

Oblikovni obrasci u programiranju međuispit

Napomena: prihvaćaju se programska rješenja u C++-u, Javi, Pythonu i C#-u.

1. Razmatramo razred `RedPoslova` koji nudi mogućnost usporednog izvođenja većeg broja poslova. Poslovi se zadaju primjercima bilo kojeg razreda koji implementira sučelje `Posao` koje deklarira jednu apstraktnu metodu `void izvrsi()`. Razred `RedPoslova` nudi metodu `void dodaj(Posao p)` koja u red poslova ubacuje predani `posao`. Razred `RedPoslova` prilikom inicijalizacije stvara jednu ili više dretvi koje iz reda vade jedan po jedan `posao` i pozivaju njihove metode `izvrsi()`. Razred `RedPoslova` isporučuje se u okviru binarne biblioteke, tj. njegov izvorni kod nije dostupan.

Javila se potreba da se pri radu s razredom `RedPoslova` omoguće sljedeće dodatne funkcionalnosti:

- mjeriti koliko vremena je trajalo izvođenje posla i ako je to vrijeme dulje od nekog zadanog, na zaslon ispisati odgovarajuću poruku;
- u log datoteku ispisati datum i vrijeme kada je započelo izvođenje posla kao i identifikator dretve koja izvodi taj `posao`;
- predvidivo je da bi ovakvih zahtjeva moglo biti još.

Koji je oblikovni obrazac prikladan za rješavanje opisanog problema? Nacrtajte strukturni dijagram i naznačite njegove sudionike.

Definirajte tri posla. P1: izračunati sumu prvih 100 parnih brojeva; P2: izračunati s koliko je manjih nenegativnih brojeva djeljiv broj 5742; P3: provjeriti je li broj 3117 prost broj. Sva tri posla odgovor ispisuju na zaslon. Pri izvršavanju posla P1 treba odraditi i funkcionalnosti (1+2); pri izvršavanju P2 treba odraditi još i funkcionalnost (2). Napišite cjelovito programsko rješenje ovog zadatka. Za rješenje koje ste napisali nacrtajte dijagram razreda.

2. Razmatramo internacionalizaciju programa s grafičkim korisničkim sučeljem. Pretpostavite da sve grafičke komponente (gumbi, labele, ...) umjesto konkretne poruke koju moraju ispisati pamte jedinstveni ključ poruke. Pripremili smo tekstovne datoteke u kojima su dani prijevodi poruka na različite jezike. Tako u poruke-hr.txt imamo 1=Pozdrav, 2=Izbornik, a u poruke-en.txt imamo 1=Hello, 2=Menu. U svakom trenutku je aktivan samo jedan jezik. Sve grafičke komponente znaju koji ključ prikazuju pa do prijevoda dolaze prozivanjem objekta `tablicaPrijevoda` koji pohranjuje mapu s ključevima i prijevodima za aktivni jezik.

Željeli bismo omogućiti mijenjanje jezika tijekom izvođenja programa i to na način da se promjene konzistentno i trenutno odražavaju u svim elementima korisničkog sučelja.

- (a) Predložite odgovarajuće rješenje i za njega nacrtajte dijagram razreda.
- (b) Koji je oblikovni obrazac prikladan?
- (c) Ne trebate pisati programsko rješenje. Umjesto toga objasnite kako bi bila ostvarena dinamička promjena trenutnog jezika. Koju metodu nad kojim objektom bi klijent morao pozvati i što bi se tada događalo (pojasnite dinamiku izvođenja)?

3. Razmatramo tekstna korisnička sučelja (ljuske, engl. shell) u kojima korisnik zadaje naredbe u obliku jednog retka teksta, a rezultati izvođenja naredbe se predstavljaju formatiranim tekstom. Potrebno je oblikovati bibliotečnu komponentu `CommandLineInterface` koja bi omogućila izradu sučelja za različite primjene.

Komponenta `CommandLineInterface` treba definirati metodu `mainloop()` koja u svakom prolazu kroz petlju od korisnika prima naredbeni redak, predobrađi ga pozivom privatne metode `preprocess()` te daljnju obradu prepušta klijentskom kodu. Kako bi klijentskom kodu omogućila dodatnu komunikaciju s korisnikom (npr. postavljanje sigurnosnih pitanja poput "jeste li sigurni da želite

pobrizati datoteku xyz”), komponenta `CommandLineInterface` također treba definirati i zaštićene metode `writeline()` i `readline()`.

Koji oblikovni obrazac bi mogao zadovoljiti prikazane zahtjeve? Predložite organizaciju prema tom obrascu i prikažite je dijagramom razreda. Bismo li zadatak mogli riješiti i nekim drugim obrascem ako bismo zahtjeve smjeli malo izmijeniti?

4. Navedite simptome loše organizacije koje možemo primijetiti u sljedećem kodu, te objasnite koja načela oblikovanja su prekršena. Predložite popravljenu organizaciju u skladu s načelima oblikovanja. Nacrtajte strukturne dijagrame prije i poslije promjene.

```
class XMLparser{
public:
    void parseXML(){
        // ...
    }
};

class MyProtocol{
    XMLparser xml;
public:
    void processRequest(){
        // ...
        xml.parseXML();
        // ...
    }
};

class MyServer{
    MyProtocol protocol;
public:
    void mainLoop(){
        // ...
        protocol.processRequest();
        // ...
    }
};
```

5. Pretpostavite da razvijamo računalni program za obradu teksta u kojem postoji razred `Document` s metodama `void addText(int row, int col, string text)` i `void removeText(int row, int col1, int col2)`. Pretpostavite nadalje da su meniji u tom programu oblikovani prema oblikovnom obrascu naredba.

Prikažite implementaciju naredbe koja se treba pozvati kada korisnik odabere meni `Edit->Paste`. Pretpostavite pri tome da razred `Document` referencira objekt razreda `ClipboardAccessor` koji ima metodu `getClipboardContents()`. Pokažite organizaciju dijagramom razreda i u njemu naznačite sve sudionike oblikovnog obrasca naredba.

Pokažite koje dodatne promjene trebamo unijeti ako želimo omogućiti opozivanje naredbi.