1. **Objekat i klasa  
   Šta je objekat, a šta klasa? Kreiranje objekta.**

**Oblast važenja referencne promenljive.**

**Koliko objekata je kreirano u datom primeru?**

**Pitanja tipa šta će biti rezultat izvršavanja sledećeg koda**

Tacka t = new Tacka(1, 1);

Tacka t2 = t;

t2.x = 12;

Console.WriteLine(t.x);

1. **Konstruktor  
   Šta je konstruktor?**

**Šta je default-ni konstuktor?**

**Da li svaka klasa mora da ima konstruktor?**

**Kada ne postoji default-ni kontruktor?**

1. **Overloading  
   Šta je i po čemu se moraju razlikovati overloadovani metodi?**

**Da li overloadovani metodi mogu vraćati razlicite tipove?**

1. **This – šta predstavlja i ko ga poseduje?**

**Prepoznati kada je neohodno koristiti ga.**

1. **Članovi objekata i Članovi klasa (nestatički podaci i metodi pripadaju objektima, a statički klasi)**

**Kako se koriste statički podaci i metodi?**

Class Tacka{

public int x;

public static int statikx;

}

Tacka t = new Tacka();

t.x=4;

Tacka.statikx = 4;

**Definisanje ID-a objekta kao int podatka čija je vrednost jednaka rednom broju kreiranog objekta.**

**Koja je razlika između nestatičkih i statičkih metoda? Šta statički/nestatički metod može da koristi od podataka i metoda definisanih u klasi?**

1. **Nasleđivanje** 
   1. **Kako se izvodi u kodu?**
   2. **Da li svaka klasa u C# ima roditeljsku klasu?**

**Nema klasa Object.**

* 1. **Šta se nasleđuje sa aspekta vidiljivosti članova?**
  2. **Kastovanje i konverzija?**

**Na datom primeru prokomentarisati da li kod prolazi prevođenje i/ili da li proizvodi run-time grešku. Pitanja tipa koliko je objekata kreirano?**

static void Main(string[] args)

{

A a = new A(); // Gradjanin

B b = new B(); // Ucenik

A c = b; // prolazi i kompajliranje i izvrsenje

B d = (B)a; // prolazi kompajl, ali uzrokuje gresku u run-time-u

B e = (B)c; // prolazi i kompajliranje i izvrsenje

a.metod();

b.metod();

c.metod(); Console.ReadKey();

}

}

class A { public void metod() {Console.WriteLine("A metod");}}

class B : A { public void metod() { Console.WriteLine("B metod"); } }

* 1. **Da li je moguće da se referencom jednog tipa ukazuje na objekat drugog tipa?**
  2. **Da li je moguće imati u dete klasi metod sa istim potpisom koji je nasleđen iz roditeljske klase? Isto pitanje i za podatke? Koji je efekat upotrebe virtual/override?**

static void Main(string[] args)

{

A a = new A(); // Gradjanin

B b = new B(); // Ucenik

A c = b; // prolazi i kompajliranje i izvrsenje

//B d = (B)a; // prolazi kompajl, ali uzrokuje gresku u run-time-u

B e = (B)c; // prolazi i kompajliranje i izvrsenje

a.metod(); b.metod(); c.metod();

a.metod2(); b.metod2(); c.metod2();

Console.ReadKey();

}

}

class A { public void metod() {Console.WriteLine("A metod");}

virtual public void metod2() { Console.WriteLine("A virtual metod"); } }

class B : A { public void metod() { Console.WriteLine("B metod"); }

override public void metod2() { Console.WriteLine("B override metod"); }

}

* 1. **Konstrukcija objekata?**

static void Main(string[] args)

{

Gradjanin a = new Gradjanin("pera peric"); // Gradjanin

Ucenik b = new Ucenik("milos pantic", "PKG"); // Ucenik

Console.ReadKey();

}

}

class Gradjanin {

public string ime;

public Gradjanin(string a) { ime = a; }

}

class Ucenik : Gradjanin {

public string skola;

public Ucenik(string ime, string skola) : base(ime) { this.skola = skola; }

}

* 1. **base – Šta predstavlja poziv base(), base.metod(), base.podatakX?**

static void Main(string[] args)

{

A a = new A(); // Gradjanin

B b = new B(); // Ucenik

}

}

class A { public void metod() {Console.WriteLine("A metod");}

virtual public void metod2() { Console.WriteLine("A virtual metod");}}

class B : A { public void metod() {

Console.WriteLine("B metod");

base.metod();

}

override public void metod2() { Console.WriteLine("B override metod");} }

1. **Modifikatori vidljivosti**

**Koji su i kakav efekat imaju na upotrebu elemenata objekata (atributa i metoda)?**

1. **Properties – kako se definišu i koriste?**