

1. U obrascu Most o kojoj komponenti ovisi komponenta s platformski ovisnim kodom?

- a) prilagojenoj apstrakciji
- b) klijentu
- c) apstrakciji
- d) konkretnom izvođaču**

2. Pitanje o kompozitu u C-u (na koji je način ostvaren)

3. Konkretni skupni objekti modernih biblioteka tipično definiraju:

- a) okvirnu metodu
- b) metodu tvornicu za konkretni iterator**
- c) metodu za prelazak na sljedeći element
- d) metodu serialize ()

4. Koje od navedenih rješenja najčešće NEĆE biti uspješno pri ispravljanju kršenja Liskovog načela supstitucije?

- a) smanjiti odgovornost osnovnog razreda
- b) preoblikovati sve uključene razrede
- c) odustati od nasljeđivanja
- d) promijeniti smjer nasljeđivanja**

5. Obrazac Posjetitelj NIJE prikladan kad imamo:

- a) nestabilnu hijerarhiju elemenata**
- b) nestabilnu hijerarhiju postupaka
- c) stabilnu hijerarhiju elemenata i postupaka
- d) stabilnu hijerarhiju elemenata

6. Zadan je razred A u C++-u s dvije virtualne funkcije i jednim cjelobrojnim podatkovnim članom. Koliko bi mjesta na stogu 32 – bitne organizacije tipično zauzimalo polje od 100 objekata tipa A?

- a) 100 bajta
- b) 400 bajta
- c) 800 bajta**
- d) 1200 bajta

7. Kako klijenti tipično pristupaju stanju konteksta u obrascu Stanje?

- a) izravno, preko vlastitog pokazivača na razred koji igra ulogu konkretnog stanja
- b) izravno, koristeći pristupnu metodu
- c) klijenti stanju konteksta ne pristupaju izravno**
- d) izravno, preko vlastitog pokazivača na razred koji igra ulogu stanja

8. O kojoj komponenti ovisi ciljni razred u obrascu Prilagodnik?

- a) ništa nije točno**
- b) vanjskom razredu
- c) klijentu
- d) prilagodniku (adapteru)

9. Posjetitelj možemo zamijeniti obrascem Most samo ako:

- a) svi elementi nasljenuju isti osnovni razred
- b) sve operacije nad elementima uspijemo prikazati metodama hijerarhije elemenata
- c) sve operacije nad elementima uspijemo prikazati preko zajedničkog implementacijskog sučelja**
- d) sve konkretne operacije nasljenuju isti osnovni razred

10. Za složeni obrazac model-pogled-upravljač vrijedi:

- a) pogled i upravljač ponekad trebaju raditi usporedno
- b) upravljač ne ovisi o pogledu
- c) model i upravljač ponekad trebaju raditi usporedno**
- d) model upravlja prikazom na zaslonu ekrana

11. O kojoj se vrsti Proxy-ja radi ako je ciljni subjekt u drugom adresnom prostoru:

- a) sinkronizacijski
- b) virtualni
- c) sigurnosni
- d) udaljeni**

12. Model-pogled-upravljač se sastoji od:

- a) promatrač i stanje (strategija, a ne stanje???)**
- b) dekorator i posjetitelj
- c) dekorator i promatrač
- d) promatrač i posjetitelj

1. Koja je namjera obrasca Prototip?

- a) upravljati pristup ciljanom objektu
- b) prilagoditi sučelje postojećeg objekta korištenjem prototipa
- c) odrediti razred novog objekta pozivom metode jedinstvenog objekta
- d) odrediti razred novog objekta odabirom odgovarajućeg postojećeg objekta**

2. Koji nam oblikovni obrazac omogućava stvaranje primjerka bez poznavanja njihovog razreda?

- a) posjetitelj
- b) most
- c) dekorator
- d) prototip**

3. Obrazac Most može biti alternativa sljedećoj tehnici:

- a) viseštruko nasljeđivanje**
- b) privatno nasljeđivanje
- c) obična kompozicija
- d) jako nasljeđivanje

4. Koja metoda je karakteristična obrascu Prototip?

- a) create
- b) attach
- c) clone**
- d) accept

5. O kojoj komponenti mora ovisiti apstrakcija u obrascu most?

- a) o klijentu
- b) o prilagođenoj apstrakciji
- c) o izvođaču**
- d) o konkretnom izvođaču

6. Što postizemo Mostom u kojem ne postoji niti jedan apstraktni razred?

- a) čvršće povezivanje klijenata s apstrakcijom
- b) mogućnost zadavanja različitih konkretnih izvođača
- c) razdvajanje klijenata od izvođača**
- d) mogućnost zadavanja različitih prilagođenih apstrakcija

1. uz koji se još obrazac koristi posjetitelj

2. koja je metoda svojstvena prototipu

3. koju metodu implementiraju posjetitelji

4. ko instancira konkretnog posjetitelja

5. što može zakomplicirati posjetitelja

6. šta omogućuje prototip

7. šta se dobiva posjetiteljem

8. o kome najviše ovisi konkretni element

1. Namjena metode tvornice je omogućiti izvedenim razredima da:

- a) koriste metode osnovnog razreda za pozivanje proizvoda
- b) pozivaju klijente preko metoda osnovnog razreda
- c) definiraju konkretan tip stvorenih objekata**
- d) pohrane pokazivač na... nekakav razred

2. Tipični podatkovni član dekoratora jest pokazivač na:

- a) apstraktnu komponentu**
- b) klijenta obrasca
- c) konkretan dekorator
- d) dekoraciju

3. Metode tvornice tipično pozivamo iz:

- a) metoda konkretnog proizvoda
- b) metoda apstraktnog proizvoda
- c) konstruktora apstraktnog kreatora
- d) metoda apstraktnog kreatora**

4. Posljedice oblikovnog obrasca Stanje:

- a) omogućava se atomarna i eksplicitna promjena ponašanja objekta**
- b) ponašanja vezana uz različita stanja implementirana su u istoj komponenti
- c) bolja nadogradivost na štetu slabije jedinstvene odgovornosti
- d) učestali uvjetni izrazi koji testiraju tekuće stanje

5. Obrazac most može biti alternativa sljedećoj tehnici:

- a) višestruko nasljeđivanje**
- b) privatno nasljeđivanje
- c) obična kompozicija
- d) nešto

6. Posjetitelj [valjda] je primjenjiv i u slučaju elemenata koji su:

- a) iteratori
- b) strategije
- c) mostovi
- d) kompoziti**

7. Gdje u Stanju definirati prijelaze ako želimo minimalnu međuovisnost:

- a) ili u apstraktnom stanju ili u kontekstu
- b) obavezno u apstraktnom stanju
- c) u kontekstu ili u pomoćnoj komponenti**
- d) ili u konkretnom ili u apstraktnom stanju

8. Koji podatkovni član tipično ima osnovni razred hijerarhije mosta [valjda] s kojim interagiraju klijenti... nešto: [pliz nek netko nadopuni ak se sjeća]

- a) listu objekata tog istog razreda
- b) pokazivač na klijent
- c) pokazivač na implementirajući objekt**
- d) pokazivač na funkciju cilja

9. Proxy koji omata objekte koje je skupo konstruirati:

- a) pametni pokazivač
- b) virtualni**
- c) udaljeni
- d) sigurnosni

10. Posjetitelj nije prikladan obrazac za oblikovanje klasa:

- a) nestabilne hijerarhije elemenata**
- b) stabilne hijerarhije elemenata
- c) nešto
- d) nešto

11. O kojem sudioniku obrasca Proxy tipično ovise klijenti?

- a) Samo o stvarnom subjektu
- b) Samo o apstraktnom subjektu**
- c) O stvarnom i o proxyju
- d) Ništa od navedenog

1. O kome ovisi apstrakcija u Mostu:

- a) prilagođenoj apstrakciji
- b) izvođaču**
- c) konkretnom izvođaču
- d) ?

2. Upotreba parametrizirane tvornice

c) lokaliziranje odgovornosti za stvaranje objekata

3. Defaultna implementacija metode *add* u apstraktnoj komponenti obrasca kompozit:

- a) baca iznimku**
- b) dodaje komponentu prosljeđenu argumentom u listu komponenti

4. Proxy koji omata objekte koje je skupo konstruirati:

- a) pametni pokazivač
- b) virtualni**
- c) udaljeni
- d) sigurnosni

5. Gdje u Stanju definirati prijelaze ako želimo minimalnu međuovisnost:

- a) u konkretnom stanju
- b) u apstr. stanju ili kontekstu
- c) u apstraktnom stanju
- d) u kontekstu**

6. Visitor se primjenjuje i u slučaju elemenata koji su:

- a) prototipovi
- b) strategije
- c) mostovi
- d) kompoziti**

7. Tko kreira konkretnu naredbu?

- a) pozivatelj
- b) primatelj
- c) apstraktna naredba
- d) klijent**

8. Metoda tvornica:

c) povezivanje stvaranja raznorodnih objekata

9. Zašto koristiti Prototip?

- a) zbog stvaranja klijentu nepoznatih razreda**
- b) zbog nekakvog brisanja ...

10. U Adapteru klijent ovisi o:

- a) adapteru
- b) ciljnom razredu**
- c) vanjskom razredu

11. Moraju li klijenti u Iteratoru ovisiti o konkretnim razredima?

- a) ne, osim kada je inverzija ovisnosti**
- b) svakako
- c) nikako
- d) nekakav kompromisni odgovor

12. Odnos Okvirne metode i Strategije:

- a) template ima šire područje primjene
 - b) kod templatea ima nužno manje objekata**
 - c) razlike u brzini izvođenja
 - d) ?
-

1. Postoji li mogućnost pozivanja konkretnih komponenata koje su napisane nakon prevođenja programa u izvršni kod?

- a) ta mogućnost ne postoji
- b) da**
- c) da, ali samo ako se koristi Jedinostveni objekt
- d) da, ali samo ako se koriste predlošci

2. O kojoj komponenti ovisi konkretna naredba u obrascu Naredba?

- a) o naručitelju
- b) o primatelju**
- c) o pozivatelju
- d) o klijentu

3. Stvaranje Prototipa može se zakomplicirati kod objekata koji:

- a) sadrže kompozite
- b) sadrže posjetitelje
- c) sadrže iteratore
- d) sudjeluju u cirkularnom lancu referenci**

4. Organizaciju s cikličkim ovisnostima možemo popraviti:

- a) bez prekrajanja ciklički ovisnih komponenti
- b) usmjeravanjem ovisnosti prema roditeljskim razredima**
- c) višestrukim nasljeđivanjem
- d) usmjeravanjem ovisnosti prema izvedenim razredima

5. Koja je razlika između obrasca Proxy i Dekorator?

- a) dekorator sadržanoj komponenti uvijek pristupa preko apstraktnog sučelja**
 - b) klijenti Proxya uvijek poznaju konkretni tip sadržane komponente
 - c) dekorator se prvenstveno bavi kontrolom pristupa sadržanom subjektu
 - d) nema razlike između dva obrasca
-

1. Što vrijedi za Okvirnu metodu:

- a) izvedeni razredi implementiraju sve metode
- b) izvedeni razredi implementiraju jednu metodu
- c) osnovni razred nema implementaciju niti jedne metode
- d) niti jedan odgovor nije točan**

2. Koji element u obrascu Promatrač ne ovisi o drugima:

- a) Subjekt**
- b) kontekst
- c) Apstraktna strategija

3. Okvirna metoda koristi se kada:

- a) imamo određeni postupak, a želimo dozvoliti mijenjanje nekih dijelova**

4. Usporedba strategije i okvirne metode, tj. kad koristimo strategiju:

- a) kada želimo više osi promjene**

5. Okvirnu metodu koristimo:

- a) moj odgovor, kada korisnik može mijenjati biblioteke ?**

6. Ako se želimo riješiti nepotrebnih obavješćavanja kod obrasca Promatrač trebamo:

- a) povećati složenost obavješćavanja
 - b) smanjiti složenost obavješćavanja
 - c) povećati složenost stvaranja i dodavanja**
-

1. Ako u obrascu promatrač imamo samo jedan subjekt i jedan promatrač to možemo poistovjetiti s obrascem:

- a) strategija ?**
- b) okvirna metoda
- c) dekorator
- d) nista od navedenog

2. Koja je namjera obrasca okvirna metoda ? - Definirati okvirni postupak koji neke korake prepušta izvedenim razredima.

3. U oblikovnom obrascu promatrač imamo: - Subjekt: izdavač, Objekt: pretplatnik

4. U oblikovnom obrascu promatrač struktura objekta(promatrača) obično u sebi sadrži: - Pokazivač na subjekt

5. Kako se u oblikovnom obrascu okvirna metoda ostvaruje prilagodljivost ?
- Nasljeđivanjem.

1. Što od sljedećeg nije prednost korištenja obrazaca:

- a) gotova funkcionalnost
- b) produktivnost oblikovanja
- c) obogacen rjecnik
- d) laka komunikacija

2. Oblikovni obrazac je :

a) općenito rješenje učestalog oblikovnog problema

3. Što je veća apstraktnost komponente, to je:

- a) veća inertnost
- b) manja inertnost
- c) veća složenost

4. Testiranje je najbolje raditi:

- a) svaku komponentu posebno
- b) Odgađati testiranje do završetka projekta
- c) Testirati zajedno sve komponente u istom paketu
- d) Sve odjednom (tako nešto)

5. Obrazac strategije temelji se na (ili može se koristiti zbog):

- a) nasljeđivanja
 - b) povjeravanja (nešto od ovoga dvoje je odgovor, nisam siguran sto)
-

1. Razred koji igra ulogu subjekta u obrascu promatrač tipično sadrži metode:

- a) attach i notify
- b) clone
- c) update
- d) create

2. Cilj načela nadogradnje bez promjene je:

- a) omogućiti proširenje funkcionalnosti komponente bez mijenjanja njene implementacije
- b) spriječiti da stari kod radi s novim kodom
- c) osigurati da nadogradnja komponente zahtijeva promjene svih ovisnih komponenata
- d) zatvoriti komponente za nadogradnju, a otvoriti za promjene

3. U obrascu Promatrač eventualna ovisnost promatrača o subjektu nas ne smeta jer:

- a) promatrači uglavnom imaju jednostavnu izvedbu
- b) subjekt se znatno manje mijenja od promatrača
- c) subjekt ima jednostavnu izvedbu
- d) promatrači se rijetko mijenjaju

4. U krutom programskom sustavu unošenje skromne nove funkcionalnosti tipično za sobom povlači:

- a) korištenje strukturirane paradigme
- b) potrebu za neplaniranim izlaskom iz programa
- c) korištenje objektno orijentirane paradigme
- d) mijenjanje velikog broja komponenata

5. Obrazac strategija pospješuje inverziju ovisnosti jer:

- a) strategija ne pospješuje inverziju ovisnosti
- b) kontekst ne mora poznavati konkretne strategije
- c) kontekst može raditi s različitim strategijama
- d) mogu se po volji kombinirati različite porodice strategija

6. Funkcija f() prima referencu na objekt razreda A te poziva njegovu javnu virtualnu metodu m(). Razred B nasljeđuje A, te definira javnu metodu m(). Glavni program poziva f() s objektom razreda B. Koja metoda će se na kraju pozvati?

- a) B::m()
- b) doći će do pogreške pri izvođenju
- c) A::m()
- d) ovisi o implementaciji razreda A

7. U kojem sudioniku obrasca dekorator su implementirane dodatne odgovornosti?

- a) u konkretnim dekoratorima
- b) u konkretnoj komponenti
- c) u klijentima apstraktne komponente
- d) u apstraktnom dekoratoru

8. Koje načelo oblikovanja garantira uspjeh poziva metode konkretnog objekta preko pokazivača na osnovni razred?

- a) načelo jedinstvene odgovornosti
- b) načelo nasljeđivanja implementacije
- c) načelo nasljeđivanja sučelja
- d) Liskovino načelo supstitucije

9. Dinamički polimorfizam u Pythonu se ostvaruje:

- a) korištenjem polja pokazivača na funkcije
- b) dinamički polimorfizam u Pythonu nije moguće ostvariti
- c) rekurzivnim prozivanjem asocijativnih spreminka metoda**
- d) ponovnim prevođenjem parametriziranog koda

10. Ključne četiri komponente oblikovnog obrasca su:

- a) zahtjevi, oblikovanje, izvedba, ispitivanje
- b) naziv, problem, rješenje, rezultat**
- c) zamisao, izvedba, optimizacija, dokumentacija
- d) strategija, promatrač, dekorator, tvornica

11. Koja je razlika između primitiva i nadomjestivih metoda (tzv. hooks) osnovnog razreda u obrascu Okvirna metoda?

- a) nadomjestive metode su implementirane u izvedenom razredu, a primitivi ne
- b) primitivi ne moramo implementirati u izvedenom razredu
- c) nadomjestive metode su implementirane u osnovnom razredu, a primitivi ne**
- d) primitivi su razredi za razliku od nadomjestivih metoda

1. Prilikom testiranja programskog sustava u razvoju, najbolje je:

- a) testirati svaku komponentu zasebnim ispitnim programom**
- b) testirati čitav sustav istovremeno
- c) testiranje ne provoditi dok čitav sustav nije dovršen
- d) testirati istovremeno sve komponente iz istog paketa

2. Kojim obrascem često stvaramo konkretne naredbe?

- a) strategijom
- b) jedinstvenim objektom
- c) parametriziranom tvornicom
- d) metodom tvornicom**

3. Metoda notify subjekta tipično poziva sljedeću metodu promatrača:

- a) create
- b) update**
- c) attach
- d) clone

4. Osnovni razred hijerarhije s kojom u obrascu Most integriraju klijenti tipično ima sljedeći podatkovni element:

- a) pokazivač na implementacijski objekt**
- b) oikazival na klijenta
- c) pokazivač na funkciju cilja
- d) listu objekata tog istog razreda

5. Kako se ostvaruje veza od objektnog Prilagodnika do prilagođenog objekta?

- a) preko klijenta
- b) delegacijom**
- c) nasljeđivanjem
- d) niti jedan odgovor nije točan

7. Koja od navedenih komponenta obrasca Posjetitelj ovisi o najvećem broju drugih komponenata?

- a) element
- b) naredba
- c) konkretni elementi
- d) konkretni posjetitelji**

8. Kako se tipično omogućava slijedni prolaz kroz sve elemente višerazinskog kompozita?

- a) dodavanjem pokazivača na roditelja u kompozitu
- b) dodavanjem pokazivača na sljedeći element kompozita
- c) dodavanjem metode getChild(int index)
- d) dodavanjem pokazivača na roditelja u apstraktnoj komponenti**

9. Odabir konkretne strategije u obrascu strategija vrši:

- a) klijent konteksta**
- b) apstraktna strategija
- c) konkretna strategija
- d) autor izvornog koda

10. Kako zovemo proxy koji briše objekte koje više nitko ne koristi?

- a) udaljeni
- b) pametni pokazivač**
- c) garbage collector
- d) sigurnosni