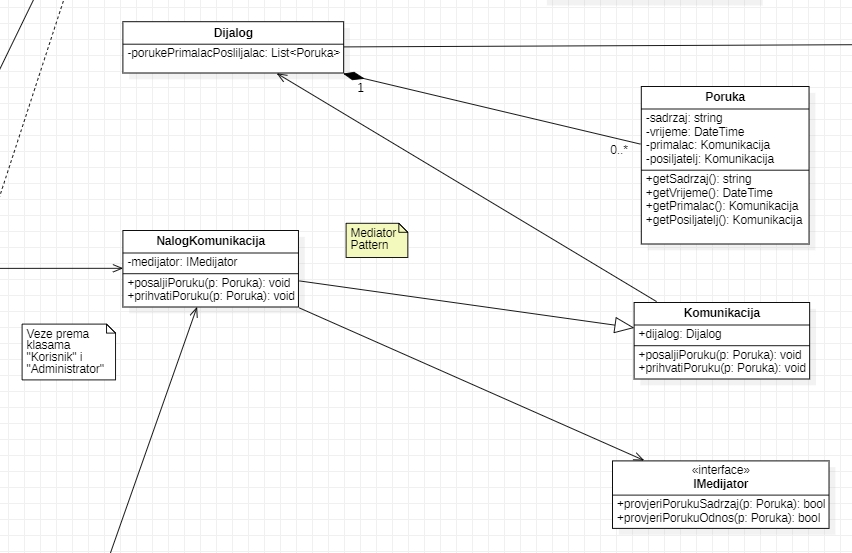
Paterni ponašanja

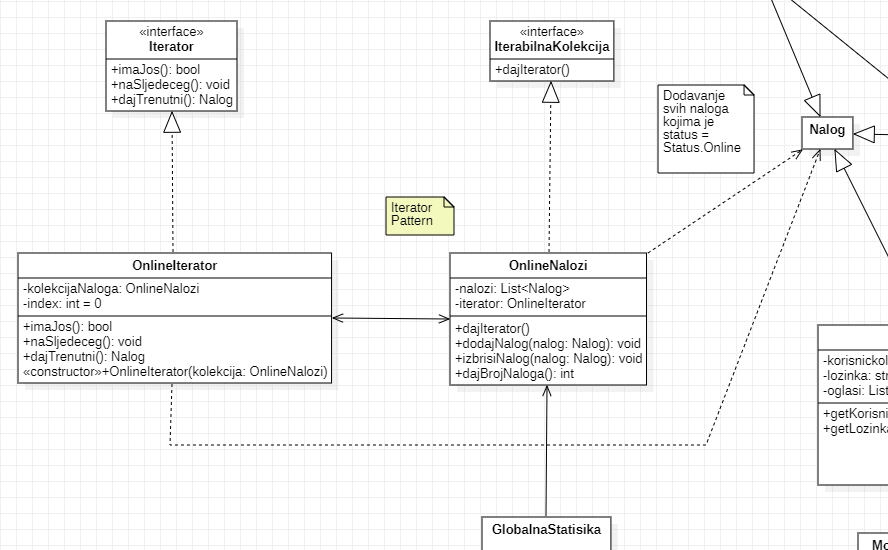
**Mediator (Primjenjen)**

Medijator patern omogućava reduciranje ovisnosti izmedu objekata. Ovaj patern ograničava direktnu komunikaciju izmedu objekata i forsira ih da kolaboriraju samo preko medijator objekta. Umjesto za direktno povezujemo veliki broj objekata, koristimo medijator, koji je zadužen za njihovu komunikaciju. Kada neki objekat želi poslati poruku drugom objektu, on šalje poruku medijatoru, a medijator zatim prosljeđuje tu poruku drugom objektu.

****

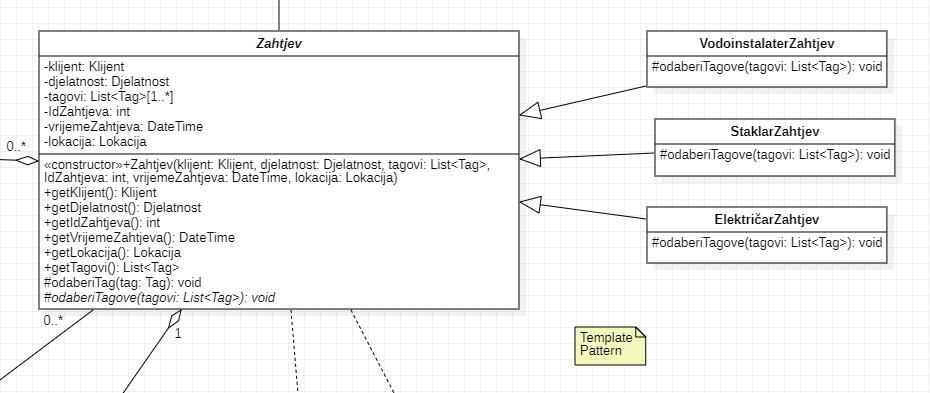
**Iterator**

Iterator patern omogućava sekvencijalni pristup elementima kolekcije bez poznavanja kako je ta kolekcija struktuirana.



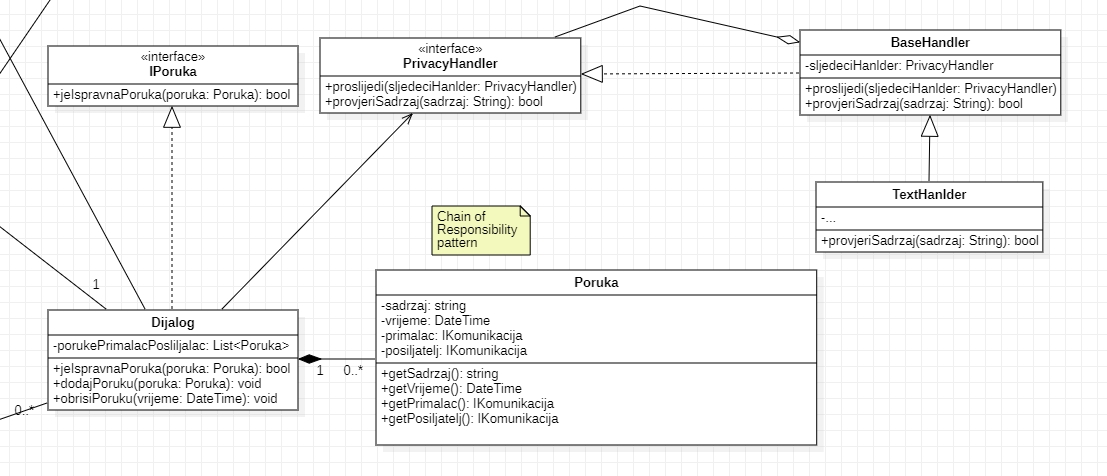
**Template (Primjenjen)**

Svrha Template method patern-a je da omogući izdvajanje odredenih koraka algoritma u odvojene podklase. Samim tim, struktura algoritma se ne mijenja, mali dijelovi operacija se izdvajaju i ti se dijelovi mogu implementirati različito. Ovaj patern se sastoji od klase Algorithm koja uključuje metodu koja izdvaja dijelove svojih operacija u druge klase. Tu se nalazi interfejs IPrimitive koji definira operacije koja pomenuta metoda izdvaja u druge klase te AnyClass koja implementira interfejs IPrimitive.



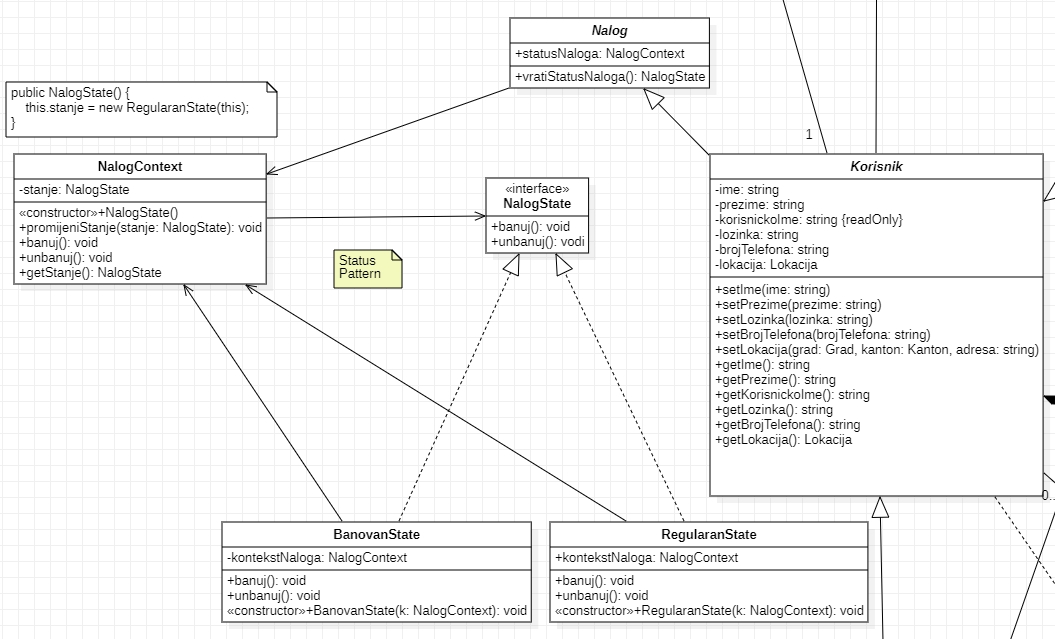
**Chain of Responsibility**

Chain of responsibility patern predstavlja listu objekata, ukoliko objekat ne može da odgovori prosljeđuje zahtjev narednom u nizu.



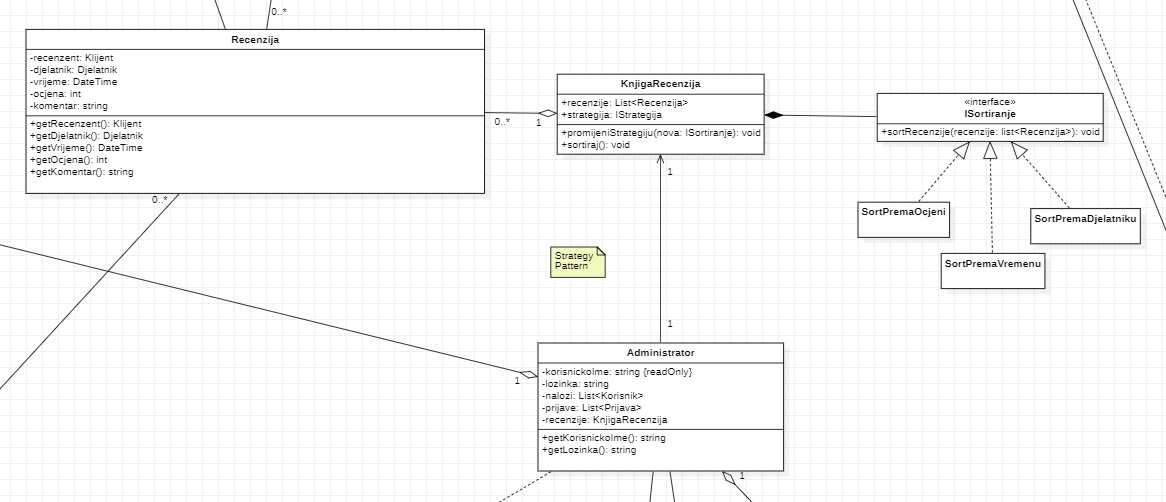
**Status**

State patern omogućava objektu da promijeni svoje ponašanje, od kojih zavisi njegovo ponašanje. Nakon promjene stanja, objekat se počinje ponašati kao da je promijenio klasu. Objekat mijenja način ponašanja na osnovu trenutnog stanja. State patern predstavlja dinamičku verziju Strategy patern-a. Postiže se promjenom podklase unutar hijerarhije klasa. Podržava open-closed princip.



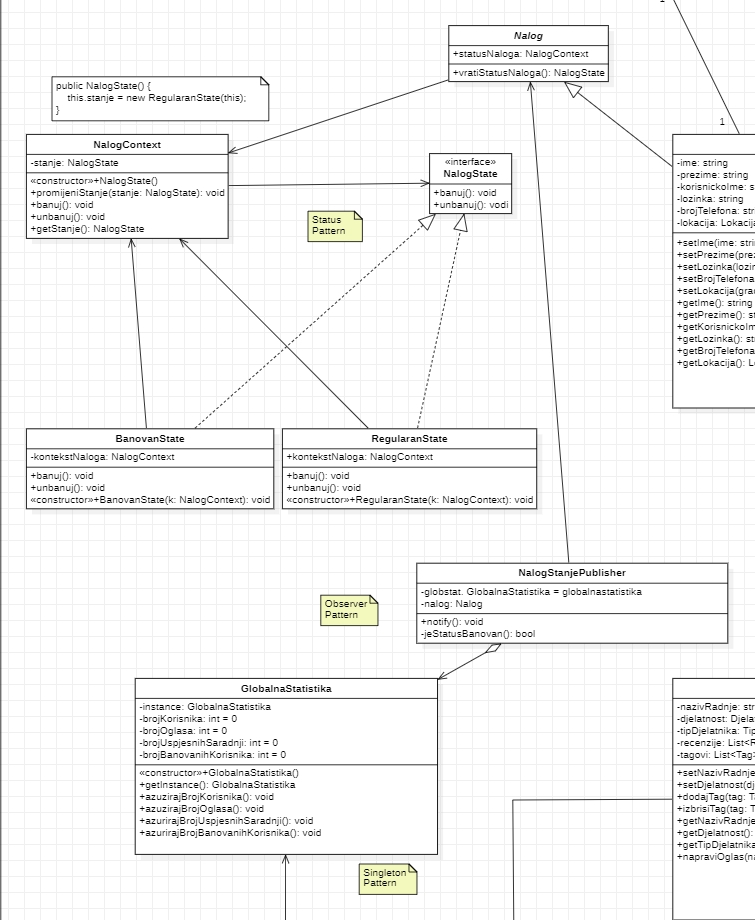
**Strategy**

Uloga Strategy patern-a je da izdvaja algoritam iz matične klase i uključuje ga u posebne klase. Koristi se kada postoje različite primjenljivne strategije za neki problem. On omogućava korisniku izbor jednog algoritma iz familije algoritama za korištenje.

****

**Observer**

Uloga Observer patern-a je da uspostavi relaciju izmedu objekata tako kada jedan objekat promijeni stanje drugi zainteresirani objekti se obavještavaju.

****