

Projektovanje softvera

Posmatrač

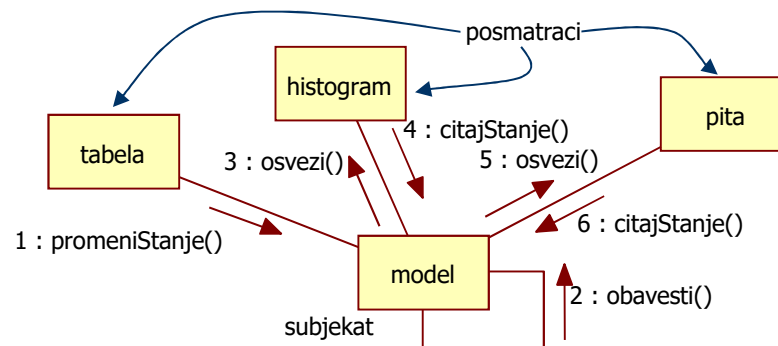


Posmatrač (1)

- Ime i klasifikacija:
 - Posmatrač (*engl. Observer*) – objektni uzorak ponašanja
- Namena:
 - definiše 1:n zavisnost između objekata takvu da kada jedan objekat promeni stanje svi zavisni se obaveste i modifikuju automatski
- Druga imena:
 - Zavisni objekti, Objavljivanje-Pretpлата, Pretpлата-Obaveštavanje (*Dependents, Publish-Subscribe, Subscribe-Notify*)

Posmatrač (2)

- Motivacija:
 - nije dobro da su klase sistema čvrsto-spregnute (smanjuje se reupotrebljivost)
 - na primer, alati za GUI razdvajaju prezentacione aspekte od aplikativnih podataka
 - klase aplikativnih podataka i one za prezentaciju se mogu reupotrebiti nezavisno
 - objekat za tabelarni prikaz i objekat za histogram mogu prikazati iste podatke
 - objekat za tabelarni prikaz i objekat za histogram ne znaju jedan za drugog
 - zbog toga dopuštaju nezavisnu reupotrebu, a ponašaju se kao da su spregnuti
 - kada se podaci promene kroz tabelu – promeni se i histogram

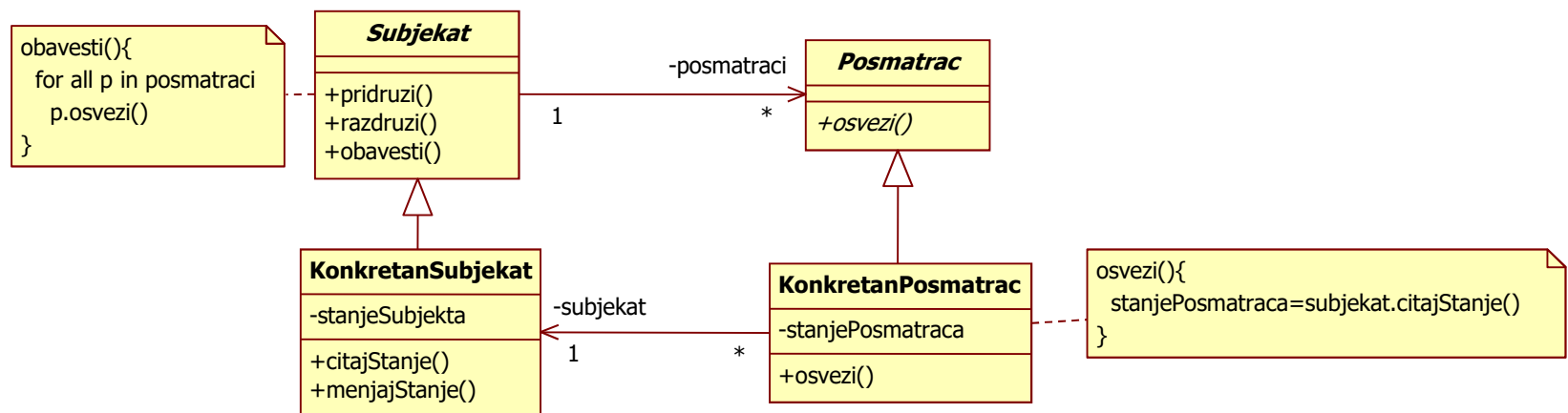


Posmatrač (3)

- Motivacija (nastavak):
 - tabela i histogram su zavisni od modela: treba ih obavestiti kad model promeni stanje
 - nema ograničenja u broju zavisnih objekata koje treba obavješavati
 - ključni objekti u uzorku su subjekat (*subject*) i posmatrač (*observer*)
 - uzorak se naziva i *Dependents*, jer posmatrač zavisi od subjekta
 - subjekat obavještava posmatrača kad promeni stanje
 - posmatrač zatim šalje upit subjektu o njegovom stanju
 - na osnovu primljenog stanja subjekta, posmatrač ažurira svoje stanje
 - uzorak se naziva i *Publish-Subscribe*, kao i *Subscribe-Notify*
 - posmatrači se "pretplaćuju" kod subjekta za obavještenja (notifikaciju)
 - subjekat šalje poruku obavještenja o promeni posmatraču
 - poruke se šalju svim posmatračima i ne znajući ko su sve posmatrači

Posmatrač (4)

- **Primenljivost:** uzorak treba koristiti u sledećim situacijama:
 - kada jedna apstrakcija ima barem dva međusobno zavisna apseka takva da promena bilo kog utiče na promenu drugih
 - kada izmena jednog objekta zahteva izmenu nepoznatog broja drugih objekata
 - kada jedan objekat treba da signalizira promenu drugim objektima ne znajući prirodu (konkretni tip) tih objekata
- **Struktura:**

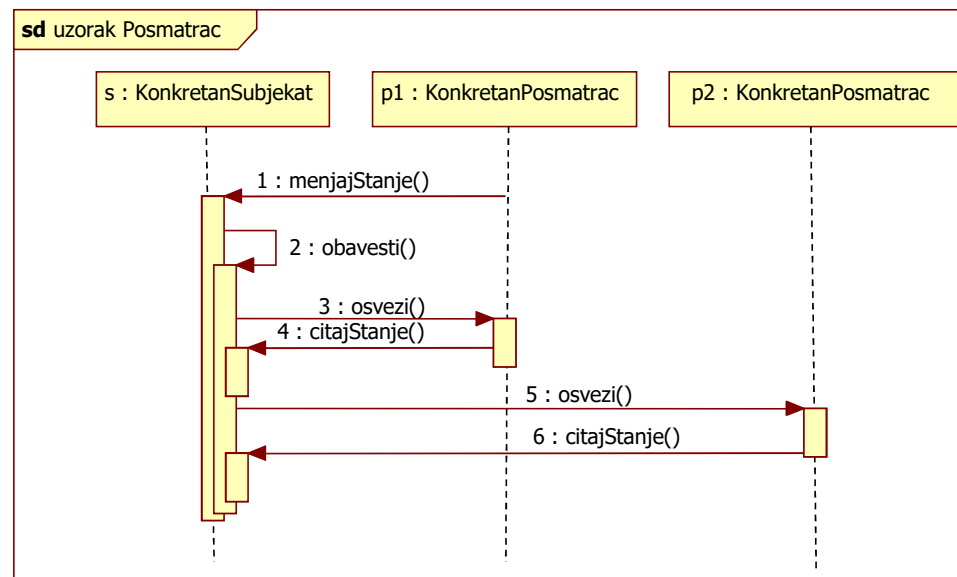


Posmatrač (5)

- Učesnici:
 - *Subjekat* (klasa *Model*)
 - zna svoje posmatrače (proizvoljan broj), ali ne i njihovu prirodu
 - obezbeđuje interfejs za pridruživanje i razdruživanje posmatrača
 - *Posmatrac* (klasa *Prikaz*)
 - definiše interfejs za signaliziranje promena subjekta
 - *KonkretanSubjekat* (klasa *Statistika*)
 - čuva stanje od interesa za konkretne posmatrače
 - šalje signal posmatračima kada se promeni stanje
 - omogućava čitanje stanja
 - *KonkretanPosmatrac* (klase *Tabela, Histogram, Pita*)
 - poseduje referencu na konkretan subjekat (subjekat može da bude i parametar)
 - čuva stanje koje treba da bude u konzistenciji sa stanjem konkretnog subjekta
 - implementira operaciju preko koje mu subjekat signalizira promenu
 - čita stanje konkretnog subjekta da bi ažurirao sopstveno stanje

Posmatrač (6)

- Saradnja:
 - konkretan subjekat signalizira svojim posmatračima svaku promenu svog stanja
 - nakon poziva `osvezi`, konkretan posmatrač traži od subjekta informaciju o stanju
 - posmatrač koristi informaciju o stanju subjekta da ažurira svoje stanje

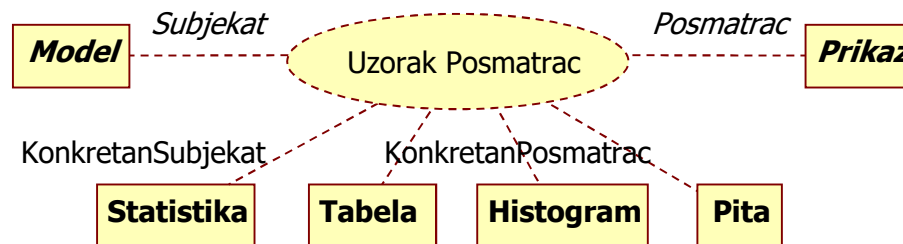


Posmatrač

25.10.2023.

Posmatrač (7)

- UML notacija



- Posledice:

- dobre strane

- apstraktno vezivanje između subjekata i posmatrača; tako se omogućava da konkretni subjekat i posmatrač budu u različitim slojevima aplikacije
- podrška sa *multicast* komunikaciju; subjekat šalje signal svim registrovanim posmatračima

- nedostatak:

- nepoznata cena promene: pošto posmatrači ne znaju ko će sve dobiti signal, oni nisu svesni cene promene subjekta

- Povezani uzorci:

- za kapsuliranje kompleksne semantike ažuriranja
Posrednik može da posreduje između subjekta i posmatrača