

Objekat i klasa. Članovi objekata.

1. Šta je objekat, a šta klasa? Kreiranje objekta.
2. Oblast važenja referencne promenljive. Životni vek objekata.
3. Koliko objekata je kreirano u datom primeru?
Šta će biti rezultat izvršavanja sledećeg koda?

```
Tacka t = new Tacka(1, 1);  
Tacka t2 = t;  
t2.x = 12;  
Console.WriteLine(t.x);
```
4. Kako se definiše konstanta u C#? Kada joj se može definisati vrednost?
5. Da li se automatske promenljive moraju definisati?
6. Koje su podrazumevane vrednosti za primitivne i referencne tipove atributa objekta?

Kompajler, interpreter, garbage collector

7. Da li se za prevođenje Java kodova koristi kompajler ili interpreter?
8. Šta je JVM, a šta JRE?
9. Šta je Java bajt kod?
10. Šta je garbage collector i čemu služi?

Konstrukcija objekata

11. Šta je konstruktor? Šta je default-ni konstruktor?
12. Da li svaka klasa mora da ima konstruktor?
13. Kada ne postoji default-ni konstruktor?
14. Na koji način je moguće sprečiti poziv konstruktora neke klase van njene definicije? Ako je to uspešno izvedeno, kako bi rešili korišćenje objekata takve klase u aplikaciji?
15. Kako obezbediti da postoji samo jedan objekat neke klase? (Kako definisati Singleton klasu?)

Overloading

16. Šta je overloading? Po čemu se moraju razlikovati overloadovani imetodi?
17. Da li overloadovani metodi mogu vraćati različite tipove?

This

18. This – šta predstavlja i ko ga poseduje?
19. Šta se dobija pozivom this()

Static

20. Šta su članovi objekata, a šta članovi klase?
21. Kako se definišu i koriste statički podaci i metodi?
22. Definisanje ID-a objekta kao int podatka čija je vrednost jednaka rednom broju kreiranog objekta.
23. Šta svi nestatički metodi poseduju, a statički ne?
24. Koja su ograničenja u korišćenju static metoda?
25. Da li nestatički metodi definisani u klasi mogu pozivati statičke/nestatičke metode te iste klase?
26. Da li se u dete klasi može definisati statički metod koji ima isti potpis kao statički metod definisan u roditeljskoj klasi?
27. Da li je moguće definisati statičku klasu?

Nasleđivanje

- 28. Šta je prepisivanje metoda?
- 29. Šta je polimorfizam? Dati primer kada se javlja polimorfno ponašanje. Uslovi za polimorfizam.
- 30. Da li se polimorfno ponašanje može primeniti na statičke metode? Dati i primer.
- 31. Pozivi kojih metoda se vezuju statički, a koji dinamički?
- 32. Da li vidljivost metoda kojim se prepisuje roditeljski metod može biti veća/manja od vidljivosti prepisanog (roditeljskog metoda)?
- 33. Upotreba **super**.
- 34. Šta se implicitno poziva kao prva komanda svakog konstruktora?
- 35. Da li privatni podatak definisan u roditeljskoj klasi postoji u objektu dete klase? Da li je moguće dobiti njegovu vrednost?
- 36. **Implicitna i eksplicitna konverzija referecnih tipova.**

Access modifiers (vidljivost)

- 37. Koji modifikatori vidljivosti su predviđeni u Javi? Na koji način svaki od modifikatora vidljivosti utiče na vidljivost članova objekata?
- 38. Da li se modifikatori vidljivosti mogu primeniti na statičke članove?
- 39. Koja je podrazumevana vidljivost članova objekata/klasa u Javi?
- 40. Koja je podrazumevana vidljivost članova interfejsa?
- 41. Da li se modifikator pristupa/vidljivosti može primeniti na automatske promenljive?

Apstraktni metodi i tipovi

- 42. Kako se deklariraju apstraktni metodi?
- 43. Da li se za apstraktan metod može postaviti bilo koja vrsta vidljivosti? Ako postoje ograničenja, obrazložiti.
- 44. Šta su apstraktne klase? Navesti glavne karakteristike njene upotrebe. Dati primer definicije jedne apstraktne klase.
- 45. Da li je moguće definisati referencu tipa apstraktne klase? Na objekte kojih klasa ona može da ukazuje?
- 46. Šta je potrebno uraditi da bi klasa izvedena iz apstraktne klase bila konkretna? Koje klase su konkretne?

Interfejsi

- 47. Šta su interfejsi? Navesti glavne karakteristike. Dati primer definicije interfejsa.
- 48. Da li interfejsi mogu sadržati podatke članove?
- 49. Da li klasa može da implementira više interfejsa?
- 50. Implicitna vidljivost članova interfejsa.
- 51. Da li se u interfejsu može definisati statički metod?
- 52. Da li je moguće definisati referencu tipa interfejsa? Na objekte kojih klasa ona može da ukazuje?
- 53. Razlika između apstraktnih klasa i interfejsa.

Final

- 54. Šta je posledica primene modifikatora final na atribut, metod, klasu?

OOP koncepti

- 55. Šta je enkapsulacija?
- 56. Na koji način se postiže enkapsulacija u C#.
- 57. Koji mehanizam u oo jezicima omogućava reusability koda?
- 58. Overriding,
- 59. Mehanizam nasleđivanja.
- 60. Da li Java podržava višestruko nasleđivanje?

Izuzeci

- 61. Try/catch/finally – objasniti na primeru šta se i kojim redosledom izvršava.
- 62. Koji su proveravani, a koji neproveravani izuzeci?

GUI

- 63. Šta su Layout manager-i i čemu služe?
- 64. Šta su komponente, a šta kontejneri?
- 65. Dati primer definisanja anonimne klase.

Ugnježdjeni tipovi

- 66. Koja je razlika između ugnježenih i unutrašnjih klasa/tipova?
- 67. Kako se instancira objekat ugnježdene/unutrašnje klase?
- 68. Primer pristupa članovima objekata unutrašnje/ugnježdene klase.
- 69. Anonimne klase.

Niti

- 70. Šta su niti? Kako se mogu definisati u Javi? Dati primer.
- 71. Stanja niti.
- 72. Objasniti mehanizam zaključavanja. Šta se 'zaključava'? Šta se dešava kada nit poziva sinhronizovan metod objekta na koji je u trenutku poziva postavljen katanac?
- 73. Na šta se može primeniti synchronized?
- 74. Šta se dešava pozivom metoda wait(), a šta pozivom notify()?
- 75. Sinhronizacija i nasleđivanje.
- 76. Primer zadatka (Pogledati i zadatke sa kolokvijuma od ranijih godina)
Neka su definisana dva tipa niti, od kojih jedna koristi objekat skladišta iz kog čita njegovo stanje u obliku celobrojne vrednosti, a druga dodeljuje celobrojnu vrednost kao stanje skladišta.

```
class Proizvodjac extends Thread {  
    private Skladiste mojeskladiste;  
    private int broj;  
    private int k;  
  
    public Proizvodjac(Skladiste sk, int broj) {  
        mojeskladiste = sk;  
        this.broj = broj;  
    }  
  
    public void run() {  
        for (int i = 0; i < 10; i++) {  
            mojeskladiste.stavi(i);  
        }  
    }  
}  
  
class Korisnik extends Thread {  
    private Skladiste mojeskladiste;  
    private int broj;  
  
    public Korisnik(Skladiste sk, int broj) {  
        mojeskladiste = sk;
```

```
    this.broj = broj;
}

public void run() {
    int procitani_broj = 0;
    for (int i = 0; i < 10; i++) {
        procitani_broj = mojeskladiste.uzmi();
    }
}
}
```

Definisati klasu skladište tako da bude thread-safe, onemogućiti istovremeni pristup/ažuriranje stanja skladišta, obezbedi naizmenično pisanje i čitanje.

Napisati telo main metoda testne klase u kom se instanciraju po dve niti korisnika i potrošača i startuju.

Legenda

crvenom bojom su obeležena pitanja dodata u odnosu na verziju 2