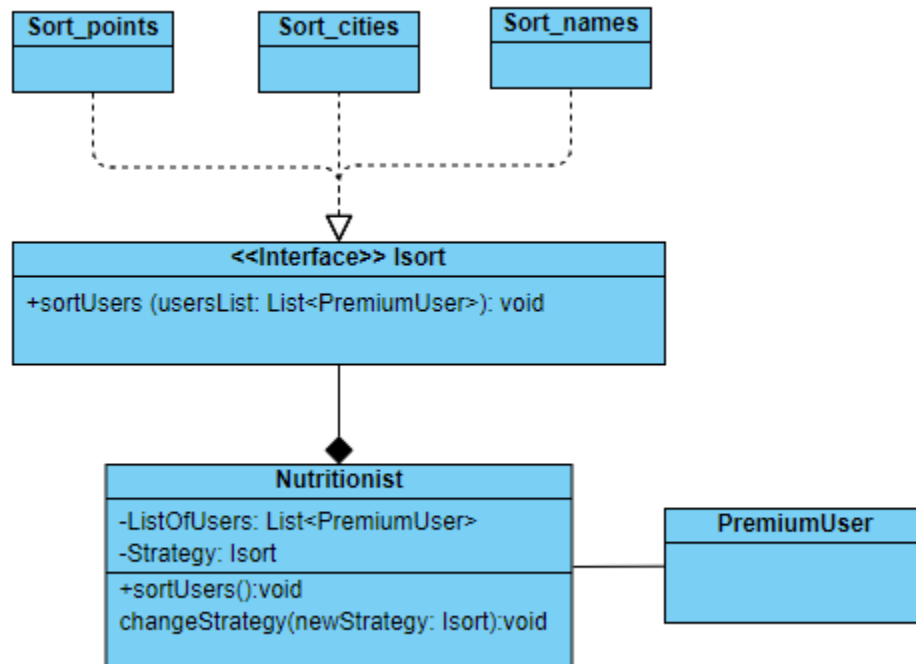


PATTERNI PONAŠANJA

Strategy pattern:

Strategy pattern se može primijeniti na način da se uvedu tri algoritma sortiranja premium korisnika koje nutricionista može koristiti. On će birati da li želi sortirati premium korisnike po bodovima, po gradovima ili abecedno.



State pattern:

State pattern se može primijeniti u MyNutritionist aplikaciji za upravljanje različitim stanjima ili modima aplikacije. U ovom slučaju, možemo definirati apstraktnu klasu "KorisnikState" koja predstavlja stanje korisnika. Imamo dvije konkretne implementacije ove klase: "RegistrovaniKorisnikState" i "PremiumKorisnikState". Klasa "Korisnik" bi imala atribut "state" koji bi bio instanca apstraktne klase "KorisnikState". Na početku, kada je korisnik registrovan, "state" atribut bi bio postavljen na "RegistrovaniKorisnikState". Kada korisnik odluči nadograditi na premium korisnika, pozivao bi se odgovarajući metod, npr. "nadogradiNaPremium()". U okviru metoda "nadogradiNaPremium()" unutar klase "RegistrovaniKorisnikState", mijenjali bismo stanje korisnika tako što bismo postavili "state" atribut na instancu "PremiumKorisnikState". Ova promjena stanja bi se odrazila na ponašanje korisnika jer bi se korisniku omogućile nove funkcionalnosti koje su dostupne samo premium korisnicima. Na ovaj način, State pattern omogućava fleksibilno upravljanje stanjima korisnika i omogućava jednostavno prelazak iz jednog stanja u drugo, kao što je prelazak iz registrovanog korisnika u premium korisnika.

Template method pattern:

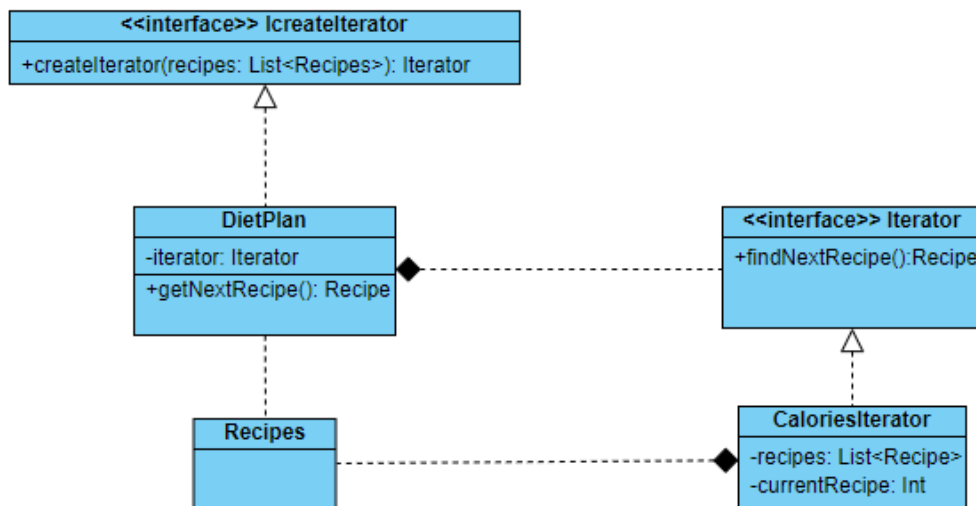
Template method pattern se može primijeniti u MyNutritionist aplikaciji za definiranje okvira ili strukture algoritama koji se koriste u različitim koracima ili procesima. Na primjer, možemo imati apstraktnu klasu "PlanIshraneTemplate" koja definiše metode poput "kreirajDorucak()", "kreirajRucak()", "kreirajVeceru()", a zatim podklase poput "JednodnevniPlanIshrane" i "SedmodnevniPlanIshrane" koje nasljeđuju "PlanIshraneTemplate" i pružaju konkretne implementacije tih metoda. Template method pattern omogućava da definišemo zajedničku strukturu algoritama, ali omogućava podklasama da implementiraju specifične korake u okviru te strukture.

Observer pattern:

Observer pattern se može primijeniti u MyNutritionist aplikaciji za praćenje promjena u podacima ili stanjima i obavještavanje relevantnih objekata. Na primjer, možemo imati subjekt "NapredakKorisnika" koji prati napredak korisnika u ishrani i tjelesnoj aktivnosti, a zatim možemo imati različite observere poput "Korisnik", "Nutricionist" ili "LeaderBoard" koji se registruju kod subjekta i primaju obavijesti o promjenama u napretku.

Iterator pattern:

Iterator pattern se može primijeniti u MyNutritionist aplikaciji za olakšavanje prolaska kroz različite kolekcije objekata ili podataka. Na primjer, možemo imati kolekciju "Recepti" ili "Namirnice" i zatim možemo imati Iterator interfejs koji definiše metode poput `pređiNaSljedeći()`. Zatim možemo imati konkretne implementacije Iteratora za te kolekcije koje omogućavaju iteriranje kroz objekte.



Chain of responsibility pattern:

Chain of responsibility pattern se može primijeniti u MyNutritionist aplikaciji za obradu upgrade-a u premium korisnika. Imamo lanac odgovornosti koja se sastoji od tri koraka: Korisnik, Administrator i Nutricionist.

Korisnik:

Klasa "Korisnik" predstavlja prvi korak u lancu odgovornosti. Kada korisnik želi nadograditi svoj račun na premium status, prvo korisnik unosi svoje podatke i odlučuje se za nadogradnju. Ako korisnik odluči nadograditi račun, tada se korak prenosi na sljedeći član lanca - Administratora.

Administrator:

Klasa "Administrator" predstavlja drugi korak u lancu odgovornosti. Administrator provjerava da li je korisnik ispravno unio broj kartice i izvršava uplatu za nadogradnju na premium status. Ako je uplata uspješna, korak se prenosi na sljedeći član lanca - Nutricionista.

Nutricionist:

Klasa "Nutricionist" predstavlja treći korak u lancu odgovornosti. Kada korisnik postane premium korisnik, Nutricionist prima obavijest o novom korisniku. Nutricionist može zatim dodijeliti personalizirani plan ishrane i pružiti dodatne informacije i podršku premium korisniku.

Svaki korak u lancu odgovornosti ima svoju logiku za provjeru i izvršavanje odgovarajućih zadataka. Ako korak nije u mogućnosti obraditi zahtjev, može prenijeti zahtjev na sljedeći korak u lancu ili ga odbiti.

Mediator pattern:

Koristeći Mediator pattern, omogućava se fleksibilna i jednostavna komunikacija između različitih sudionika u aplikaciji MyNutritionist. KomunikacijskiMedijator preuzima odgovornost za distribuciju poruka i omogućava različitim klasama da komuniciraju bez potrebe da izravno znaju jedni za druge. Kada korisnik šalje poruku putem Mediatora, ona se proslijeđuje svim ostalim sudionicima koji mogu biti zainteresirani za tu poruku. Na primjer, ako korisnik šalje poruku o nadogradnji računa, ta poruka se šalje administratoru i nutricionistu.