

Oblikovni obrasci u programiranju

drugi međuispit

1. Razmatramo sustav za izmjenu signalnih poruka između brodova i operatera na obali. Brod radiovezom odašilje poruku, poruka se prima na jednoj od antena uz obalu i putem TCP/IP veze šalje na operatersko računalo. Postoje različite vrste poruka, a vrsta poruke se određuje na temelju prvih 10 primljenih bitova. Za svaku poruku je potrebno odraditi drugačiju obradu, ali sve poruke moraju obrađene.

Potrebno je izraditi strukturni dijagram ključnog dijela organizacije programa koji će koristiti operateri na obali. Pri tome je potrebno zadovoljiti sljedeće uvjete:

- čitanje poruka odvija se u metodi `mainloop()` razreda `TheMessageApplication`;
- rukovanje svakom pojedinom vrstom poruke potrebno je implementirati u okviru zasebnog razreda izvedenog iz `MessageProcessor`;
- za svaku poruku moramo podržati sljedeće metode:
 - inicijalizaciju s primljenim binarnim kôdom,
 - formatirani ispis poruke prikladan za inspekciju od strane operatera,
 - arhiviranje poruke u bazi podataka,
 - paljenje odgovarajućeg zvučnog alarma koji ovisi o trenutnim okolnostima i konkretnoj poruci;
- (bonus) dodavanje novih vrsta poruka u sustav ne smije uzrokovati promjenu niti jedne postojeće komponente programa.

(10 bodova)

2. Sustav za logiranje pogrešaka namijenjen je ispisivanju poruka o pogrešci u datoteku. Postoji više vrsta podataka koji se mogu ispisati poput: kratka poruka o pogrešci, duga poruka o pogrešci, redak u kojem je došlo do pogreške, vrijeme u kojem je došlo do pogreške, naziv računala u kojem je došlo do pogreške.

Kod svake pogreške nužno se ispisuje kratka poruka o pogrešci. Potrebno je izgraditi strukturni dijagram sustava i napisati odgovarajući kod gdje korisnici u toku izvođenja aplikacije mogu proizvoljno određivati koji podaci o pogreškama se ispisuju.

Svaka vrsta podatka o pogrešci treba biti ostvarena u zasebnom razredu. Zabranjeno je korištenje višestrukog naslijeđivanja, te je potrebno omogućiti nadogradnju sustava nekim novim podacima o pogrešci (poput ispisa IP adrese računala na kojem se pogreška desila) bez modificiranja postojećih komponenata.

(10 bodova)

3. Razmatramo razred za pronalaženje određenih vrsta znakovnih nizova u tekstu, koji je implementiran kao konačni automat. Podaci za stanja i prijelaze automata se učitavaju iz datoteke u konstruktoru razreda i ne mijenjaju se tijekom izvođenja programa. Metoda `find` vraća prvu poziciju na kojoj je pronađena zadana vrsta znakovnog niza ili -1 u slučaju da niz nije pronađen.

```
class Automat{
public:
    Automat(const std::string &file);
public:
    int find(const std::string &text);
};
```

U aktualnoj primjeni veći broj klijenata koristi usluge automata, i to svaki preko svoje privatne instance. S obzirom na to da su traženi uzorci tipično vrlo složeni, javio se problem da podatci stanja i prijelaza zauzimaju značajnu količinu memorije.

Predloži izmjene tako da se osigura što efikasnija uporaba memorijskih resursa, te što brže pokretanje cjelokupnog programa. (10 bodova)

4. Razred `x` poziva metodu apstraktne tvornice `T`, te šalje dobiveni rezultat `P` konfiguracijskoj metodi konteksta `K` obrasca strategije.

Nacrtaj dijagram razreda opisane organizacije, označi koji razred predstavlja kojeg sudionika dvaju obrazaca te skiciraj odgovarajući izvorni kod.

(10 bodova)

5. Pretpostavimo da oblikujemo paket za evidenciju klijenata internetskog dućana. Kad god se u bazu klijenata unese novi klijent, potrebno je napraviti sljedeće akcije:

- (a) poslati novom korisniku e-mail dobrodošlice
- (b) provjeriti da li je unesena valjana poštanska adresa
- (c) aktualizirati prosječnu dob svih korisnika

Očekujemo da ćemo u skoroj budućnosti računati i druge statističke parametre korisnika (npr, kućna adresa), kako bismo mogli prilagođavati naslovnici dućana registriranim korisnicima. Htjeli bismo omogućiti dodavanje takvih akcija bez potrebe za mijenjanjem postojećeg koda. Koji obrazac bi se mogao primijeniti u ovom slučaju? Nacrtaj dijagram razreda, te skiciraj programsku izvedbu. (10 bodova)

6. Zadana je funkcija za ostvarivanje veze s udaljenim računalom preko modema.

```
void LogOn(Modem& m, string& pno,
           string& user, string& pw)
{
    if (m.type == Modem::hayes)
        DialHayes((Hayes&)m, pno);
    else if (m.type == Modem::courrier)
        DialCourrier((Courrier&)m, pno);
    else if (m.type == Modem::ernie)
        DialErnie((Ernie&)m, pno)
    // ...
}
```

Primjećujemo da funkcija ne omogućava transparentno dodavanje novih vrsta modema. Kojim obrascem bismo to mogli postići? Ideju ilustriraj prikladnim strukturnim dijagramom i izvornim kodom. (10 bodova)

7. U koji oblikovni obrazac bi se mogao uklopiti sljedeći razred? Opiši uloge razreda X, Y, Z i W u okviru obrasca. Iz kojih komponenti se pozivaju metode m1 i m2? Nacrtaj dijagram razreda obrasca, te skiciraj programske implementacije ostalih sudionika.

```
class X{
    // ...
};
class Y{
public:
    void m1(); //calls m2()
    //...
private:
    virtual X* m2()=0;
    //...
};
class Z: public X{
    // ...
};
class W: public Y{
    virtual X* m2();
};
```

(10 bodova)