

2 Objektno orijentisano projektovanje

1

Metodologija:

- Način na koji je nešto urađeno (t.j. strategija, koraci, smernice ili akcije).
- Metodologije mogu biti
 - kupljene,
 - kreirane,
 - kombinacija obe prethodne.
- Postoje hiljade metodologija za razvoj informacionih sistema.

2

Pregled metodologija:

- *Klasifikacija metodologija:*
 - *Tradicionalna*
 - *Strukturna analiza i projektovanje*
 - *Modelovanje/obrada informacija*
 - *Objektna-orijentacija*

3

Pregled metodologija:

- *Prototipovanje je tehnika (mada ga neki ubrajaju u metodologiju)*

4

Tradicionalna metodologija: (od 1950-tih do danas)

- **Primenljiva kod malih timova na malim projektima**
- **Funkcionalna perspektiva oblasti problema**
- **To je informativna, nestrukturirana, neponovljiva, nemerljiva, ad-hoc metodologija**
- **Alati za podršku su adekvatni.**

5

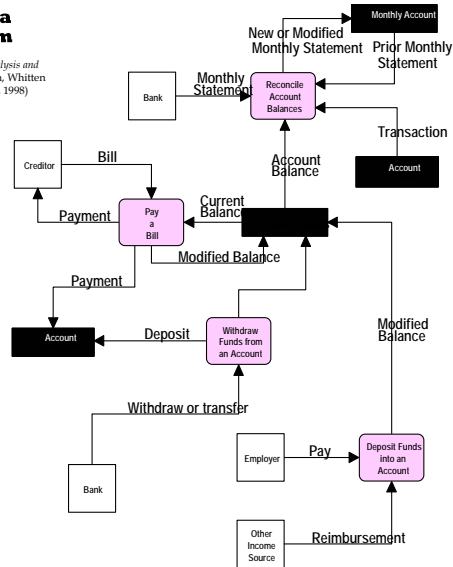
Metodologija strukturne analize i projektovanja (od sredine 1970-tih do sada):

- **Sinonim: metodologija toka podataka**
- **Zahvaljujući strukturnom programiranju**
- **Vrlo popularna – možda vodeća u biznisu**
- **Može biti ponovljiva, merljiva i automatizovana**
- **IDE & CASE alati predstavljaju značajnu pomoć**
- **Funkcionalna perspektiva oblasti problema**
- **Stvarni svet opisuje kao tok podataka kroz informacijski sistem sa transformacijom između ulaza i izlaza**

6

A Simple Data Flow Diagram

(adapted from *Systems Analysis and Design Methods*, 4th Edition, Whitten and Bentley, McGraw-Hill, 1998)



7

Metodologija modelovanja informacija (od ranih 1980-tih do sada) :

- **Modelovanje podataka & konstruisanje informacija**
- **Opisuje stvarni svet pomoću podataka, atributa podataka i odnosa između podataka**
- **Može biti ponovljiva, merljiva i automatizovana**
- **Perspektiva podataka o domenu problema**

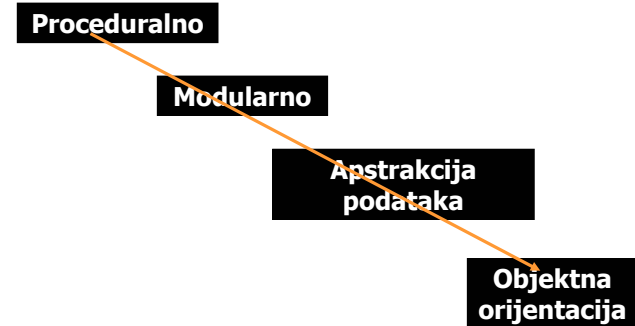
8

Objektno orijentisana metodologija (od kasnih 1980-tih do sada):

- Objektno modelovanje
- Zahvaljujući objektno orijentisanom programiranju
- Može biti ponovljiva, merljiva i automatizovana
- Objektna perspektiva oblasti problema
- Opisuje stvarni svet pomoću objekata, atributa, operacija i odnosa
- Podaci i funkcije su inkapsulirani zajedno

9

Put do OO projektovanja:



10

Objektna orijentacija:

■ Definicija:

Objektna orijentacija je posmatranje i modelovanje sveta (ili sistema) kao skupa objekata koji su međusobno povezani i utiču jedan na drugi.

11

Koncepti objektno orijentacije:

- Opšte metode organizacije*
- Apstrakcija*
- Inkapsulacija*
(skrivanje informacija)
- Nasleđivanje*
- Polimorfizam*
- Komunikacija porukama*
- Udruživanje (spajanje)*
- Ponovna upotreba (Reuse)*

12

Koncepti objektne orijentacije :

- a) **Opšte metode organizacije**
- b) Apstrakcija
- c) Inkapsulacija
(skrivanje informacija)
- d) Nasleđivanje
- e) Polimorfizam
- f) Komunikacija porukama
- g) Udruživanje (spajanje)
- h) Ponovna upotreba (Reuse)

13

a) Opšte metode organizacije:

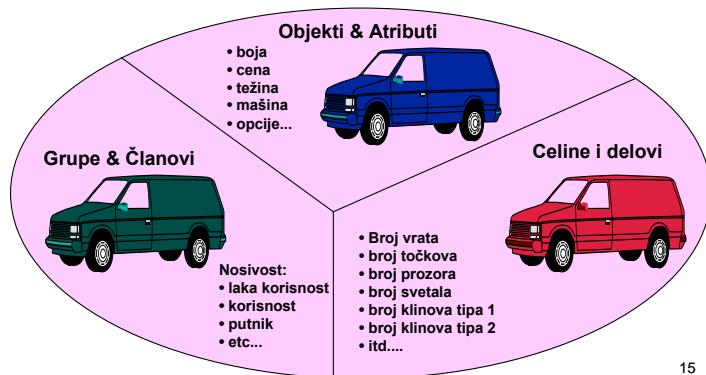
(Teorija klasifikacija)

- **Objekti i njihove karakteristike**
- **Celine i delovi**
- **Grupe (klase) i članovi**

14

Opšte metode organizacije:

Ljudi su navikli da razmišljaju pomoću ...



15

Objekti (1):

Coad i Yourdan (1990):

Objekat – apstrakcija nečega u domenu projekta što pokazuje osobine sistema da čuva informaciju o njemu, ima interakciju sa njim, ili oba.

16

Objekti (2):

Rumaugh et al. (1991):

Objekat je koncept, apstrakcija, ili stvar sa jasnim granicama i značenjem za aktuelni problem. Objekti imaju dve svrhe: oni unapređuju razumevanje stvarnog sveta i obezbeđuju osnovu za računarsku implementaciju.

17

Objekti (3):

Booch (1994):

Objekat ima

stanje (posebni uslovi u kojima se trenutno nalazi objekat),

ponašanje (šta objekat može da uradi), i

identitet (svaki objekat je jedinstven).

18

Objekti (4):

Wirfs-Brock et al:

Ako bi objekat bio osoba, uloga u sistemu bila bi zasnovana na odgovorima na tri pitanja:

- Ko sam ja?
- Šta mogu da radim?
- Šta znam?

19

Objekti:

Dakle,

objekat je koncept, apstrakcija, ili stvar sa granicama i značenjem **za problem koji se upravo razmatra**. Objekti imaju dva cilja: da unaprede razumevanje stvarnog sveta i obezbede praktičnu osnovu za računarsku implementaciju.

20

Objekti (5):

Dva aspekta: informacija i ponašanje

■ *Informacija:*

- 1) ima jedinstveni identitet
- 2) ima opis svoje strukture ili informacije koji se koristi za njegovo kreiranje
- 3) ima stanje koje predstavlja njegove tekuće uslove, t.j. vrednosti nekih njegovih svojstava

■ *Ponašanje:*

- 1) šta objekat može da radi?
- 2) šta može biti urađeno objektu?

21

Objekti:

- Objekti međusobno komuniciraju porukama.
- Objektna orijentacija uvela je razdvajanje podataka u sistemu od procesa koji rade na podacima.
- Svaki proces smešta se sa podacima koje koristi.
- Objekti 'znaju' svoje podatke i kako da zahtevaju usluge od drugih objekata (od kojih objekata da zahtevaju i format zahteva).

22

Primer objekta – štampač:

1) *informacija:*

- a) serijski broj
- b) model
- c) brzina
- d) memorija
- e) stanje

2) *ponašanje:*

- a) štampa fajla
- b) stop štampa
- c) ukloni fajl iz reda čekanja



23

Klase i uzorci:

Objekat predstavlja poseban *uzorak klase*.

Objekti koji su međusobno dovoljno slični pripadaju istoj *klasi*.

Svaki objekat je *uzorak* neke klase.

24

Šta je klasa?

Klasa je opis skupa objekata koji imaju iste attribute, operacije, metode, odnose i semantiku.

Zapravo, zadatak klase je da deklariše skup metoda, operacija i atributa koji potpuno opisuju strukturu i ponašanje objekata.

25

Šta je klasa?

- 1) **jedinstveno identifikovana apstrakcija skupa logički povezanih uzoraka koji imaju iste ili slične karakteristike;**
- 2) **pravila koja definišu objekte;**
- 3) **definicija ili pravilo koje opisuje kako napraviti tačnu predstavu specifičnog tipa objekata;**
- 4) **primeri: agencija, građani, automobil, ...**

Objekti se uzorkuju (kreiraju) koristeći definicije klase kao šablone.

26

Primer klase – štampač:

Štampač
Informacije
Ponašanje



27

Atributi:

- **Atribut je ime za svojstvo klase koje opisuje opseg vrednosti tog svojstva koje mogu da imaju uzorci.**
- **Atribut ima tip i definiše tip njegovog uzorka.**
- **Samo sam objekat može da menja vrednosti njegovih atributa.**
- **Dati skup vrednosti atributa definiše stanje objekta.**

28

Koncepti objektne orijentacije:

- a) Opšte metode organizacije
- b) **Apstrakcija**
- c) Inkapsulacija
(skrivanje informacija)
- d) Nasleđivanje
- e) Polimorfizam
- f) Komunikacija porukama
- g) Udruživanje (spajanje)
- h) Ponovna upotreba (Reuse)

29

b) Apstrakcija:

- Mentalna sposobnost koja omogućava ljudima da vide probleme stvarnog sveta sa različitim stepenima detalja u zavisnosti od tekućeg konteksta problema.
- Pomaže ljudima da misle o onome što rade.
- Funkcionalna apstrakcija i apstrakcija podataka

30

Apstrakcija:

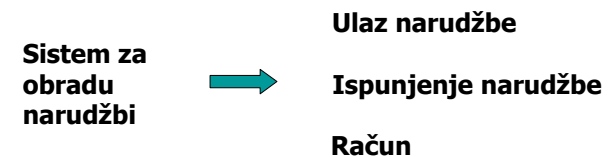
Model uključuje najvažnije aspekte datog sistema, a ignoriše manje važne detalje.

Apstrakcija dopušta upravljanje kompleksnošću koncentrišući se na suštinske karakteristike koje neki entitet razlikuju od drugih.

31

Modularnost:

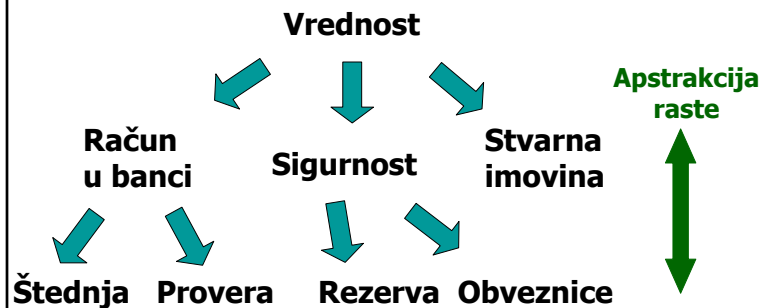
Modularnost se odnosi na proces razbijanja kompleksnog sistema na manje delove koje sadrži, a kojima se može lako upravljati.



32

Hijerarhija:

Uređivanje apstrakcija u strukturu stabla.



33

Koncepti objektne orijentacije:

- Opšte metode organizacije
- Apstrakcija
- Inkapsulacija**
(skrivanje informacija)
- Nasleđivanje
- Polimorfizam
- Komunikacija porukama
- Udruživanje (spajanje)
- Ponovna upotreba (Reuse)

34

c) Inkapsulacija (skrivanje informacija):



35

Prednosti inkapsulacije:

- Inkapsulacija odvaja implementaciju od korisnika ili klijenata, tako da jedan interfejs može imati različite implementacije.
- Tehnika u kojoj se podaci pakuju zajedno sa odgovarajućim procedurama.
- Podaci čuvani u jednom objektu ne mogu se pokvariti od strane drugog objekta.

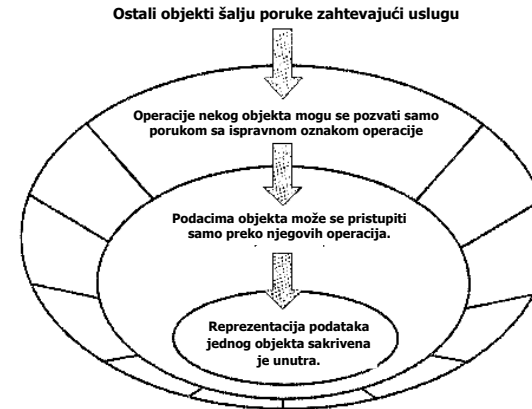
36

Zahtevi inkapsulacije:

- 1) da izloži cilj objekta
- 2) da izloži interfejs objekta
- 3) da sakrije implementaciju koja obezbeđuje ponašanje kroz interfejs
- 4) da unutar objekta sakrije podatke koji definišu njegovu strukturu i podržavaju njegovo ponašanje
- 5) da sakrije podatke koji prate njegovop stanje sakrije unutar objekta

37

Inkapsulacija:



38

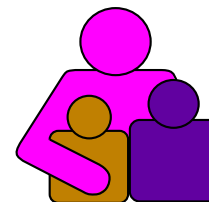
Koncepti objektne orijentacije:

- a) Opšte metode organizacije
- b) Apstrakcija
- c) Inkapsulacija
(skrivanje informacija)
- d) **Nasleđivanje**
- e) Polimorfizam
- f) Komunikacija porukama
- g) Udruživanje (spajanje)
- h) Ponovna upotreba (Reuse)

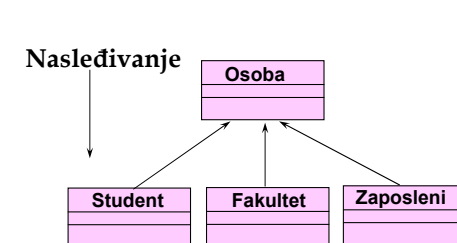
39

d) Nasleđivanje:

Mehanizam za izražavanje sličnosti između stvari, dakle za pojednostavljivanje njihove definicije.



- izgled
- ponašanje
- stavovi
- itd...



40

Nasleđivanje#1:

Nasleđivanje je mehanizam za implementiranje generalizacije i specijalizacije u OO jezicima. Kada su dve klase povezane ovim odnosom, opštija klasa naziva se **superklasa**, a specijalizovanija klasa naziva se **potklasa**.

41

Nasleđivanje#2:

Pravila OO nasleđivanja:

- 1) Potklasa nasleđuje sve karakteristike njene superklase.
- 2) Definicija potklase uvek uključuje najmanje jedan detalj koji nije proistekao iz superklase.

42

Nasleđivanje#3:

Operacija tranzitivnosti:

- 1) Potklasa nasleđuje sve karakteristike svih njenih superklasa.
- 2) Definicija potklase uvek uključuje najmanje jedan detalj koji nije proistekao ni iz jedne od njenih superklasa.


43


Koncepti objektne orijentacije:

- a) Opšte metode organizacije
- b) Apstrakcija
- c) Inkapsulacija
(skrivanje informacija)
- d) Nasleđivanje
- e) **Polimorfizam**
- f) Komunikacija porukama
- g) Udruživanje (spajanje)
- h) Ponovna upotreba (Reuse)

44

e) Polimorfizam:

 Mogućost da se sakriju različite implementacije iza zajedničkog interfejsa.

 Mogućnost da dva ili više objekata odgovore na isti zahtev, svaki na svoj način.

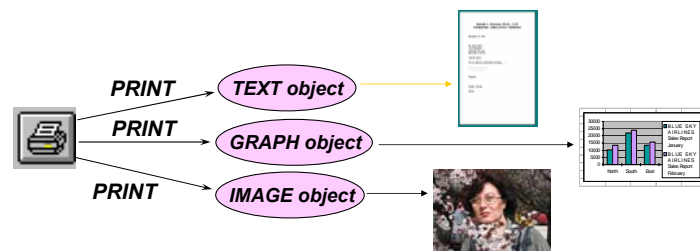
45

Polimorfizam:

- H₂O = voda, led, para (tečnost, čvrsto, magla)
- Jelo
- Ugljenik kristališe kao grafit & dijamant

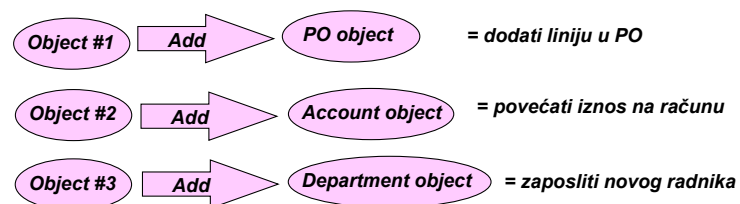
46

Polimorfizam – primer1:



47

Polimorfizam – primer2:



48

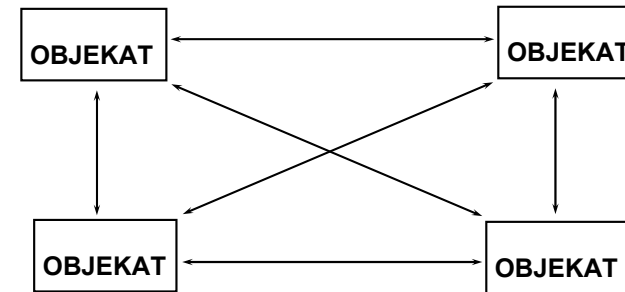
Koncepti objektne orijentacije:

- a) Opšte metode organizacije
- b) Apstrakcija
- c) Inkapsulacija
(skrivanje informacija)
- d) Nasleđivanje
- e) Polimorfizam
- f) **Komunikacija porukama**
- g) Udruživanje (spajanje)
- h) Ponovna upotreba (Reuse)

49

f) Komunikacija porukama:

Objekti komuniciraju porukama.



50

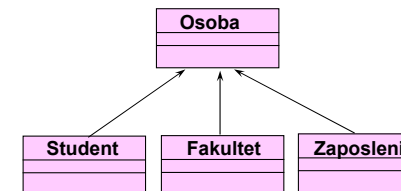
Koncepti objektne orijentacije:

- a) Opšte metode organizacije
- b) Apstrakcija
- c) Inkapsulacija
(skrivanje informacija)
- d) Nasleđivanje
- e) Polimorfizam
- f) Komunikacija porukama
- g) **Udruživanje (spajanje)**
- h) Ponovna upotreba (Reuse)

51

g) Udruživanje (spajanje):

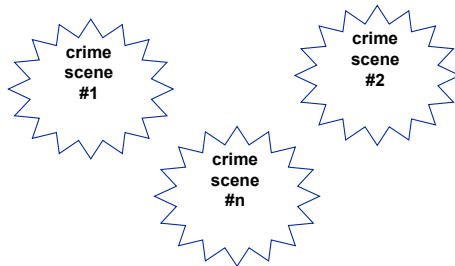
Unija ili veza ideja ili stvari.
(Objekti teba da imaju međusobnu interakciju)



52

Udruživanje (spajanje):

pod istim okolnostima:



53

Koncepti objektne orijentacije:

- a) Opšte metode organizacije
- b) Apstrakcija
- c) Inkapsulacija
(skrivanje informacija)
- d) Nasleđivanje
- e) Polimorfizam
- f) Komunikacija porukama
- g) Udruživanje (spajanje)
- h) **Ponovna upotreba (Reuse)**

54

h) Ponovna upotreba (reuse):

Mogućnost ponovne upotrebe objekata.

Različiti stepeni ponovne upotrebe:



- **kompletna ili zajednička**
- **kopiranje, kupovina ili kloniranje**
- **delimična ili podešavanje**
- **ništa**

Software:

- "Komadići"
- Komponente
- Kontrola
- Modeli

55

Ponovna upotreba:

- Komponente moraju biti ponovo korišćene tri do pet puta **da bi se pokrila cena kreiranja i podrške.**
- Kreiranje i podrška komponente za samo jednu ponovnu upotrebu košta **jedan i po do tri puta više od kreiranja komponente za samo jednu upotrebu.**
- Ponovno korišćenje komponente **košta 25% cene kreiranja nove.**
- Potrebna su tri proizvodna ciklusa (oko tri godine) **pre nego korist od ponovne upotrebe postane značajna.**

56

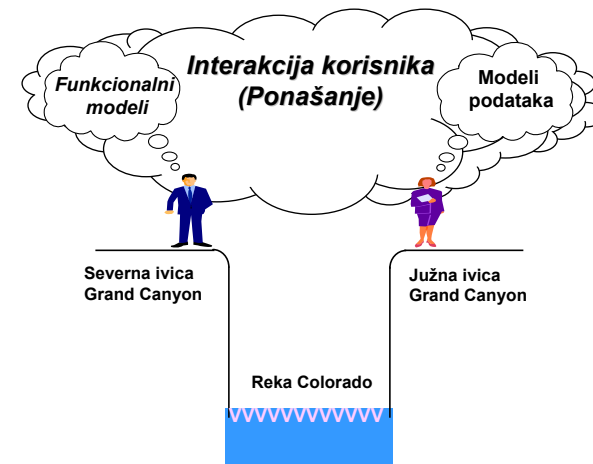
OO Metodologija analize i projektovanja sistema:

Tri klasična izazova u analizi i projektovanju sistema :

- ❖ Model podataka nasuprot funkcionalnom modelu
- ❖ Prelazak sa analize na projektovanje
- ❖ Podržavanje izvornog koda

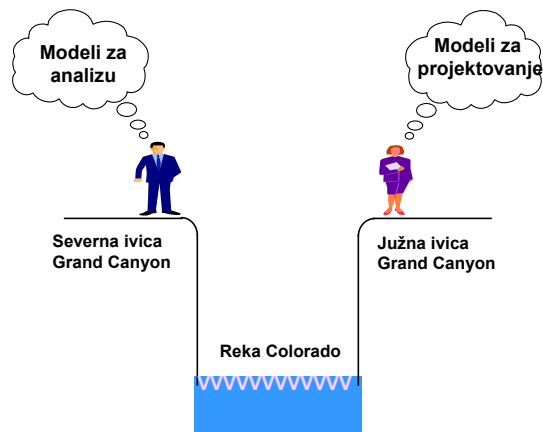
57

Klasični izazov#1 – Različiti modeli:



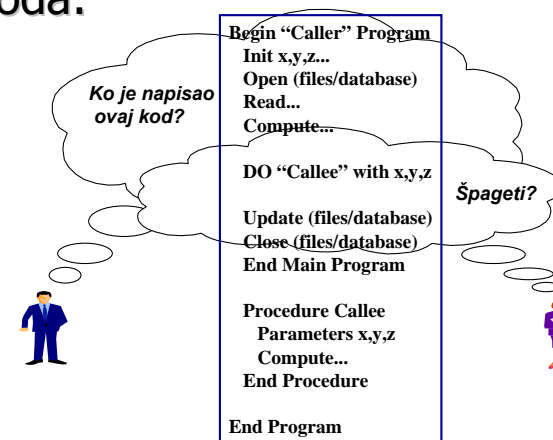
58

Klasični izazov#2 – Transformacija modela:



59

Klasični izazov #3 – Podržavanje koda:



60

Rešenje:

INTEGRISANI MODEL(I)
(funkcija, podaci, ponašanje)
(analiza, projektovanje i
implementacija)

**Kružno putovanje
ENGINEERING**

