Zavod za elektroniku, mikroelektroniku
računalne i inteligentne sustave

12. srpnja 2017.

# Oblikovni obrasci u programiranju

## Pismeni ispit

1. Razmatramo obitelj algoritama s korijenom `Algorithm` koji definira apstraktnu metodu `apply`. Javila se potreba za povezivanjem klijenata tog algoritma s funkcionalnostima u binarnim bibilotekama. Preciznije, potrebno je omogućiti da poziv metode `apply` bude implementiran slijednim pozivima funkcija `f1` i `f2` binarne biblioteke koju zadaje glavni program. Izvorni kod postojećih algoritama i svih klijenata koji koriste apstraktne algoritme ne smije biti mijenjan.

   Predložite oblikovno rješenje koje zadovoljava predložene zahtjeve. Skicirajte strukturni dijagram razreda, te navedite o kojem se oblikovnom obrascu (ili obrascima) radi. Povežite elemente vaše organizacije sa sudionicima obrasca (ili obrazaca) te opišite koja načela pospješuje ovakva ogranizacija. Skicirajte ključne dijelove izvornog koda. Napišite glavni program koji kreira konkretni algoritam vezan uz biblioteku `xyzzy.so` i šalje ga klijentu.

2. Razmatramo elementarni program za uređivanje teksta. Razred `Model` sadrži listu stringova koji odgovaraju retcima te elementarne metode za umetanje i brisanje teksta.

   Predložite oblikovno rješenje koje bi omogućilo opozivanje i ponovno izvođenje uređivačkih radnji bez pamćenja čitavog teksta za svaki mogući opoziv. Skicirajte strukturni dijagram razreda, navedite o kojem se oblikovnom obrascu (ili obrascima) radi. Povežite elemente vaše organizacije sa sudionicima obrasca (ili obrazaca) te opišite koja načela pospješuje ovakva ogranizacija.

   Napišite izvorni kod demonstracijskog programa. Glavni program treba inicijalizirati dokument na četiri retka teksta, unijeti riječ `dodatak` na početak drugog redka te zatim izbrisati znakove trećeg retka u stupcima 5-10 (dvije uređivačke radnje). Nakon toga treba obje radnje opozvati, te na kraju ponovo izvesti prvu radnju.

3. Oblikujte rješenje koje omogućava standardnu iteraciju po svim elementima kompozita: listovima, podkompozitima i, rekurzivno, njihovim elementima. Rješenje mora biti primjenljivo i za klijente koji ne razlikuju listove od kompozita.

   Skicirajte strukturni dijagram razreda, navedite o kojem se oblikovnom obrascu (ili obrascima) radi. Povežite elemente vaše organizacije sa sudionicima obrasca (ili obrazaca) te opišite koja načela pospješuje ovakva ogranizacija.

   Napišite glavni program koji kreira kompozit koji se sastoji od dva lista i jednog podkompozita s jednim listom. Napišite funkciju koje prima kompozit i vraća sveukupni broj listova i podkompozita.

4. Neka je porodica geometrijskih likova modelirana sučeljem GraphicalObject, a model dokumenta razredom DocumentModel koji sadrži metode za upravljanje geometrijskim likovima. Oblikujte i prikažite programsku organizaciju koja će omogućiti da klikom miša postignemo stvaranje i brisanje geometrijskih likova, ovisno o aktivnom alatu grafičkog korisničkog sučelja. Vaša organizacija mora omogućiti jednostavno nadograđivanje programa novim likovima.

   Skicirajte strukturni dijagram razreda, navedite o kojem se oblikovnom obrascu (ili obrascima) radi. Povežite elemente vaše organizacije sa sudionicima obrasca (ili obrazaca) te opišite koja načela pospješuje ovakva ogranizacija. Prikažite relevantne dijelove programskog koda.

5. Prikažite organizaciju prikazanog izvornog kôda strukturnim dijagramom (OMT) i grafom ovisnosti komponenta. Navedite o kojem se oblikovnom obrascu radi te povežite razrede sa sudionicima obrasca. Napišite minimalni program koji provjerava mogu li se prikazani razredi prevesti i povezati.

```
class F{
public:
  // ...
};

class E{
protected:
  virtual void m()=0;
public:
  void n(F& f){
    // ...
    m();
  }
};
```

```
class G: public E{
public:
  G(){}
protected:
  virtual void m(){/* ... */}
};

class C{
  F f;
public:
  void o(G& g){
    g.n(f);
  }
};
```