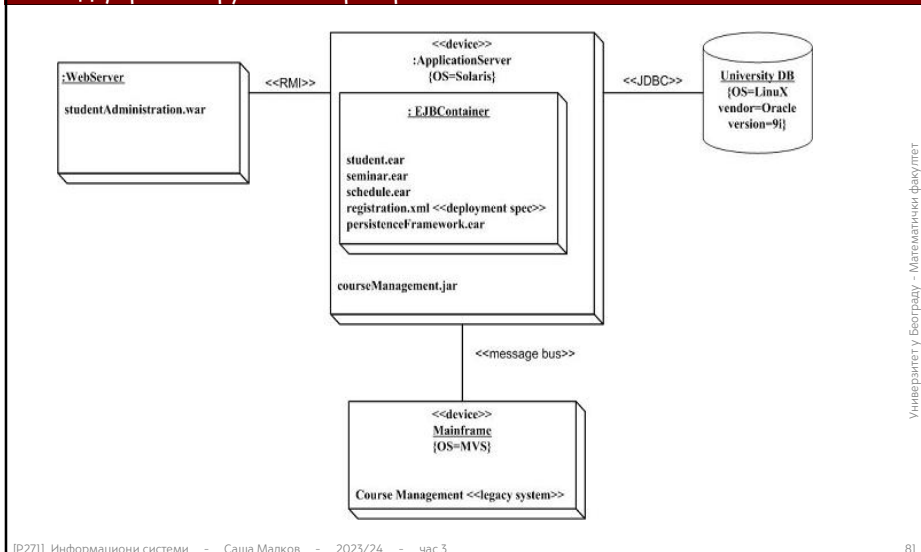


Универзитет у Београду - Математички факултет

### UML – Дијаграм испоручивања – пример 2



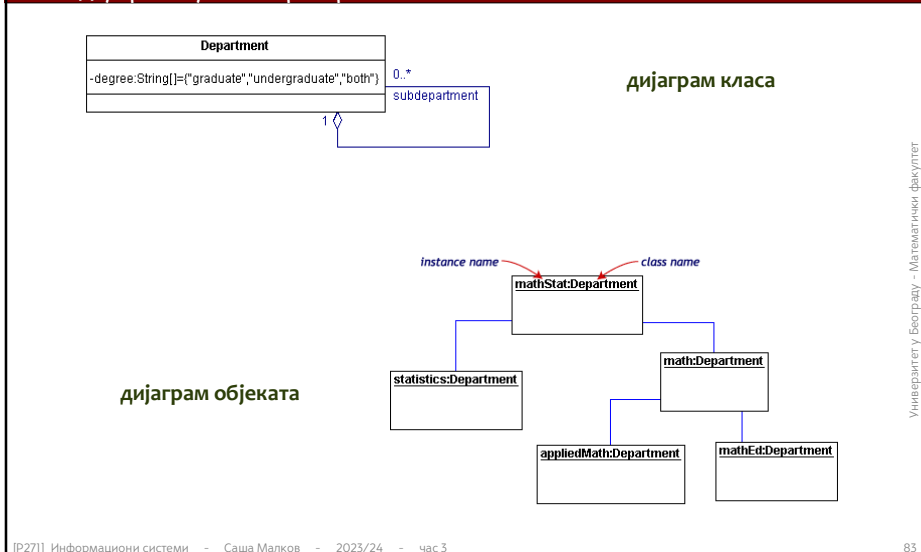
### UML – Дијаграм испоручивања – пример 3



## UML – Дијаграм објеката

- Представља објекте и њихове односе у једном тренутку времена. Користи се као допуна дијаграма класа и комуникације за описивање динамичких система.
- Садржи
  - називе класа
  - називе објеката
  - имена и вредности атрибута
  - односе

### UML – Дијаграм објеката – пример 1

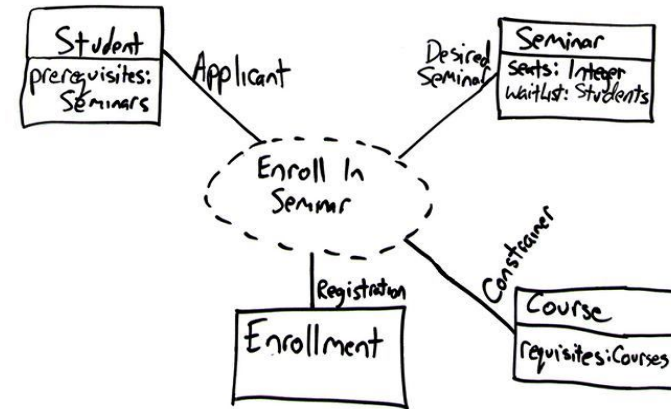




## УМЛ – Дијаграм сложене структуре

- Представља интерну структуру класе, објекта, компоненте или случаја употребе.
- Садржи
  - сложене компоненте и њихове елементе
  - тачке интеракције са другим елементима система

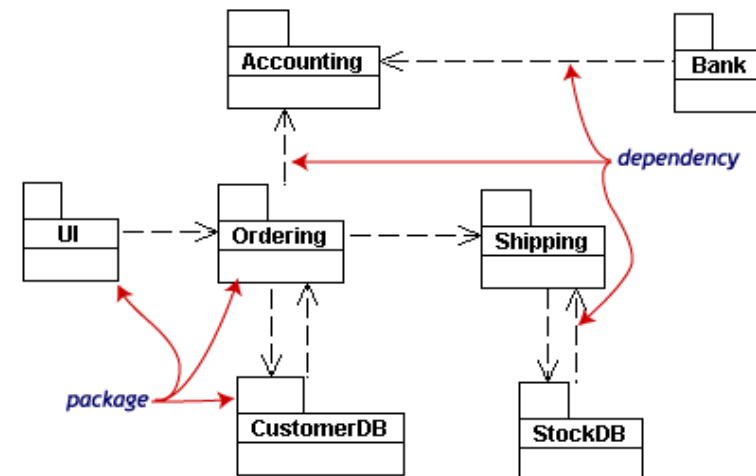
## УМЛ – Дијаграм композитних структура – пример 1



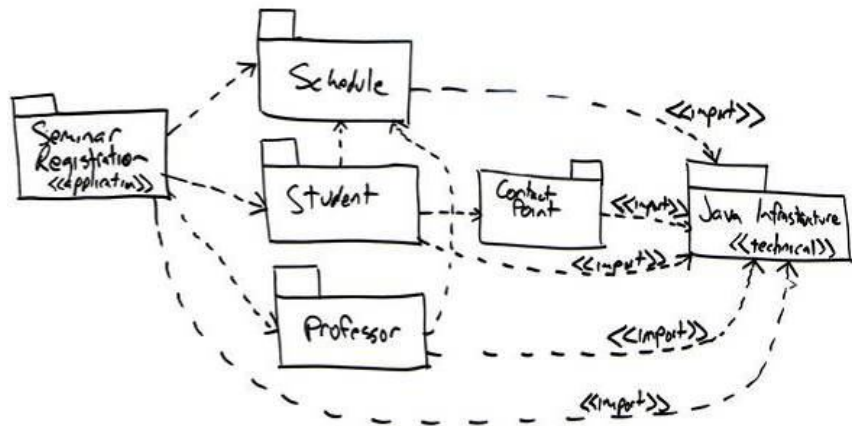
## УМЛ – Дијаграм пакета

- Илуструје како су елементи логичког модела организовани у пакете, као и међузависности пакета.
- Садржи
  - називе и границе пакета
  - класе у пакетима
  - међусобне односе класа
  - међусобне зависности пакета
  - може да се користи и у домену случајева употребе

## УМЛ – Дијаграм пакета – пример 1

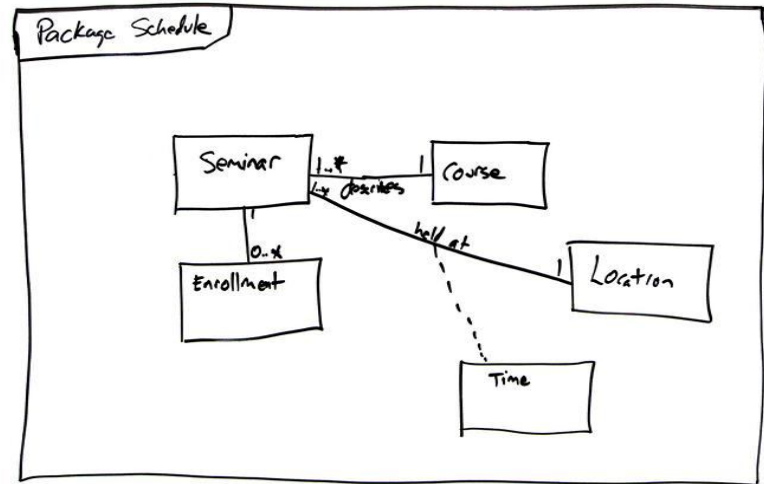


UML – Дијаграм пакета – пример 2



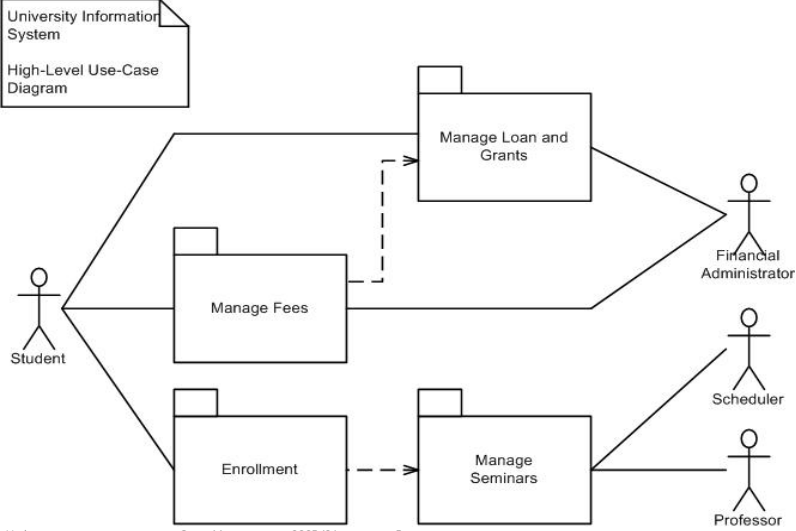
Универзитет у Београду - Математички факултет

UML – Дијаграм пакета – пример 3



Универзитет у Београду - Математички факултет

UML – Дијаграм пакета – пример 4



Универзитет у Београду - Математички факултет

## Литература за тему

- Avison, Fitzgerald, **Information Systems Development (4.ed)**, McGraw Hill, 2005
- Simon Bennett, Steve McRobb, Ray Farmer, **Object Oriented Systems Analysis and Design (Using UML)**, McGraw Hill, 2002
- Ambler, <http://www.agilemodeling.com/essays/umlDiagrams.htm>
- OMG UML: <http://www.uml.org/>
- <http://edn.embarcadero.com/article/31863>



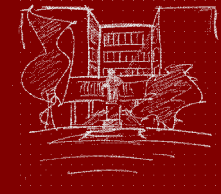
Универзитет у Београду - Математички факултет

## Додатни извори



- Booch, *Object oriented design with applications (1.ed)*, Benjamin/Cummings, 1991.
- Coad, Yourdon, *Object Oriented Analysis (2.ed)*, Prentice Hall, 1991.
- Martin, Odell, *Object Oriented Information Engineering*, Prentice Hall, 1992.
- Yourdon, Argila, *Case Studies in Object-Oriented Analysis and Design*, Prentice Hall, 1996
- [http://atlas.kennesaw.edu/~dbraun/csis4650/A&D/UML\\_tutorial/](http://atlas.kennesaw.edu/~dbraun/csis4650/A&D/UML_tutorial/)
- <http://uml-tutorials.trireme.com/>
- <http://www.uml.org/>

Хвала на пажњи!



**МАТФ**  
Универзитет у Београду  
Математички факултет

