

PITANJA ZA II KOLOKVIJUM

NASLJEĐIVANJE, VIRTUELNE FUNKCIJE I POLIMORFIZAM





NASLJEĐIVANJE, VIRTUELNE FUNKCIJE I POLIMORFIZAM

1. Definicija svake klase sadrzi ključnu rec _____ iza koje se navodi ime klase:

- **class**
- **public**
- **extends**



NASLJEĐIVANJE, VIRTUELNE FUNKCIJE I POLIMORFIZAM

1. Definicija svake klase sadrzi ključnu rec _____ iza koje se navodi ime klase:



class

- **public**
- **extends**



NASLJEĐIVANJE, VIRTUELNE FUNKCIJE I POLIMORFIZAM

2. Kada je funkcija članica definisana izvan definicije klase, zaglavlje funkcije mora da sadrzi ime klase i _____, iza koje sledi ime funkcije koja povezuje funkciju članicu sa definicijom klase.

- ::
- **public**
- **class**

NASLJEĐIVANJE, VIRTUELNE FUNKCIJE I POLIMORFIZAM

2. Kada je funkcija članica definisana izvan definicije klase, zaglavlje funkcije mora da sadrži ime klase i _____, iza koje sledi ime funkcije koja povezuje funkciju članicu sa definicijom klase.



::

- **public**
- **class**

NASLJEĐIVANJE, VIRTUELNE FUNKCIJE I POLIMORFIZAM

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

- **tacno**
- **netacno**



NASLJEĐIVANJE, VIRTUELNE FUNKCIJE I POLIMORFIZAM

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



tacno

- **netacno**



NASLJEĐIVANJE, VIRTUELNE FUNKCIJE I POLIMORFIZAM

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

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



NASLJEĐIVANJE, VIRTUELNE FUNKCIJE I POLIMORFIZAM

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

- **Tacno**

 **Netacno**



NASLJEĐIVANJE, VIRTUELNE FUNKCIJE I POLIMORFIZAM

5. Pokazivaci na različite tipove nikada se ne mogu dodeljivati jedan drugom bez konverzije.

- **tacno**
- **netacno**



NASLJEĐIVANJE, VIRTUELNE FUNKCIJE I POLIMORFIZAM

5. Pokazivaci na različite tipove nikada se ne mogu dodeljivati jedan drugom bez konverzije.

- **tacno**



- netacno**





NASLJEĐIVANJE, VIRTUELNE FUNKCIJE I POLIMORFIZAM

6. Clanovi klase deklarirani kao _____ dostupni su svuda gde je objekat klase dostupan.

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



NASLJEĐIVANJE, VIRTUELNE FUNKCIJE I POLIMORFIZAM

6. Clanovi klase deklarirani kao _____ dostupni su svuda gde je objekat klase dostupan.



public

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



NASLJEĐIVANJE, VIRTUELNE FUNKCIJE I POLIMORFIZAM

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

- =
- ==
- konstruktor



NASLJEĐIVANJE, VIRTUELNE FUNKCIJE I POLIMORFIZAM

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



=

- ==
- **konstruktor**



NASLJEĐIVANJE, VIRTUELNE FUNKCIJE I POLIMORFIZAM

8. Funkcija koja nije članica klase mora se deklarirati kao prijatelj klase da bi mogla da pristupi privatnim podacima članovima klase.

- **tacno**
- **netacno**



NASLJEĐIVANJE, VIRTUELNE FUNKCIJE I POLIMORFIZAM

8. Funkcija koja nije članica klase mora se deklarirati kao prijatelj klase da bi mogla da pristupi privatnim podacima članovima klase.



tacno

- **netacno**



NASLJEĐIVANJE, VIRTUELNE FUNKCIJE I POLIMORFIZAM

9. Ključna rec _____ označava da se objekat ili promenljiva ne mogu menjati nakon inicijalizacije.

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



NASLJEĐIVANJE, VIRTUELNE FUNKCIJE I POLIMORFIZAM

9. Ključna rec _____ označava da se objekat ili promenljiva ne mogu menjati nakon inicijalizacije.



const

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





NASLJEĐIVANJE, VIRTUELNE FUNKCIJE I POLIMORFIZAM

10. Ključna reč _____ označava definiciju preklopljene operatorske funkcije.

- **operator**
- **new**
- **static**



NASLJEĐIVANJE, VIRTUELNE FUNKCIJE I POLIMORFIZAM

10. Ključna rec _____ označava definiciju preklopljene operatorske funkcije.



operator

- **new**
- **static**



NASLJEĐIVANJE, VIRTUELNE FUNKCIJE I POLIMORFIZAM

11. U jeziku C++ mogu se preklopiti samo postojeći operatori.

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



NASLJEĐIVANJE, VIRTUELNE FUNKCIJE I POLIMORFIZAM

11. U jeziku C++ mogu se preklopiti samo postojeći operatori.



Tacno

- **Netacno**



NASLJEĐIVANJE, VIRTUELNE FUNKCIJE I POLIMORFIZAM

12. Funkcije koje se mogu redefinisati deklarisu se pomocu kljucne reci

- **virtual**
- **const**
- **public**



NASLJEĐIVANJE, VIRTUELNE FUNKCIJE I POLIMORFIZAM

12. Funkcije koje se mogu redefinisati deklarisu se pomocu kljucne reci

-  **virtual**
- **const**
 - **public**



NASLJEĐIVANJE, VIRTUELNE FUNKCIJE I POLIMORFIZAM

13. Dati rezultat izvršavanja sledećeg koda.

```
#include <iostream>
using namespace std;
class ParentClass {
public:
    int id;
    ParentClass(int id) {
        this->id = id;
    }
    void print() {
        cout << id << endl;
    }
};
```

```
class ChildClass: public ParentClass {
public:
    int id;
    ChildClass(int id): ParentClass(1) {
        this->id = id;
    }
};
int main() {
    ChildClass c(2);
    c.print();
    return 0;
}
```

- 0
- 1
- 2
- ništa

NASLJEĐIVANJE, VIRTUELNE FUNKCIJE I POLIMORFIZAM

13. Dati rezultat izvršavanja sledećeg koda.

```
#include <iostream>
using namespace std;
class ParentClass {
public:
    int id;
    ParentClass(int id) {
        this->id = id;
    }
    void print() {
        cout << id << endl;
    }
};
```

```
class ChildClass: public ParentClass {
public:
    int id;
    ChildClass(int id): ParentClass(1) {
        this->id = id;
    }
};
int main() {
    ChildClass c(2);
    c.print();
    return 0;
}
```

- 0
-  1
- 2
- ništa

NASLJEĐIVANJE, VIRTUELNE FUNKCIJE I POLIMORFIZAM

14. Neka su Circle i Rectangle klase izvedene od klase GeometricObject i neka je zadato:

```
void displayGeometricObject(GeometricObject shape){  
    cout << shape.toString() << endl;  
}
```

Naznačiti validne pozive date funkcije.

- **displayGeometricObject(GeometricObject("black", true));**
- **displayGeometricObject(Circle(5));**
- **displayGeometricObject(Rectangle(2, 3));**
- **displayGeometricObject(string());**

NASLJEĐIVANJE, VIRTUELNE FUNKCIJE I POLIMORFIZAM

14. Neka su Circle i Rectangle klase izvedene od klase GeometricObject i neka je zadato:

```
void displayGeometricObject(GeometricObject shape){  
    cout << shape.toString() << endl;  
}
```

Naznačiti validne pozive date funkcije.

-  **displayGeometricObject(GeometricObject("black", true));**
-  **displayGeometricObject(Circle(5));**
-  **displayGeometricObject(Rectangle(2, 3));**
- **displayGeometricObject(string());**

NASLJEĐIVANJE, VIRTUELNE FUNKCIJE I POLIMORFIZAM

15. Dati rezultat izvršavanja sledećeg koda.

```
#include <iostream>
using namespace std;
class B {
public:
~B() {
cout << "B";
}
};
class A: public B {
public:
~A() {
cout << "A";
}
};
```

```
int main() {
A a;
return 0;
}
```

- AB
- BA
- A
- B
- AA

NASLJEĐIVANJE, VIRTUELNE FUNKCIJE I POLIMORFIZAM

15. Dati rezultat izvršavanja sledećeg koda.

```
#include <iostream>
using namespace std;
class B {
public:
    ~B() {
        cout << "B";
    }
};
class A: public B {
public:
    ~A() {
        cout << "A";
    }
};
```

```
int main() {
    A a;
    return 0;
}
```



AB

- **BA**
- **A**
- **B**
- **AA**

NASLJEĐIVANJE, VIRTUELNE FUNKCIJE I POLIMORFIZAM

16. Šta je pogrešno u sledećem kodu?

```
class Fruit {  
public:  
    Fruit(int id)  
{  
}  
};  
  
class Apple: public Fruit {  
public:  
    Apple() {  
}  
};
```

- Program će se prevesti (engl. compile) ako se za Fruit doda konstruktor bez argumenata
- Program ima grešku prevođenja, jer Fruit nema konstruktor bez argumenata.
- Program će se prevesti ako se izbriše konstruktor za Fruit.
- Program će se prevesti ako se zameni “Apple()” sa “Apple(): Fruit(4)”

NASLJEĐIVANJE, VIRTUELNE FUNKCIJE I POLIMORFIZAM

16. Šta je pogrešno u sledećem kodu?

```
class Fruit {  
public:  
    Fruit(int id)  
{  
}  
};  
  
class Apple: public Fruit {  
public:  
    Apple() {  
}  
};
```

- 😊 Program će se prevesti (engl. compile) ako se za Fruit doda konstruktor bez argumenata
- 😊 Program ima grešku prevođenja, jer Fruit nema konstruktor bez argumenata.
- 😊 Program će se prevesti ako se izbriše konstruktor za Fruit.
- 😊 Program će se prevesti ako se zameni "Apple()" sa "Apple(): Fruit(4)"

NASLJEĐIVANJE, VIRTUELNE FUNKCIJE I POLIMORFIZAM

17. Koje su od sledećih izjava tačne?

- Za redefiniranje funkcije, funkcija mora biti definisana u izvedenoj klasi koristeći isti potpis (engl. signature) i povratni tip (engl. return type) kao u svojoj osnovnoj klasi.
- Preklapanje (engl. overloading) funkcije omogućava više od jedne funkcije s istim imenom, ali s različitim potpisima koji prave razliku između njih
- Ako se dvije funkcije razlikuju se samo po tipu povratne vrednosti to je greška.
- Privatna funkcija ne može biti redefinisana. Ako funkcija, definisana u izvedenoj klasi, je privatna u osnovnoj klasi, dvije funkcije su potpuno nepovezane.

NASLJEĐIVANJE, VIRTUELNE FUNKCIJE I POLIMORFIZAM

17. Koje su od sledećih izjava tačne?

-  Za redefiniranje funkcije, funkcija mora biti definisana u izvedenoj klasi koristeći isti potpis (engl. signature) i povratni tip (engl. return type) kao u svojoj osnovnoj klasi.
-  Preklapanje (engl. overloading) funkcije omogućava više od jedne funkcije s istim imenom, ali s različitim potpisima koji prave razliku između njih
-  Ako se dvije funkcije razlikuju se samo po tipu povratne vrednosti to je greška.
-  Privatna funkcija ne može biti redefinisana. Ako funkcija, definisana u izvedenoj klasi, je privatna u osnovnoj klasi, dvije funkcije su potpuno nepovezane.

NASLJEĐIVANJE, VIRTUELNE FUNKCIJE I POLIMORFIZAM

18. Koje su od sledećih izjava tačne?

- **Funkcija može biti preklopljena (engl. overloaded) u istoj klasi.**
- **Funkcija može biti redefinisana u istoj klasi.**
- **Ako je funkcija vrši preklapanje druge funkcije, te dvije funkcije moraju imati isti potpis.**
- **Ako je funkcija redefinisana drugoj funkciji, te dvije funkcije moraju imati isti potpis.**
- **Konstruktori izvedene klase se mogu naslediti od osnovne klase**
- **Destruktori izvedene klase se mogu naslediti od osnovne klase**

NASLJEĐIVANJE, VIRTUELNE FUNKCIJE I POLIMORFIZAM

18. Koje su od sledećih izjava tačne?



Funkcija može biti preklopljena (engl. overloaded) u istoj klasi.

- **Funkcija može biti redefinisana u istoj klasi.**
- **Ako je funkcija vrši preklapanje druge funkcije, te dvije funkcije moraju imati isti potpis.**



Ako je funkcija redefinisana drugoj funkciji, te dvije funkcije moraju imati isti potpis.

- **Konstruktori izvedene klase se mogu naslediti od osnovne klase**
- **Destruktori izvedene klase se mogu naslediti od osnovne klase**

NASLJEĐIVANJE, VIRTUELNE FUNKCIJE I POLIMORFIZAM

19. Za poziv `toString()` funkcije koja definisana u `GeometricObject` iz `Circle` objekta `c`, koristiti:

- `super.toString()`
- `c.super.toString()`
- `c.GeometricObject::toString()`
- `c->GeometricObject::toString()`

NASLJEĐIVANJE, VIRTUELNE FUNKCIJE I POLIMORFIZAM

19. Za poziv `toString()` funkcije koja definisana u `GeometricObject` iz `Circle` objekta `c`, koristiti:

- `super.toString()`
- `c.super.toString()`
- 😊 `c.GeometricObject::toString()`
- `c->GeometricObject::toString()`

NASLJEĐIVANJE, VIRTUELNE FUNKCIJE I POLIMORFIZAM

20. Ako je u osnovnoj klasi deklarirana čista virtualna funkcija, onda izvedena klasa mora da implementira tu funkciju da bi bila konkretna klasa.

- Tačno
- Pogrešno

NASLJEĐIVANJE, VIRTUELNE FUNKCIJE I POLIMORFIZAM

20. Ako je u osnovnoj klasi deklarirana čista virtualna funkcija, onda izvedena klasa mora da implementira tu funkciju da bi bila konkretna klasa.



Tačno

- **Pogrešno**

NASLJEĐIVANJE, VIRTUELNE FUNKCIJE I POLIMORFIZAM

21. Pokazivačima na objekat osnovne klase može se dodeljivati adresa objekta izvedene klase.

- Tačno
- Pogrešno

NASLJEĐIVANJE, VIRTUELNE FUNKCIJE I POLIMORFIZAM

21. Pokazivačima na objekat osnovne klase može se dodeljivati adresa objekta izvedene klase.



Tačno

- **Pogrešno**

NASLJEĐIVANJE, VIRTUELNE FUNKCIJE I POLIMORFIZAM

22. Ako osnovna klasa ima neku virtuelnu funkciju, onda i njen destruktor treba da bude virtuelan.

- Tačno
- Pogrešno



NASLJEĐIVANJE, VIRTUELNE FUNKCIJE I POLIMORFIZAM

22. Ako osnovna klasa ima neku virtuelnu funkciju, onda i njen destruktor treba da bude virtuelan.



Tačno

- **Pogrešno**

NASLJEĐIVANJE, VIRTUELNE FUNKCIJE I POLIMORFIZAM

23. Dati rezultat izvršavanja sledećeg koda.

```
#include <iostream>
#include <string>
using namespace std;
class Person {
public:
void printInfo() {
cout << getInfo() << endl;
}
virtual string getInfo() {
return "Person";
}
};
```

```
class Student: public Person {
public:
virtual string getInfo() {
return "Student";
}
};
```

```
int main() {
Person().printInfo();
Student().printInfo();
}
```

- **Person Person**
- **Person Student**
- **Student Student**
- **Student Person**

NASLJEĐIVANJE, VIRTUELNE FUNKCIJE I POLIMORFIZAM

23. Dati rezultat izvršavanja sledećeg koda.

```
#include <iostream>
#include <string>
using namespace std;
class Person {
public:
void printInfo() {
cout << getInfo() << endl;
}
virtual string getInfo() {
return "Person";
}
};
```

```
class Student: public Person {
public:
virtual string getInfo() {
return "Student";
}
};
```

```
int main() {
Person().printInfo();
Student().printInfo();
}
```

- 
- **Person Person**
 - **Person Student**
 - **Student Student**
 - **Student Person**

NASLJEĐIVANJE, VIRTUELNE FUNKCIJE I POLIMORFIZAM

24. Koje su od sledećih izjava tačne?

- Ako je funkcija definisana kao virtualna u osnovnoj klasi, automatski je virtualna i u svim svojim izvedenim klasama. Nije potrebno dodavati ključnu reč u definiciji virtualne funkcije u izvedenoj klasi.
- Ako funkcija neće biti redefinisana, efikasnije je bez njenog definisanja kao virtualne funkcije, jer je potrebno više vremena i resursa sistema za dinamičko povezivanje (engl. bind) virtualnih funkcija za vrijeme izvršavanja (engl. at runtime)
- Virtualna funkcija se može implementirati u nekoliko izvedenih klasa. C ++ dinamički povezuje (engl. bind) implementacije funkcije za vrijeme izvršavanja, što odlučuje stvarna klasa objekta koja je referencirana preko varijable
- Prevodilac pronalazi odgovarajuću funkciju prema tipu parametra, broju parametara, te redosledu parametara u vrijeme prevođenja (engl. at compile time).

NASLJEĐIVANJE, VIRTUELNE FUNKCIJE I POLIMORFIZAM

24. Koje su od sledećih izjava tačne?

-  Ako je funkcija definisana kao virtualna u osnovnoj klasi, automatski je virtualna i u svim svojim izvedenim klasama. Nije potrebno dodavati ključnu reč u definiciji virtualne funkcije u izvedenoj klasi.
-  Ako funkcija neće biti redefinisana, efikasnije je bez njenog definisanja kao virtualne funkcije, jer je potrebno više vremena i resursa sistema za dinamičko povezivanje (engl. bind) virtualnih funkcija za vrijeme izvršavanja (engl. at runtime)
-  Virtualna funkcija se može implementirati u nekoliko izvedenih klasa. C ++ dinamički povezuje (engl. bind) implementacije funkcije za vrijeme izvršavanja, što odlučuje stvarna klasa objekta koja je referencirana preko varijable
-  Prevodilac pronalazi odgovarajuću funkciju prema tipu parametra, broju parametara, te redosledu parametara u vrijeme prevođenja (engl. at compile time).

NASLJEĐIVANJE, VIRTUELNE FUNKCIJE I POLIMORFIZAM

25. Analizirati sledeći kod:

```
#include <iostream>
using namespace std;
class A {
public:
A() {
t();
// cout << "i od A je " << i << endl;
}
void t() {
setl(20);
}
virtual void setl(int i) {
this->i = 2 * i;
}
int i;
};
```

```
class B: public A {
public:
B() {
cout << "i od B je " << i << endl;
}
virtual void setl(int i) {
this->i = 3 * i;
}
};
int main() {
A* p = new B();
return 0;
}
```

- **Konstruktor klase A nije pozvan.**
- **Konstruktor klase A se poziva i prikazuje "i od B je 0".**
- **Konstruktor klase A se poziva i prikazuje "i od B je 40".**
- **Konstruktor klase A se poziva i prikazuje "i od B je 60".**

NASLJEĐIVANJE, VIRTUELNE FUNKCIJE I POLIMORFIZAM

25. Analizirati sledeći kod:

```
#include <iostream>
using namespace std;
class A {
public:
A() {
t();
// cout << "i od A je " << i << endl;
}
void t() {
setl(20);
}
virtual void setl(int i) {
this->i = 2 * i;
}
int i;
};
```

```
class B: public A {
public:
B() {
cout << "i od B je " << i << endl;
}
virtual void setl(int i) {
this->i = 3 * i;
}
};
int main() {
A* p = new B();
return 0;
}
```

- **Konstruktor klase A nije pozvan.**
- **Konstruktor klase A se poziva i prikazuje "i od B je 0".**
- **Konstruktor klase A se poziva i prikazuje "i od B je 40".**
-  **Konstruktor klase A se poziva i prikazuje "i od B je 60".**

NASLJEĐIVANJE, VIRTUELNE FUNKCIJE I POLIMORFIZAM

26. Koje je od sledećih apstraktna funkcija?

- **virtual double getArea();**
- **virtual double getArea() = 0;**
- **double getArea() = 0;**
- **double getArea();**



NASLJEĐIVANJE, VIRTUELNE FUNKCIJE I POLIMORFIZAM

26. Koje je od sledećih apstraktna funkcija?

- **virtual double getArea();**
- 😊 **virtual double getArea() = 0;**
- **double getArea() = 0;**
- **double getArea();**



NASLJEĐIVANJE, VIRTUELNE FUNKCIJE I POLIMORFIZAM

27. Koje su od sledećih izjave tačne?

- **Apstraktne klase se definišu korišćenjem ključne reči `abstract`.**
- **Klasa je apstraktna ako sadrži čistu virtuelnu funkciju.**
- **Apstraktna klasa je ista kao kao standardna klasa, osim što ne može kreirati objekte.**
- **Može se definisati apstraktna klasa čak i ako ne sadrži apstaktne funkcije**

NASLJEĐIVANJE, VIRTUELNE FUNKCIJE I POLIMORFIZAM

27. Koje su od sledećih izjave tačne?

- **Apstraktne klase se definišu korišćenjem ključne reči `abstract`.**
-  **Klasa je apstraktna ako sadrži čistu virtuelnu funkciju.**
-  **Apstraktna klasa je ista kao kao standardna klasa, osim što ne može kreirati objekte.**
- **Može se definisati apstraktna klasa čak i ako ne sadrži apstaktne funkcije**

NASLJEĐIVANJE, VIRTUELNE FUNKCIJE I POLIMORFIZAM

28. Članovi klase deklarirani kao _____ dostupni su samo funkcijama članicama klase.

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

NASLJEĐIVANJE, VIRTUELNE FUNKCIJE I POLIMORFIZAM

28. Članovi klase deklarirani kao _____ dostupni su samo funkcijama članicama klase.

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

NASLJEĐIVANJE, VIRTUELNE FUNKCIJE I POLIMORFIZAM

29. Nasleđivanje je vrsta ponovnog korišćenja softvera u kome nove klase obuhvataju svojstva i ponašanje postojećih klasa i obogaćuju ih novim mogućnostima.

- Tačno
- Pogrešno

NASLJEĐIVANJE, VIRTUELNE FUNKCIJE I POLIMORFIZAM

29. Nasleđivanje je vrsta ponovnog korišćenja softvera u kome nove klase obuhvataju svojstva i ponašanje postojećih klasa i obogaćuju ih novim mogućnostima.



Tačno

- **Pogrešno**

NASLJEĐIVANJE, VIRTUELNE FUNKCIJE I POLIMORFIZAM

30. Nasleđivanjem se olakšava ponovno korišćenje softvera koji se dokazao kao kvalitetan.

- Tačno
- Pogrešno

NASLJEĐIVANJE, VIRTUELNE FUNKCIJE I POLIMORFIZAM

30. Nasleđivanjem se olakšava ponovno korišćenje softvera koji se dokazao kao kvalitetan.



Tačno

- **Pogrešno**

NASLJEĐIVANJE, VIRTUELNE FUNKCIJE I POLIMORFIZAM

31. Članovi deklarirani kao _____ u osnovnoj klasi dostupni su u osnovnoj i izvedenim klasama.

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

NASLJEĐIVANJE, VIRTUELNE FUNKCIJE I POLIMORFIZAM

31. Članovi deklarisan kao _____ u osnovnoj klasi dostupni su u osnovnoj i izvedenim klasama.

- public
- 😊 protected
- private

NASLJEĐIVANJE, VIRTUELNE FUNKCIJE I POLIMORFIZAM

32. Klasa Ptica u deklaraciji class Ljubimac : Ptica je izvedena:

- Javno
- Privatno

NASLJEĐIVANJE, VIRTUELNE FUNKCIJE I POLIMORFIZAM

32. Klasa Ptica u deklaraciji class Ljubimac : Ptica je izvedena:

- Javno



- Privatno

NASLJEĐIVANJE, VIRTUELNE FUNKCIJE I POLIMORFIZAM

33. 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**

NASLJEĐIVANJE, VIRTUELNE FUNKCIJE I POLIMORFIZAM

33. 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

NASLJEĐIVANJE, VIRTUELNE FUNKCIJE I POLIMORFIZAM

34. Izvedena klasa ne nasleđuje konstruktore osnovne klase.

- Tačno
- Pogrešno



NASLJEĐIVANJE, VIRTUELNE FUNKCIJE I POLIMORFIZAM

34. Izvedena klasa ne nasleđuje konstruktore osnovne klase.



Tačno

- **Pogrešno**

NASLJEĐIVANJE, VIRTUELNE FUNKCIJE I POLIMORFIZAM

35. Konstruktor izvedene klase je dužan da prosledi sve potrebne parametre konstruktoru osnovne klase.

- Tačno
- Pogrešno

NASLJEĐIVANJE, VIRTUELNE FUNKCIJE I POLIMORFIZAM

35. Konstruktor izvedene klase je dužan da prosledi sve potrebne parametre konstruktoru osnovne klase.



Tačno

- **Pogrešno**

NASLJEĐIVANJE, VIRTUELNE FUNKCIJE I POLIMORFIZAM

36. Kada se uništava objekat izvedene klase, destruktori se pozivaju u obrnutom redosledu od konstruktora.

- Tačno
- Pogrešno

NASLJEĐIVANJE, VIRTUELNE FUNKCIJE I POLIMORFIZAM

36. Kada se uništava objekat izvedene klase, destruktori se pozivaju u obrnutom redosledu od konstruktora.



Tačno

- **Pogrešno**

NASLJEĐIVANJE, VIRTUELNE FUNKCIJE I POLIMORFIZAM

37. Ako klasa sadrži barem jednu čistu virtuelnu metodu, onda je takva klasa apstraktna.

- Tačno
- Pogrešno

NASLJEĐIVANJE, VIRTUELNE FUNKCIJE I POLIMORFIZAM

37. Ako klasa sadrži barem jednu čistu virtuelnu metodu, onda je takva klasa apstraktna.



Tačno

- **Pogrešno**

NASLJEĐIVANJE, VIRTUELNE FUNKCIJE I POLIMORFIZAM

38. Ne mogu da se kreiraju objekti apstraktnih klasa.

- Tačno
- Pogrešno

NASLJEĐIVANJE, VIRTUELNE FUNKCIJE I POLIMORFIZAM

38. Ne mogu da se kreiraju objekti apstraktnih klasa.



Tačno

- **Pogrešno**

NASLJEĐIVANJE, VIRTUELNE FUNKCIJE I POLIMORFIZAM

39. Sve virtuelne funkcije u apstraktnoj osnovnoj klasi moraju se deklarirati kao čiste virtuelne funkcije.

- Tačno
- Pogrešno

NASLJEĐIVANJE, VIRTUELNE FUNKCIJE I POLIMORFIZAM

39. Sve virtuelne funkcije u apstraktnoj osnovnoj klasi moraju se deklarirati kao čiste virtuelne funkcije.

- Tačno

 Pogrešno



KRAJ!

