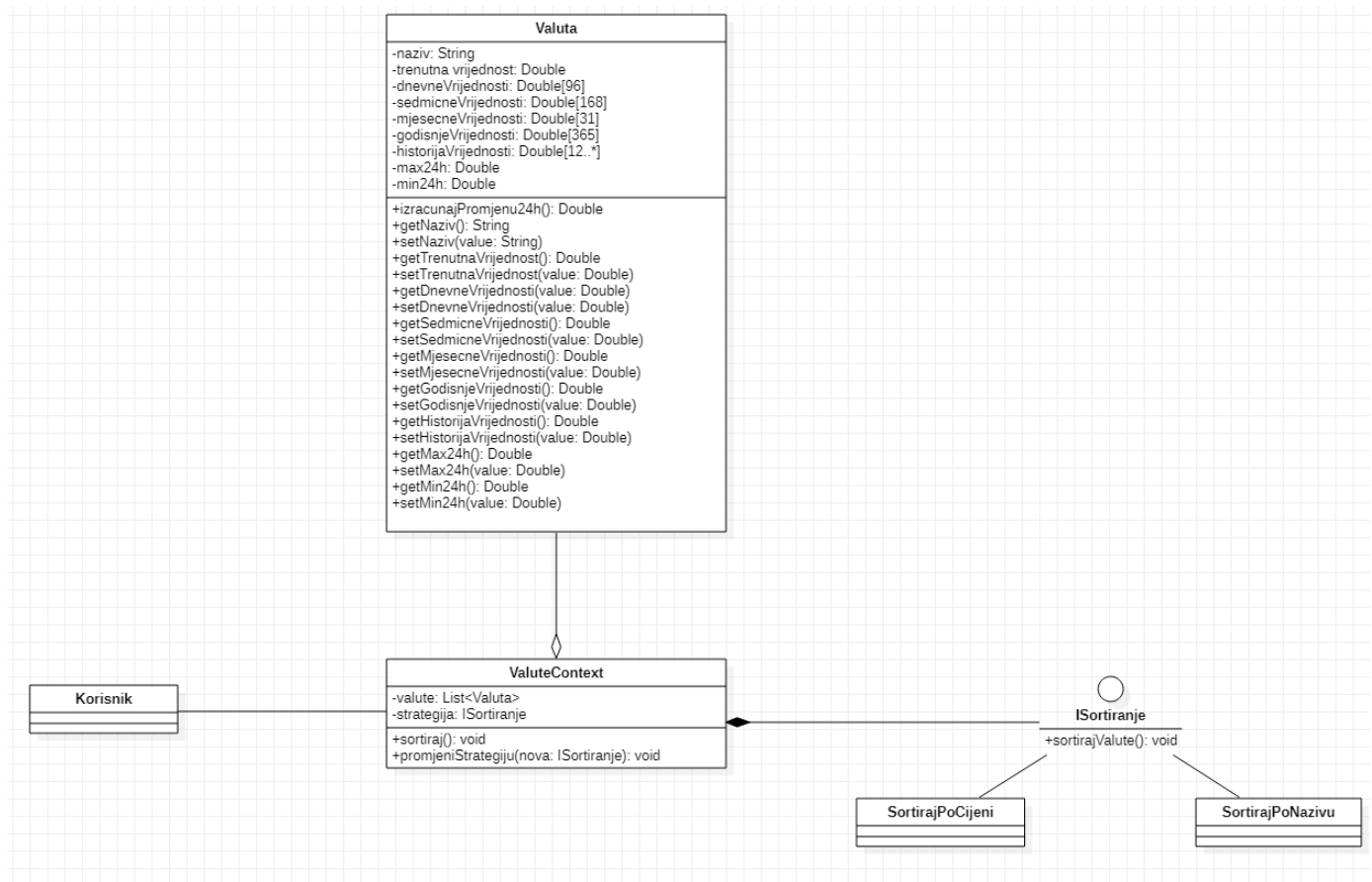


Paterni ponašanja

Strategy patern

Strategy patern izdvaja algoritam iz matične klase i uključuje ga u posebne klase. Pogodan je kada postoje različiti primjenljivi algoritmi (strategije) za neki problem. Strategy patern omogućava klijentu izbor jednog od algoritama iz familije algoritama za korištenje. Algoritmi su neovisni od klijenata koji ih koriste. Podržava open-closed princip.

- Unutar aplikacije *CryptoInvest*, *Strategy patern* ćemo primijeniti na način da svaki korisnik ima mogućnost prikaza kriptovaluta sortiranih po trenutnoj vrijednosti ili po nazivu. Za tu svrhu potrebno je dodati klasu *ValutaContext* koja će interfejsu *ISortiranje* proslijediti potrebne podatke. Navedeni interfejs sadrži metodu *sortirajValute*.



State patern

State patern mijenja način ponašanja objekta na osnovu trenutnog stanja. State patern predstavlja dinamičku verziju Strategy paternu. Postiže se promjenom podklase unutar hijerarhije klasa.

- *Unutar aplikacije CryptoInvest, State patern bi mogao biti upotrijebljen u okviru transakcije, kada korisnik ima mogućnost odabira LimitBuyOrder odnosno limitiranje naloga kupovine. Naime, ukoliko korisnik odluči da ne želi odmah izvršiti kupovinu može iskoristiti navedenu opciju kako bi odabrao kriptovalute koje želi kupiti ali tek kada dostignu određenu cijenu. To znači da u slučaju kada korisnik odabere kriptovalute koje bi želio kupiti, kupovina je neaktivna sve dok ne dostignu cijenu koju je korisnik postavio. Kada su uslovi ispunjeni kupovina postaje aktivna, te se izvršava automatski.*

Template method patern

Template Method patern omogućava algoritmima da izdvoje pojedine korake u podklase. Struktura algoritma se ne mijenja – mali dijelovi operacija se izdvajaju i ti dijelovi se mogu implementirati različito.

- *Unutar aplikacije CryptoInvest, nije korišten Template method patern, ali u slučaju da smo iskoristili navedeni patern to bi moglo biti u okviru pristupa kursevima. Korisnik najprije bira koji kurs želi pohađati, pri čemu postoje 3 vrste kurseva Beginner, Medium i Hard. Nezavisno od kursa, korisnik bira da li nakon preslušanih lekcija želi pristupiti testu. Beginner ima 3 lekcije, Medium 4, dok Hard ima 5 lekcija. Također, unutar Hard kursa postoji mogućnost izbora da li korisnik želi jednu dodatnu lekciju dok unutar Beginner i Medium levela ta mogućnost ne postoji.*

Observer patern

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

- *Unutar aplikacije CryptoInvest, Observer bi mogao biti iskorišten u okviru uključivanja obavještenja o promjenama na tržištu. Naime, unutar aplikacije korisnici bi trebali biti obaviješteni o promjenama na tržištu kako bi jednostavnije kupovali i prodavali kriptovalute. U tom slučaju potreban je interfejs IKorisnik koji bi imao metodu posaljiObavjestenje. Unutar klase Transakcija potreban je broj svih korisnika koji su izvršili transakciju, te metoda obavijestiKorisnika.*

Iterator patern

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

- *Unutar aplikacije CryptoInvest, Iterator patern bi mogao biti iskorišten u okviru Portfolia gdje imamo listu Transakcija. U tom slučaju, TransakcijeIterator će kao privatni atribut imati listu transakcija, kroz koju će se iterirati. Također, potreban je i interfejs IEnumerable koji sadrži metodu GetEnumerator().*

Chain of responsibility patern

Chain of responsibility patern predstavlja listu objekata, ukoliko objekat ne može da odgovori prosljeđuje zahtjev narednom u nizu.

- *Unutar aplikacije CryptoInvest, Chain of responsibility patern bi mogao biti iskorišten ukoliko bi postojala mogućnost da korisnici između sebe razmjenjuju kriptovalute. U tom slučaju jedan korisnik bi poslao zahtjev drugom. Kako bi navedeno bilo moguće potrebna je potvrda od korisnika te administratora. Naime, jedan korisnik bi poslao drugom zahtjev za kupovinu, te bi za tu svrhu bila potrebna klasa ZahtjevZaKupovinu, interfejs IHandler, te određene klase koje bi bile naslijeđene iz klase ZahtjevZaKupovinu.*

Mediator patern

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

- *Unutar aplikacije CryptoInvest, Mediator patern bi mogao biti iskorišten u slučaju kada bi bilo potrebno ostvariti interakciju između korisnika. U tom slučaju, korisnici bi na određenom forumu mogli ostavljati svoje utiske o trgovanju, gdje bi mogućnost čitanja komentara imali svi korisnici, dok bi mogućnost pisanja imali samo korisnici koji imaju barem jednu izvršenu transakciju. Također, komentari koji imaju neprimjeren sadržaj bili bi odbačeni. Za tu svrhu, potreban je interfejs IKomentarMedijator, te unutar klase Korisnik atribut IKomentarMedijator.*