Zavod za elektroniku, mikroelektroniku
računalne i inteligentne sustave

28. lipnja 2018.

Oblikovni obrasci u programiranju

Završni ispit

Napomena uz sve zadatke: povežite rješenja s odgovarajućim **oblikovnim obrascima** i **načelima oblikovanja**; izvorni kôd možete skicirati u C-u, C++-u, C#-u, Javi ili Pythonu, osim ako nije drukčije navedeno.

1. Razmatramo funkciju `make_log_stats` koja prima standardni iterator po retcima dnevnika (engl. log file) i vraća statističke podatke koji nisu tema ovog zadatka. Potrebno je omogućiti da funkcija `make_log_stats` obradi samo one retke dnevnika koji u sebi sadrže zadanu ključnu riječ, bez izmjene izvornog koda te funkcije. Pretpostavite da ključna riječ nije unaprijed poznata (npr. čita se iz konfiguracijske datoteke).

   - Predložite implementaciju iteratora po retcima datoteke (ako vaš jezik ima takav iterator u biblioteci, pravite se da on ne postoji).
   - Predložite organizaciju i implementaciju rješenja oblikovnog problema navedenog u tekstu zadatka.

2. Razmatramo sučelje `FancyFloats` koje enkapsulira kolekciju realnih brojeva. Sučelje ima virtualne metode `duplicate`, `multiply`, `id` i neke druge koje nisu bitne za ovaj zadatak. Metoda `duplicate` vraća kopiju matičnog objekta istog konkretnog tipa. Metoda `multiply` mijenja elemente kolekcije tako da ih pomnoži sa zadanim realnim argumentom. Metoda `id` vraća jedinstveni identifikator kolekcije (npr. pokazivač na prvi element).

   U praksi se javio problem pretjerane potrošnje memorije uslijed kopija nastalih pozivima funkcije `duplicate`. Riješite problem na način da uvedete transparentno automatsko kopiranje nakon mijenjanja (engl. copy on write). Vaše rješenje treba biti primjenljivo na sve konkretne razrede koji nasljeđuju `FancyFloats` te zadovoljiti sljedeći test:

   ```
   void test(FancyFloats* pff1){
     FancyFloats* pff2 = pff1->duplicate();
     assert(pff2->id() == pff1->id());
     pff2->multiply(-1);
     assert(pff2->id() != pff1->id());
   }
   ```

3. Organizacijska struktura neke kompanije može se prikazati hijerarhijom koja sadrži odjele i pojedinačne zaposlenike. Svaki odjel ima kolekciju članova koji mogu biti pododjeli i zaposlenici.

   Oblikujte komponentu za predstavljanje te organizacijske strukture. Predložite rješenje za obilazak svih zaposlenika kompanije jednostrukom petljom. Iskoristite to rješenje kako biste izračunali ukupan trošak plaća svih zaposlenika pod pretpostavkom da zaposlenici imaju metodu `get_pay`, a odjeli ne.

4. Implementirajte funkciju u programskom jeziku C koja prima slijed cijelih brojeva i vraća njihov zbroj. Poopćite vašu funkciju tako da pored zbroja može vratiti i neku drugu redukciju, npr. umnožak ili maksimum. Za prazan slijed funkcija treba vratiti podrazumijevanu vrijednost koja ovisi o vrsti redukcije. Implementirajte ispitni program koji generira polje od deset slučajnih brojeva od 1 do 10 te pozivom vaše funkcije određuje njihov umnožak, zbroj i maksimum.

5. U okviru treće laboratorijske vježbe razvijali smo komponentu za uređivanje teksta. Kao podsjetnik, spomenimo značajnije razrede: `TextEditor`, `TextEditorModel` te sučelja `TextObserver` i `CursorObserver`. `TextEditorModel` čuvao je listu redaka teksta (podatkovni član `lines`), pamtio položaj kursora (podatkovni član `cursorLocation`) te nudio metodu `insert(char c)`.

   - Nacrtajte dijagram razreda u kojem su sadržane komponente: `TextEditor`, `TextEditorModel`, `TextObserver` i `CursorObserver`. Odredite o kojem se oblikovnom obrascu radi te povežite komponente sa sudionicima obrasca.
   - Uz zanemarivanje selekcije, prikažite implementaciju metode `insert(char c)` razreda `TextEditorModel` iz koje je vidljiv način na koji je omogućena podrška za opoziv te akcije. Relevantan razred bio je `UndoManager` koji je imao dva stoga. Radi jednostavnosti pretpostavite da je argument uvijek slovo. Prikažite dijagram razreda na kojem je vidljiv `UndoManager` te ostali (za ovaj podzadatak) relevantni sudionici. Odredite o kojem se oblikovnom obrascu radi te povežite komponente sa sudionicima obrasca.