Odlučili smo se za implementaciju sljedeća četiri patterna:

- Proxy
- Composite
- Singleton
- Prototype

Proxy

Ovaj pattern smo implementriali na sljedeći način. Pošto je Uprava jedina ta koja ima privilagije da upravlja studentskim domom, Proxy je implementiran tako da sve funkcionalnosti vezane za studentski dom, izvršava jedino ako se zaključi da je korisnik koji je ulogovan na sistem zapravo Uprava. To se provjeri tako što se provjeri username i password koji je unesen, koji se čuva kao atribut u klasi Proxy (atribut korisnik), i ako on odgovara onome koji je dodijeljen upravi, dozvoljene su modifikacije u studentskom domu. Imamo interfejs IStudentskiDom koji implementiraju i klasa StudenstskiDom, kao i Proxy klasa, koji u sebi sadrži sve operacije koje se vrše nad studenstkim domom. U klasi Proxy, ako se ustanovi da je username i password onaj koji odgovara upravi, postavlja se nivoPristupa na odgovarajuću vrijednost. Također sve metode u klasi Proxy, prvo provjeravaju da li nivoPristupa ima odgovarajuću vrijdnost, a zatim ako ima, pozivaju se odgovarajuće metode nad atributom studentskiDom.

Composite

Ovaj pattern smo implementirali na sljedeći način. Uveli smo dvije vrste studenata: Redovni i Ponovac koji implementiraju interfejs IStudent. Jedina metoda u interfejsu je metoda uplatiDomZaOdabraniMjesec. Sada je Blagajna urađena na taj način da ima atribut tipa IStudent, i da prilikom uplate doma za određeni mjesec poziva se metoda uplatiDomZaOdabraniMjesec nad atributom koji je tipa IStudent i u zavisnosti od toga da li je atribut tipa Redovni ili Ponovac, pozivat će se različita implementacija metode.

Singleton

Odlučili smo da singleton klasa u našem sistemu bude StudenskiDom, jer nam je potrebna samo jedna instanca i treba biti dostupna svima. Implementirali smo je tako što smo u StudenstkiDom dodali referencu na sebe, odnosno dodali statički atribut instance tipa StudenstskiDom, privatni konstruktor i statičku metodu getInstance koja služi da dohvaćanje atributa instance. U getInstance ako je atribut instance null referenca, odnosno ako već nije instanciran, pozivamo konstruktor, odnosno vršimo instantaciju objekta. Ako nije null, jednostavno vratimo iz metode atribut instance.

Prototype

Prototype pattern je implementiran na način tako što smo napravili interfejs ISoba koji sadrži metodu clone. Klasa Soba implementira taj interfejs kao i metodu clone, koja služi za kreiranje klonova objekta Soba. Ovim putem ćemo umjesto instaciranjem novog objekta, vršiti kloniranje određenog objekta i samo izvršiti promjenu atributa brojSobe. Time smo izbjegli potrebu za stalnim instanciranjem novih objekata klase Soba.