APLIKATIVNI SOFTVER

Razvoj - UML - Slojevi - OOP

dr Miloš Dobrojević

školska 2013/14. godina

Sadržaj

- Aplikativni softver
  - Uvod
  - Tipovi
  - Podela
- Razvoj
  - Tehnologije
  - Zdrav razum
  - Rizik
  - Faze razvoja
  - Modelovanje procesa razvoja
  - Klasične metode
  - Agilne metode
- UML Modelovanje
  - Uvod
  - Dijagrami
- Slojevi
  - Logički
  - Fizički
  - Prednosti i mane
- OOP
  - Uvod
  - Pravila formiranja klasa

Aplikativni softver

Uvod

Aplikativni softver

Uvod

- To je softver pomoću koga kompjuter izvršava korisne, tačno određene zadatke ili grupe zadataka
  - ***Obrada teksta*** MS Word, Open Office Writer, ...
  - ***Obrada slika*** Photoshop, Gimp, ...
  - ***Obrada zvuka, videa*** Adobe Premiere, ...
  - ***Reprodukcija zvuka, videa*** Winamp, VLC, ...
  - ***Grupe zadataka*** MS Office, Open Office, ...
  - ...
- Napisani su pomoću jednog od mnoštva programskih jezika dostupnih za konkretan tip kompjutera i operativni sistem.
- Podela se može izvršiti na različite načine, prema tipu i nameni.
- Važna grupa koja je nastala popularizacijom Interneta su **Web aplikacije.**

Aplikativni softver

Tipovi

- ***Programski paketi.*** Više međusobno povezanih aplikacija. MS Office, iWork itd.
- ***Enterprise software.*** Pokriva organizacione potrebe i protok podataka u firmi. Knjigovodstvo, finansije, CRM, nabavka.
- ***Obrazovni softver.*** Pristup obrazovnom materijalu sa opcijama prilagođenim nastavnicima i učenicima. Testiranje, evaluacija, praćenje napretka, saradnja...
- ***Simulacioni softver.*** Simulacija stvarnih ili apstraktnih sistema, za potrebe istraživanja, obuke ili zabave.
- ***Media development software.*** Obrada slika, video i zvučnog materijala, HTML editori, animacija itd.
- ***Product engineering software.*** CAD/CAE, editori programskog jezika, kompajleri, IDE.

Aplikativni softver

Podela

Horizontalne	Vertikalne
Opšte namene	Fokusirane na određeni tip industrije ili biznisa.
Primer: obrada teksta, baze podataka, email, CMS	Primer: Osiguranje, nekretnine, zdravstvo, transport, državna uprava

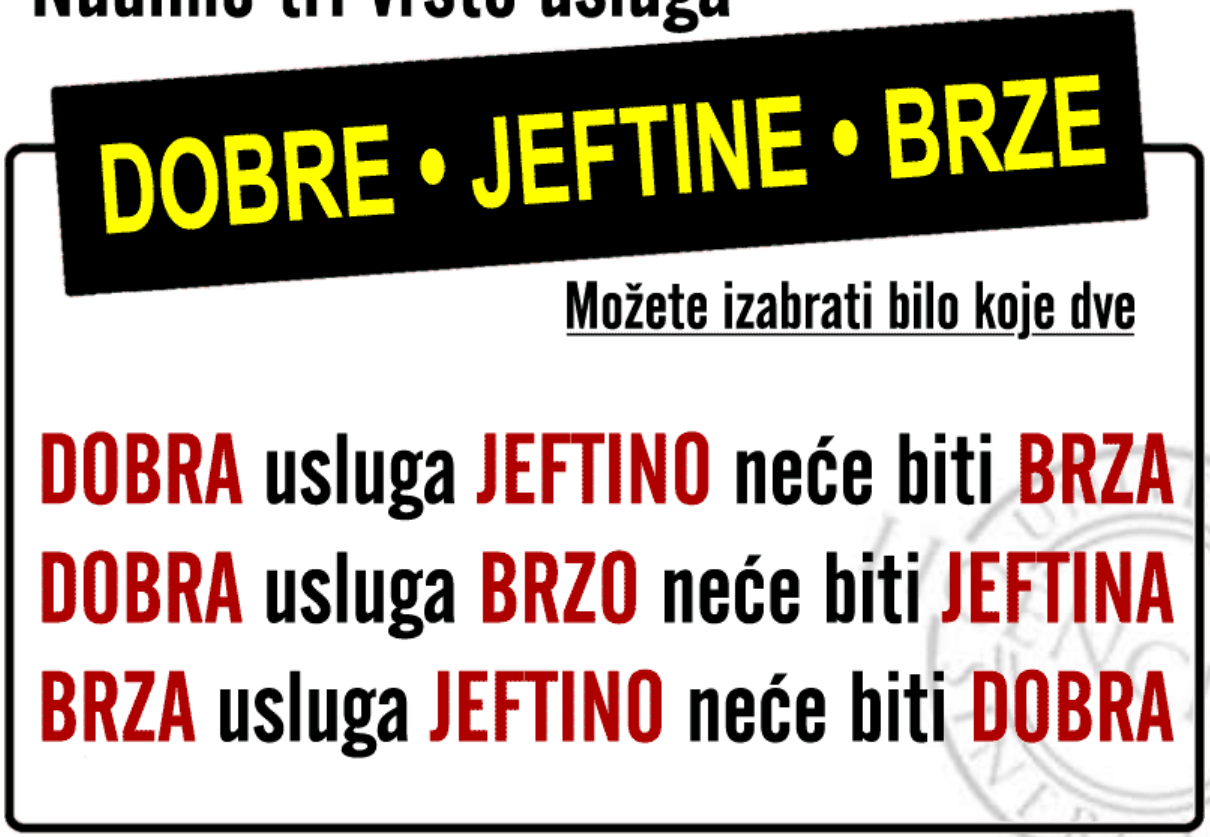
- Prema operativnom sistemu
  - Windows, Linux, Android, iOS, ...
- Prema kompjuterskoj platformi
  - Wintel (x86 procesor + Windows)
  - Mobilni uređaji. Android, Firefox OS, Palm, Symbian, iOS, Windows Mobile, ...
  - Macintosh (Apple + Mac OS)
  - Igračke konzole. Sony PS, Microsoft Xbox, Nintendo Wii, ...
  - Super kompjuteri

Aplikativni softver

Podela

- Prema softverskoj platformi
  - Adobe (AIR, Flash, Shockwave)
  - Java (Micro Ed, Standard Ed, Enterprise Ed, JavaFX, JavaFX Mobile)
  - .NET Framework
  - LAMP / WAMP
  - Silverlight
  - SAP NetWeaver
- Cross-platform aplikacije
  - Aplikacija koja funkcioniše na najmanje dve kompjuterske platforme, bilo da je u pitanju arhitektura hardvera ili operativni sistem.
  - Primer: **web aplikacije**

Razvoj


Nudimo tri vrste usluga
DOBRE • JEFTINE • BRZE
Možete izabrati bilo koje dve
DOBRA usluga JEFTINO neće biti BRZA
DOBRA usluga BRZO neće biti JEFTINA
BRZA usluga JEFTINO neće biti DOBRA

Razvoj

- Tehnologije i alati za razvoj aplikacija
  - Open Source / Licencne
  - Programski jezici
  - Frameworks
  - Source code editori
  - Kompajleri
  - IDE

Razvoj

- Zdrav razum (1)
  - Sagledavanje sopstvenih resursa i potencijala
    - Šta znam da radim, koliko mogu da uradim?
  - Projektni zadatak
    - Samostalni projekat, novi proizvod ili rad za klijenta?
  - Realizacija, upravljanje projektom
    - Sistematičan pristup. Rokovi, kvalitet, finansije?
  - Dobra komunikacija
    - Unutar samog tima i sa klijentom koji često nije tehnički obrazovan.
  - Kvalitet
    - Klijent i/ili korisnici zadovoljni.
    - **Zamka: over-engineering!**

Razvoj

- Zdrav razum (2)

Sopstveni razvoj	Rad za klijenta
Istraživanje tržišta, ciljna grupa	Razumevanje zadatka
Biznis plan, budžet	Rokovi
- Koliko će da me košta	Budžet
- Kolike će prihode da donese i kada?	Dobra komunikacija, consulting
- Životni vek? Održavanje?	Angažovani resursi
- Šta ću od resursa morati da angažujem?	Obuka, postprodaja

Taktika? Hit & Run nikako!

Razvoj

- Rizik
  - Poslovni model, biznis plan
  - Planirani rast kompanije
  - Resursi
    - Tehnički
    - Ljudski
    - Finansijski
    - Vremenski, time to market
  - Bezbednost
  - Bagovi

Razvoj

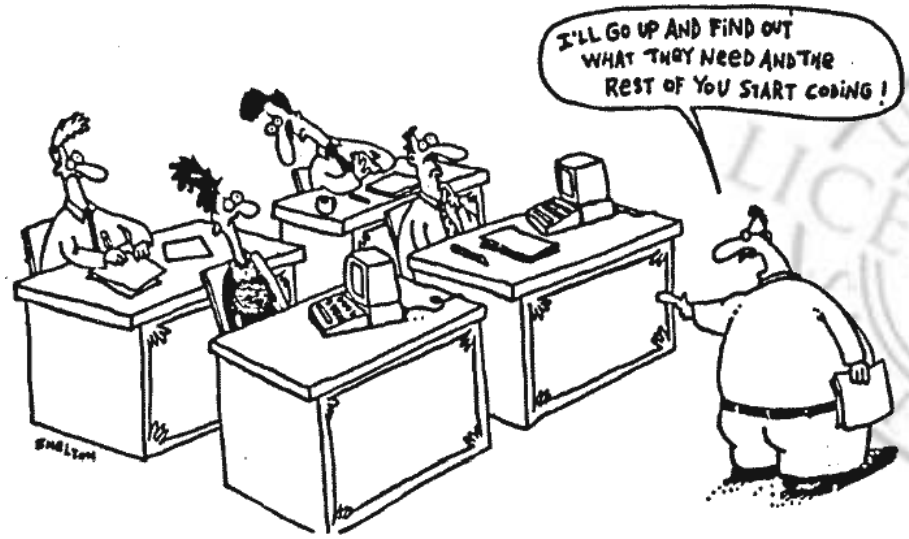
- Faze razvoja (1)
  - Analiza i definisanje zahteva
    - Saradnja sa kupcem i korisnikom
    - Analiza zahteva
    - Interakcija sistema sa okruženjem
    - Budžetiranje
    - Rezultat faze → lista korisničkih zahteva

Klijenti nikada ne znaju šta im treba...

...dok ne vide šta su dobili

Razvoj

- Faze razvoja (2)
  - Projektovanje sistema
    - Formiranje timova
    - Izrada projekta sistema prema listi korisničkih zahteva
    - Izbor platforme i arhitekture softvera, definisanje strukture baze podataka, komponente, preliminarna optimizacija, algoritmi, korisnički interfejs, dizajn
    - Rapid prototyping



Razvoj

- Faze razvoja (3)
  - Projektovanje programa
    - Formiranje modula (podprojekata)
    - Definisanje veze između modula
    - Metodologija razmene podataka između modula
  - Realizacija
    - Izrada programskog koda prema projektnom zadatku
  - Testiranje (TDD), debug
    - Jedinično — nezavisno testiranje svake programske komponente
    - Integraciono — da li sistemske komponente sarađuju prema specifikaciji?
    - Funkcionalno — da li sistem izvršava funkcije prema specifikaciji?
    - Performanse — poređenje sa hardverskim i softverskim zahtevima u realnom radnom okruženju

Razvoj

- Faze razvoja (3)
  - Implementacija
    - Instalacija aplikacije u radno okruženje
  - Postprodaja, održavanje
    - Ispravljanje grešaka
    - Modifikacije i unapređenje. Funkcionalnost, dizajn, korisnički interfejs

Razvoj

- Modelovanje procesa razvoja
  - Opis sistema postaje zajedničko shvatanje svih učesnika → lakša komunikacija, manje nesporazuma i pogrešnih tumačenja
  - Odražava ciljeve razvoja. Pre implementacije se procenjuje usklađenost predviđenih aktivnosti sa postavljenim zahtevima
  - Pomaže u nalaženju nedoslednosti, suvišnih ili izostavljenih elemenata → bolja efikasnost
  - Iako je urađen za potrebe konkretnog projekta, model se može primeniti i u drugim situacijama. Sačuvati ga!

Razvoj

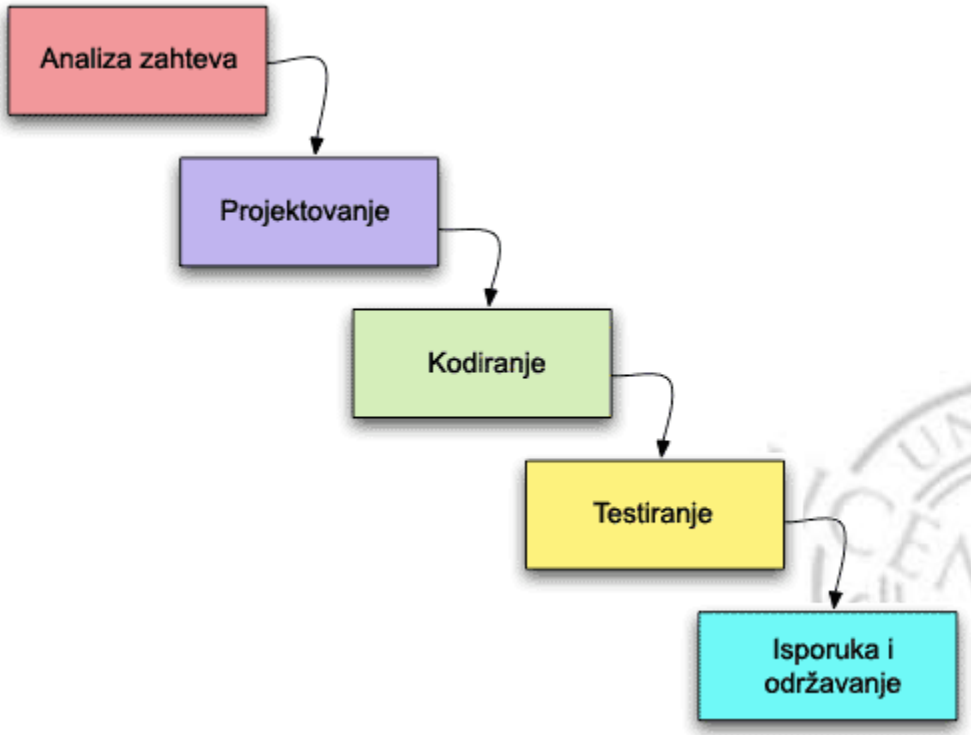
Klasične metode modelovanja

- Klasične metode
  - Kaskadni model (Royce, 1970)
  - V model (Nemačka vlada, 1992)
  - Fazni razvoj (inkrementalni i iterativni)
  - Prototipski model
  - Transformacioni model
  - Spiralni model
  - RUP

Razvoj

Klasične metode modelovanja

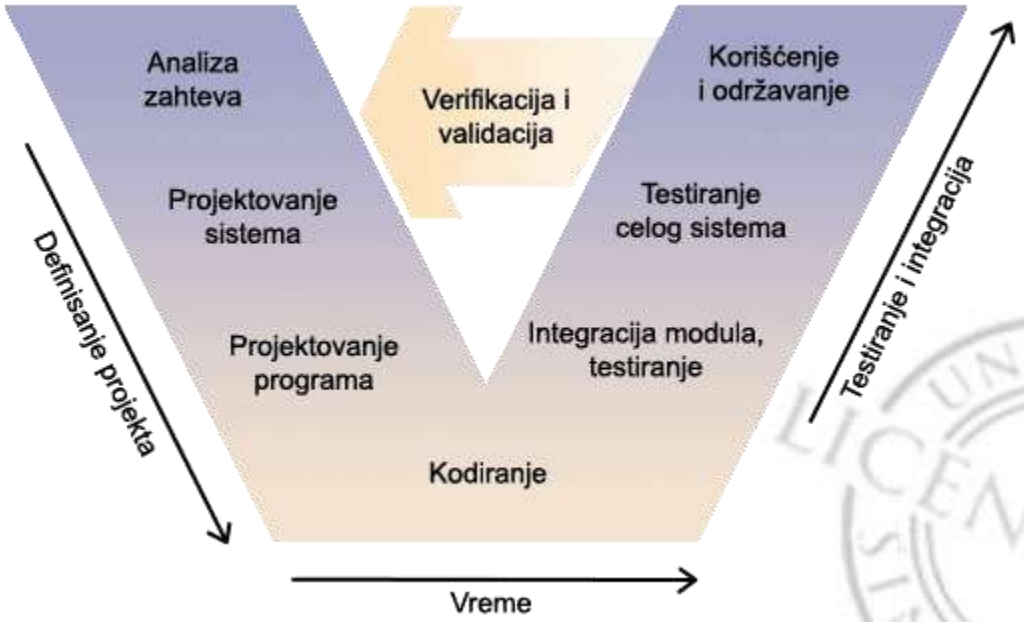
- Kaskadni model



Razvoj

Klasične metode modelovanja

- V model



Razvoj

Klasične metode modelovanja

- Metode pristupa problemu
  - Top-down

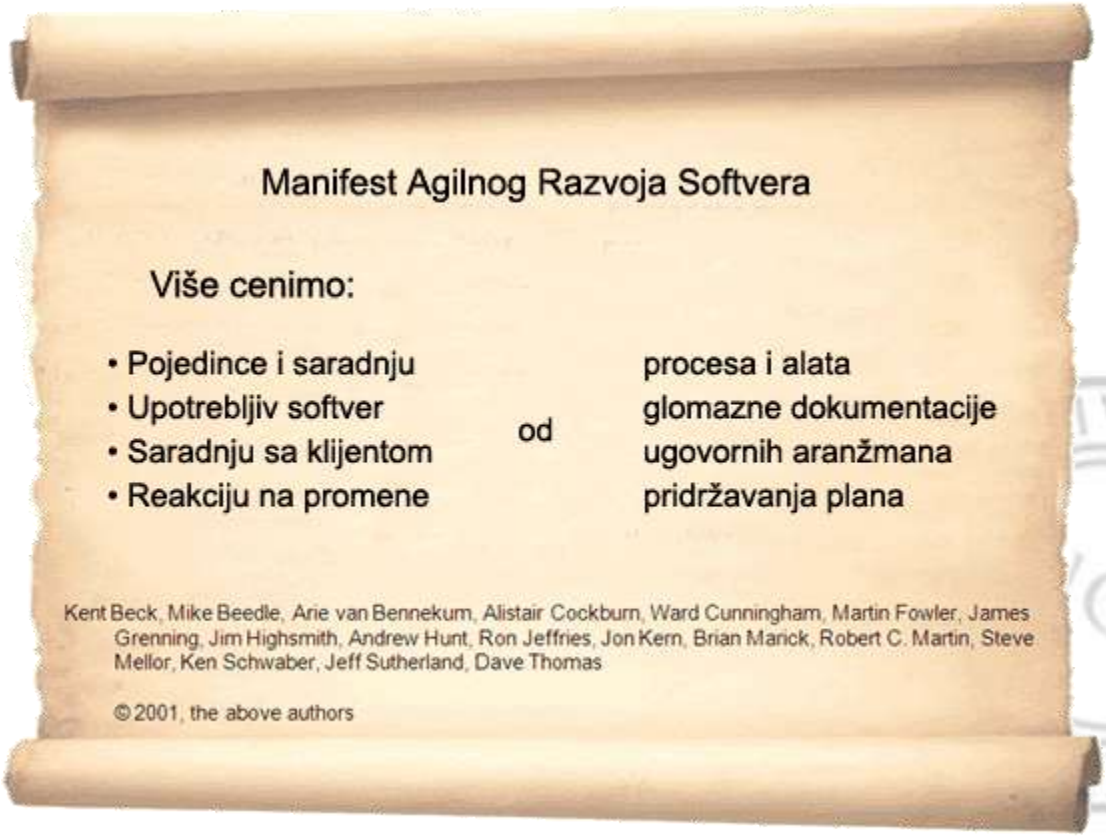
    Iterativno razbijanje sistema na podsisteme, sve dok cela specifikacija nije svedena na osnovne elementa.
  - Bottom-up

    Slaganje kockica. Integracija sistema u veći sistem. Prvi korak je detaljna specifikacija osnovnih elemenata koji se povezuju u podsisteme. Proces se ponavlja sve dok se ne formira top-level sistem.

Razvoj

Agilne metode

- Agilne metode su bazirane na brzini i spretnosti, a nastale su kao posledica otpora tradicionalnim metodama modelovanja.



Manifest Agilnog Razvoja Softvera

Više cenimo:

• Pojedince i saradnju		procesa i alata
• Upotrebljiv softver	od	glomazne dokumentacije
• Saradnju sa klijentom		ugovornih aranžmana
• Reakciju na promene		pridržavanja plana

Kent Beck, Mike Beedle, Arie van Bennekum, Alistair Cockburn, Ward Cunningham, Martin Fowler, James Grenning, Jim Highsmith, Andrew Hunt, Ron Jeffries, Jon Kern, Brian Marick, Robert C. Martin, Steve Mellor, Ken Schwaber, Jeff Sutherland, Dave Thomas

Razvoj

Agilne metode

- Principi (1)
  - Zadovoljan klijent. Brza isporuka upotrebljvog softvera.
  - Zahtevi za izmenu su dobrodošli. Čak i u poodmaklim fazama razvoja.
  - Upotrebljiv softver se brzo isporučuje. Kao vremenska jedinica, nedelju dana je prihvatljivije nego mesec dana.
  - Upotrebljiv softver je glavno merilo napretka.
  - Tempo razvoja treba da bude održiv i konstantan.
  - Dnevna saradnja na relaciji menadžment-razvojni tim.
  - Licem-u-lice je najbolji vid komunikacije.
  - Osnova projekta su motivisani pojedinci kojima se može verovati.

Razvoj

Agilne metode

- Principi (2)
  - Neprekidno voditi računa o kvalitetu.
  - Jednostavnost.
  - Samo-organizujući timovi.
  - Fleksibilnost. Prilagođavanje novonastalim okolnostima.

Razvoj

Agilne metode

- Metodologija
  - Akcenat je na razbijanju zadatka na manje korake i bez detaljnog ili dugoročnog planiranja.
  - Iteracije se vrše u kratkim vremenskim intervalima u trajanju od jedne nedelje do mesec dana.
  - Za svaku iteraciju je zadužen višefunkcionalan tim koji obavlja zadatke planiranja, analize zahteva, projektovanja, kodiranja i testiranja. Na kraju iteracije, upotrebljiv proizvod se predstavlja klijentu.
  - Svakodnevni kratki jutarnji sastanci, gde članovi tima jedni druge obaveštavaju o obavljenom poslu prethodnog dana, šta planiraju za danas i na koje su eventualno probleme naišli.
  - Time se rizik svodi na minimum, a ceo projekat je fleksibilan i lako prilagodljiv eventualnim izmenama.

Razvoj

Agilne metode

- Tehnike
  - Ekstremno programiranje (XP)

    Skup tehnika kojima se naglašava kreativnost timskog rada uz minimalno administriranje
  - Scrum

    Propisuje načine upravljanja zahtevima, iteracijama razvoja, implementacijom i isporukom
  - Timeboxing

    Određivanje vremenskog perioda za svaku aktivnost
  - Feature-driven development

    Inkrementalni i iterativni proces razvoja

Razvoj

- Pročitati (1)
  - Razvoj aplikativnog softvera, V. Tomašević, Singidunum 2012.
    Poglavlje 2 → Proces razvoja softvera
  - Wikipedia

    Klasične metode razvoja
    - http://en.wikipedia.org/wiki/Software_development_methodology
    - http://en.wikipedia.org/wiki/Waterfall_model
    - http://en.wikipedia.org/wiki/V-Model_(software_development)
    - http://en.wikipedia.org/wiki/Software_prototyping
    - http://en.wikipedia.org/wiki/Iterative_and_incremental_development
    - http://en.wikipedia.org/wiki/Spiral_model

Razvoj

- Pročitati (2)
  - Wikipedia

    Agilni razvoj
    - http://en.wikipedia.org/wiki/Agile_software_development
    - http://en.wikipedia.org/wiki/Extreme_programming
    - http://en.wikipedia.org/wiki/Scrum_(software_development)
    - http://en.wikipedia.org/wiki/Timeboxing
    - http://en.wikipedia.org/wiki/Feature-driven_development

UML Modelovanje

- UML (Unified Modeling Language)
  - Opšteprihvaćeni jezik za modelovanje. Predstavlja skup grafičkih notacija zasnovanih na jedinstvenom metamodelu.
    - ***Grafička notacija*** - skup grafičkih elemenata koji definišu sintaksu programskog jezika
    - ***Metamodel*** - dijagram koji opisuje koncepte jezika za modelovanje.
  - Grafičke notacije i metamodel se ne moraju strogo primenjivati, već predstavljaju samo pokušaj uvođenja discipline.
  - Korisnost modela je od primarnog značaja.

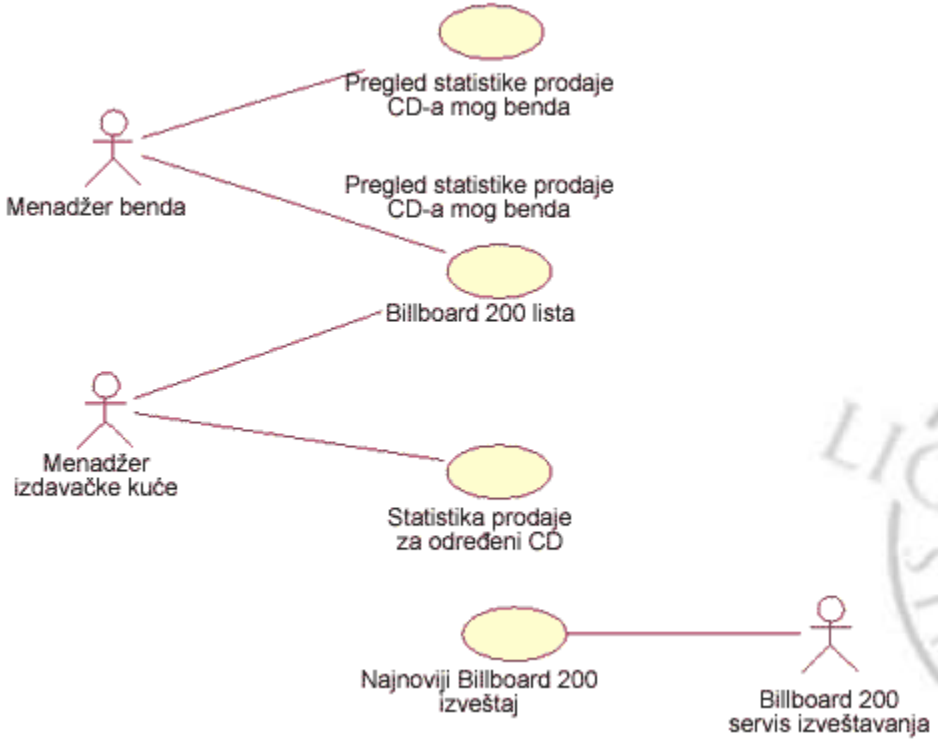
UML Modelovanje

- Dijagrami
  - Slučajevi korišćenja
    Opis postupaka interakcije između korisnika i sistema
  - Klase
    Tipovi entiteta u sistemu, vrste statičkih veza između entiteta, ograničenja u načinu njihovog povezivanja
  - Sekvence
    Interakcija između objekata po redosledu dešavanja
  - Aktivnosti
    Tok kontrole između dve ili više sistemskih komponenti.
  - Komponente
    Fizička organizacija softverskih komponenti u sistemu i zavisnosti između njih
  - Raspoređivanje
    Fizička organizacija sistema, hardverska i softverska arhitektura

UML Modelovanje

- Dijagram slučaja korišćenja

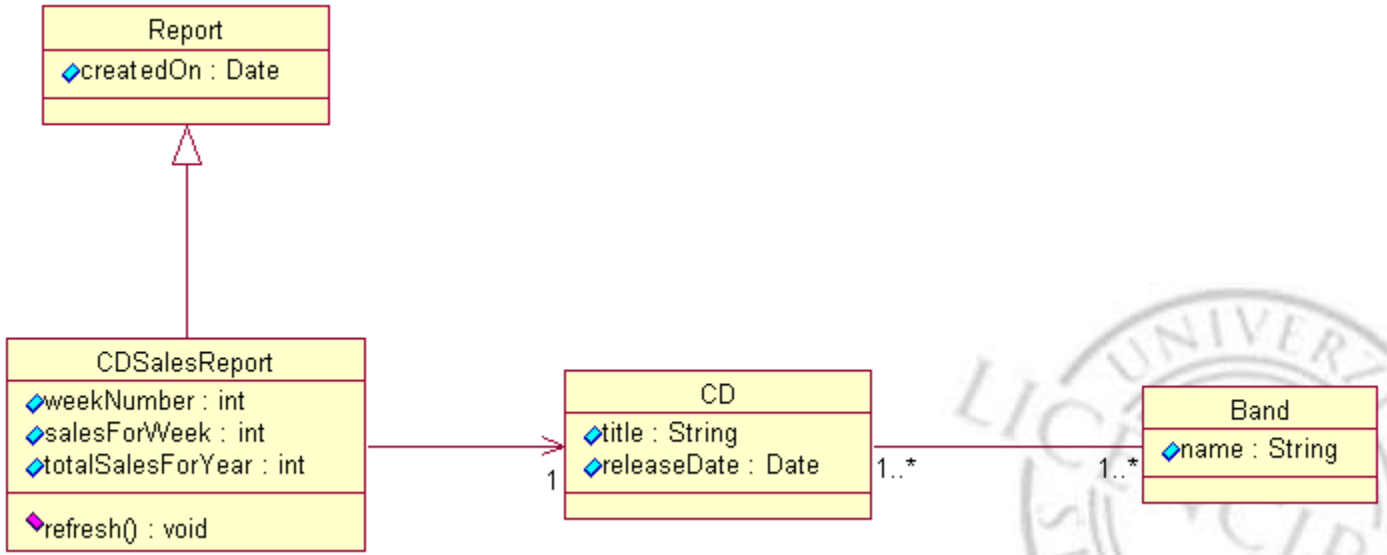
  http://www.ibm.com/developerworks/rational/library/769.html



UML Modelovanje

- Dijagram klasa

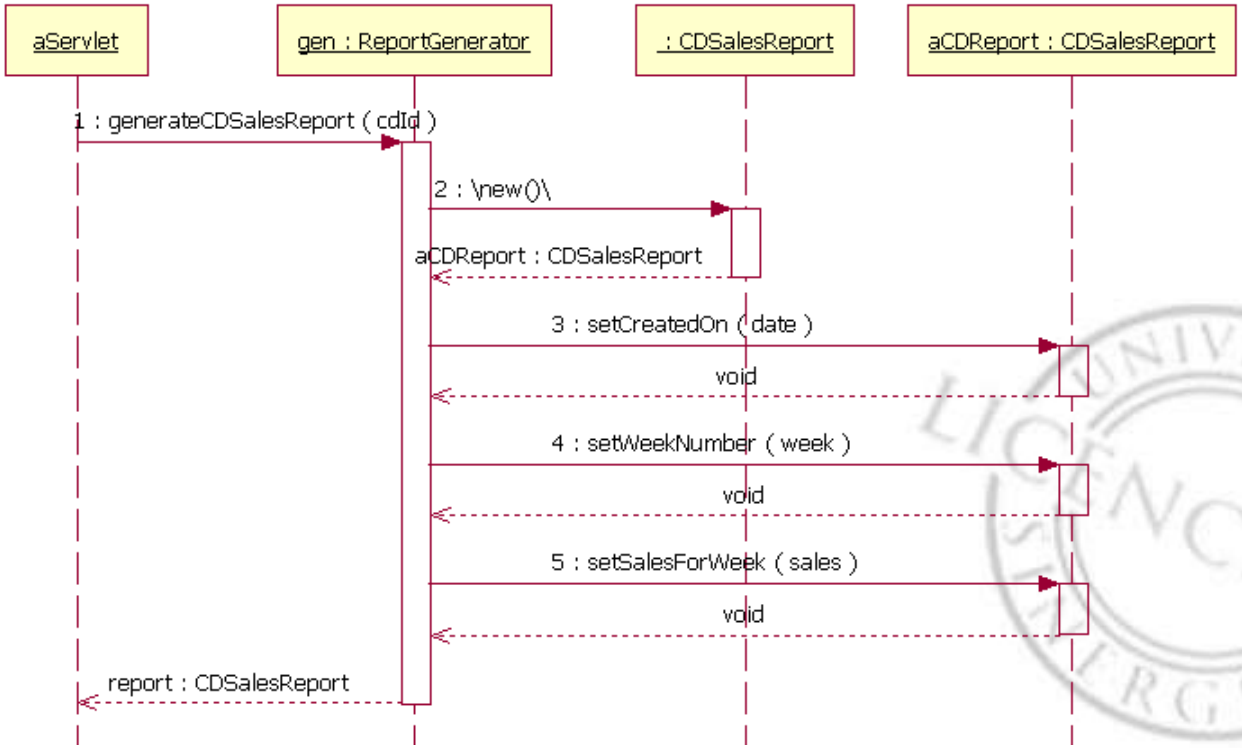
  http://www.ibm.com/developerworks/rational/library/769a.html



UML Modelovanje

- Dijagram sekvenci

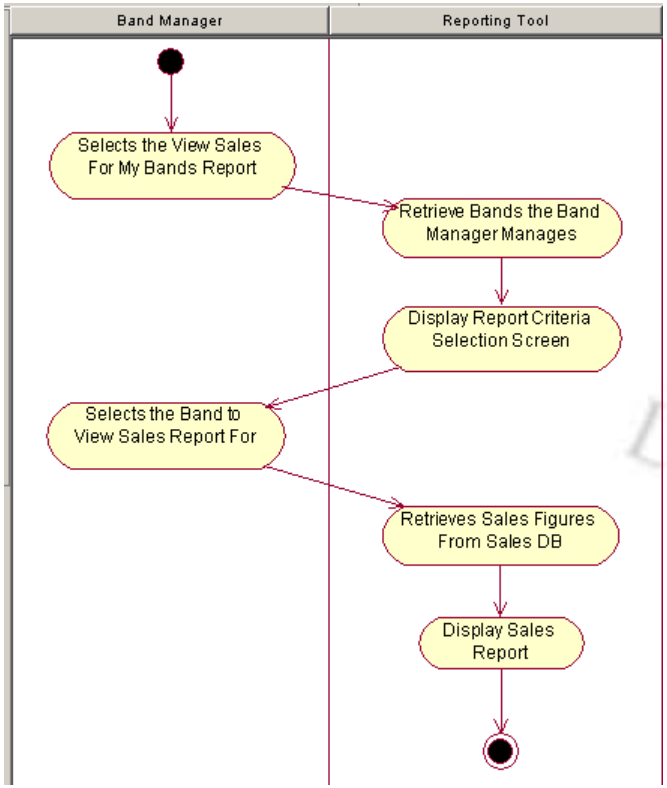
  http://www.ibm.com/developerworks/rational/library/769b.html



UML Modelovanje

- Dijagram aktivnosti

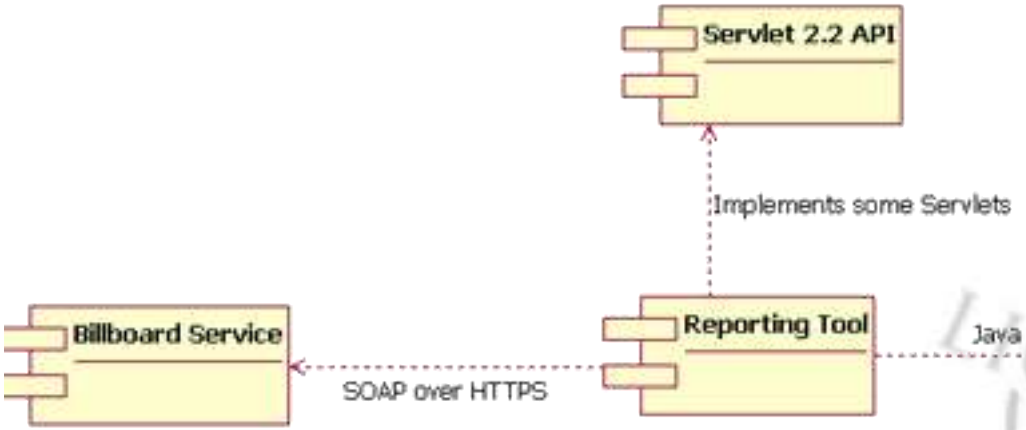
  http://www.ibm.com/developerworks/rational/library/769.html



UML Modelovanje

- Dijagram komponenti

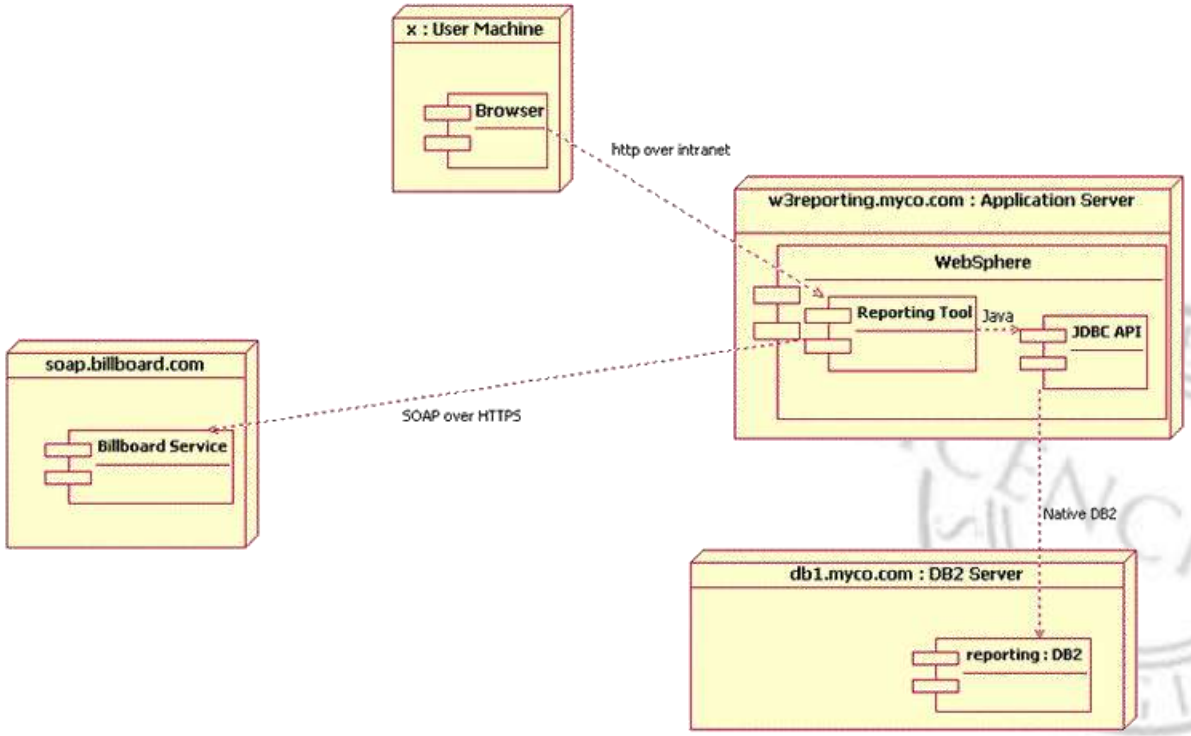
  http://www.ibm.com/developerworks/rational/library/769.html



UML Modelovanje

- Dijagram raspoređivanja

  http://www.ibm.com/developerworks/rational/library/769d.html



UML Modelovanje

- Pročitati
  - Razvoj aplikativnog softvera, V. Tomašević, Singidunum 2012.
    Poglavlje 5 → UML Modelovanje
  - Web
    - http://ibm.com/developerworks/rational/library/769.html
    - http://ibm.com/developerworks/rational/library/3101.html
    - http://en.wikipedia.org/wiki/Unified_Modeling_Language
    - http://giuliozambon.blogspot.com/2010/09/oo-uml-behavior-diagrams.html

Slojevi

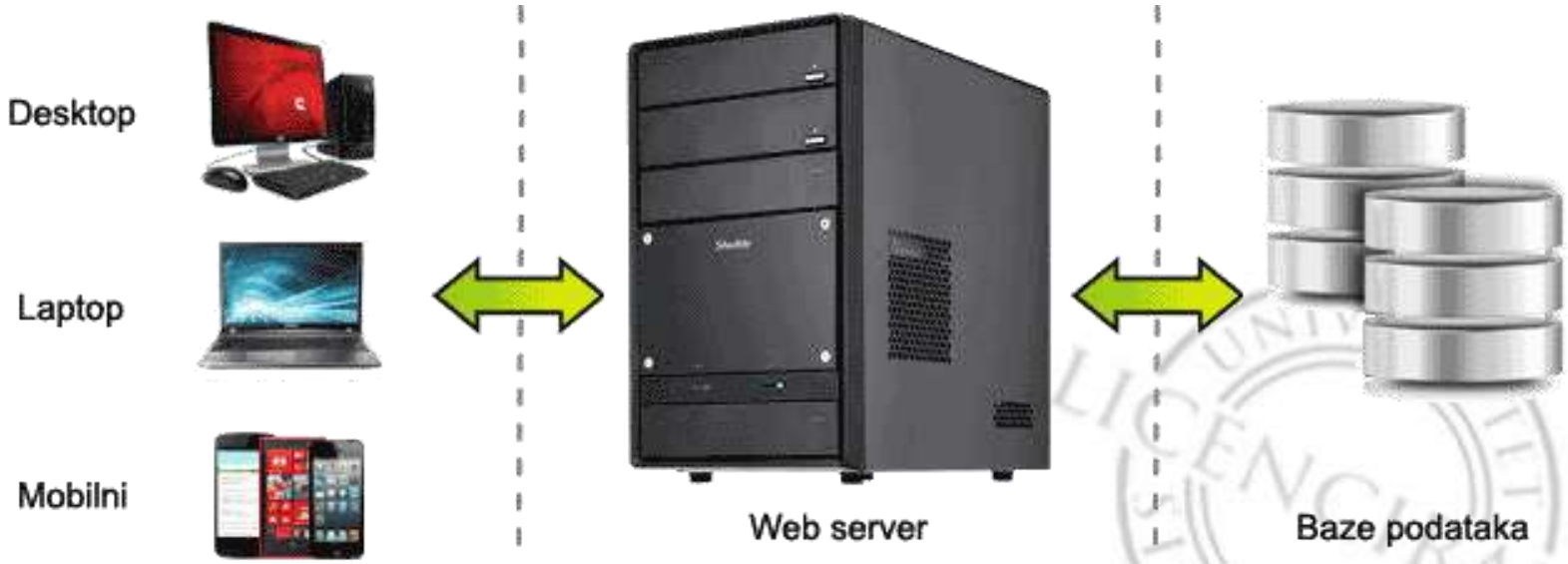
- Logički slojevi (Layers)
  - Način na koji se organizuje programski kod
  - Zašto? Radi bolje preglednosti koda, skalabilnosti, lakšeg održavanja, bržeg razvoja, smanjenja troškova razvoja i održavanja.
  - Primer
    - Korisnički interfejs
    - Funkcionalna logika
      - Load balancing (softverski)
      - Bezbednost
      - Autorizacija
      - Templejti
      - ...
    - Skladištenje podataka
    - Pristup podacima
    - ...

Slojevi

- Fizički nivoi (Tiers)
  - Mesta na kojima se slojevi nalaze i gde se izvršavaju.
  - Zašto? Prvenstveno da bi se ostvarila ravnoteža između performansi, skalabilnosti, tolerancije na greške i bezbednosti.
  - Primer:
    - Prezentacioni sloj
    - Web server
    - CDN
    - email server
    - DB server
    - ...

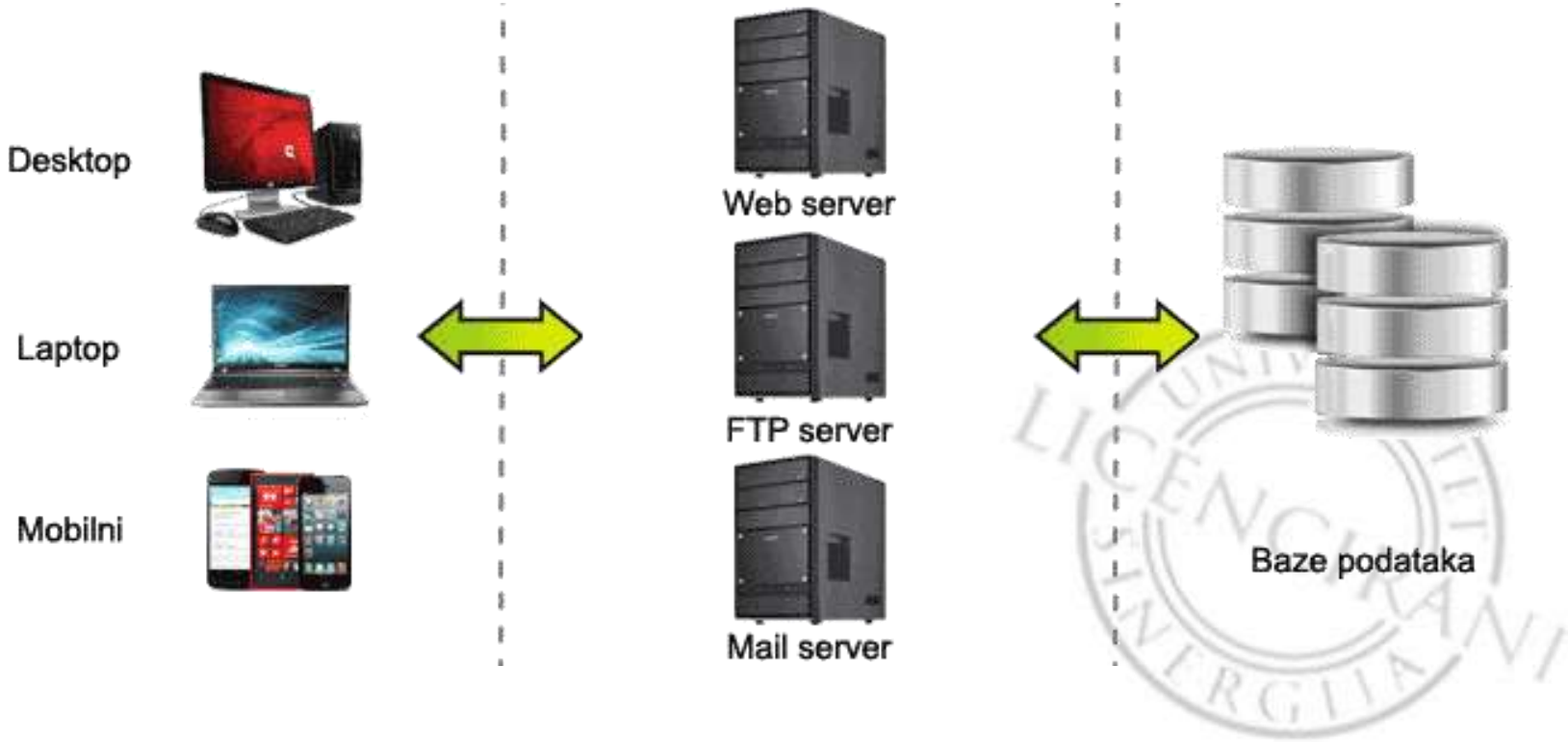
Slojevi

- Troslojne aplikacije



Slojevi

- Višeslojne aplikacije



Slojevi

- Prednosti

  Omogućava razvoj fleksibilnih aplikacija, čiji se delovi zajedno ili nezavisno mogu iznova koristiti u različitim projektima.

  Izmene u jednom sloju su izolovane u odnosu na ostale slojeve.

  Protok informacija između slojeva je deo arhitekture i zasnovan je na jednom ili više protokola (sockets, web servisi...).

- Mane

  Sa porastom broja slojeva, i što je aplikacija složenija, sve je teže pratiti protok podataka kroz slojeve.

Slojevi

- Izbor arhitekture
  - Na osnovu trenutnih, ali i budućih potreba sistema.
  - Za statičke web sajtove i jednostavne web aplikacije sa malim brojem korisnika, dovoljna je dvoslojna arhitektura.
  - Za web aplikacije sa većim brojem korisnika, SaaS ili poslovne aplikacije, neophodni minimum je troslojna arhitektura.
  - Kriterijumi za izbor
    - Projektovani broj korisnika
    - Primenjena platforma i softverski alati
    - Tip i veličina baze podatak
    - Broj i veličina dokumenata
    - ...

Slojevi

- Pročitati
  - Razvoj aplikativnog softvera, V. Tomašević, Singidunum 2012.

    4.2.2 Slojevita arhitektura
  - Web
    - http://www.lhotka.net/weblog/ShouldAllAppsBeNtier.aspx
    - http://stackoverflow.com/questions/120438/whats-the-difference-between-layers-and-tiers

OOP

Klase tipovi koji opisuju strukturu objekta
Objekti konkretni podaci

- Uvod

  Objektno-orijentisano programiranje (OOP) je metod programiranja upotrebom "objekata", koji su instance klasa i poseduju

  - skupove atributa (osobina) koji ih opisuju i
  - funkcije (operacije) - metode

  OO program obično sadrži različite tipove objekata, svaki sa skupom podataka koji opisuju neku realnu pojavu:

  - Automobil
  - Račun u banci
  - Profil sportiste
  - ...

OOP

- Pravila formiranja klasa
  - Razdvojiti bitno od nebitnog za dati problem. Ono što je bitno, implementirati u klasi.
  - Svi podaci u klasi su skriveni osim onih koji su eksplicitno deklarisani kao javni.
  - Enkapsulacija
    - kontrola pristupa komponentama objekta
    - public, private, protected
  - Modularnost
    - Razbijanje koda na manje delove koji mogu samostalno da funkcionišu.
  - Nasleđivanje
    - Kada jedna klasa nasledi drugu ona zadržava kompletan sadržaj klase koju nasleđuje, i taj sadržaj može redefinisati ili proširiti.

OOP

- Metode
  - Modifier (mutator)
  - Accessor

```
class MyClass
{
    private $prop;
    // Accessor (or Getter)
    public function getProp() { return $this->prop; }
    // Mutator (or Setter)
    public function setProp($value) { $this->prop = $value; }
}
```

  - Constructor
    - Poziva se prilikom kreiranja objekta
  - Destructor
    - Poziva se prilikom gašenja (destrukcije) objekta

OOP

- Motorno vozilo (osnovna)
  - Tip — auto, kombi, kamion
  - Gorivo — benzin, dizel, TNG
  - Boja — bela, crvena, ...
  - Broj vrata — 2, 3, 4, ...
  - Težina — x kg
  - Godina proizvodnje — godina
  - Registarski broj
  - Registracija — datum

OOP

- Auto (extends)
  - Karoserija — sedan, hatch, karavan
  - Broj sedišta — 2, 3, 4, 5

OOP

- Kombi (extends)
  - Tip — putnički, teretni
  - Broj sedišta — 2, 3, 4, 5, 6, 7, ...
  - Nosivost — 500, 800, ...

OOP

- Kamion (extends)
  - Tip — tegljač, cisterna, ...
  - Broj sedišta — 2, 3
  - Nosivost — 1.5t, 5t, 7.5t, ...
  - Max. doz. težina — x kg
  - Osovinsko opterećenje — x kg

OOP

- Motor (extends)
  - Kubikaža x cm^3
  - Broj cilindara 2-12
  - Ulje tip mineralno, sint.
  - Ulje marka Castrol, Total, ...
  - Ulje gradacija 5W40, 15W40, SAE30
  - Svećice ...

OOP

- Pročitati
  - Razvoj aplikativnog softvera, V. Tomašević, Singidunum 2012.
    4.2.6 Objektno-orijentisani pristup
  - Web
    - http://www.php.net/manual/en/language.oop5.php
    - http://www.aonaware.com/OOP2.htm