PATERNI PONAŠANJA

Strategy patern - izdvaja algoritam iz matične klase i uključuje ga u posebne klase. Strategy patern omogućava klijentu izbor jednog od algoritma iz familije algoritama za korištenje.

U našem sistemu Strategy patern se može iskoristiti za klasu Korisnik gdje bi Korisnik prilikom zahtjevaZaVakcinaciju mogao filtrirati vrste vakcina po raznim kriterijima, kao što su najkorištenije, najbrojnije, vakcine sa jednom dozom, vakcine sa dvije doze.

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

U našem sistemu, Observer patern je moguće primijeniti ukoliko bismo Korisnicima omogućili da izraze zainteresovanost za određenu vrstu vakcine, te bi sistem prilikom dolaska većeg broja određene vrste vakcine za koju se korisnik interesuje, slao obavještenje korisniku.

State patern - dinamička verzija Strategy paterna. Objekat mijenja način ponašanja na osnovu trenutnog stanja.

U našem sistemu, State patern moguće je primijeniti ukoliko dodatno proširimo klasu Statistika da prati i broj lakših i težih slučajeva. Broj lakših i težih slučajeva bi zavisio od trenutnog stanja pacijenta, stanje bi bilo praćeno te ukoliko pacijent ne bi tražio nikakvu pomoć u prvih 10 dana bio bi povećan broj lakših slučajeva. S druge strane, ukoliko bi bila zatražena hitna pomoć doktora pacijent bi bio svrstan u broj težih slučajeva. Potrebno bi bilo pratiti i kontrolirati zahtjeve i dane od početka vođenja nekog pacijenta u sistemu.

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.

U našem sistemu, **TemplateMethod patern** bismo mogli iskoristiti prilikom podnošenja zahtjeva za vakcinaciju. Naime kako je u pitanju vakcinacija građana, najveći priroritet imaju ljudi stariji od 75 godina. Tako bi ljudi stariji od 75 godina bili podklasa klase Korisnik, sam tok prijave za vakcinaciju i podnošenje zahtjeva išao bi istim putem, a jedina razlika bila bi u tome da ako je Korisnik zapravo StarijiKorisnik njegov zahtjev bi odmah bio odobren i zakazani datum bi bio najbliži.

Iterator patern - Iterator patern omogućava sekvencijalni pristup elementima kolekcije bez poznavanja kako je kolekcija strukturirana.

U našem sistemu, **Iterator patern** je moguće iskoristiti u klasi Doktor koja sadrži listu različitih zahtjeva, **Iterator paternom** bi bilo omogućeno različito kretanje kroz zahtjeve npr. prvo zahtjevi za testiranje, zahtjevi za vakcinaciju, zahtjevi starijih ljudi itd...

Memento patern - Omogućava spašavanje internog stanja nekog objekta van sistema i njegovo ponovno vraćanje.

U našem sistemu, **Memento patern** je moguće iskoristiti prilikom popunjavanja i biranja simptoma koje Osoba osjeća, nakon odabira svih simptoma i klika na OK, korisniku bi bilo ponuđeno da potvrdi svoju prijavu simptoma ili da se se BACK vrati na biranje simptoma gdje bi mogao označene simptome odznačniti ili neoznačene naknadno označiti.

Mediator patern – Enkapsulira protokol za komunikaciju među objektima dozvoljavajući da objekti komuniciraju bez međusobnog poznavanja interne strukture objekta.

U našem sistemu, Mediator patern je moguće iskoristiti prilikom komunikacije pacijenta sa doktorom i prilikom opisivanja stanja pacijenta. Poruka bi se slala Mediatoru, koji bi pacijentovu poruku proslijedio dalje određenom doktoru.