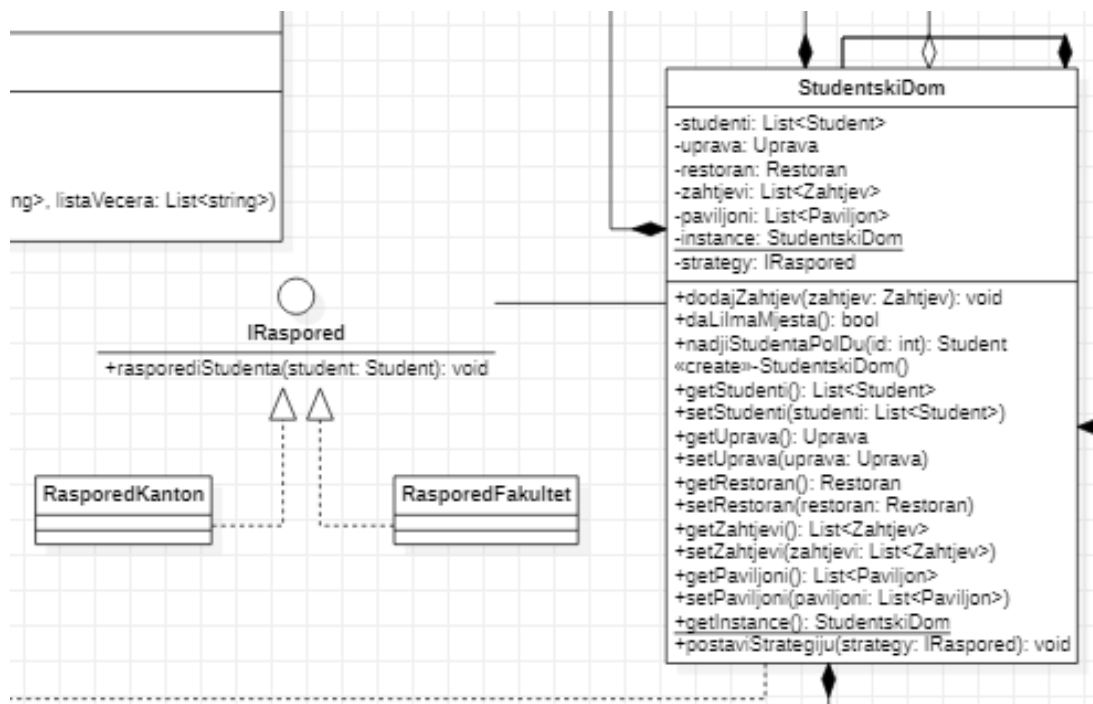


## Strategy patern:

Napravili smo interfejs `IStrategy` koji ima metodu `rasporediStudenta`. Klasa `Stragetgy` nasljeđuje intefejs `IStrategy` koja implementira metodu `rasporediStudenta`. U zavisnosti od načina na koji želimo rasporediti studenta možemo implementirati više klasa `Strategy` (klasa `Strategy` je samo generalni primjer). Mi smo se odlučili da trenutno implementiramo klase `RasporedKanton` i `RasporedFakultet`, i one implementiraju interfejs `IStrategy`, tako što klasa `RasporedKanton` predstavlja kriterij za raspoređivanje studenata koji dolaze iz istog kantona, kod `RasporedFakultet` predstavlja kriterij za raspoređivanje studenata koji idu na isti fakultet.

Metoda `rasporediStudenta` se implementira na način da traži sobu koja ima praznih mjesta i u kojoj se nalazi student koji ima isti kriterij raspoređivanja. Ako je uspješno pronađena soba, onda se taj student smjesti u tu sobu, u suprotnom se traži prazna soba u koju će biti smješten taj student. Ako se ne nađe ni prazna soba student se raspoređuje po drugom kriteriju.

Klasa `StudentskiDom` posjeduje metodu `upisiStudenta` koja će dodati studenta u listu svih studenata, i preko privatnog atributa `strategy` poziva metodu `rasporediStudenta`, koja će po određenom kriteriju rasporediti studenta u neku od soba. Ove funkcionalnosti možemo vrlo lahko proširiti tako što deklariramo novu klasu koja implementira interfejs `IRaspored` i implementiramo metodu `rasporediStudenta`.



## Template Method patern:

Napravili smo klasu apstraktnu klasu algoritam koja ima tri metode, i to: templateMethod, sortiraj i filtriraj. Sve tri klase imaju svoju implementaciju, ali će klasa Algoritam biti apstraktna iz razloga da se ne bi mogla instancirati, jer ona nema smisla ako se bar jedna metode ne override. Metode sortiraj i filtriraj primaju listu tipa Student, i vraćaju listu tipa Student. Inicijalno metode ne rade ništa, to jest vraćaju istu listu koju primaju kao parametar, ali će one imati neku funkciju tek kada se klasa izvede. Metoda templateMethod prvo poziva funkciju filtriraj, pa onda funkciju sortiraj. U zavisnosti da li izvedena klasa override metodu sortiraj ili filtriraj, jedino ta metoda će i imati uticaja na kranji rezultat, odnosno samo će se studenti filtrirati ili sortirati, a ne oboje, s tim da ako izvedena klasa override obje metode onda će se lista studenata i filtrirati i sortirati.

Imamo i dvije izvedene klase KantonFilter i GodinaStudijaSort. Klasa KantonFilter prima parametar u konstruktoru tipa string, koji predstavlja ime kantona, i ona override metodu filtriraj koja vraća sve studente koji su iz tog kantona. Klasa GodinaStudijaSort prima parametar u konstruktoru tipa bool, koji predstavlja ime način sortiranja, to jest da li će se studenti sortirati po opadajućem ili rastućem kriteriju, i ona override metodu sortiraj koja vraća istu listu studenata samo sortiranu u odgovarajućem redoslijedu.

Klasa StudentskiDom trenutno ima dvije metode sortirajPoKantonu i filtrirajPoGodini, koje primaju listu studenata i jos jedan odgovarajući parametar, a vraća malo izmjenjenu listu studenata. U ovim metodama se instancira klasa algoritam odgovarajućom izvedenom klasom i na osnovu nje se ta lista sortira, odnosno filtrira.

Sve ove opcije za sortiranje i filtriranje možemo vrlo lahko proširiti tako što samo izvedemo nove klase koje nasljeđuju klasu Algoritam, i samo implementiramo nove metode u klasi StudentskiDom koje se interno izvršavaju preko te izvedene klase.

