



Završni ispit iz Oblikovnih obrazaca u programiranju

	A	B	C	D
1.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
2.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
3.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
4.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
5.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
6.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
7.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
8.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
9.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
10.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
11.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
12.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

1. Organizaciju s cikličkim ovisnostima možemo popraviti:

- (a) usmjeravanjem ovisnosti prema izvedenim razredima
- (b) bez prekrajanja ciklički ovisnih komponenta
- (c) smještanjem ciklički ovisnih komponenata u odvojene pakete
- (d) usmjeravanjem ovisnosti prema roditeljskom razredu

2. Što postićemo mostom u kojem ne postoji niti jedan apstraktni razred?

- (a) mogućnost zadavanja različitih konkretnih izvodača
- (b) razdvajanje klijenata od sudionika Apstrakcija
- (c) mogućnost zadavanja različitih prilagođenih apstrakcija
- (d) razdvajanje klijenata od sudionika Izvodač

3. Koja je razlika između obrascu Proxy i Dekorator?
 - (a) proxy se prvenstveno bavi modificiranjem ponašanja sadržanog subjekta
 - (b) dekorator sadržanoj komponenti uvijek pristupa preko apstraktnog sučelja
 - (c) nema razlike između dva obrasca
 - (d) dekorator se prvenstveno bavi kontrolom pristupa sadržanom subjektu
4. O kojoj komponenti ovisi konkretna naredba u obrascu Naredba?
 - (a) o naručitelju
 - (b) o primatelju
 - (c) o klijentu
 - (d) o posivatelju
5. Postoji li mogućnost pozivanja konkretnih komponentata koje su napisane nakon prevođenja programa u izvršni kod?
 - (a) da, ali samo ako se koristi Metoda tvornica koja objekte stvara operatorom new
 - (b) da
 - (c) ta mogućnost ne postoji
 - (d) da, ali samo ako se koristi Prototip
6. Moraju li klijenti obrasca Iterator ovisiti o konkretnim razredima?
 - (a) ni u kojem slučaju
 - (b) ne, ako klijenti spremnik stvaraju odgovarajućom tvornicom
 - (c) svakako, ali samo o konkretnom iteratoru
 - (d) svakako, ali samo o konkretnom spremniku
7. Stvaranje prototipa može se zakomplicirati kod objekata koji:
 - (a) sudjeluju u cirkularnom lancu referenci
 - (b) sadrže kompozite
 - (c) sadrže iteratore
 - (d) sadrže posjetitelje
8. O kojoj komponenti ovisi ciljni razred (engl. target) u obrascu Prilagodnik?
 - (a) o prilagodniku (engl. adapter)
 - (b) samo o klijentu
 - (c) i o klijentu i o vanjskom razredu (engl. adapter)
 - (d) niti jedan od ponuđenih odgovora nije točan
9. Obrasci Posjetitelj možemo razmijeniti obrascem Most ako:
 - (a) sve operacije nad elementima uspijemo prikazati metodom hijerarhije elemenata
 - (b) sve konkretne operacije nasljeđuju isti osnovni razred
 - (c) svi elementi nasljeđuju isti osnovni razred
 - (d) operacije nad elementima prikazemo preko zajedničkog implementacijskog sučelja
10. Metode tvornice tipično pozivamo iz:
 - (a) metoda konkretnog proizvoda
 - (b) klijentskog koda
 - (c) konstruktora apstraktnog kreatora
 - (d) konstruktora konkretnog proizvoda
11. Što može napraviti podrazumijevana implementacija metode add apstraktnih komponente obrasca Kompozit?
 - (a) dodati sebe u kompozit zadan argumentom
 - (b) baciti iznimku
 - (c) dodati sebe u roditelja komponente zadane argumentom
 - (d) niti jedan od ponuđenih odgovora nije točan
12. U kojem sudioniku obrasca Stanje treba definirati prijelaze među stanjima ako želimo minimizirati međuovisnost konkretnih stanja?
 - (a) obavezno u apstraktnom stanju
 - (b) ili u kontekstu ili u nekoj pomoćnoj komponenti
 - (c) obavezno u konkretnom stanju
 - (d) ili u apstraktnom stanju ili u kontekstu

Oblikovni obrasci u programiranju

Završni ispit

1. Razmatramo obitelj operacija s osnovnim razredom `Command` koji definira apstraktnu metodu `execute` bez argumenta. Irvina se potreba za omogućivanjem poštovanja operacija nesortiranim pozivateljima. Operacija se smije izvršiti samo ako pozivatelj ima odgovarajuće dozvole. Pretpostavite sljedeće:

- Većina klijenata konkretne operacije dobiva izvana primjenom injicije ovisnosti.
- Dolazi trenutnog identiteta i dozvola pozivatelja omogućava globalni objekt `auth` razreda `Authentication`. Dozvole se mogu dohvatiti u obliku liste stringova pozivom metode `permissions()`. Identitet pozivatelja vraća metoda `name()`.
- Potrebne dozvole za izvođenje operacije moraju se moći dinamički konfigurirati u obliku liste stringova. Operacija se smije izvršiti samo ako su potrebne dozvole podskup trenutnih dozvola, a u suprotnosti se mora dopustiti iznimka.
- Izvorni kod cjelokupne obitelji operacija i svih klijenata koji koriste apstraktne operacije se smije biti njeznan.

Predložite oblikovni rješenje koje zadovoljava predložene zahtjeve. Skicirajte ključne dijelove izvornog koda. Napišite glavni program koji kreira dvije operacije, dodjeljuje im dozvole, te pokrene njihovo izvođenje tako da jedna operacija uspije a druga ne. Skicirajte strukturalni dijagram razreda, navedite o kojem se oblikovnom obrascu radi. Povežite elemente vaše organizacije sa sudionicima obrasca te opišite koja načela pospješuje ovakva organizacija.

2. Razmatramo dio programske organizacije iz 4. laboratorijske vježbe koji uključuje sljedeće komponente:

`GraphicalObject`, `GraphicalObjectListener` i izvedene razrede, `DocumentModel` te `DocumentModelListener` i izvedene razrede. Skicirajte implementacije metoda konkretnih razreda izvedenih iz `GraphicalObjectListener` i `DocumentModelListener`. Skicirajte strukturalni dijagram te navedite koji se oblikovni obrascu mogu prepoznati. Povežite elemente vaše organizacije sa sudionicima obrasca te opišite koja načela pospješuje ovakva organizacija.

3. Razmatramo razred `TextEditorModel` iz 3. laboratorijske vježbe koji sadrži listu redaka tipa `List<String>`. Pretpostavite da sadrži listu sadi samo dvije metode:

- `int size()` koja vraća broj elemenata liste, te
- `String get(int index)` koja vraća element s predane pozicije (numeracija ide od 0).

Razred `TextEditorModel` svojim klijentima nudi metodu `Iterator<String> linesRange()`. Prikažite implementaciju te metode te procenite relevantan kod ako se za potrebe iteriranja ne smije stvarati duplikati kolekcija. Za korišteni oblikovni obrasc prikazite strukturalni dijagram i navedite njegove sudionike. Koja načela pospješuje ovakva organizacija koda? Pojasnite. Napišite primjer klijentskog koda koji ima referencu na model te uporabom spomenute metode odšarupa ispis prvih $n/2$ redaka iz modela.

4. Razmatramo obitelji `GraphicalObject` i `Renderer` koje su u 4. laboratorijskoj vježbi korištene za prikazivanje i isticavanje geometrijskih likova.

- Nacrtajte dijagram razreda koji prikazuje odnose tih razreda (`AbstractGraphicalObject` možete zanemariti).
- Napišite kod koji bi listu geometrijskih likova nacrtao na zaslonu.
- Vrata između dviju obitelji bliska je dvama oblikovnim obrascima ali se odgovara u potpunosti niti jednom od njih. O kojim se oblikovnim obrascima radi? Nacrtajte dijagram razreda koji prikazuje kako bi izgledala organizacija prema tim obrascima. Za svaki od ta dva slučaja povežite imena tipova sa sudionicima oblikovnog obrasca te prikazite kako bi izgledao kod koji crta listu grafičkih objekata.

5. Oblikujte komponentu za selektiranje elemenata pobrojivih (iterabilnih) spremnika prema kriterijima koje definira klijent. Komponenta treba omogućiti standardni običajak selektiranih elemenata bez njihovog kopiranja. Vaša organizacija mora omogućiti jednostavno kombiniranje elementarnih kriterija selekcije uz izbjegavanje dupliranja koda. Ilustrirajte rad vaše komponente na primjeru selekcije elemenata spremnika koji sadrži brojeve od 1 do 10. Prikazite kod koji odabire i na zaslon ispisuje: (a) parne brojeve; (b) brojeve veće od 5; (c) parne brojeve veće od 5. Uputa: definirajte dva elementarna kriterija selekcije koje ćete koristiti u prva dva podzadatka. Treći podzadatak riješite kombiniranjem elementarnih kriterija. Skicirajte strukturalni dijagram razreda, navedite o kojem se oblikovnom obrascu radi. Povežite elemente vaše organizacije sa sudionicima obrasca te opišite koja načela pospješuje ovakva organizacija.