PATERNI PONAŠANJA

U paterne ponašanja spadaju: Strategy, State, TemplateMethod, Observer i Iterator.

- ## STRATEGY PATERN

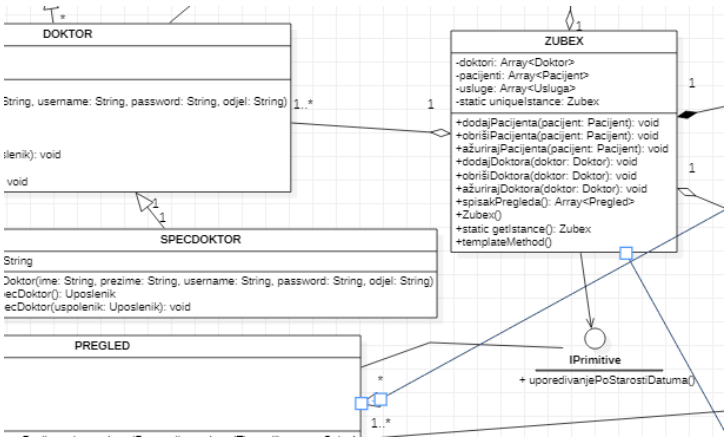
Strategy patern izdvaja algoritam iz matične klase i uključuje ga u posebne klase.Pogodan je kada postoje različiti primjenjivi algoritmi(strategije) za neke probleme.On omogućava klijentu izbor jednog od agloritama iz familije algoritama za korištenje.Algoritmi su neovisni od klijenata koji ih koriste.Ovaj patern bi se mogao iskoristiti npr. kod biranja nekih algoritama za sortiranje pregleda, trebalo bi dodati klase koje implementiraju algoritme(tj. Istrategijy intefejs) kao i Context klasu.U Istrategy interfejsu se defeniše metoda sortiranja koji je isti za sve algoritme i njega nasljeđuju sve klase koje vrše različita sortiranja.

- ## STATE PATERN

State patern je dinamička verzija Strategy paterna.Objekat mijenja način ponašanja na osnovu trenutnog stanja.Postiže se promjenom podklase unutar hijerarhije klasa.Kada je osoba koja koristi aplikaciju offline on samo može dobiti obavijest da mu je pregled u roku od 24h i ništa drugo ne može raditi, dok ukoliko je on online može zakazivat nove preglede,primati obavijesti da mu je pregled uskoro itd.Dakle imali bi dvije State klase StateOffline i StateOnline i u njima implementirana stanja korisnika.Potrebno je i dodati klasu Obavijest koja bi se vezala sa interfejsom Istate na koji se dalje vežu StateOffline i StateOnline.

- ## TEMPLATEMETHOD PATERN

TemplateMethod patern omogućava izdvajanje određenih koraka algoritma u odvojene podklase.Struktura algoritma se ne mijenja – mali dijelovi operacija se izdvajaju i ti se dijelovi mogu implementirati različito.Ovaj patern smo iskoristili za mogućnost sortiranja pregleda, i to soritanje po starosti datuma pregleda.Klasa Zubex predstavlja klasu Algoritham i kod nje ubacujemo operaciju TemplateMethod(), u ovoj klasi se vrši sortiranje po zadatom kriteriju.Napravljen je interfejs IPrimitive sa operacijom uporedivanjePoStarostiDatuma() i njega implementira klasa Pregled.



- OBSERVER PATERN

Uloga Observer paterna je da uspostavi relaciju između objekata tako kada jedan objekat promijeni stanje drugi zainteresovani objekti se obavještavaju.Ukoliko bi uveli online prijavljivanje i da za svaku osobu koja se prvi put prijavi omogući se besplatan regularni pregled ili da postoji opcija da se šalju obavijesti pacijentima sa napomenom kad im je idući pregled, ali s obzirom da nemamo tih mogućnosti nećemo koristiti ovaj patern.

- ITERATOR PATERN

Iterator pater omogućava sekvencijalni pristup elementima kolekcije bez poznavanja kako je kolekcija struktuirana.Ovaj patern također možemo iskoristiti za pretraživanje, sortiranje pregleda,doktora ili pacijenta.Klasa Client bi bila naša Zubex gdje se nalazi spisak svih pacijenata,doktora i pregleda.Zatim kreira se interfejs iduciPregled koji nasljeđuje ostale iteratore npr. reverseIterator,poDatumuIterator,poBrojuKartonaIterator..Potrebno je dodati ICreateIterator koji se povezuje sa klasom Zubex zbog pristupa kolekciji kroz koju se treba kretati.