

**Za implementaciju smo odabrali strukturalne paterne proxy i decorator.**

## **Adapter**

Adapter omogućava povezivanje klasa koje imaju nespojive interfejsa. Omotava postojeću klasu i prevodi njen interfejs u oblik koji druga klasa može koristiti.

### **Potreba u našem sistemu?**

Eksterni sistem (npr. email/SMS servis) koristi drugačiji interfejs od klase obavijest. Umjesto direktne integracije može se koristiti adapter koji omogućava da naš sistem šalje obavijesti koristeći postojeći interfejs klase obavijest, ali uz pozadinsko korištenje eksternog API-ja.

## **Facade**

Facade pruža pojednostavljen interfejs za složen sistem tj. objedinjuje više funkcionalnosti u jednu ulaznu tačku i time smanjuje složenosti i povećava čitljivost.

### **Potreba u našem sistemu?**

Bilo koja složenija operacija poput: evidencije prisustva, slanja maila uz neku akciju ili generisanja mjesečnog izvještaja, se može objединiti u neku facade klasu koja smanjuje složenost operacija.

## **Decorator**

Decorator omogućava dinamičko dodavanje dodatnih odgovornosti objektima bez mijenjanja njihove osnovne strukture.

### **Potreba u našem sistemu?**

Obavijestima se mogu dodavati neki posebni ukrasi. Umjesto da mijenjamo klasu obavijest, može se koristiti nekoliko dekoratora koji dodaju nove funkcionalnosti.

## **Bridge**

Bridge odvaja apstrakciju od njene implementacije kako bi obje strane mogle nezavisno da se mijenjaju.

### **Potreba u našem sistemu?**

Kada bi imali aktivnosti koje se mogu izvoditi na različite načine sa različitim izvedbama, onda umjesto da svaka varijanta ima posebnu klasu, koristili bi bridge pattern.

## **Composite**

Composite omogućava da se objekti organizuju u neku hijerarhiju, gdje se pojedinačni i složeni objekti tretiraju na isti način.

### **Potreba u našem sistemu?**

Služilo bi nam da možemo djecu grupisati po nekim kriterijima i onda da možemo izvršavati istu operaciju nad svakom djetetu iz tog composite-a.

## **Proxy**

Proxy kontrolira pristup stvarnom objektu, može dodavati sigurnosne provjere, keširanje, logovanje itd.

### **Potreba u našem sistemu?**

Pristup podacima o djetetu treba biti ograničen, pa bi trebali koristiti zaštitni proxy nad klasom djetete.