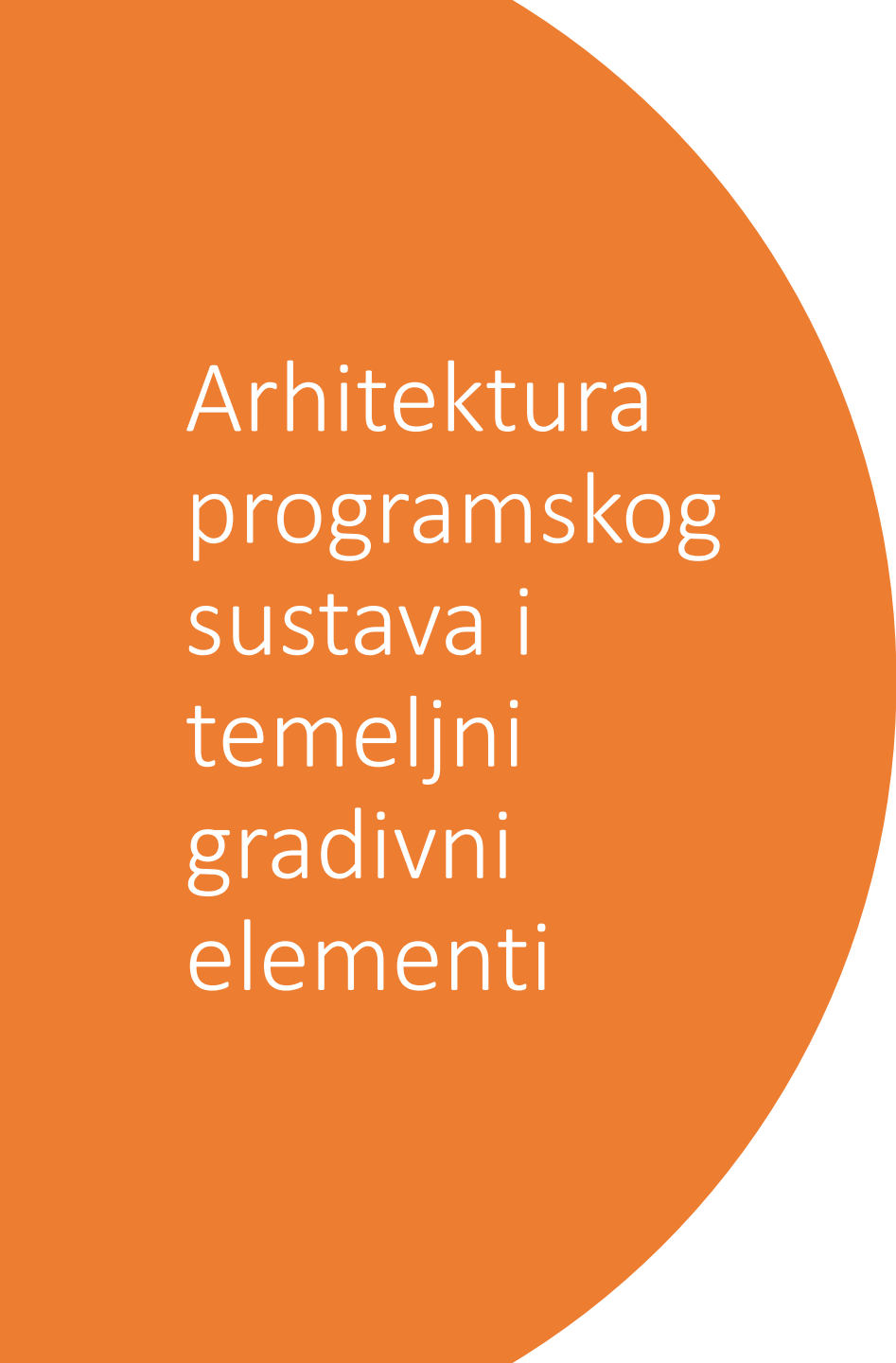




Modeliranje programskog sustava

Zlatko Stapić, dr. sc.
Sveučilište u Zagrebu
Fakultet Organizacije i Informatike,
Varaždin

A large orange circle on the left side of the slide, partially cut off by the edge.

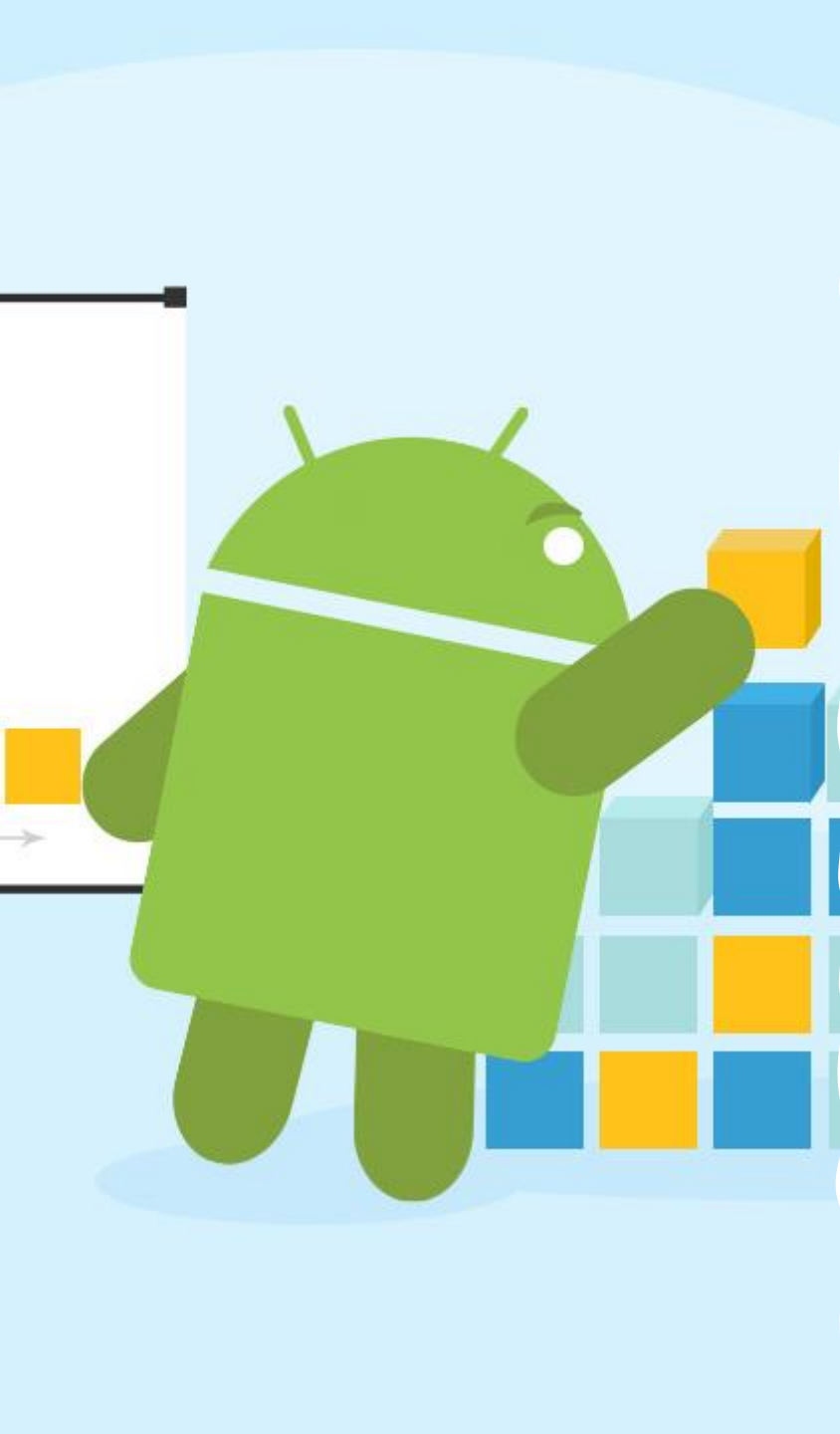
Arhitektura
programskog
sustava i
temeljni
gradivni
elementi

Raznolikost karakteristika
arhitekture

Nedostatak univerzalnih
standarda

Dinamična priroda softverskog
ekosistema

Kategorizacija karakteristika
arhitekture



Primjeri karakteristika arhitekture

modularnost

skalabilnost

elastičnost

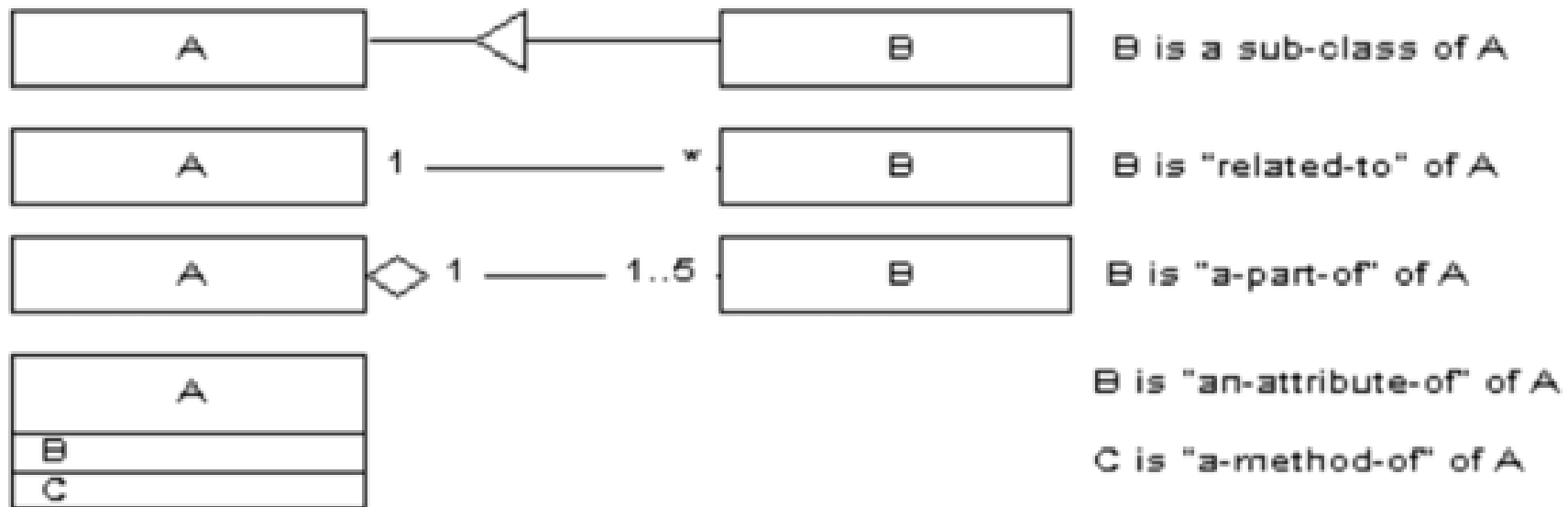
Strukturna arhitektura



- Odgovornost arhitekata za strukturu koda
- Kvaliteta koda kao briga arhitekata
- Dobro modularana struktura
- Kontrolirano spajanje između komponenata
- Čitljiv kod
- Unutarnje procjene kvalitete

Dijagramske tehnike i notacije za modeliranje programskog sustava

- UML dijagrami
- notacija neovisna o jeziku koja omogućava specifikaciju razreda, njihovih podataka ili atributa (privatnih) i metoda (javno), nasljeđivanje i druge općenitije odnose između razreda

Summary Of UML Object Modelling Notation





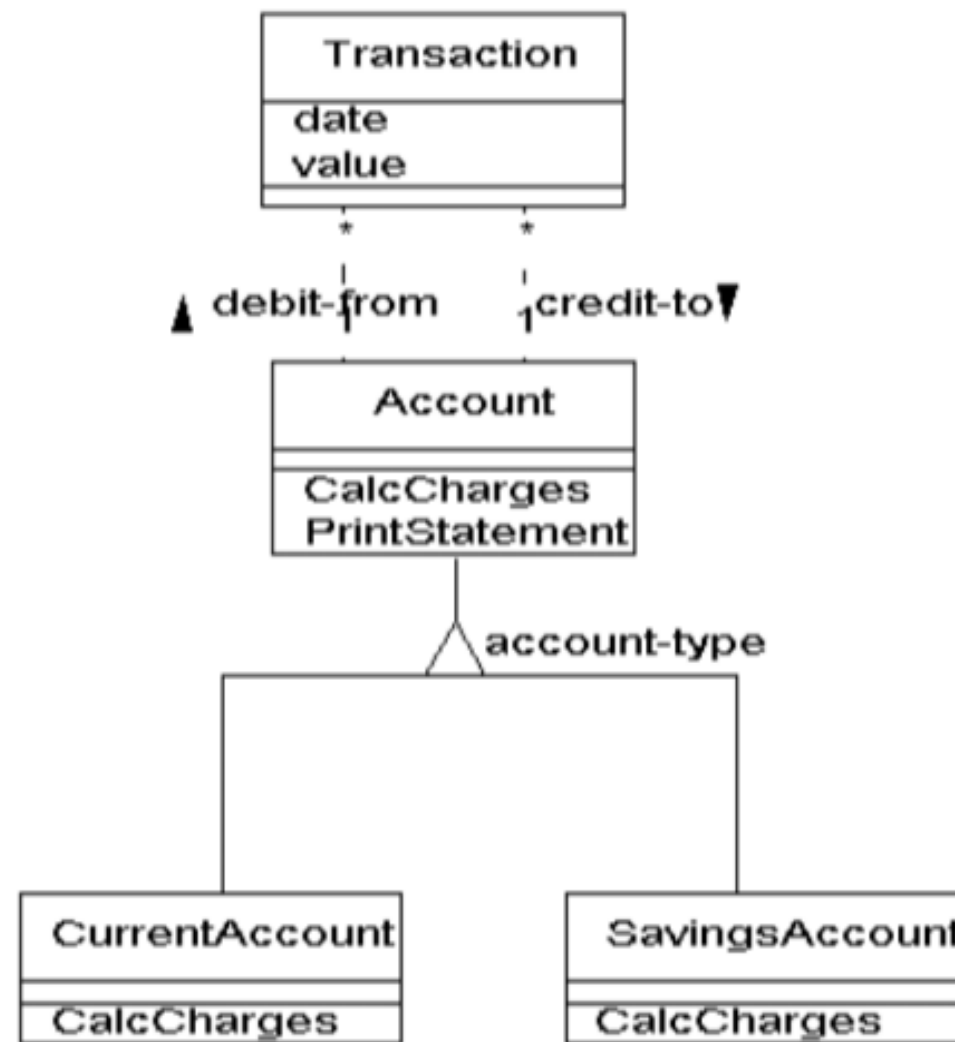
Objektni model

- Struktura klasa i atributa
- Odnosi između klasa
- Naslijeđivanje i predefiniranje metoda
- Upotreba polimorfizma u objektno orijentiranim jezicima
- Veze između različitih klasa i entiteta
- Značaj run-time veza u objektnom modeliranju
- Kako objektni model olakšava razumijevanje i implementaciju sustava
- Principi objektno orijentiranog dizajna u praksi

Bennett, 2001

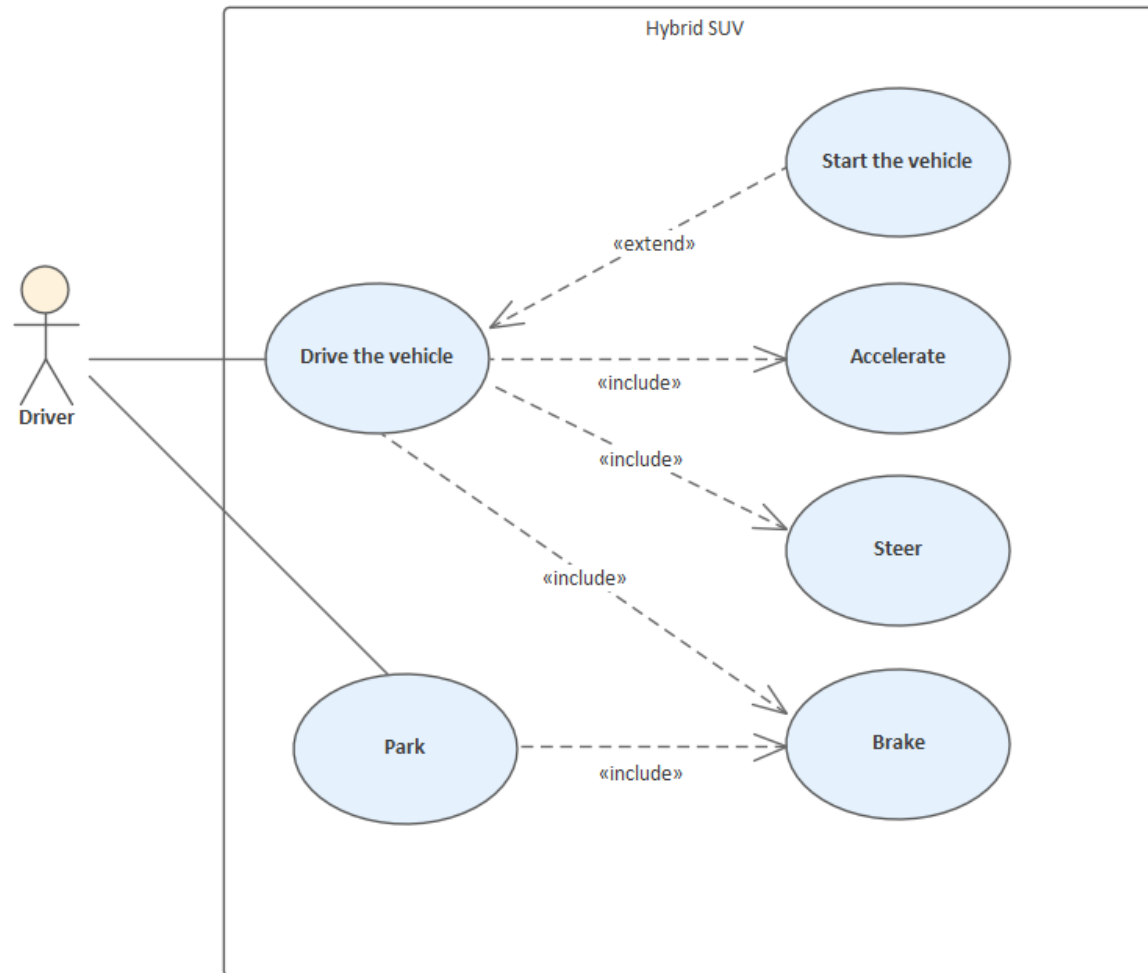
Implementacija objektnih modela (primjer)

- Potrebne klase u objektnom modelu
- Klase: Transaction, Account, Current Account i Savings Account
- Povezanost računa s transakcijama
- Implementacija metode PrintStatement
- Primjena naslijeđivanja veza između konceptualnog objektnog modela i konkretnih programskih jezika



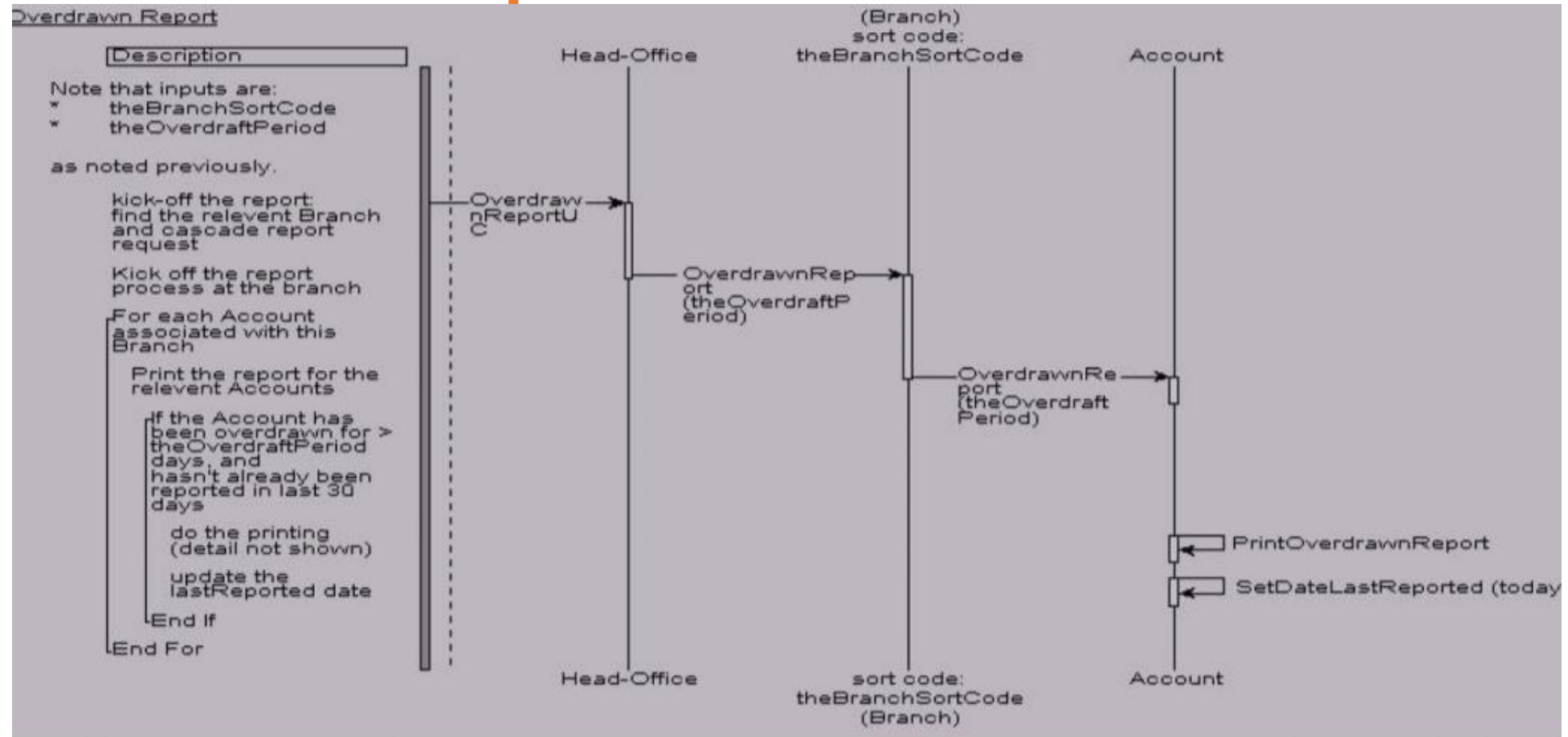
Objektno orijentirana notacija

- USE CASES

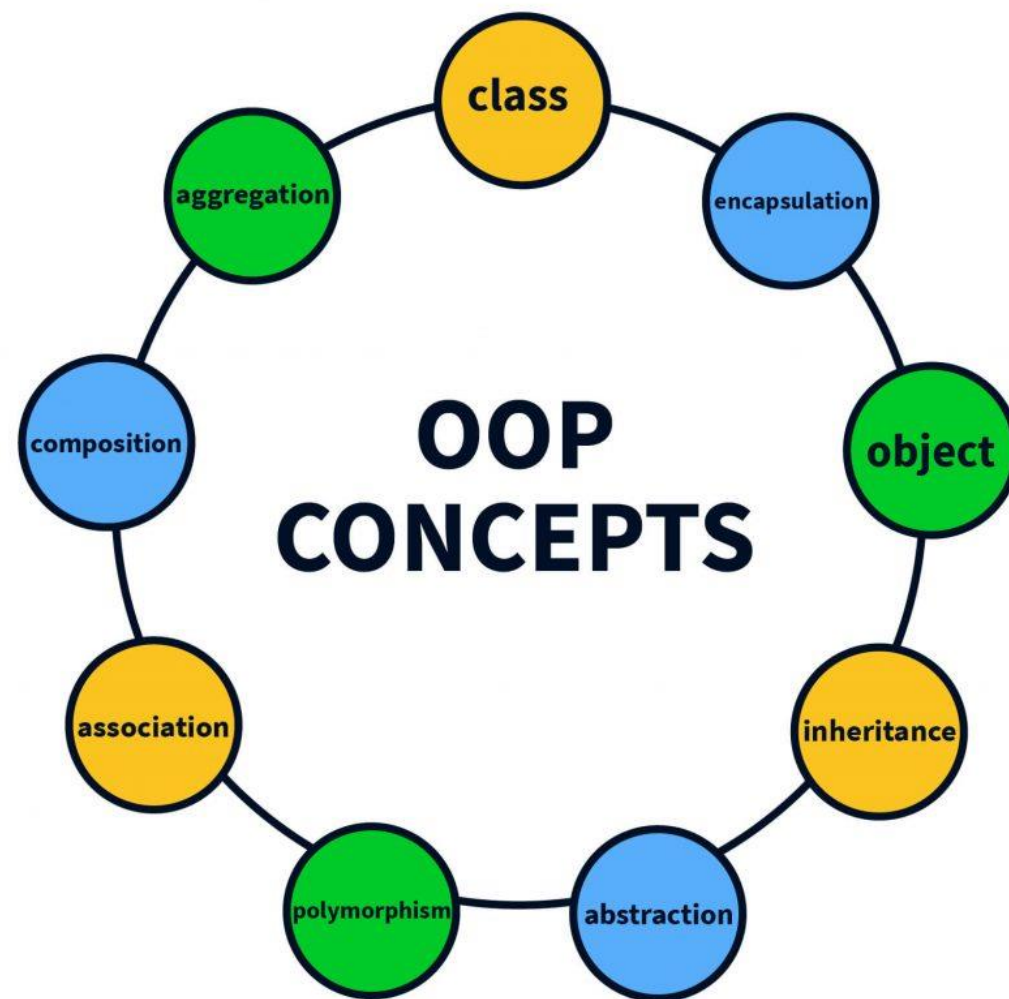


Objektno orijentirana notacija

- Dijagrami slijeda

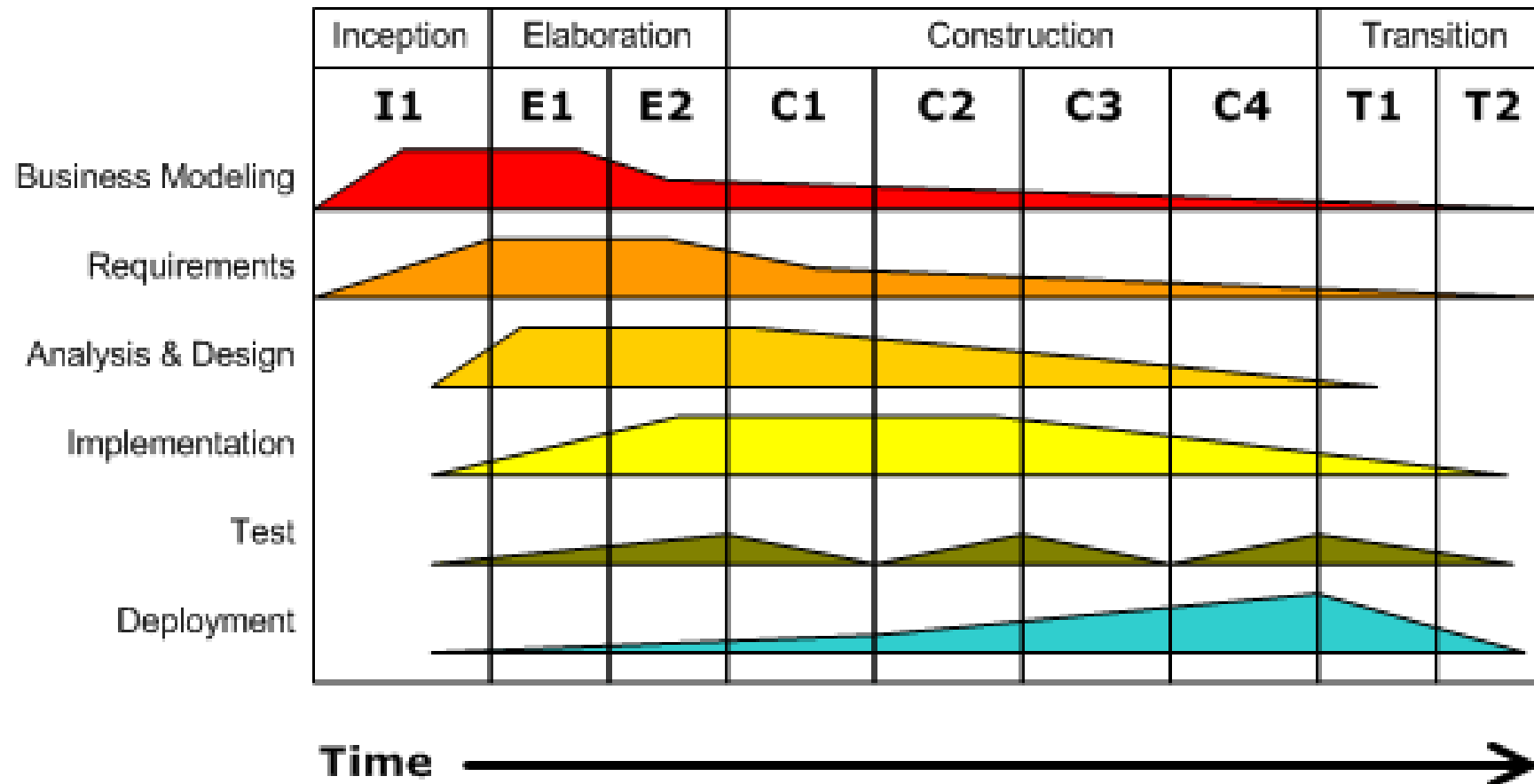


Temeljni koncepti OO pristupa

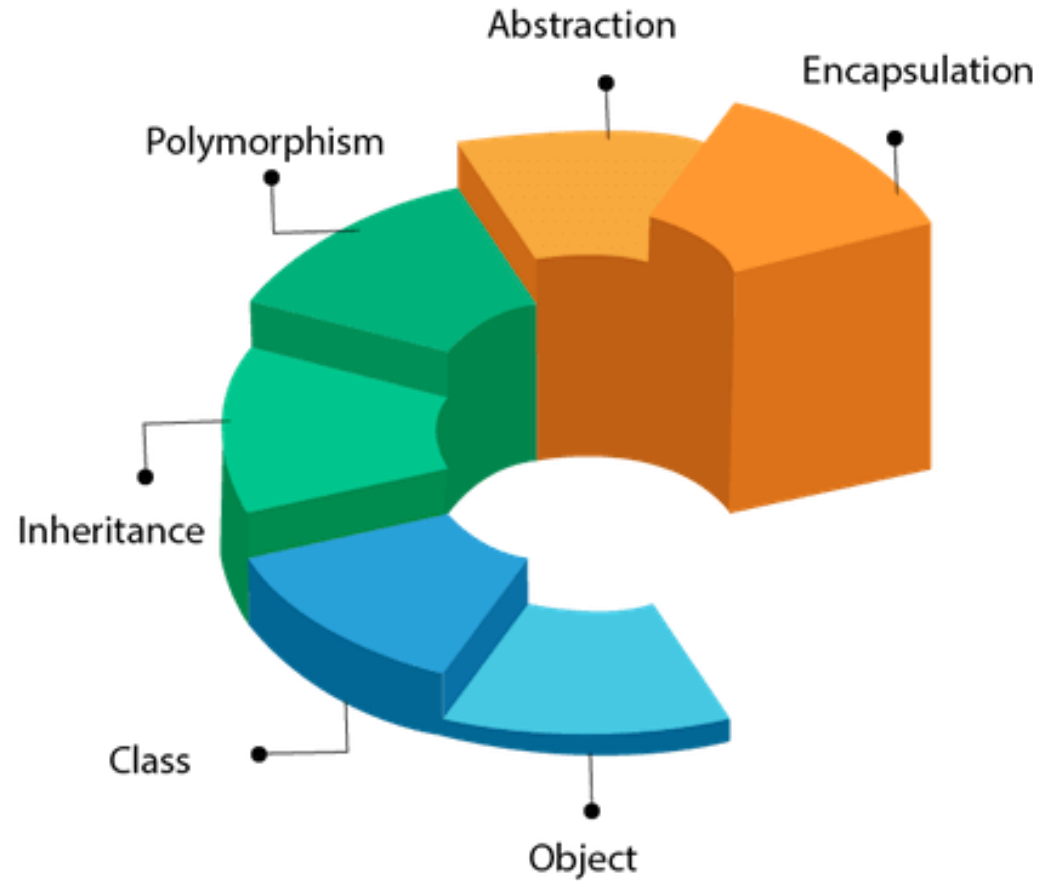


Iterative Development

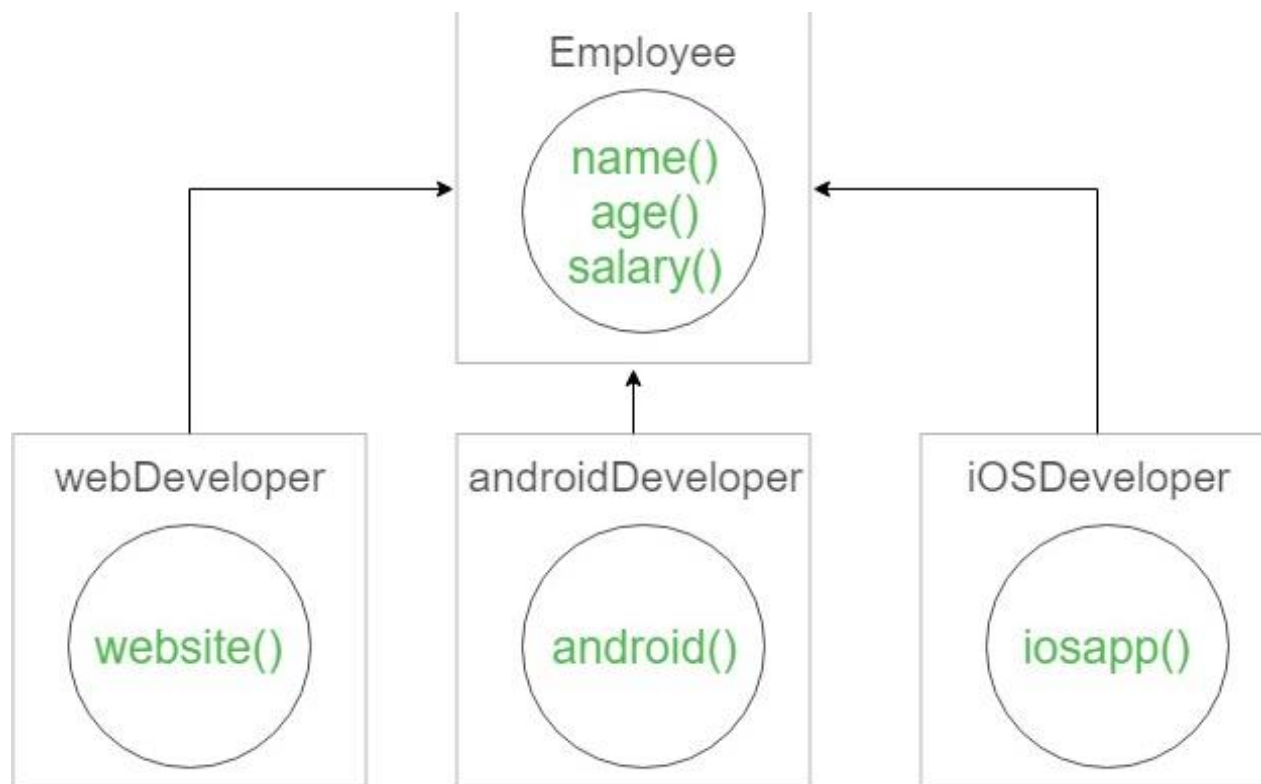
Business value is delivered incrementally in time-boxed cross-discipline iterations.



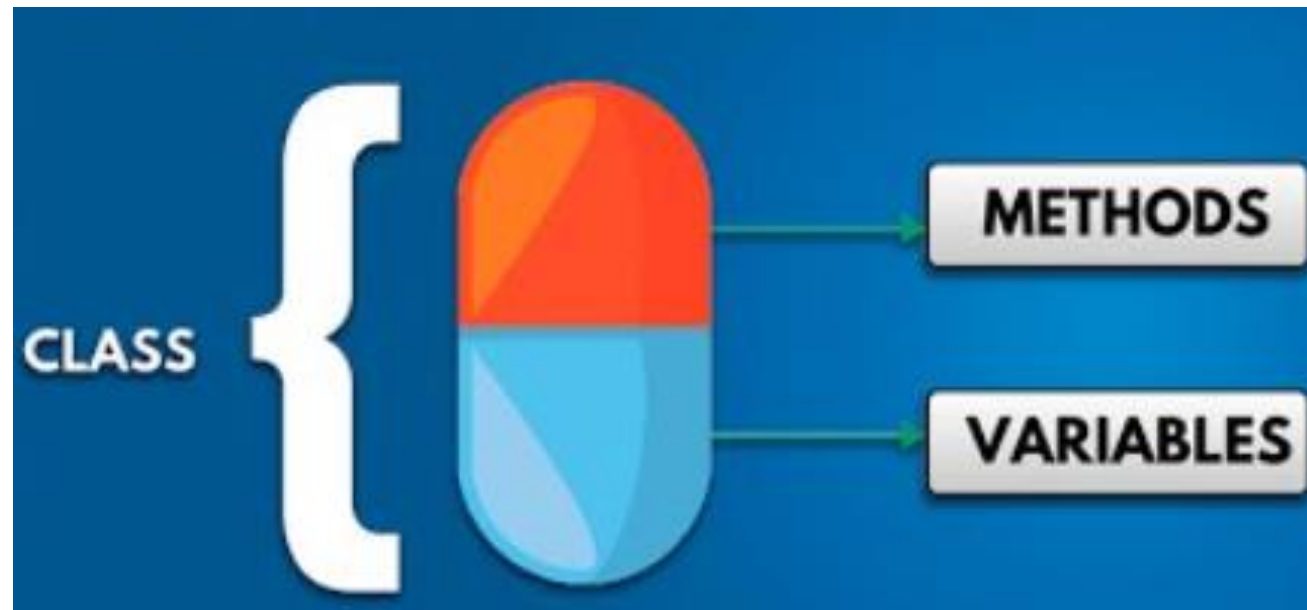
OOPs (Object-Oriented Programming System)



Nasljeđivanje



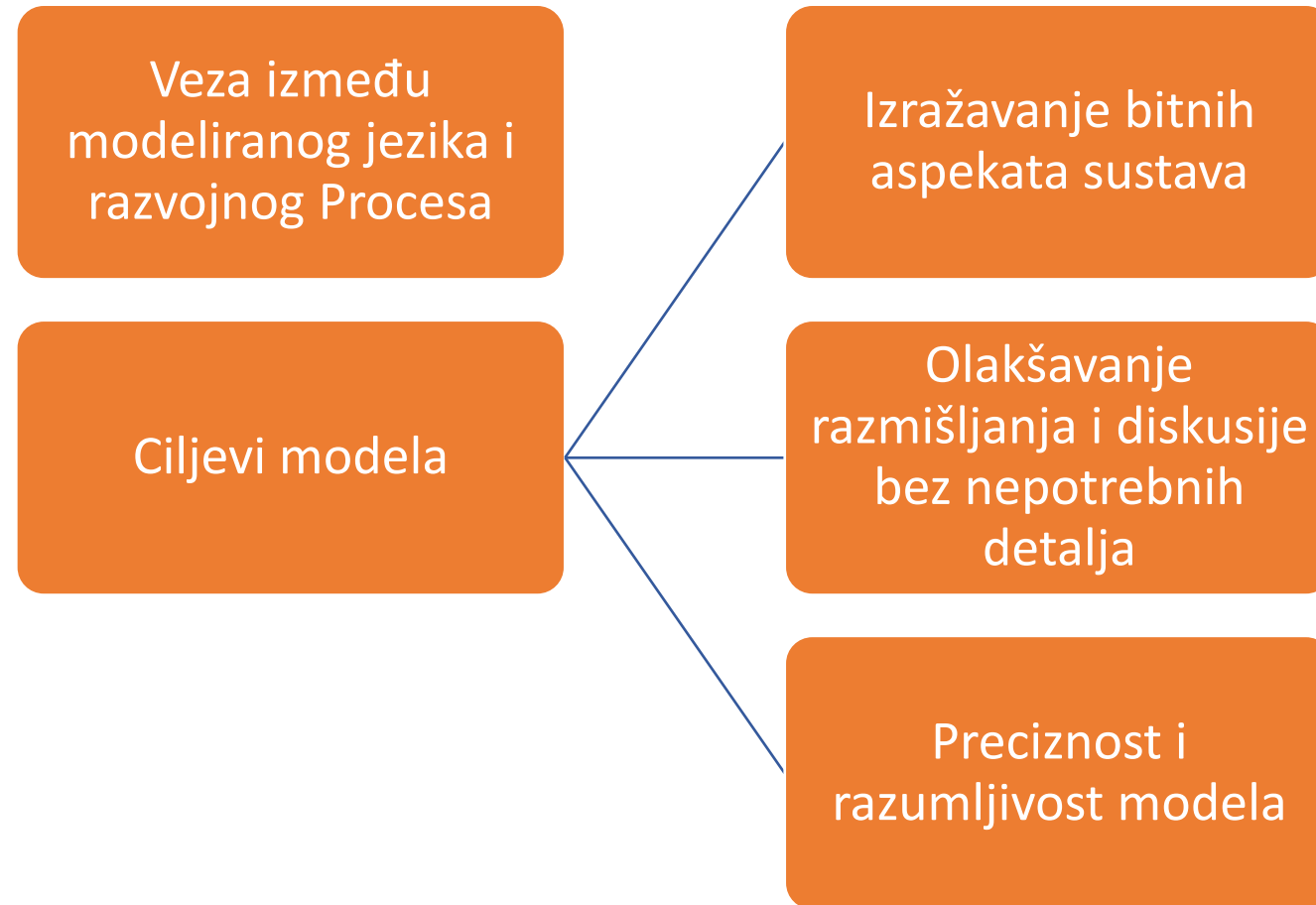
Ekapsulacija



Polimorfizam

- Omogućava različitim objektima da se ponašaju na različite načine kroz zajednički interface
- Doprinosi fleksibilnosti i reusabilnosti koda
- Postoji statički (compile-time) polimorfizam i dinamički (runtime) polimorfizam
- Temeljni koncept u razvoju softvera za različite uređaje i aplikacije

Objektno orijentiran pristup u programskim jezicima i alatima



UML paradigma i primjena pri arhitekturnom i strukturnom dizajnu

Inženjering softvera

- Inženjering softvera primjenjuje inženjerske principe na izradu softverskih sustava.
- Uključuje definirani proces s fazama i jasnim zahtjevima.
- Naglašava važnost verifikacije i validacije te korištenje alata.

UML paradigma i primjena

- Objektno modeliranje u UML-u je središnja tehnika.
- Primjenjuje se na arhitektonski i strukturni dizajn.
- Pomaže u organizaciji objekata i razreda temeljenih na stvarnim entitetima, ulogama, događajima i interakcijama.

Dizajn sustava

- Važno je izbjegavati monolitni dizajn u kojem je jedan razred odgovoran za sve u sustavu.
- U OO pristupu, cijeli sustav čine objekti i njihove interakcije.
- Klase se trebaju prilagoditi određenim zadacima i ne smiju sadržavati nepotrebne informacije.

Pitanja



surveylegend.com

Literatura

1. Richards M.; Ford N.: Fundamentals of Software Architecture: An Engineering Approach, O'Reilly, 2020.
2. Martin, C. R.: Clean architecture : a craftsman's guide to software structure and design, Prentice Hall, 2017.
3. Bennett, S.; McRobb,S.; Farmer, R. Object-Oriented Systems Analysis and Design using UML. McGraw-Hill, 2001
4. Stevens, P., & Pooley, R. *Using UML: software engineering with objects and components*. Addison-Wesley Longman Publishing Co., Inc., 1999