

Q: Kako izgleda inicijalizacija za virtualnu tablicu?

A: {dogGreet, dogMenu}

Q: Koliko CALL je generirano od A *pa = new A; ?

A: 2

Q: Kada je poznato koja zivotinja se zove?

A: Tijekom izvođenja unutar tijela funkcije animalPrintGreeting

Q: Kada je poznato odrediste od f1->tabulate() ?

A: Prilikom prevodenja

Q: Koliko eksplicitnih definicija funkcija sadrzava virtualna tablica od Square?

A: 2

Q: Koliko puta moramo dereferencirati polimorfni objekt kako bismo dosli do virtualne tablice?

A: 2

Q: U kojoj funkciji se postavlja virtualna tablica za razred Dog?

A: U funkciji constructDog.

Q: Koliko bi najmanje memorije zauzimala klasa Linear ako je prevedena za 32-bitno računalo?

A: Linear ima 2 int, 2 double i 1 ptr, to je $2 \times 4 + 2 \times 8 + 4$, kao što je napisano iznad.

Q: U kojoj funkciji se pozivaju metode animalPrintGreeting i animalPrintMenu?

A: Funkciji testAnimals.

Q: [isječak asemblera s pozivom virtualne funkcije] Što se nalazi na adresi [neka adresa, možda je čak bio rbp u pitanju]?

A: Tu sam se dvoumio između "virtualna tablica" i "element virtualne tablice"

Q: Zašto razredi CoolClass i PlainOldClass ne zauzimaju istu količinu memorije?

A: Jer CoolClass ima virtualne metode pa onda i pokazivač na tablicu virtualnih funkcija

1. f2 je pointer na Square, f2->negative_value_at(1.0). U kojem razredu se nalazi taj poziv?

Unary_Function

2. Ako imamo pokazivač na polimorfan objekt, koliko ga puta trebamo dereferencirati prilikom poziva virtualne funkcije?

2 ili 3

3. Koliko razred Linear najmanje zauzima memorije u 32-bitnom sustavu?

28 (objašnjenje: $2 \times \text{int} + 2 \times \text{double} + \text{pokazivač na virtualnu tablicu}$)

4. U funkciji animalPrintGreeting se poziva funkcija catGreet ili dogGreet. Kada se određuje koja se od tih funkcija poziva?

Tijekom izvođenja, u funkciji animalPrintGreeting.

5. U kojoj funkciji se provodi inicijalizacija virtualne tablice?

constructDog

6. A* a = new A. Koliko puta će puta biti CALL u strojnom kodu ako znamo da A ima 3 virtualne funkcije, konstruktor i destruktorktor.

2

7. Razlike između normalnih i virtualnih funkcija, spominjao se nativni strojni kod, u jednom odgovoru je bilo nešto s NBP.

idk nisam odgovorila

8. Kad se inicijalizira tablica podrazreda?

neposredno nakon poziva konstruktora roditeljskog razreda, prije provođenja metoda u konstruktoru podrazreda

9. Ako želimo dodati još jednu životinju, što trebamo promijeniti kod funkcije animalPrintGreeting?

ništa, izvorni i izvršni kod za funkciju ostaju isti

10. Koja od sljedećih je dobro navedena deklaracija za element virtualne tablice?

niti jedan od ponuđenih

- niti jedan nije bio ovo: typedef char const* (*PTRFUN)();

Q: Koliko puta moramo dereferencirati polimorfni objekt kako bismo dosli do pokazivača na virtualnu tablicu

1

Q: Koja je uloga oblikovnog obrasca koju ima Cell u 6. zadatku?

A: Promatrač i subjekt

Q: Koja od navedenih funkcionalnosti se preslikava u klasu koja predstavlja oblikovni obrazac u zadatku s percentilima?

A: Generiranje slučajnog niza (ponuđeno je i pristupanje nizu, to je krivo)

Q: Ako bismo htjeli dodati novi način stvaranja slučajnog niza, šta bismo trebali učiniti?

A: Napisati novu klasu i omogućiti stvaranje njezinih instanci

Q: Koju ulogu ima klasa koja generira fibonaccijev niz?

A: Strategija (konkretna strategija)

Q: Koliko argumenata prima mymax u C-u?

A: 4

Q: Kojeg su tipa prva dva argumenta od mymax u 2. zadatku ako je treći argument gt_int?

A: int* i int* (ostali ponuđeni svi imaju char* pa su zato krivi)

Q: Kada polimorfna verzija funkcije drawShapes saznaje tip podatka?

A: Mislim da tu ide odgovor da funkcija ne treba znati konkretni tip

Q: Koju ulogu igra razred Sheet u zadnjem zadatku?

A: Konkretni razred

Q: Koje obrasce smo koristili u zadatku sa slijedom brojeva (5. zadatak)?

A: Strategija i promatrač

Q: Zašto nije problem kad promatrač ima referencu na subjekt (to je cirkularna ovisnost)?

A: Jer najčešće nitko osim subjekta ne ovisi o promatračima

Q: Koja metoda vraća polje objekata na koje se treba pretplatiti (6. zadatak)?

A: getrefs

Q: Dan je rječnik u pythonu, nešto kao `D = {'a':3, 'b':4, 'c':12, 'e':1}`, što je rezultat naredbe `max(D) + max(D, key=D.get)`?

A: ec, jer prvi max uspoređuje po ključevima, a drugi po vrijednostima

Q: Koji od navedenih poziva funkcije mymax će baciti grešku? (4 ponuđena)

A: `mymax([1,2,3], key=lambda x: len(x))`, jer ovaj x mora biti istog tipa kao element niza, u ovom slučaju int, a int nema metodu len

Kojeg tipa je ? u c++ mymax(int,int,?)?

→ funkcija koja prima dva pointera na int

Nesto je bilo za 6 zadatak koliko petlji treba za cell

2 ili 0

koja metoda baca exception u sestom zadatku?

- a) Cell.atach
- b) Sheet.getrefs
- c) Cell.update
- Sheet.set ili Cell.set

koliko forova za cell set -> 0???

Na sto treba castat argumente u fji gt_char

-> const char*

Zasto nije tolko bitno sto promatrac ovisi o subjektu

→ jer se subjekt rijetko mijenja

Sta ce vratit fja mymax(D) ako je D = (b=5,s=4)

-> 5

ako su u mymax c++ prva dva argumenta tipa int sta je treci?

-> (int*, int*), osobno meni boolean (int*,int*)

Ako je rijec o polimorfnoj metodi drawShapes() di ce bit poznato o kojem objektu se radi

-> prilikom izvođenja

Koliko apstraktnih strategija ima u zadatku s percentilima (0,1,2 ili 3)

-> 2 (generator i percentili)

Koje obrasce primjenjujemo u 5. zadatku

-> promatrac i strategiju

1. 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
2. Tipični podatkovni član dekoratora jest pokazivač na:
 - a) apstraktnu komponentu
 - b) klijenta obrasca
 - c) konkretan dekorator
 - d) dekoraciju
3. Odnos Okvirne metode i Strategije:
 - a) template ima šire područje primjene
 - b) kod templatea ima nužno manje objekata
 - c) razlike u brzini izvođenja
 - d) ?
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. 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
6. Koji element u obrascu Promatrač ne ovisi o drugima:
 - a) Subjekt
 - b) kontekst
 - c) Apstraktna strategija
7. Okvirna metoda koristi se kada:
 - a) imamo određeni postupak, a želimo dozvoliti mijenjanje nekih dijelova`1
8. Usporedba strategije i okvirne metode, tj. Kad koristimo strategiju:
 - a) kada želimo više osi promjene
9. Okvirnu metodu koristimo:
 - a) moj odgovor, kada korisnik može mijenjati biblioteke ?
10. Ako se želimo riješiti nepotrebnih obavještanja kod obrasca Promatrač trebamo:
 - a) povećati složenost obavještanja
 - b) smanjiti složenost obavještanja
 - c) povećati složenost stvaranja i dodavanja
11. 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

12. Koja je namjera obrasca okvirna metoda ?

- Definirati okvirni postupak koji neke korake prepušta izvedenim razredima.

13. U oblikovnom obrascu promatrač imamo:

- Subjekt: izdavač, Objekt: pretplatnik

14. U oblikovnom obrascu promatrač struktura objekta(promatrača) obično u sebi sadrži:

- Pokazivač na subjekt

15. Kako se u oblikovnom obrascu okvirna metoda ostvaruje prilagodljivost ?

- Nasljeđivanjem.

16. Što od sljedećeg nije prednost korištenja obrazaca:

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

17. Oblikovni obrazac je:

- a) općenito rješenje učestalog oblikovnog problema

18. Što je veća apstraktnost komponente, to je:

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

19. 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)

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

- a) nasljeđivanja
- b) povjeravanja

21. 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

22. Metoda notify subjekta tipično poziva sljedeću metodu promatrača:

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

23. Odabir konkretne strategije u obrascu strategija vrši:

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

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

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

2. Cilj nacela 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. 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) 1200 bajta
- c) 1600 bajta
- d) 800 bajta

1x pokazivač na tablicu virtualnih fja = 4 bajta

1xint = 4 bajta

ukupno = 8 bajta $\rightarrow 100 \cdot 8 = 800$

4. U obrascu Promatrač eventualna ovisnost promatrača o subjektu 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

5. U krutom programskom sustavu unosenje skromne nove funkcionalnosti tipično za sobom povlači:

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

6. 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

7. Funkcija f() prima referencu na objekt razreda A te poziva njegovu javnu virtualnu metodu m(). Razred B nasljedjuje 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

8. U kojem sudioniku obrasca dekoratora su implementirane dodatne odgovornosti?

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

9. Koje načelo oblikovanja garantira uspjeh poziva metode konkretnog objekta preko pokazivaca na osnovni razred?

- a) načelo jedinstvene odgovornosti
- b) načelo nasljedjivanja implementacije
- c) načelo nasljedjivanja sučelja
- d) Liskovinovo načelo supstitucije

10. Dinamički polimorfizam u Pythonu ostvaruje se:

- a) korištenjem polja pokazivaca na funkcije
- b) dinamički polimorfizam u Pythonu nije moguće ostvariti
- c) rekurzivnim pozivanjem asocijativnih spremnika metoda
- d) ponovnim prevodjenjem parametriziranog koda

11. 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, tvornice

12. Koja je razlika između primitiva i nadomjestivih metoda (eng. 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