



PITANJA ZA II KOLOKVIJUM

KLASE I OBJEKTI





KLASE I OBJEKTI

1. Enkapsulacija je _____ podataka.

- **skrivanje**
- **apstrakcija**
- **nasledivanje**



KLASE I OBJEKTI

1. Enkapsulacija je _____ podataka.



skrivanje

- **apstrakcija**
- **nasledivanje**



KLASE I OBJEKTI

2. Unutar deklaracije klase navode se:

- **definicije funkcija članica**
- **prototipovi (deklaracije) funkcija članica**



KLASE I OBJEKTI

2. Unutar deklaracije klase navode se:

- definicije funkcija članica
- 😊 prototipovi (deklaracije) funkcija članica



KLASE I OBJEKTI

3. Sakrivanje detalja realizacije klase zove se:

- **apstrakcija**
- **enkapsulacija**
- **polimorfizam**



KLASE I OBJEKTI

3. Sakrivanje detalja realizacije klase zove se:

- apstrakcija
- 😊 enkapsulacija
- polimorfizam



KLASE I OBJEKTI

4. Objekat izvedene klase istovremeno je i objekat osnovne klase.

- **Tacno**
- **Netacno**



KLASE I OBJEKTI

4. Objekat izvedene klase istovremeno je i objekat osnovne klase.



Tacno

- **Netacno**



KLASE I OBJEKTI

5. Skriveni pokazivac na objekat za koji je pozvana funkcija clanica zove se:

- **this**
- **that**
- **the**



KLASE I OBJEKTI

5. Skriveni pokazivac na objekat za koji je pozvana funkcija clanica zove se:



this

- **that**
- **the**



KLASE I OBJEKTI

6. Funkcija koja ima isto ime kao klasa zove se:

- **funkcija clanica**
- **konstruktor**
- **metod**



KLASE I OBJEKTI

6. Funkcija koja ima isto ime kao klasa zove se:

- funkcija clanica
-  konstruktor
- metod



KLASE I OBJEKTI

7. Objekti klase:

- **imaju sopstvene kopije vrednosti podataka clanova**
- **dele zajednicke vrednosti podataka clanova**



KLASE I OBJEKTI

7. Objekti klase:

- 😊 imaju sopstvene kopije vrednosti podataka članova
- dele zajednicke vrednosti podataka članova



KLASE I OBJEKTI

8. Nasledivanje klasa je nacin na koji se postojeća klasa može:

- **zameniti**
- **prosiriti**
- **popraviti**
- **ukloniti**



KLASE I OBJEKTI

8. Nasledivanje klasa je nacin na koji se postojeća klasa može:

- zameniti
- 😊 prosiriti
- popraviti
- ukloniti



KLASE I OBJEKTI

9. Konstruktor kopije se poziva u slucaju kada se objekat klase inicijalizuje drugim objektom:

- **iste klase**
- **druge klase**



KLASE I OBJEKTI

9. Konstruktor kopije se poziva u slucaju kada se objekat klase inicijalizuje drugim objektom:

- 😊 iste klase
- druge klase



KLASE I OBJEKTI

10. Kada se objekat vraca kao rezultat funkcije, pravi se njegova privremena kopija.

- **Tacno**
- **Netacno**



KLASE I OBJEKTI

10. Kada se objekat vraca kao rezultat funkcije, pravi se njegova privremena kopija.



Tacno

- **Netacno**





KLASE I OBJEKTI

11. Svaka klasa ima konstruktor i destruktor.

- **Tacno**
- **Netacno**



KLASE I OBJEKTI

11. Svaka klasa ima konstruktor i destruktor.



Tacno

- **Netacno**







KLASE I OBJEKTI

12. Koje su od sledećih tvrdnji tačne?

- **Podrazumevani konstruktor bez argumenata (eng. default no-arg constructor) je automatski obezbeđen ako konstruktor nije jasno definisan u klasi.**
- **Najmanje jedan konstruktor mora uvek biti eksplicitno definisan.**
- **Konstruktori nemaju povratni tip (engl. return type), čak ni void.**
- **Konstruktori moraju da imaju isto ime kao i odgovarajuća klasa.**
- **Konstruktori se pozivaju kada se kreira jedan objekat.**

KLASE I OBJEKTI

12. Koje su od sledećih tvrdnji tačne?

-  **Podrazumevani konstruktor bez argumenata (eng. default no-arg constructor) je automatski obezbeđen ako konstruktor nije jasno definisan u klasi.**
 - Najmanje jedan konstruktor mora uvek biti eksplicitno definisan.**
-  **Konstruktori nemaju povratni tip (engl. return type), čak ni void.**
-  **Konstruktori moraju da imaju isto ime kao i odgovarajuća klasa.**
-  **Konstruktori se pozivaju kada se kreira jedan objekat.**

KLASE I OBJEKTI

13. Analizirati sledeći kôd:

```
#include <iostream>
using namespace std;
class A {
public:
int s;
A(int newS) {
s = newS;
}
void print() {
cout << s;
}
};
```

```
int main() {
A a;
a.print();
}
```

- Program ima grešku kompilacije, jer klasa A nije javna (engl. public) klasa.
- Program ima grešku kompilaciju, jer klasa A nema podrazumevani konstruktor (engl. default constructor).
- Program će se prevesti (engl. compile), pokreće se, ali se ništa ne štampa.
- Program će se prevesti i pokrenuti ako se “A a” promeni na “A a (5)”.

KLASE I OBJEKTI

13. Analizirati sledeći kôd:

```
#include <iostream>
using namespace std;
class A {
public:
int s;
A(int newS) {
s = newS;
}
void print() {
cout << s;
}
};
```

```
int main() {
A a;
a.print();
}
```

- Program ima grešku kompilacije, jer klasa A nije javna (engl. public) klasa.
- 😊 Program ima grešku kompilaciju, jer klasa A nema podrazumevani konstruktor (engl. default constructor).
- Program će se prevesti (engl. compile), pokreće se, ali se ništa ne štampa.
- 😊 Program će se prevesti i pokrenuti ako se “A a” promeni na “A a (5)”.

KLASE I OBJEKTI

14. Šta je pogrešno u sledećem kôdu ?

```
#include <iostream>
using namespace std;
class TempClass {
public:
int i;
TempClass() {
int i = 5;
}
};
int main() {
TempClass temp(2);
}
```

- **Program ima grešku prevođenja (engl. compile), jer TempClass nema podrazumevani konstruktor (engl. default constructor).**
- **Program ima grešku prevođenja, jer TempClass nema konstruktor sa int argumentom.**
- **Program se prevodi, ali se ne izvršava, jer klasa C nije javna (engl. public).**
- **Program se prevodi i pokreće bez problema.**
- **Nijedno od datih rešenja**

KLASE I OBJEKTI

14. Šta je pogrešno u sledećem kôdu ?

```
#include <iostream>
using namespace std;
class TempClass {
public:
int i;
TempClass() {
int i = 5;
}
};
int main() {
TempClass temp(2);
}
```

- Program ima grešku prevođenja (engl. compile), jer TempClass nema podrazumevani konstruktor (engl. default constructor).
- 😊 Program ima grešku prevođenja, jer TempClass nema konstruktor sa int argumentom.
- Program se prevodi, ali se ne izvršava, jer klasa C nije javna (engl. public).
- Program se prevodi i pokreće bez problema.
- Nijedno od datih rešenja



KLASE I OBJEKTI

15. Ako je data izjava Circle x, koja je od sledećih izjava najpreciznija?

- **x sadrži int vrednost.**
- **x je objekat tipa Circle.**
- **Može se dodeliti int vrijednost varijabli x**
- **x je referenca na objekat Circle .**
- **Nijedno od datih rešenja**

KLASE I OBJEKTI

15. Ako je data izjava Circle x, koja je od sledećih izjava najpreciznija?

- x sadrži int vrednost.
-  x je objekat tipa Circle.
-  Može se dodeliti int vrijednost varijabli x
- x je referenca na objekat Circle .
- Nijedno od datih rešenja


16. Analizirati sledeći kôd:

```
#include <iostream>
using namespace std;
class Test {
public:
int x;
Test() {
cout << "Test";
}
};
int main() {
Test test;
cout << test.x;
}
```

- **Program ima grešku prevođenja, jer test nije inicijaliziran.**
- **Program ima grešku prevođenja jer x nije bio inicijaliziran.**
- **Program se pokreće bez problema, ali test.x ima nepredvidivu vrednost.**
- **Program ima grešku prevođenja, jer Test nema podrazumevani konstruktor (engl. default constructor).**
- **Nijedno od datih rešenja**

16. Analizirati sledeći kôd:

```
#include <iostream>
using namespace std;
class Test {
public:
int x;
Test() {
cout << "Test";
}
};
int main() {
Test test;
cout << test.x;
}
```

- **Program ima grešku prevođenja, jer test nije inicijaliziran.**
- **Program ima grešku prevođenja jer x nije bio inicijaliziran.**
-  **Program se pokreće bez problema, ali test.x ima nepredvidivu vrednost.**
- **Program ima grešku prevođenja, jer Test nema podrazumevani konstruktor (engl. default constructor).**
- **Nijedno od datih rešenja**

KLASE I OBJEKTI





17. Koje su od sledećih izjava istinite?

- **Lokalne promenljive nemaju podrazumevane vrednosti (engl. default values).**
- **Polja podataka u klasi nemaju podrazumevane vrednosti**
- **Varijabla primitivnog tipa ima vrednost primitivnog tipa.**
- **Naziv objekta je kao konstanta kojoj se ne može dodeliti novi objekat.**



KLASE I OBJEKTI

17. Koje su od sledećih izjava istinite?

-  **Lokalne promenljive nemaju podrazumevane vrednosti (engl. default values).**
-  **Polja podataka u klasi nemaju podrazumevane vrednosti**
-  **Varijabla primitivnog tipa ima vrednost primitivnog tipa.**
-  **Naziv objekta je kao konstanta kojoj se ne može dodeliti novi objekat.**



KLASE I OBJEKTI

18. Neka su circle1 i circle2 dva Circle objekta. Šta je rezultat izvršavanja sledeće izjave?

```
circle2 = circle1;
```

- **Kopira sadržaj circle1 u circle2**
- **Čini da circle2 i circle1 postanu isti objekat.**
- **Kopira sadržaj circle2 u circle1.**
- **Ova izjava nije validna.**



KLASE I OBJEKTI

18. Neka su circle1 i circle2 dva Circle objekta. Šta je rezultat izvršavanja sledeće izjave?

circle2 = circle1;



Kopira sadržaj circle1 u circle2

- Čini da circle2 i circle1 postanu isti objekat.
- Kopira sadržaj circle2 u circle1.
- Ova izjava nije validna.







KLASE I OBJEKTI

19. Koje su od sledećih izjava istinite?

- **Imena objekta su kao imena nizova. Kada se jednom definiše ime objekta, oni referenciraju jedan objekat.**
- **Ime objekta se ne može ponovo dodeliti da referencira neki drugi objekat.**
- **Naziv objekta je konstanta, iako se sadržaj objekta može promeniti.**
- **Objekat je pridružen samom imenu objekta.**

19. Koje su od sledećih izjava istinite?






-  **Imena objekta su kao imena nizova. Kada se jednom definiše ime objekta, oni referenciraju jedan objekat.**
-  **Ime objekta se ne može ponovo dodeliti da referencira neki drugi objekat.**
-  **Naziv objekta je konstanta, iako se sadržaj objekta može promeniti.**
-  **Objekat je pridružen samom imenu objekta.**

KLASE I OBJEKTI

20. Koje su od sledećih izjava istinite?

- Izjava **Circle circle = Circle()** kreira Circle objekat koristeći konstruktor bez argumenta (engl. no-arg constructor) te kopira svoj sadržaj u circle.
- Izjava **Circle circle = Circle(5)** kreira Circle objekat sa radijusom čija je vrednost 5 te kopira svoj sadržaj u circle.
- Izjavu **Circle circle = Circle()** bi trebalo zameniti izjavom **Circle circle**.
- Izjavu **Circle circle = Circle(5)** bi trebalo zameniti izjavom **Circle circle(5)**.
- Nema podrazumevanih vrednosti za polja podataka u klase
- Nijedno od datih rešenja

20. Koje su od sledećih izjava istinite?

-  Izjava `Circle circle = Circle()` kreira `Circle` objekat koristeći konstruktor bez argumenta (engl. no-arg constructor) te kopira svoj sadržaj u `circle`.
-  Izjava `Circle circle = Circle(5)` kreira `Circle` objekat sa radijusom čija je vrednost 5 te kopira svoj sadržaj u `circle`.
-  Izjavu `Circle circle = Circle()` bi trebalo zameniti izjavom `Circle circle`.
-  Izjavu `Circle circle = Circle(5)` bi trebalo zameniti izjavom `Circle circle(5)`.
-  Nema podrazumevanih vrednosti za polja podataka u klase
 - Nijedno od datih rešenja

KLASE I OBJEKTI

21. Analizirati sledeći kôd:


```
#include <iostream>
using namespace std;
class B {
public:
B() { };
int k;
};
int main() {
B b;
cout << b.k << endl;
return 0;
}
```

- **Program ima grešku prevođenja (engl. compile error), jer se ne može pristupiti b.k**
- **Program prikazuje 0.**
- **Program prikazuje 1.**
- **Program prikazuje nepredvidljiv broj.**
- **Program ima grešku kod izvršavanja programa (engl. runtime error), jer b.k nema vrednost.**

KLASE I OBJEKTI

21. Analizirati sledeći kôd:

```
#include <iostream>
using namespace std;
class B {
public:
B() { };
int k;
};
int main() {
B b;
cout << b.k << endl;
return 0;
}
```

- Program ima grešku prevođenja (engl. **compile error**), jer se ne može pristupiti **b.k**
- Program prikazuje 0.
- Program prikazuje 1.
-  Program prikazuje nepredvidljiv broj.
- Program ima grešku kod izvršavanja programa (engl. **runtime error**), jer **b.k** nema vrednost.






KLASE I OBJEKTI

22. Koje su od sledećih izjava istinite?

- **C ++ omogućava razdvajanje deklaracije klase od njene implementacije.**
- **Deklaracija klasa opisuje detalje klase, a implementacija klase detalje implementacije.**
- **Deklaracija klasa je jednostavno spisak svih polja podataka, prototipova konstruktora, prototipova funkcija. Implementacija klase implementira konstruktore i funkcije.**
- **Deklaracija klasa i implementacija klase se nalaze u dvije odvojene datoteke. Obe datoteke treba da imaju isto ime, ali sa različitim ekstenzijama imena (engl. extension names).**
- **Deklaracija klase ima ekstenziju imena .h i datoteka implementacije klase ima ekstenziju imena .cpp.**

KLASE I OBJEKTI




22. Koje su od sledećih izjava istinite?

-  C ++ omogućava razdvajanje deklaracije klase od njene implementacije.
-  Deklaracija klasa opisuje detalje klase, a implementacija klase detalje implementacije.
-  Deklaracija klasa je jednostavno spisak svih polja podataka, prototipova konstruktora, prototipova funkcija. Implementacija klase implementira konstruktore i funkcije.
-  Deklaracija klasa i implementacija klase se nalaze u dvije odvojene datoteke. Obe datoteke treba da imaju isto ime, ali sa različitim ekstenzijama imena (engl. extension names).
-  Deklaracija klase ima ekstenziju imena .h i datoteka implementacije klase ima ekstenziju imena .cpp.

23. Koje su od sledećih izjava istinite?

- **Simbol `::` se zove operator dosega (engl. scope operator).**
- **Operator binarnog dosega (engl. binary scope operator) se može koristiti kao `ClassName::member` da označi prevodiocu da član pripada klasi.**
- **Unarni operator dosega (engl. unary scope operator) se može koristiti kao `::var` da se pokaže prevodiocu da je neka varijabla globalna varijabla.**

23. Koje su od sledećih izjava istinite?

-  Simbol `::` se zove operator dosega (engl. scope operator).
-  Operator binarnog dosega (engl. binary scope operator) se može koristiti kao `ClassName::member` da označi prevodiocu da član pripada klasi.
-  Unarni operator dosega (engl. unary scope operator) se može koristiti kao `::var` da se pokaže prevodiocu da je neka varijabla globalna varijabla.

KLASE I OBJEKTI

24. Dati rezultat sledećeg kôda:

```
#include <iostream>
using namespace std;
class A {
public:
int x;
int y;
int z;
A(): x(1), y(2), z(3) {
}
};
```

```
int main() {
A a;
cout << a.x << " " << a.y << " " << a.z;
return 0;
}
```

- 1 1 1
- 1 1 2
- 1 2 3
- 2 2 2
- 3 3 3

KLASE I OBJEKTI

24. Dati rezultat sledećeg kôda:

```
#include <iostream>
using namespace std;
class A {
public:
int x;
int y;
int z;
A(): x(1), y(2), z(3) {
}
};
```

```
int main() {
A a;
cout << a.x << " " << a.y << " " << a.z;
return 0;
}
```

- 1 1 1
- 1 1 2
- 😊 1 2 3
- 2 2 2
- 3 3 3

KLASE I OBJEKTI

25. Neka su dvije datoteke zaglavlja (engl. header files) t1.h i t2.h koje sadrže deklaracije za klasu T. Šta će se desiti ako obje uključite u program?

- **Generiše se višestruka greška deklaracije ako datoteke zaglavlja ne uključuju guard operator.**
- **Prevodilac će automatski odlučiti koja će implementacija biti korišćena**
- **Program će se izvršiti bez problema i koristiće se prva datoteka zaglavlja koja je uključena.**
- **Program će se izvršiti bez problema i prva datoteka zaglavlja koja je uključena će se koristiti ako datoteke zaglavlja imaju guard operator.**

KLASE I OBJEKTI

25. Neka su dvije datoteke zaglavlja (engl. header files) t1.h i t2.h koje sadrže deklaracije za klasu T. Šta će se desiti ako obje uključite u program?

- **Generiše se višestruka greška deklaracije ako datoteke zaglavlja ne uključuju guard operator.**
- **Prevodilac će automatski odlučiti koja će implementacija biti korišćena**
- **Program će se izvršiti bez problema i koristiće se prva datoteka zaglavlja koja je uključena.**
- ☺ **Program će se izvršiti bez problema i prva datoteka zaglavlja koja je uključena će se koristiti ako datoteke zaglavlja imaju guard operator.**

26. Konstruktori _____ i funkcije _____ su definisane kao inline u klase A.

```
class A {  
public:  
A() {  
value = 0;  
}  
A(double);  
double f1() {  
// Return a number  
return value;  
}  
double f2();  
private:  
double value;  
};
```

- **A()**
- **A(double)**
- **f1()**
- **f2()**

26. Konstruktori _____ i funkcije _____ su definisane kao inline u klase A.

```
class A {  
public:  
A() {  
value = 0;  
}  
A(double);  
double f1() {  
// Return a number  
return value;  
}  
double f2();  
private:  
double value;  
};
```



A()

- **A(double)**







f1()

- **f2()**

27. Koje su od sledećih izjava istinite?

- **Koristiti ključnu reč private da se obuhvate (engl. encapsulate) polja podataka.**
- **Kapsulacija polja podataka omogućava da program bude jednostavan za održavanje.**
- **Kapsulacija polja podataka čini program kratkim.**
- **Kapsulacija polja podataka pomaže u sprečavanju programskih grešaka.**
- **Ako se ne koristi ključna reč public, vidljivost je podrazumevano (engl. default) private.**

27. Koje su od sledećih izjava istinite?

-  Koristiti ključnu reč **private** da se obuhvate (engl. **encapsulate**) polja podataka.
-  Kapsulacija polja podataka omogućava da program bude jednostavan za održavanje.
 - Kapsulacija polja podataka čini program kratkim.
-  Kapsulacija polja podataka pomaže u sprečavanju programskih grešaka.
-  Ako se ne koristi ključna reč **public**, vidljivost je podrazumevano (engl. **default**) **private**.

KLASE I OBJEKTI


28. Pretpostavimo da se želi obezbediti akcesor funkcija (engl. accessor function) za bulovsku funkcionalnost “finished”, koji potpis (engl. signature) funkcije treba izabrati?

- **void getFinished()**
- **bool getFinished()**
- **bool isFinished()**
- **void isFinished()**



KLASE I OBJEKTI

28. Pretpostavimo da se želi obezbediti akcesor funkcija (engl. accessor function) za bulovsku funkcionalnost “finished”, koji potpis (engl. signature) funkcije treba izabrati?

- **void getFinished()**
- **bool getFinished()**
-  **bool isFinished()**
- **void isFinished()**

KLASE I OBJEKTI

29. Dati rezultat izvršavanja sledećeg kôda:

```
#include <iostream>
using namespace std;
class Foo {
public:
int x; // data field
int y; // data field
Foo() {
x = 10;
y = 10;
}
void p() {
int x = 20; // local variable
cout << "x je " << x << " ";
cout << "y je " << y << endl;
}
};
```

```
int main() {
Foo foo;
foo.p();
return 0;
}
```


- **x je 10 y je 10**
- **x is 20 y je 20**
- **x is 20 y je 10**
- **x is 10 y je 20**

KLASE I OBJEKTI

29. Dati rezultat izvršavanja sledećeg kôda:

```
#include <iostream>
using namespace std;
class Foo {
public:
int x; // data field
int y; // data field
Foo() {
x = 10;
y = 10;
}
void p() {
int x = 20; // local variable
cout << "x je " << x << " ";
cout << "y je " << y << endl;
}
};
```

```
int main() {
Foo foo;
foo.p();
return 0;
}
```

- **x je 10 y je 10**
- **x is 20 y je 20**
-  **x is 20 y je 10**
- **x is 10 y je 20**

KLASE I OBJEKTI

30. Analizirati sledeći kôd:

```
#include <iostream>
using namespace std;
class B {
public:
    B() { };
private:
    int k;
};
int main() {
    B b;
    cout << b.k << endl;
    return 0;
}
```

- Program prikazuje 0.
- Program prikazuje 1.
- Program prikazuje nepredvidljiv broj.
- Program ima grešku prevođenja jer se ne može pristupiti b.k
- Program ima grešku kod izvršavanja (engl. runtime error) jer b.k nema vrednost.

KLASE I OBJEKTI

30. Analizirati sledeći kôd:

```
#include <iostream>
using namespace std;
class B {
public:
    B() { };
private:
    int k;
};
int main() {
    B b;
    cout << b.k << endl;
    return 0;
}
```

- Program prikazuje 0.
- Program prikazuje 1.
- Program prikazuje nepredvidljiv broj.
- 😊 Program ima grešku prevođenja jer se ne može pristupiti b.k
- Program ima grešku kod izvršavanja (engl. runtime error) jer b.k nema vrednost.

KLASE I OBJEKTI

31. Koje su od sledećih izjava tačne?

- **privatni članovi klase moraju se deklarirati pre javnih**
- **javnim podacima članovima objekta izvan objekta pristupa se pomoću bilo koje funkcije članice.**
- **objekti klase imaju sopstvene kopije vrednosti podataka članova**
- **objekti klase dele zajedničke vrednosti podataka članova**
- **unutar deklaracije klase navode se prototipovi (deklaracije) funkcija članica**
- **klasa može da ima više preklopljenih destruktora.**

KLASE I OBJEKTI

31. Koje su od sledećih izjava tačne?

- privatni članovi klase moraju se deklarirati pre javnih
- javnim podacima članovima objekta izvan objekta pristupa se pomoću bilo koje funkcije članice.
- ☺ objekti klase imaju sopstvene kopije vrednosti podataka članova
- objekti klase dele zajedničke vrednosti podataka članova
- ☺ unutar deklaracije klase navode se prototipovi (deklaracije) funkcija članica
- klasa može da ima više preklopljenih destruktora.

KLASE I OBJEKTI

32. Koje su od sledećih tvrdnji tačne?

- U jeziku C++ postoji funkcija bez povratnog tipa i argumenata.
- Podaci članovi ili funkcije članice deklarirani nakon ključne reči `private` dostupni su funkcijama članicama klase u kojoj su deklarirani.
- Promenljive deklarirane u telu neke funkcije članice zovu se podaci članovi i mogu se koristiti u svim funkcijama članicama klase.
- Članovi klase deklarirani kao `protected` dostupni su svuda gde je objekat klase dostupan.
- Skriveni pokazivač na objekat za koji je pozvana funkcija članica zove se `this`
- Moguće je dodeljivati objekte iste klase jedan drugome

KLASE I OBJEKTI

32. Koje su od sledećih tvrdnji tačne?



U jeziku C++ postoji funkcija bez povratnog tipa i argumenata.

- **Podaci članovi ili funkcije članice deklarirani nakon ključne reči private dostupni su funkcijama članicama klase u kojoj su deklarirani.**
- **Promenljive deklarirane u telu neke funkcije članice zovu se podaci članovi i mogu se koristiti u svim funkcijama članicama klase.**
- **Članovi klase deklarirani kao protected dostupni su svuda gde je objekat klase dostupan.**
- **Skriveni pokazivač na objekat za koji je pozvana funkcija članica zove se his**



Moguće je dodeljivati objekte iste klase jedan drugome

KLASE I OBJEKTI

33. Ključna reč _____ označava da se objekat ili promenljiva ne mogu menjati nakon inicijalizacije.

- **konstanta**
- **static**
- **volatile**
- **private**
- **no_copy**
- **ništa od ponuđenog**

KLASE I OBJEKTI

33. Ključna reč _____ označava da se objekat ili promenljiva ne mogu menjati nakon inicijalizacije.

- **konstanta**
- **static**
- **volatile**
- **private**
- **no_copy**
- 😊 **ništa od ponuđenog**

KLASE I OBJEKTI




34. Koje su od sledećih tvrdnji tačne?

- **klasa može imati više preklopljenih konstruktora.**
- **klasa može da ima više konstruktora.**
- **destruktor klase može da ima argumente.**
- **svaka klasa ima konstruktor i destruktor.**
- **sve predhodno navedeno**



KLASE I OBJEKTI

34. Koje su od sledećih tvrdnji tačne?

-  **klasa može imati više preklopljenih konstruktora.**
-  **klasa može da ima više konstruktora.**
 - **destruktor klase može da ima argumente.**
-  **svaka klasa ima konstruktor i destruktor.**
 - **sve predhodno navedeno**



KLASE I OBJEKTI

35. Koje su od sledećih tvrdnji tačne?

- **destruktor klase Primer je `Primer : : ~ Primer () ;` ;**
- **Potrebno je eksplicitno pozivati destruktor za lokalnu promenljivu.**
- **Za svaku klasu mora se napisati konstruktor kopije.**
- **Ako se za klasu napiše konstruktor sa argumentima, podrazumevani (default) konstruktor više nije dostupan.**
- **Konstruktor kopije se poziva u slučaju kada se objekat klase inicijalizuje drugim objektom.**
- **Spajanje (enfl. Concatenation) string objekata može se postići i pomoću operatora `+=`.**

KLASE I OBJEKTI

35. Koje su od sledećih tvrdnji tačne?

- ☺ **destruktor klase Primer je `Primer : : ~ Primer () ;`**
 - **Potrebno je eksplicitno pozivati destruktor za lokalnu promenljivu.**
 - **Za svaku klasu mora se napisati konstruktor kopije.**
- ☺ **Ako se za klasu napiše konstruktor sa argumentima, podrazumevani (default) konstruktor više nije dostupan.**
- ☺ **Konstruktor kopije se poziva u slučaju kada se objekat klase inicijalizuje drugim objektom.**
- ☺ **Spajanje (enfl. Concatenation) string objekata može se postići i pomoću operatora `+=`.**

36. Konkatencija (nedovezivanje) string objekata može se postići i pomoću operatora +=.

- **tačno**
- **pogrešno**



36. Konkatencija (nedovezivanje) string objekata može se postići i pomoću operatora +=.



tačno

- **pogrešno**








KLASE I OBJEKTI

37. Koje su od sledećih tvrdnji tačne?

- **Nasleđivanje je vrsta ponovnog korišćenja softvera u kome nove klase obuhvataju svojstva i ponašanje postojećih klasa i obogaćuju ih novim mogućnostima.**
- **Nasleđivanjem se olakšava ponovno korišćenje softvera koji se dokazao kao kvalitetan.**
- **Izvedena klasa ne nasleđuje konstruktore osnovne klase.**
- **Konstruktor izvedene klase je dužan da prosledi sve potrebne parametre konstruktoru osnovne klase.**
- **Članovi deklarisan kao protected u osnovnoj klasi dostupni su u osnovnoj i izvedenim klasama**

KLASE I OBJEKTI

37. Koje su od sledećih tvrdnji tačne?

-  Nasleđivanje je vrsta ponovnog korišćenja softvera u kome nove klase obuhvataju svojstva i ponašanje postojećih klasa i obogaćuju ih novim mogućnostima.
-  Nasleđivanjem se olakšava ponovno korišćenje softvera koji se dokazao kao kvalitetan.
-  Izvedena klasa ne nasleđuje konstruktore osnovne klase.
-  Konstruktor izvedene klase je dužan da prosledi sve potrebne parametre konstruktoru osnovne klase.
-  Članovi deklarirani kao `protected` u osnovnoj klasi dostupni su u osnovnoj i izvedenim klasama

KLASE I OBJEKTI

38. Kada se kreira objekat izvedene klase, poziva se _____ osnovne klase koji obavlja sve neophodne inicijalizacije podataka članova osnovne klase u objektu izvedene klase.

- **konstruktor**
- **konstruktor kopije**
- **destruktor**
- **konstruktor kopiranja**
- **virtualni konstruktor**

KLASE I OBJEKTI

38. Kada se kreira objekat izvedene klase, poziva se _____ osnovne klase koji obavlja sve neophodne inicijalizacije podataka članova osnovne klase u objektu izvedene klase.



konstruktor

- **konstruktor kopije**
- **destruktor**
- **konstruktor kopiranja**
- **virtualni konstruktor**






KLASE I OBJEKTI

39. Naznačiti tačne odgovore:

- **Kada se uništava objekat izvedene klase, destruktori se pozivaju u obrnutom redosledu od konstruktora.**
- **Ne mogu da se kreiraju objekti apstraktnih klasa**
- **Sve virtuelne funkcije u apstraktnoj osnovnoj klasi moraju se deklarirati kao čiste virtuelne funkcije.**
- **Ako je u osnovnoj klasi deklarirana čista virtuelna funkcija, onda izvedena klasa mora da implementira tu funkciju da bi bila konkretna klasa.**
- **Pokazivačima na objekat osnovne klase može se dodeljivati adresa objekta izvedene klase**
- **Ako osnovna klasa ima neku virtuelnu funkciju, onda i njen destruktork treba da bude virtuelan.**

KLASE I OBJEKTI

39. Naznačiti tačne odgovore:

-  Kada se uništava objekat izvedene klase, destruktori se pozivaju u obrnutom redosledu od konstruktora.
-  Ne mogu da se kreiraju objekti apstraktnih klasa
 - Sve virtuelne funkcije u apstraktnoj osnovnoj klasi moraju se deklarirati kao čiste virtuelne funkcije.
-  Ako je u osnovnoj klasi deklarirana čista virtuelna funkcija, onda izvedena klasa mora da implementira tu funkciju da bi bila konkretna klasa.
-  Pokazivačima na objekat osnovne klase može se dodeljivati adresa objekta izvedene klase
-  Ako osnovna klasa ima neku virtuelnu funkciju, onda i njen destruktork treba da bude virtuelan.

KLASE I OBJEKTI

40. _____ predstavlja celinu u stvarnom svetu koja može biti jasno identifikovana.

- **klasa**
- **objekat**
- **funkcija**
- **polje podataka (engl. data field)**

KLASE I OBJEKTI

40. _____ predstavlja celinu u stvarnom svetu koja može biti jasno identifikovana.

- klasa
- 😊 objekat
- funkcija
- polje podataka (engl. data field)

KLASE I OBJEKTI

41. _____ je struktura koja definiše objekte istog tipa.

- **klasa**
- **objekat**
- **funkcija**
- **polje podataka (engl. data field)**

41. _____ je struktura koja definiše objekte istog tipa.



klasa

- **objekat**
- **funkcija**
- **polje podataka (engl. data field)**

KLASE I OBJEKTI

42. _____ se poziva kada se želi kreirati objekat.

- **konstruktor**
- **funkcija main ()**
- **funkcija sa povratnim tipom (engl. return type)**
- **virtualni konstruktor**

42. _____ se poziva kada se želi kreirati objekat.

-  **konstruktor**
- **funkcija main ()**
 - **funkcija sa povratnim tipom (engl. return type)**
 - **virtualni konstruktor**

KLASE I OBJEKTI

43. Object-orijentisano programiranje omogućava izvođenje nove klase od postojećih klasa. To se zove:

- **kapsulacija**
- **nasleđivanje**
- **generalizacija**
- **polimorfizam**



43. Object-orijentisano programiranje omogućava izvođenje nove klase od postojećih klasa. To se zove:

- kapsulacija
- 😊 nasleđivanje
- generalizacija
- polimorfizam

44. Kapsuliranje je _____ podataka.

- **Skrivanje**
- **Apstrakcija**
- **Nasleđivanje**



44. Kapsuliranje je _____ podataka.



Skrivanje

- **Apstrakcija**
- **Nasleđivanje**



45. Uočavanje zajedničkih svojstava objekata i njihovo grupisanje u klasu naziva se:

- **Apstrakcija**
- **Kapsuliranje**
- **Polimorfizam**



45. Uočavanje zajedničkih svojstava objekata i njihovo grupisanje u klasu naziva se:



Apstrakcija

- **Kapsuliranje**
- **Polimorfizam**





KLASE I OBJEKTI

46. Objekat je primerak (instanca) neke:

- **Klase**
- **Promenljive**
- **Metode**



KLASE I OBJEKTI

46. Objekat je primerak (instanca) neke:



Klase

- **Promenljive**
- **Metode**




KLASE I OBJEKTI

47. Za definisanje klase koristi se ključna reč:

- **method**
- **class**
- **main**
- **object**

KLASE I OBJEKTI

47. Za definisanje klase koristi se ključna reč:

- **method**
-  **class**
- **main**
- **object**

48. Privatni članovi klase moraju se deklarirati pre javnih.

- Tačno
- Pogrešno



KLASE I OBJEKTI

48. Privatni članovi klase moraju se deklarirati pre javnih.

- Tačno



- Pogrešno



49. Javnim podacima članovima objekta izvan objekta pristupa se pomoću:

- **Bilo koje funkcije članice**
- **Operatora**
- **Operatora razrešenja dosega (::)**

49. Javnim podacima članovima objekta izvan objekta pristupa se pomoću:

- **Bilo koje funkcije članice**
-  **Operatora**
- **Operatora razrešenja dosega (::)**



KLASE I OBJEKTI

50. Sakrivanje detalja realizacije klase zove se:

- **Apstrakcija**
- **Kapsuliranje**
- **Polimorfizam**



KLASE I OBJEKTI

50. Sakrivanje detalja realizacije klase zove se:

- Apstrakcija
- 😊 Kapsuliranje
- Polimorfizam





KLASE I OBJEKTI

51. Javni članovi klase deklarišu se kao:

- **public**
- **protected**
- **private**



KLASE I OBJEKTI

51. Javni članovi klase deklarišu se kao:

- 😊 **public**
- **protected**
- **private**




52. Specijalna funkcija članica koja se poziva prilikom kreiranja objekta klase zove se:

- **Glavni metod**
- **Metod bez argumenata**
- **Konstruktor**
- **Rekurzivni metod**

KLASE I OBJEKTI

52. Specijalna funkcija članica koja se poziva prilikom kreiranja objekta klase zove se:

- **Glavni metod**
- **Metod bez argumenata**
-  **Konstruktor**
- **Rekurzivni metod**

KLASE I OBJEKTI

53. U jeziku C++ postoji funkcija bez povratnog tipa i argumenata:

- Tačno
- Pogrešno



53. U jeziku C++ postoji funkcija bez povratnog tipa i argumenata:



Tačno

- **Pogrešno**



KLASE I OBJEKTI

54. Podaci članovi ili funkcije članice deklarirani nakon ključne reči private dostupni su funkcijama članicama klase u kojoj su deklarirani:

- **Tačno**
- **Pogrešno**



54. Podaci članovi ili funkcije članice deklarirani nakon ključne reči private dostupni su funkcijama članicama klase u kojoj su deklarirani:



Tačno

- **Pogrešno**



KLASE I OBJEKTI

55. Promenljive deklarisanе u telu neke funkcije članice zovu se podaci članovi i mogu se koristiti u svim funkcijama članicama klase:

- **Tačno**
- **Pogrešno**



KLASE I OBJEKTI

55. Promenljive deklarisanе u telu neke funkcije članice zovu se podaci članovi i mogu se koristiti u svim funkcijama članicama klase:

- Tačno
- 😊 Pogrešno



KLASE I OBJEKTI

56. Članovi klase deklarirani kao _____ dostupni su svuda gde je objekat klase dostupan.

- **public**
- **protected**
- **private**



KLASE I OBJEKTI

56. Članovi klase deklarirani kao _____ dostupni su svuda gde je objekat klase dostupan.



public

- **protected**
- **private**

KLASE I OBJEKTI

57. Operator _____ može se koristiti za dodelu objekta neke klase drugom objektu iste klase.

- =
- ==
- **konstruktor**

KLASE I OBJEKTI

57. Operator _____ može se koristiti za dodelu objekta neke klase drugom objektu iste klase.



=

- ==

- **konstruktor**

KLASE I OBJEKTI

58. Ključna reč _____ označava da se objekat ili promenljiva ne mogu menjati nakon inicijalizacije.

- **const**
- **static**
- **volatile**

KLASE I OBJEKTI

58. Ključna reč _____ označava da se objekat ili promenljiva ne mogu menjati nakon inicijalizacije.



const

- **static**
- **volatile**


KLASE I OBJEKTI

59. Pod pretpostavkom da je sapun objekat klase Kozmetika, koji od sledećih je važeći poziv funkcije operi()?

- **operi ();**
- **sapun::operi();**
- **sapun.operi();**
- **sapun:operi();**

KLASE I OBJEKTI

59. Pod pretpostavkom da je sapun objekat klase Kozmetika, koji od sledećih je važeći poziv funkcije operi()?

- **operi ();**
- **sapun::operi();**
-  **sapun.operi();**
- **sapun:operi();**

KLASE I OBJEKTI

60. Ključna reč inline:

- **Ugrađuje telo funkcije neposredno u kod na mestu poziva**
- **Umeće ceo objekat u jedan red**
- **Izvršava kod jednog reda**



60. Ključna reč inline:

- 😊 **Ugrađuje telo funkcije neposredno u kod na mestu poziva**
 - **Umeće ceo objekat u jedan red**
 - **Izvršava kod jednog reda**

61. Operator :: služi za:

- **Pristup objektu**
- **Razrešenje dosega**
- **Poziv metode**



61. Operator :: služi za:

- **Pristup objektu**
-  **Razrešenje dosega**
- **Poziv metode**



62. Operator strelica →:

- **Pokazuje na objekat**
- **Pokazuje na operator množenja**
- **Pristupa članu objekta preko pokazivača**

KLASE I OBJEKTI

62. Operator strelica →:

- Pokazuje na objekat
- Pokazuje na operator množenja
- 😊 Pristupa članu objekta preko pokazivača



KLASE I OBJEKTI

63. Podrazumevani (default) konstruktor:

- **Mora da ima argumente**
- **Ne sme da ima argumente**
- **Može, ali ne mora da ima argumente**



KLASE I OBJEKTI

63. Podrazumevani (default) konstruktor:

- Mora da ima argumente
- 😊 Ne sme da ima argumente
- Može, ali ne mora da ima argumente





KLASE I OBJEKTI

64. Klasa može da ima samo jedan konstruktor.

- Tačno
- Pogrešno



64. Klasa može da ima samo jedan konstruktor.

- Tačno
- 😊 Pogrešno



65. Destruktor klase može da ima argumente.

- Tačno
- Pogrešno



KLASE I OBJEKTI

65. Destruktor klase može da ima argumente.

- Tačno

-  Pogrešno



66. Klasa može da ima više preklopljenih destruktora.

- Tačno
- Pogrešno



66. Klasa može da ima više preklopljenih destruktora.

- Tačno

-  Pogrešno



67. Klasa može imati više preklopljenih konstruktora.

- Tačno
- Pogrešno



67. Klasa može imati više preklopljenih konstruktora.



Tačno

- **Pogrešno**



68. Ako je funkcija i definisana, a ne samo deklarirana unutar definicije klase, ona je:

- **Preklopljena**
- **inline**
- **Statička**



68. Ako je funkcija i definisana, a ne samo deklarirana unutar definicije klase, ona je:

- **Preklopljena**
-  **inline**
- **Statička**

KLASE I OBJEKTI

69. Skriveni pokazivač na objekat za koji je pozvana funkcija članica zove se that.

- Tačno
- Pogrešno



KLASE I OBJEKTI

69. Skriveni pokazivač na objekat za koji je pozvana funkcija članica zove se that.

- Tačno

-  Pogrešno



KLASE I OBJEKTI

70. Moguće je dodeljivati objekte iste klase jedan drugom.

- **Tačno**
- **Pogrešno**



KLASE I OBJEKTI

70. Moguće je dodeljivati objekte iste klase jedan drugom.



Tačno

- **Pogrešno**





KLASE I OBJEKTI

71. Destruktor klase Primer je:

- **Primer::Primer();**
- **Primer::~~Primer();**



71. Destruktor klase Primer je:

- **Primer::Primer();**
- 😊 **Primer::~~Primer();**



72. Potrebno je eksplicitno pozivati destruktor za lokalnu promenljivu.

- Tačno
- Pogrešno



72. Potrebno je eksplicitno pozivati destruktor za lokalnu promenljivu.

- Tačno

 Pogrešno



KLASE I OBJEKTI

73. Za svaku klasu mora se napisati konstruktor kopije.

- Tačno
- Pogrešno



KLASE I OBJEKTI

73. Za svaku klasu mora se napisati konstruktor kopije.

- Tačno

-  Pogrešno



KLASE I OBJEKTI

74. Ako se za klasu napiše konstruktor sa argumentima, podrazumevani (default) konstruktor više nije dostupan.

- **Tačno**
- **Pogrešno**



74. Ako se za klasu napiše konstruktor sa argumentima, podrazumevani (default) konstruktor više nije dostupan.



Tačno

- **Pogrešno**



KLASE I OBJEKTI

75. Konstruktor kopije se poziva u slučaju kada se objekat klase inicijalizuje drugim objektom.

- **Tačno**
- **Pogrešno**



75. Konstruktor kopije se poziva u slučaju kada se objekat klase inicijalizuje drugim objektom.



Tačno

- **Pogrešno**



76. Za klasu string mora se uključiti zaglavlje:

- **cstring**
- **string**
- **iostream**



KLASE I OBJEKTI

76. Za klasu string mora se uključiti zaglavlje:

- **cstring**
-  **string**
- **iostream**





KLASE I OBJEKTI

77. Funkcija _____ briše znakove iz stringa.

- **erase**
- **delete**
- **at**



77. Funkcija _____ briše znakove iz stringa.



erase

- **delete**
- **at**

KLASE I OBJEKTI

78. Funkcija _____ pronalazi prvo pojavljivanje bilo kog znaka u stringu:

- **at**
- **find**
- **open**

78. Funkcija _____ pronalazi prvo pojavljivanje bilo kog znaka u stringu:

- at
-  find
- open



KRAJ!

