

## Projektovanje funkcionalnosti

Metodi funkcionalne dekompozicije (FD):

- Funkcionalna dekompozicija I vrste
- Funkcionalna dekompozicija II vrste
- Funkcionalna dekompozicija III vrste

Suština metoda:

- Dekomponovanje funkcionalnosti na podfunkcije sa različitim nivoom detalja.

## Projektovanje funkcionalnosti – FD1

Funkcionalna dekompozicija I vrste (FD1):

- Funkcionalnosti se dekomponuju na skup potencijalnih podfunkcija bez ikakvih dodatnih detalja.

Dve varijante FD1:

- Neanotirana FD1: bez naznake tokova podataka od/ka okruženju i podacima i između funkcionalnosti.
- Anotirana FD1: sa naznakom tokova podataka od/ka okruženju i podacima i između funkcionalnosti – horizontalno i vertikalno.

## Projektovanje funkcionalnosti – FD1

Primer neanotirane FD1 - BIBLIOTEKA:

BIBLIOTEKA

OdrzavanjeMaticnihPodataka

ObradaPrometa

ObradaTraznjeNaslova

ObradaVracanjaKnjige

ObradaGubitkaKnjige

Izvestavanje

## Projektovanje funkcionalnosti – FD1

U daljoj dekompoziciji:

### BIBLIOTEKA

...

ObradaPrometa

ObradaTraznjeNaslova

OcitavanjeNeophodnihPodataka

ProveraVecIma

ProveraImaSlobodno

EvidentiranjeIzdavanja

EvidentiranjeRezervacije

...

...

## Projektovanje funkcionalnosti – FD1

Anotirana FD1:

Uz funkcionalnosti se u zagradama naznačuju:

- imenovani i usmereni tokovi podataka:
  - => I.Tok    ulazni tok od okruženja (na vrhu samo I)
  - <= O.Tok    izlazni tok ka okruženju (na vrhu samo I)
  - > Tok        ulazni tok ka funkcije od ostatka sistema
  - < Tok        izlazni tok od funkcije ka ostatku sistema
  - <> Tok        ulazno-izlazni tok funkcije ka ostatku sistema

## Projektovanje funkcionalnosti – FD1

- imenovani i usmereni pristupi podacima IS:
  - > D.Tok ulazni tok ka podacima (na vrhu samo D)
  - < D.Tok izlazni tok od podataka (na vrhu samo D)
  - <> D.Tok ulazno-izlazni tok (na vrhu samo D)

I kod tokova i kod podataka simbol . označava odnos sadržanosti – ono iza je na bilo koji način komponenta onog ispred.

Pri dekompoziciji mora biti ispoštovana usaglašenost.

## Projektovanje funkcionalnosti – FD1

### Primer anotirane FD1 - BIBLIOTEKA:

BIBLIOTEKA (=>I,<=O,<>D)

OdrzavanjeMaticnihPodataka

(=>I.Maticno,<=O.IshodMaticno,<>D.Maticno)

ObradaPrometa

(=>I.Promet,<=O.IshodPromet,<>D.Promet)

Izvestavanje

(=>I.Izvestaj,<=O.Izvestaj,<>D)

## Projektovanje funkcionalnosti – FD1

U daljoj dekompoziciji:

BIBLIOTEKA (=>I,<=O,<>D)

...

ObradaPrometa

(=>I.Promet,<=O.IshodPromet,<>D.Promet)

ObradaTraznjeNaslova

(=>I.Promet.Trazi,<=O.IshodPromet.Trazi,<>D.Promet)

ObradaVracanjaKnjige

(=>I.Promet.Vrati,<=O.IshodPromet.Vrati,<>D.Promet)

ObradaGubitkaKnjige

(=>I.Promet.Gubi,<=O.IshodPromet.Gubi,<>D)

...



## Projektovanje funkcionalnosti – FD2

Funkcionalna dekompozicija II vrste (FD1):

- Funkcionalnosti se dekomponuju na skup potencijalnih podfunkcija sa dodatnim detaljima:
  - \* redosled                      **n** redni broj
  - \* uslovljenost                **?** ispred rednog broja
  - \* isključivost                **e** iza rednog broja
  - \* ponavljanje                **\*** ispred rednog broja

Dve varijante FD2, isto kao i kod FD1:

- Neanotirana FD2
- Anotirana FD2

# SI3IS1 2.1.3

j

## Projektovanje funkcionalnosti – FD2

Primer neanotirane FD2 - BIBLIOTEKA:

### BIBLIOTEKA

- 1e OdrzavanjeMaticnihPodataka

- 1e ObradaPrometa

  - 1e ObradaTraznjeNaslova

  - 1e ObradaVracanjaKnjige

  - 1e ObradaGubitkaKnjige

- 1e Izvestavanje

## Projektovanje funkcionalnosti – FD1

U daljoj dekompoziciji:

### BIBLIOTEKA

...

1e ObradaPrometa

1e ObradaTraznjeNaslova

1 OcitavanjeNeophodnihPodataka

2 ProveraVecIma

?3 ProveraImaSlobodno

?4e EvidentiranjeIzdavanja

?4e EvidentiranjeRezervacije

...

...

## Projektovanje funkcionalnosti – FD2

### Anotirana FD2:

- Ista anotacija kao kod FD2
- Kada je razvijen i model podataka, notacija odnosa sa podacima se zamenjuje detaljnijom notacijom odnosa sa entitetima modela podataka, gde uz naziv entiteta ide:

**C:Entitet** kreiranje instance

**R:Entitet** uvid u instancu/klasu

**U:Entitet** izmena instance

**D:Entitet** uništavanje instance

Moguće su i kombinacije više odnosa.