Univerzitet u Sarajevu
Elektrotehnički Fakultet
Objektno Orijentisana Analiza i Dizajn

STRUKTURALNI PATERNI U PROJEKTU "CHARITY FOUNDATION"

Strukturalni paterni omogućavaju fleksibilniju i efikasniju organizaciju sistema kombinovanjem klasa i objekata u veće strukture. U nastavku je prikazana primjena strukturnih paterna u kontekstu sistema za upravljanje humanitarnom fondacijom.

1. Adapter Pattern

Omogućava povezivanje klasa s nekompatibilnim interfejsima.

U sistemu imamo različite načine unosa podataka o donacijama (ručni unos, uvoz iz Excel fajla, API iz drugih platformi). Podaci dolaze u različitim formatima. Možemo koristiti DonacijaAdapter koji implementira zajednički interfejs IDonacija i omogućava konverziju u standardizovani format koji sistem koristi. Time se omogućava proširivost bez izmjena postojećih klasa.

2. Decorator Pattern

Omogućava dinamičko dodavanje funkcionalnosti postojećim objektima.

Decorator patern može se primijeniti na sistem obavještavanja korisnika. Osnovni Notifier može slati email obavijesti, ali možemo dodati dekoratore kao što su SMSNotifier, PushNotifier, InAppNotifier, koji proširuju osnovno ponašanje. Na taj način, bez izmjena postojećeg koda, možemo slati višekanalne obavijesti korisnicima i volonterima.

3. Proxy Pattern

Omogućava kontrolisani pristup stvarnim objektima.

Pristup osjetljivim podacima kao što su lični podaci donatora i volontera treba biti ograničen. UserProxy može implementirati provjere ovlaštenja prije nego što se dozvoli pristup klasama kao što su DonorProfile ili VolunteerInfo. Tako npr. samo administratori mogu vidjeti sve podatke, dok obični korisnici vide samo osnovne informacije.

4. Composite Pattern

Omogućava da se pojedinačni objekti i njihove kompozicije tretiraju jednako.

Pri raspodjeli donacija, često se grupišu korisnici po kategorijama (npr. porodice s djecom, starije osobe, osobe s invaliditetom). Možemo napraviti hijerarhiju korisnika korištenjem Composite paterna, gdje Korisnik i GrupaKorisnika implementiraju zajednički interfejs IRecipient, i poziv metode raspodijeliDonaciju() funkcioniše jednako bez obzira da li se radi o pojedincu ili grupi.

5. Facade Pattern

Pojednostavljuje interakciju s kompleksnim podsistemima nudeći jedinstveni interfejs.

Sistem sadrži više podsistema – upravljanje donacijama, korisnicima, obavijestima i izvještajima. Možemo kreirati CharityFacade koji pruža metode poput kreirajDonaciju(), posaljiObavijest(), generisiIzvjestaj(). Klijenti sistema (npr. UI sloj) komuniciraju samo s fasadom, bez potrebe da direktno poznaju sve podsisteme.

6. Flyweight Pattern

Dijeljenje zajedničkog stanja među velikim brojem sličnih objekata.

U sistemu može postojati veliki broj korisnika koji pripadaju istim kategorijama pomoći. Npr. korisnici s istim tipom pomoći (“paket hrane”, “školski pribor”) mogu dijeliti isti objekat TipPomoci. Umjesto da svaki korisnik ima zaseban objekat, koristi se FlyweightFactory za dijeljenje istog objekta kad je to moguće – što štedi memoriju.

7. Bridge Pattern

Omogućava nezavisno mijenjanje apstrakcije i implementacije.

Pri generisanju izvještaja, možemo koristiti Bridge patern kako bismo odvojili vrstu izvještaja (Izvjestaj) od načina generisanja (PDFGenerator, ExcelGenerator, WebReportGenerator). Kroz interfejs IReportGenerator, izvještaj može koristiti bilo koji format bez potrebe za direktnom vezom između svih kombinacija tipova izvještaja i izlaza.