# DIZAJN PATERNI PONAŠANJA

## Strategy

**Svrha**: Omogućava različite načine prikaza statistike u zavisnosti od uloge korisnika.

**Uloga u sistemu**:

* Pregled statistike uspjeha učenika (nastavnik, roditelj, učenik).

**Klase**:

* StrategijaStatistike *(interfejs ili apstraktna klasa)*
* StatistikaNastavnik, StatistikaRoditelj, StatistikaUcenik
* Korisnik ima atribut strategija : StrategijaStatistike

**Metode**:

* prikaziStatistiku(), izracunajProsjek(), brojIzostanaka(), vladanje()

## State

**Svrha**: Ponašanje objekta se mijenja zavisno od njegovog trenutnog stanja (prisutan/odsutan učenik).

**Uloga u sistemu**:

* Prisustvo učenika – stanje određuje da li se može ocijeniti.

**Klase**:

* StanjePrisustva (interfejs)
* Prisutan, Odsutan (implementacije)
* Ucenik ima atribut stanjePrisustva : StanjePrisustva

**Metode**:

* ocjeni(), promijeniStanje(StanjePrisustva novoStanje)

## Template Method

**Svrha**: Definisanje koraka algoritma u nadklasi, uz delegiranje konkretnih koraka podklasama.

**Uloga u sistemu**:

* Generisanje izvještaja o učeniku (ocjene, izostanci, vladanje).

**Klase**:

* IzvjestajGenerator (apstraktna klasa)
* IzvjestajRazrednika, IzvjestajRoditelja
* Ucenik, Ocjena, Izostanak, Razred

**Metode**:

* generisiIzvjestaj(), dodajOcjene(), dodajIzostanke(), dodajVladanje()

## Observer

**Svrha**: Notifikacija zainteresovanih klasa kada dođe do promjene stanja.

**Uloga u sistemu**:

* Notifikacija razredniku kad nastavnik unese ocjenu.

**Klase**:

* Observer (interfejs), Razrednik, Roditelj
* Ocjena poziva notify()
* Subject interfejs, implementiran u npr. DnevnikOcjena

**Metode**:

* dodajObserver(Observer o), notifyAll(), update()

## Iterator

**Svrha**: Sekvencijalni pristup elementima bez otkrivanja interne strukture.

**Uloga u sistemu**:

* Pregled ocjena, učenika, izostanaka.

**Klase**:

* OcjeneIterator, UceniciIterator, IzostanciIterator
* Razred implementira Iterable

**Metode**:

* hasNext(), next(), getIterator()

## Command

**Svrha**: Enkapsulira sve informacije potrebne za izvršenje komande.

**Uloga u sistemu**:

* Dodavanje učenika od strane administratora kroz više koraka.

**Klase**:

* Command (interfejs), AddStudentCommand, EditStudentCommand
* Invoker – klasa AdminController
* Receiver – StudentService

**Metode**:

* execute(), undo()

## Memento

**Svrha**: Čuvanje prethodnog stanja objekta.

**Uloga u sistemu**:

* Razrednik uređuje izvještaj i može vratiti prethodnu verziju.

**Klase**:

* Izvjestaj (Originator), IzvjestajMemento, Caretaker

**Metode**:

* spasiStanje(), vratiStanje()

## Mediator

**Svrha**: Centralizuje komunikaciju između objekata.

**Uloga u sistemu**:

* Chat sistem (učenik ↔ nastavnik ↔ roditelj).

**Klase**:

* ChatMediator (interfejs), GlobalniChat (konkretna klasa)
* Korisnik (učenik, roditelj, nastavnik...) šalju poruke kroz mediator

**Metode**:

* posaljiPoruku(String poruka, Korisnik posiljalac), registrujKorisnika(Korisnik k)