Programski jezici

Primeri kodova za test i kolokvijume

<!-- Ispravke, sugestije, mišljenja i ostalo šaljite na download@puskice.org -->

2013.

Hijavata

Predgovor

U ovoj skripti su skupljeni primeri kodova koji kruže okolo, kao i primeri testova iz 2012. i 2013. godine. Kao što ćete videti, pitanja se često ponavljaju, ali to može biti najveća varka ovog ispita, jer na testu može doći "neznatno" izmenjeno pitanje koje je jako slično nekom od pitanja u ovoj skripti, ali dovoljno drugačije da je odgovor drugi. Zato nemojte učiti ova pitanja napamet, već prvo prođite skripte za prvi i drugi kolovkijum da biste skapirali o čemu se radi ovde, a zatim prođite ove kodove.

Ograničavam se na to da je raspodela gradiva na prvi i drugi kolokvijum koje sam se držao, ona koja je važila u školskoj 2012/2013 godini, i da se može menjati po nahođenju asistenata.

Zadatak je napisan u obliku:

```
Redni broj pitanja u ovoj skripti
```

Pitanje (zahtev)

Kod

Rešenje

Objašnjenje

Prvi kolokvijum

1 Šta će se ispisati?

```
class Student
    {
        public int ocena;
    }
    static void Main(string[] args)
    {
       Student s = new Student();
       int ocena = s.ocena;
       Console.WriteLine(ocena);
    }
```

0

Ispisaće se 0. To je zato što prvo formiramo objekat student. Pravljenjem objekta, svakom polju se dodeljuje default vrednost za taj tip promenljive (osim ako u konstruktoru nije drugačije navedeno!). Za int, default vrednost je 0. Zatim se novoj promenljivoj ocena (lokalna promenljiva u Main metodi), prenosi sadržaj promenljive ocena iz objekta s, i to preko vrednosti (ne preko reference)

2 Šta će se ispisati?

```
class Program
    {
        static void Metoda(int x)
        {
            x--;
        }
        static void Main(string[] args)
        {
            int broj = 1;
            Metoda(broj);
            Console.WriteLine(broj);
        }
    }
```

1

Ispisaće se 1. To je zato što se u metodi Metoda(int x) vrši prenos preko vrednosti. To znači da se pir izvršavanju ove metode kreira nova promenljiva na novoj lokaciji u memoriji i u nju se kopira vrednost int-a broj. Pri tom, vrednost originalne promenljive broj ostaje ista.

3 Šta će se ispisati?

```
class Program
    {
        public void Izmeni(int a)
        {
            a = a + 4;
        }
        public void Izmeni1(int a)
        {
            a = a - 7;
        }

        static void Main(string[] args)
        {
            Program pro = new Program();
            int ulaz = 20;
            pro.Izmeni(ulaz);
            pro.Izmeni1(ulaz);
            Console.WriteLine(ulaz);
        }
    }
```

20

Ispisaće se 20. Razlog je taj što obe metode rade sa kopijom promenljive ulaz, tako da ona zapravo ostaje netaknuta.

4 Šta će se ispisati?

```
class Program
    {
        public void Izmeni(int a)
        {
            a = a + 4;
        }
        public void Izmeni1(ref int a)
        {
            a = a - 7;
        }
        static void Main(string[] args)
        {
            Program pro = new Program();
            int ulaz = 20;
            pro.Izmeni(ulaz);
            pro.Izmeni1(ref ulaz);
            Console.WriteLine(ulaz);
        }
    }
```

13

Ispisaće se 13. Metoda Izmeni radi sa kopijom promenljive ulaz tako da ne utiče direktno na nju. S druge strane, u metodi Izmeni1 vrši se prenos promenljive preko reference tako da će ona moći da izmeni promenljivu.

5 Šta će se ispisati?

```
class Student
    {
        public int brojIndeksa = 112;
    }

    ----
static void Main(string[] args)
{
     Student s = new Student ();
     Student s1 = s;
     s.brojIndeksa = 100;
     Console.WriteLine(s1.brojIndeksa);
}
```

100

Ispisaće se 100. Pošto je Student referentni tip, nakon pravljenja studenta s1, on će pokazivati na isto mesto u heap-u gde pokazuje i s, tj. na isti objekat Student. Zato će svaka izmena, bilo s, bilo s1 uticati na taj objekat.

6 Šta će se ispisati?

```
class Broj
    { public int vred; }
  ----
static void Main(string[] args)
{
     Broj br1 = new Broj();
     Broj br2 = br1;
     br1.vred = 100;
     br2.vred = 50;
     Console.WriteLine(br1.vred);
}
```

50

Ispisaće se 50. Pošto je Broj referentni tip, br2 će pokazivati na isto mesto u heap-u gde pokazuje i br1, tj. na isti objekat Broj u heap-u. To je zato što se kreiranjem objekta b2 u njega upisuje objekta b1 na steku, a to je adresa na objekat u heap-u. Zato će svaka izmena, bilo br1, bilo br2 uticati na taj objekat.

7 Koji kod je ispravan:

```
a)
int a = 33;
const int broj = a;
b)
const int broj = 33;
int b = broj;
```

a)

Ispravan je deo koda pod b) zato što se promenljivoj može dodeliti vrednost konstante a obrnuto ne može. Konstanti se može dodeliti samo vrednost druge konstante

8

Šta će se ispisati:

```
class Krug
    {
        public int precnik;
    }
    class Program
    {

        static void Main(string[] args)
        {
            Krug k;
            k.precnik = 50;
            Console.WriteLine(k.precnik);
        }
    }
```

Greška

Ispisaće se greška zato što ne postoji instanca objekta Krug. Sve što postoji je rezervisana adresa u heapu za pokazivač na Krug k (kad on bude inicijalizovan). Da bi se ispisalo 50, moralo bi da stoji:

```
Krug k = new Krug();
```

9

Šta će se ispisati:

```
class Broj
    {
        public int vred;
    }

    class Program
    {

        static void Main(string[] args)
        {
            Broj br;
            br.vred = 20;
            int prom = br.vred + 1;
            Console.WriteLine(prom);
        }
    }
```

Greška

Ispisaće se greška zato što ne postoji instanca objekta Broj. Sve što postoji je rezervisana adresa u heapu za pokazivač na Broj br (kad on bude inicijalizovan).

10 Šta će se ispisati:

```
class Broj
    {
        public int vred;
    }

    class Program
    {

        static void Main(string[] args)
        {
            Broj br = new Broj();
            br.vred = 20;
            int prom = br.vred + 1;
            Console.WriteLine(prom);
        }
    }
```

21

--

11 Šta će se ispisati:

```
string kod = "PMF";
            switch (kod)
            {
                default: Console.WriteLine("FON");
                    break;
                case "ETF":
                case "PMF":
                    Console.WriteLine("PMF");
                    break;

            }
```

PMF

Imamo switch naredbu koja skače na onaj kejs koji je jednak parametru prosleđenom switch naredbi (kod). Dakle, izvršava se komanda posle case-a “PMF” jer je to vrednost promenljive kod.

12 Šta će se ispisati:

```
string kod = "ETF";
            switch (kod)
            {
                default: Console.WriteLine("FON");
                    break;
                case "ETF":
                case "PMF":
                    Console.WriteLine("PMF");
                    break;

            }
```

PMF

Imamo switch naredbu koja skače na onaj kejs koji je jednak parametru prosleđenom switch naredbi (kod). Dakle, skačemo na izvršavanje naredbe case “ETF“. Pošto tu nije navedena nijedna naredba, nastavlja se dalje, pa se izvrši naredba za sledeći case.

13

Šta će se ispisati:

```
string kod = "FPN";
            switch (kod)
            {
                default: Console.WriteLine("FON");
                    break;
                case "ETF":
                case "PMF":
                    Console.WriteLine("PMF");
                    break;

            }
```

FON

Imamo switch naredbu koja skače na onaj kejs koji je jednak parametru prosleđenom switch naredbi (kod). Pošto nijedan case nije jednak sadržaju promenljive kod, onda se ide na default vrednost i ispisuje se FON.

14

Šta će se ispisati:

```
for (int i = 0; i < 4; i++)
            {
                int a = i;
                while (a < i + 2)
                {
                    a++;
                }
                Console.Write(a);
            }
```

2345

U prvoj iteraciji for petlje, u promenljivu i se upiše vrednost 0. Zatim promenljiva a dobije istu tu vrednost. Zatim se ulazi u while petlju gde se a inkrementuje sve dok je manja od i+2:
I: 0<2 -> a=1
II: 1<2 -> a=2
III 2 nije manje od 2 -> iskačemo iz while petlje.
Zatim se ispisuje vrednost promenljive a.
Ceo proces se ponavlja i za i=1,2,3.

15 Koliko puta će se izvršiti sledeća petlja:

```
int a = 1;
            do a = a * 2;
            while (a < 15);
```

4

Izvršiće se četiri puta zato što se pre while-a uvek izvrši do. Promenljiva a će redom uzimati sledeće vrednosti: 2,4,8,16

16 Koliko puta će se izvršiti sledeća petlja i šta će ispisati:

```
int a = 0;
            while (a < 10)
            {   a++;
                Console.WriteLine(a);
                if (a == 5)
                    break;
                a++;  }
            Console.WriteLine(a);
```

---

4 puta,

ispis:

1 3 5 5

17 Šta se ispisuje:

```
string str1 = "AAA";
            string str2 = str1;
            string str3 = str2;
            str2 = "CCC";
            Console.WriteLine(str3);
            Console.WriteLine(str2);
```

AAA

CCC

Iako je string referentni tip podataka, za njega se uvek pravi nova kopija u memoriji jer je niz zapravo implementiran kao niz karaktera (dakle imaćemo tri memorijska dela sa istim podacima). Zatim se str2 promeni u CCC.

18 Šta se ispisuje?

```
string str1 = "ABC";
string str2 = str1.Substring(2, 2);
Console.WriteLine(str2);
```

Greška

Ovo će proći buildovanje ali će pri izvršavanju pojaviti greška. To je zato što metoda Substring(int a, int b) prima dva parametra: prvi je indeks karaktera u stringu od kog će početi odsecanje a drugi je dužina odsečka.
U ovom primeru, taj karakter je C (jer indeksiranje počinje od 0 jer je string zapravo niz) a pošto odsečak treba da bude dužine 2 izbaciće grešku jer ne može da odseče nepostojeće karaktere.

19 Šta se ispisuje?

```
string str1 = "ABCDEF";
string str2 = str1.Substring(2, 2);
Console.WriteLine(str2);
```

```
----
```

CD

20 Šta se ispisuje?

```
string str1 = "ABCDEF";
string str2 = str1.Insert(3, "mm");
Console.WriteLine(str2);
```

ABCmmDEF

Prvi parametar je indeks karaktera na koji se umeće drugi parametar, a drugi parametar je string koji umećemo.

21 Šta se ispisuje?

```
string a = "AB";
            string b = "AB";

            if (a == b)
            {
                Console.WriteLine("Isti");
            }
            else
            {
                Console.WriteLine("Razliciti");
            }
```

Isti

U C#, == radi isto što i Equals,tako da im proverava sadržaj a ne adrese.

22 Šta se ispisuje?

```
            string a = "Beograd";
            string b = a;
            a = "Novi Sad";
            Console.WriteLine(b[3]);
```

g

Pošto je string referentni tip (niz) uvek se pravi nova kopija pri dodeli. Posle izvršenja druge linije koda, u memoriji će postojati dva niza Beograd - na jedan pokazuje string a, a na drugi b. Zatim će se string a (tačnije niz na koji pokazuje a) „izbrisati“ iz memorije i napraviće se novi niz Novi Sad, na koji će string a da pokazuje. String b i dalje pokazuje na „svoj“ Beograd.

23 Da li je sledeći kod ispravan?

```
public struct Index
    {
        public int broj;
        public int godina;
        public void Prikaz()
        {
            Console.WriteLine("{0}/{1}", broj, godina);
        }
    }
```

Da

Jeste, proći će buildovanje.

24 Vrednost konstante Zeleno je?

```
enum Semafor
    {
        Crveno,
        Zuto,
        Zeleno
    }
```

2

Iza svake konstante u enum-u se nalazi broj. Ako nije drugačije navedeno, brojanje kreće od 0.

25 Šta će se ispisati?

```
enum Pol
    {
        zenski,
        muski = 5

    }

static void Main(string[] args)
        {
            Pol p = Pol.muski;
            Console.WriteLine(p - 5);

        }
```

zenski

Konstanti zenski dodeljena je pozadinska vrednost 0 (automatski) dok je konstanti muski eksplicitno dodeljena vrednost 5.

26 Vrednost konstante Zeleno je?

```
enum Pol
    {
        zenski,
        muski = 5
    }

static void Main(string[] args)
        {
            Pol p = Pol.muski;
            Console.WriteLine(p - 3);
        }
```

2

Konstanti zenski dodeljena je pozadinska vrednost 0 (automatski) dok je konstanti muski eksplicitno dodeljena vrednost 5. Pokušaj da se ispiše konstanta sa brojem 2 daće rezultat 2 jer ne postoji takva konstanta (neće izbaciti grešku).

27 Vrednost konstante za Zeleno je?

```
enum Semafor
    {   Crveno,
        Zuto = 2,
        Zeleno    }
```

3

Vrednost konstante kod enuma je uvek za 1 veća od vrednosti za prethodnu promenljivu

28 Koje metode mogu stajati u klasi uz sledeću metodu?

```
public int Metoda(int a, string b) {return 0; }

Ponuđeno:
a) public int Metoda(int c, string e) { return 0; }
b) public bool Metoda(int a, string b){return false;}
c) private int Metoda(int a, string b) { return 0; }
d) public int Metoda(string a, int b) { return 0; }
e) public int Metoda(int a, int b) { return 0; }
f) public int Metoda(int a) { return 0; }
g) public int Metoda(ref int a, string b) { return 0; }
h) public int Metoda(out int a, string b) { return 0; }
```

d,e,f,g,h

Dve metode ne smeju imati isti potpis (broj, tip i redosled parametara), iako im je povratna vrednost drugačija. Ako je parametar prenošen preko reference (ref ili out) to je različito u odnosu na prenos preko reference. Bitan je tip parametra koji se prenosi, a ne naziv parametra.

29 Koje metode mogu stajati u klasi uz sledeću metodu?

```
public int Metoda(ref int a, string b) {return 0; }
```

Ponuđeno:

```
a) public int Metoda(int c, string e) { return 0; }
b) public bool Metoda(int a, string b){return false;}
c) private int Metoda(int a, string b) { return 0; }
d) public int Metoda(string a, int b) { return 0; }
e) public int Metoda(int a, int b) { return 0; }
f) public int Metoda(int a) { return 0; }
g) public int Metoda(out int a, string b) { return 0; }
```

a,b,c,d,e,f

Mogu sve osim poslednje, jer ona ima isti potpis (out i ref se posmatraju isto, kao parametar prenešen preko reference)

30 Koje metode ne mogu stajati u klasi uz sledeću metodu?

```
public double Metoda(double d, bool b) { return 3; }
```

Ponuđeno:

```
a) public double Metoda(bool b, double d) { return 3; }
b) public double Metoda(ref double d, bool b) { return 3; }
c) private double Metoda(double d, bool b) { return 3; }
d) protected double Metoda(double d, bool b) { return 3; }
e) public double Metoda(double d,double b) { return 3; }
f) public long Metoda(double d, bool b) { return 3; }
g) public double Metoda(double d) { return 3; }
h) public double Metoda1(double d, bool b) { return 3; }
```

c,d,f

---

31 Koje metode imaju isti potpis?

```
a) void Metoda(double a) { }
b) void Metoda(double b) { }
c) int Metoda(double a) { return 3; }
d) void Metoda(out double a) { }
e) void Metoda(double a, int b) { }
```

a,b,c

Potpis metode je određen imenom, brojem, tipom i redosledom parametara.

32

Šta se ispisuje?

```
public void Izmeni(int a)
        {   a = a + 10;
        }
        public void Izmeni(ref int a)
        {   a = a - 10;
        }
        static void Main(string[] args)
        {
            Program pro = new Program();
            int ulaz = 20;
            pro.Izmeni(ulaz);
            pro.Izmeni(ref ulaz);
            Console.WriteLine(ulaz);
        }
```

10

U prvoj metoda vrši se prenos preko vrednosti. U memoriji će da se napravi novi int kom se doeli vrednost ulaza (20) i zatim se nad njim izvrši Izmeni(int) koji mu poveća vrednost na 30. Pošto se taj int više nigde ne koristi on nestaje. Za sve to vreme, ulaz ostaje ne promenjen. Druga metoda prosleđuje ulaz preko reference, što znači da će se operacija u metodi obaviti nad tim brojem.

33

Šta se ispisuje?

```
public void Izmeni(int a)
        {   a = a + 10;
        }
        public void Izmeni(out int a)
        {   a = 15;
        }
        static void Main(string[] args)
        {
            Program pro = new Program();
            int ulaz = 20;
            pro.Izmeni(ulaz);
            pro.Izmeni(out ulaz);
            Console.WriteLine(ulaz);
        }
```

15

U prvoj metoda vrši se prenos preko vrednosti. U memoriji će da se napravi novi int kom se doeli vrednost ulaza (20) i zatim se nad njim izvrši Izmeni(int) koji mu poveća vrednost na 30. Pošto se taj int više nigde ne koristi on nestaje. Za sve to vreme, ulaz ostaje ne promenjen. Druga metoda prosleđuje ulaz preko reference, i to tako što izbriše vrednost ulaza, i onda joj dodeli novu vrednost (15).

34 Šta se ispisuje?

```
static void Metoda(out int a)
      {
          a--;
      }
      static void Main(string[] args)
      {
          Program pro = new Program();
          int broj = 1;
          Metoda(out broj);
          Console.WriteLine(broj);
      }
```

Greška

Kada prenosimo promenljivu preko reference out, onda se vednost te promenljive izbriše, i mi moramo negde unutar metode dodeliti vrednost toj promenljivoj. Dakle trebalo bi da stoji negde u metodi a = 10; ili nešto slično.

35 Šta se ispisuje?

```
class Proizvod
    {
        public double cena = 100;
    }
    class Program
    {
        static void Povecaj(out Proizvod p, double iznos)
        {
            p.cena += iznos;
        }
        static void Main(string[] args)
        {
            Proizvod p = new Proizvod();
            Povecaj(out p, 50);
            Console.WriteLine(p.cena);
        }
    }
```

Greška

Kada prenosimo promenljivu preko reference out, onda se vednost te promenljive izbriše, i mi moramo negde unutar metode dodeliti vrednost toj promenljivoj.

36 Šta se ispisuje?

```
class Proizvod
    {
        public double cena = 100;
    }
    class Program
    {
        static void Povecaj(out Proizvod p, double iznos)
        {
            p = new Proizvod();
            p.cena += iznos;
        }
        static void Main(string[] args)
        {
            Proizvod p = new Proizvod();
            Povecaj(out p, 50);
            Console.WriteLine(p.cena);
        }
    }
```

150

Kada prenosimo promenljivu preko reference out, onda se vednost te promenljive izbriše. Zatim je u metodi toj promenljivoj dodata nova vrednost (tačnije, napravljen je novi objekat). Zatim mu je dodata vrednost na cenu.

37 Šta se ispisuje?

```
static void A(int x)
        {
            x += 20;
        }
        static void B(ref int x)
        {
            x -= 10;
        }
        static void Main(string[] args)
        {
            int broj = 30;
            A(broj);
            B(ref broj);
            Console.WriteLine(broj);
        }
```

20

U prvoj metoda vrši se prenos preko vrednosti. U memoriji će da se napravi novi int kom se doeli vrednost ulaza (30) i zatim se nad njim izvrši A(int). Pošto se taj int više nigde ne koristi on nestaje. Za sve to vreme, broj ostaje ne promenjen. Druga metoda prosleđuje ulaz preko reference, tako da će se nad tom promenljivom koja se prosleđuje izvršiti operacija.

38 Šta se ispisuje?

```
class Automobil
{
    public int regBroj;
    public Automobil(int regBroj)
    {
        regBroj = regBroj;
    }
}
class Program
{
    static void Main(string[] args)
    {
        Automobil a = new Automobil(220);
        Console.WriteLine(a.regBroj);
    }
}
```

0

U konstrukotru Automobil-a, vrednost ulaznog parametra se dodeljuje sama sebi (što je besmisleno). Zato će taj konstruktor zapravo dodeljivati atributu regBroj instance klase Automobil vrednost 0 (default). Ako ovaj kod iskucate u VS, dobićete upozorenje (ne grešku) da to što radite nema smisla.

39 Šta se ispisuje?

```
class Automobil
{
    public int regBroj;
    public Automobil(int regBroj)
    {
        this.regBroj= regBroj;
    }
}
class Program
{

    static void Main(string[] args)
    {
        Automobil a = new Automobil(220);
        Console.WriteLine(a.regBroj);
    }
}
```

220

---

40 Šta se ispisuje?

```
class Trougao
    {
        public int visina;
        public Trougao(int visina)
        {
            visina = 100;
        }
    }
    class Program
    {
        static void Main(string[] args)
        {
            Trougao t = new Trougao(50);
            Console.WriteLine(t.visina);
        }
    }
```

0

U okviru konstruktora, ulaznom parametru dodeljujemo vrednost 100, a ne atributu objekta. Da bismo pristupili atributu objekta, koristimo ključnu reč this. Ovaj konstrukor visini dodeljuje default vrednost (0 jer je int).

41 Šta se ispisuje na ekranu?

```
public class TAdresa
    {
        public int broj;
    }
    public class Primer
    {
        static void Proba(TAdresa a)
        {   a.broj += 10;   }
        static void Proba(ref TAdresa a)
        {   a.broj -= 5;    }
        public static void Main(string [] args)
        {
            TAdresa adresa = new TAdresa();
            Proba(adresa);
            Proba(ref adresa);
            Console.WriteLine(adresa.broj);
        }
    }
```

a) 5

a) 5
b) 10
c) -5
d) 0
e) Greška

Prilikom krerianja objekta adresa, default konstruktor ce atributu broj dodeliti default vrednost 0 (jer je u pitanju int). Zatim se objekat prosleđuje preko vrednosti. To znači da se pravi nova "adresa" objekta koja pokazuje na atribut `broj`, objekta koji je prosleđen. Zato će se nad tim atributom izvršiti dodavanje 10 atributu `broj`. Nakon toga, stari objekat se prenosi preko reference, što znači da se sve izmene vrše direktno nad njim. Izvršiće se odgovarajuća operacija (oduzimanje broja 5 od 10) pa dobijamo da je vrednost atributa broj jednaka 5.

42

Šta se ispisuje na ekranu?

```
public class Primer
    {
        static void Proba(int a)
        {
            a += 10;
        }
        static void Proba(ref int a)
        {
            a -= 5;
        }
        public static void Main(string [] args)
        {
            int b = 0;
            Proba(b);
            Proba(ref b);
            Console.WriteLine(b);
        }
    }
```

a) 5
b) 10
c) -5
d) 0
e) Greška

c) -5

Ovo je jako sličan primer gornjem. Ovaj put, vrši se prenos vrednosnog, a ne referentnog tipa. Prva metoda, `Proba(b)`, pravi lokalnu kopiju tog broja (novi int sa vrednošću 0). Zatim toj novoj vrednosti dodaje 10. Nakon izvršenja ovoga, pošto se promena nigde ne pamti, taj novi int nestaje. Ostaje onaj stari, neizmenjen int b koji ima vrednost 0. Zatim se u sledećoj metodi on prosleđuje preko reference, što znači da se izmene vrše direktno nad njim. Rezultat je 0 - 5 = -5.

43 Šta se ispisuje na ekranu?

```
public class TAdresa
    {
        public int broj;
    }
    public class Primer
    {
        static void Proba(TAdresa a)
        {  a.broj += 10;  }
        static void Proba(out TAdresa a)
        {  a.broj -= 5;   }
        public static void Main(string [] args)
        {
            TAdresa adresa = new TAdresa();
            Proba(adresa);
            Proba(out adresa);
            Console.WriteLine(adresa.broj);
        }
    }
```

a) 5 b) 10
c) -5 d) 0 e) Greška

d) Greška

U ovom slučaju prijaviće se greška u metodi `Proba(out TAdresa a)`. Da bi se ovo buildovalo, prosleđeni objekat mora biti inicijalizovan, jer pri prosleđivanju preko reference out, briše se njegov sadrzaj.

44 Šta se ispisuje na ekranu?

```
public class TAdresa
    {
        public string adresa;
        public int broj;
    }
    public class Primer
    {
        public static void Main(string [] args)
        {
            TAdresa a1 = new TAdresa();
            TAdresa a2 = new TAdresa();
            a1 = a2;
            a1.adresa = "Pere Perica";
            a1.broj = 1;
            a2.adresa = "Mike Mikica";
            a2.broj = 2;
            if (a1 == a2)
            {  Console.Write("Isti");   }
            else
            {  Console.Write("Razliciti");   }
        }
    }
```

Isti

Komandom `a1 = a2`, oba pokazivača se će pokazivati na isti skup atributa u heap memoriji.

45

Šta se ispisuje?

```
public class Usluga
    {
        public int cena = 200;
    }

    public class Primer
    {
        static void Uplati(out Usluga u, int iznos)
        {
            u.cena -= iznos;
        }
        public static void Main(string[] args)
        {
            Usluga u = new Usluga();
            Uplati(out u, 50);
            Console.Write(u.cena);
        }
    }
```

Greška

Prijavljuje grešku pri deklarisanju metode Uplati jer traži da parametar ima vrednost pre nego što se prosledi.

46

Šta se ispisuje?

```
public class Broj
    {
        public int vred;
    }
    public class Primer
    {
        public static void Main(string[] args)
        {
            Broj br1 = new Broj();
            Broj br2 = br1;
            br1.vred = 50;
            br2.vred = 100;
            Console.WriteLine(br1.vred);
        }
    }
```

100

Oba pokazivača pokazuju na isti skup promenljivih (u ovom slučaju, to je samo jedan int).

Drugi kolokvijum

47 Da li je kod ispravan?

```
class A
    {
        int broj;
    }
class B : A
    {
        public void Anuliraj()
        {
            broj = 0;
        }
    }
```

Ne

Prijaviće se greška zato što je podrazumevani (default) modifikator pristupa za polja u klasi private. Zbog toga, ovo polje neće biti vidljivo u izvedenim klasama. Ukoliko bi modifikator pristupa bio internal, onda ne bi bilo greške.

48 Šta će se ispisati?

```
class A
    {
        public void Ispisi()
        {
            Console.WriteLine("X");
        }
    }
class B : A { }
      ----------------------------------------
static void Main(string[] args)
{
     B b = new B();
     b.Ispisi();
}
```

50

Ispisaće se X. Nasleđena klasa, nasleđuje polja i metode bazne klase. Pošto je modifikator metode public, vidljiva je u izvedenoj klasi

49 Šta će se ispisati?

```
double d = 3.7;
int i = (int)d;
Console.WriteLine(i);
```

3

Ispisaće se 3 jer se eksplicitnim cast-ovanjem odseca decimalni deo double broja.

50 Šta će se ispisati?

```
double d = 3.7;
int i = Convert.ToInt32(d);
Console.WriteLine(i);
```

4

Ispisaće se 3 jer metoda Convert.ToInt32 vrši zaokrugljivanje po matematičkom pravilu.

51

Koje metode se mogu pridružiti delegatu?

```
public delegate void Delegat (Broj a, int b);

        public void A(Broj c, int d){}
        public static int B (Broj c, int d){return 0;}
        public string C(Broj a, int b){return "";}
        public int D(int a, Broj b){return 0;}
```

A

Da bi se metoda pridružila delegatu, mora imati isti potpis (broj, tip i redosled parametara) i istu povratdnu vrednost.

52

Šta se ispisuje?

```
class A
    {
        public A(): this("b")
        { Console.WriteLine("c");}

        public A(string i)
        { Console.WriteLine(i); }
    }

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

  ----------
        static void Main(string[] args)
        {
            B b = new B();
        }
```

bca

Pozivom konstruktora potklase, prvo se izvšava odgovarajući (isti tip i redosled parametara) konstruktor nadklase. U ovom slučaju, poziva se konstruktor B(). Prvo se izvršava konstruktor A(). Međutim, on poziva drugi konstruktor te klase A(string i). Zato se prvo izvršava konstruktor A(string i) sa parametrom “b”, zatim konstruktor A() - ispisuje c, i konačno B() koji ispiše a.
Isto bi bilo da je pisalo `A b = new B();`

53 Šta se ispisuje?

```
class A
    {
        public A(): this("b")
        { Console.WriteLine("c");}

        public A(string i)
        { Console.WriteLine(i); }
    }
class B : A
    {
        public B()
        {  Console.WriteLine("a"); }
    }
  ----------
        static void Main(string[] args)
        {  A b = new A(); }
```

bc

---

54 Šta se ispisuje?

```
class A
    {
        public A(): this("b")
        { Console.WriteLine("c");}

        public A(string i)
        { Console.WriteLine(i); }
    }
class B : A
    {
        public B()
        {  Console.WriteLine("a"); }
    }
  ----------
        static void Main(string[] args)
        {  A b = new B("M");       }
```

Greška

Ovo izbacuje grešku, jer nemamo odgovarajući konstruktor.

55 Šta se ispisuje?

```
class A
    {
        public A()
        { Console.WriteLine("1");}
    }

    class B : A
    {
        public B():this(1)
        {  Console.WriteLine("2"); }
        public B(int i)
        {  Console.WriteLine("3"); }
    }
-----
        static void Main(string[] args)
        {  B b = new B();           }
```

132

Smisao ovoga je samo da se prepozna kojim redom se pozivaju konstruktori. Dakle pravimo objekat B(), međutim ne poziva se taj konstruktor prvi, već odgovarajući (isti parametri) konstruktor iz nadklase A(). Zatim se poziva konstruktor B(int i) jer je naznačeno da njega naseđuje konstruktor B(). Tek nakon njih, poziva se konstruktor B().

56

Šta se ispisuje?

```
class A
    {
        public A(int broj)
        { Console.WriteLine("A");}
    }
class B : A
    {
        public B()
        { Console.WriteLine("B"); }
    }
class Program
    {
        static void Main(string[] args)
        {
            B b = new B();
        }
```

Greška

Prijaviće grešku pri definiciji konstruktora B() u klasi B. To je zato što ne postoji odgovarajući konstruktor nadklase, jer se pravljenjem bilo kog konstruktora koji je bez parametara, više ne podrazumeva da postoji podrazumevani konstrukotr (bez parametara).

57

Da li je sledeći kod ispravan?

```
class A
    {
        public A() : base() { }
    }
```

Da

Kod je ispravan. Svaka klasa, u nedostatku drugih konstruktora ima podrazumevani (default) konstruktor koji nema parametre i dodeljuje svim atributima default vrednosti (npr. 0 za int). Pri pravljenju novih konstruktora mogu se pozivati drugi konstruktori (radi veće čitljivosti koda). Zato je ovaj kod ispravan (mada potpuno besmislen ☺)

58 Šta se ispisuje?

```
class A
    {
        protected void Ispis()
        {
            Console.WriteLine("Tekst");
        }
    }
class B : A
    {
    }
-----
class Program
    {
        static void Main(string[] args)
        {
            B b = new B();
            b.Ispis();
        }
    }
```

Greška

Ovo je greška. Metoda Ispis() ne postoji u klasi B. Nemojte se zbuniti time što je protected: To znači da je ona vidljiva iz klase B, ali ne i iz klase Program.

59 Šta se ispisuje?

```
    class A
    {
        protected void Ispis()
        {  Console.WriteLine("Tekst");  }
    }
class B:A
    {
        static void Main(string[] args)
        {
            B b = new B();
            b.Ispis();
        }
    }
```

Tekst

Ispisaće se Tekst, tj. pozvaće se metoda Ispis().

60 Da li je kod ispravan?

```
    class A
    {
        public static int broj;
    }
---------
        static void Main(string[] args)
        {
            A a = new A();
            a.broj = 5;
        }
```

Ne

Nije zato što se statičkom polju pristupa preko imena klase, a ne istance. Treba `A.broj = 5;`

61

Da li je kod ispravan?

```
class Matematika
    {
        public static int Kvadriraj(int broj)
        {  return broj * broj;   }
    }
class Program
    {
        static void Main(string[] args)
        {
            Matematika m = new Matematika();
            int a = m.Kvadriraj(5);
            Console.WriteLine(a);
        }
    }
```

Ne

Nije ispravan zato što se statičkom polju pristupa preko imena klase, a ne istance. Treba

```
int a = Matematika.Kvadriraj(5);
```

62

Zaokruži tačna tvrđenja

a) Statički članovi se mogu pozivati i preko klase i preko instance
b) Statički članovi se mogu pozivati iako prethodno nije kreirana instanca date klase
c) Kada se statičkom polju jednom doeli vrednost, ta vrednost se ne može više menjati
d) Statička polja se čuvaju u memoriji samo jednom bez obzira na to koliko ima instanci date klase.

b,d

63

Šta se ispisuje?

```
class Broj
{
   const string id = "123";

   static void Main(string[] args)
    {
      string code = "321";
      string id = string.Format("{0},{1}", id[1], code[1]);
      Console.WriteLine(id);
    }
}
```

Greška

Prijavljuje grešku zato što postoje dva stringa sa istim imenom. Kad bi stajalo npr:

```
string a = string.Format("{0},{1}", id[1], code[1]);
Console.WriteLine(a);
```

Ispis bi bio: **2,2**

64

Šta je ispravno?

a)

```
public class A { }
    public class B : A { }
--------------
A a = new B();
```

b)

```
public class A { }
    public class B : A { }
--------------
B b = new A();
```

a)

Klasa B je izvedena iz klase A, i samim tim ona istovremeno jeste tipa A. Zato će prvi kod moći da se iskompajlira

65

Šta se ispisuje?

```
static void Main(string[] args)
        {
            try
            {   string a = "AAA";
                try
                {
                    int c = Convert.ToInt32(a);
                    Console.Write("f");
                }
                catch (DivideByZeroException)
                {
                    Console.WriteLine("n");
                }
                finally
                {
                    Console.WriteLine("d");
                }
                Console.WriteLine("c");
            }
            catch (FormatException)
            {
                Console.WriteLine("b");
            }
            finally
            {
                Console.WriteLine("a");
            }
        }
```

dba

Samo pratite ispis i izvršavanje i to je to, nema neke mudrosti. ConvertToInt32 baca FormatException.

66

Šta se ispisuje?

```
class Broj
{  public int vrednost = 1; }
---
        static void Main(string[] args)
        {
            Broj[] niz = new Broj[3];
            Console.WriteLine(niz[0].vrednost);
        }
```

Greška

Ovaj kod će se kompajlirati. Međutim, pri inicijalizaciji niza, samo je rečeno koliki će taj niz biti (rezervisan je određeni memorijski prostor). Pri inicijalizaciji niza, svakom elementu se dodeli njegova default vrednost (u slučaju Broj-a, to je null, jer je Broj objekat, a samim tim referentni tip). Zato će se pri pokretanju ovog programa izbaciti NullPointerException.

67

Šta se ispisuje?

```
static void Main(string[] args)
        {
            try
            {
                string a = "FON";
                try
                {
                    int c = Convert.ToInt32(a);
                    Console.Write(7);
                }
                catch (DivideByZeroException)
                {
                    Console.Write(6);
                }
                catch (Exception)
                {
                    Console.Write(5);
                }
                finally
                {
                    Console.Write(4);
                }
                Console.Write(3);
            }
            catch (FormatException)
            {
                Console.Write(2);
            }
            finally
            {
                Console.Write(1);
            }
        }
```

5431

Samo pratite ispis i izvršavanje i to je to, nema neke mudrosti. Izuzetak se hvata u `catch (Exception)` i to je to, ne stiže do FormatExceptiona. Finally blok se uvek izvršava ako se izvršilo bilo šta u try bloku.

68

Šta se ispisuje?

```
            int[,] niz = new int [,]{{4,3},{2,1}};
            Console.WriteLine(niz[1, 1]);
```

Greška

Pravimo dvodimenzionalni niz (matricu). Indeksi kreću od 0 do n u obe dimenzije. Traženi element se nalazi u drugom redu i drugoj koloni.

69 Šta se ispisuje?

```
public class A
{
        public virtual void Prikazi()
        {   Console.WriteLine("a");    }
}
public class B:A
{
        public override void Prikazi()
        {   Console.WriteLine("b");    }
    }
        static void Main(string[] args)
        {
            A a = new A();
            a.Prikazi();
            a = new B();
            a.Prikazi();
}
```

a

b

Klasa A ima svoju virtuelnu metodu koja se izvršava kad se pozove, nad svakim objektom klase A. Klasa B redefiniše ovu metodu ključnom rečju override. To znači da će svaki objekat klase B, pozivom metode Prikazi(), izvšriti onu metodu koja je redefinisana (koja je u klasi B) a ne odgovarajuću metodu nadkalse.

70 Šta se ispisuje?

```
public class A
    {
        public virtual void Prikazi()
        {   Console.WriteLine("A");     }
    }

    public class B:A
    {
        public override void Prikazi()
        {   Console.WriteLine("B");     }
    }
---------------
        static void Main(string[] args)
        {
            A a1 = new A();
            a1.Prikazi();
            A a2 = new B();
            a2.Prikazi();

            B b1 = new B();
            b1.Prikazi();
        }
```

A

B

B

Klasa A ima svoju virtuelnu metodu koja se izvršava kad se pozove, nad svakim objektom klase A.
Klasa B redefiniše ovu metodu ključnom rečju override. Kada napravimo objekat klase A, preko konstrukotra podklase, on istovremeno preuzima metode te podklase. Zato je drugo ispisano slovo B.
Konačno, objekat klase B, izvršava svoju metodu.

80

Šta se ispisuje?

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

    public class B:A
    {
        public void Prikazi()
        {
            Console.WriteLine("B");
        }
    }
--
        static void Main(string[] args)
        {
            A a1 = new A();
            a1.Prikazi();
            A a2 = new B();
            a2.Prikazi();
            B b1 = new B();
            b1.Prikazi();
        }
```

A

A

B

Klasa A ima svoju metodu koja se izvršava kad se pozove, nad svakim objektom klase A.
Klasa B ima metodu istog imena koja sakriva metodu nadklase. Međutim, za objekat klase A, uvek se izvršava metoda definisana u toj klasi, bez obzira na konstruktor (tip reference određuje koje metode mogu da se pozovu).
Konačno, objekat klase B, izvršava svoju metodu.

81

Šta se ispisuje?

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

    public class B:A
    {
        new public void Prikazi()
        {   Console.WriteLine("B");   }
    }
----------
        static void Main(string[] args)
        {
            A a1 = new A();
            a1.Prikazi();
            A a2 = new B();
            a2.Prikazi();
            B b1 = new B();
            b1.Prikazi();
```

A

A

B

Klasa A ima svoju virtuelnu metodu koja se izvršava kad se pozove, nad svakim objektom klase A.
Klasa B ima metodu istog imena i ključnom rečju new smo naznačili da ona ne sakriva metodu nadklase, već da je potpuno nova metoda. Prilikom poziva metode, tip reference određuje koje metode mogu da se pozovu, kao kod nevirtuelnih metoda.
Konačno, objekat klase B, izvršava svoju metodu.

82

Šta se ispisuje na ekranu?

```
public static void Main(string [] args)
        {
            string tekst = "Naziv je:\t FON.";
            Console.Write(tekst);
            tekst = @"Naziv je:\t FON.";
            Console.Write(tekst);
        }
```

Naziv je: FON.Naziv je:\t FON.

Naziv je: FON.Naziv je:\t FON.

Ovo rešenje je zato što stringovi koji počinju sa @ (verbatim stringovi) ne interpretiraju escape sequence karaktere, već se ceo string čita znak po znak.

83

Šta se ispisuje?

```
public class A
    {
        public virtual int Promeni(int a)
        {   return a++;    }
    }
    public class B:A
    {
        public override int Promeni(int a)
        {   return --a;    }
    }
---------
        static void Main(string[] args)
        {
            int b = 2;
            A a = new A();
            Console.Write(a.Promeni(b));
            a = new B();
            Console.Write(a.Promeni(b));
        }
```

21

Pošto su metode virtual i override, pozivaju se na osnovu toga koji je konstruktor datog objekta. Trik je jedino što se u prvoj metodi parametar prvo ispiše, pa mu se tek onda inkrementuje. Drugi trik je u tome što je prenos preko vrednosti, pa se promenljiva b nikad zapravo ne menja.

84

Da li je kod ispravan?

```
public class Broj
    {
        public int broj = 3;
        public static int operator +(Broj a, int b)
        {
            return a.broj * b;

        }

        public static void Main()
        {
            Broj broj1 = new Broj();
            int a = 3 + broj1;
        }
    }
```

NE

Kod je neispravan zato što se operator + ne može primeniti u ovom slučaju jer redosled nije dobar. Kod bi bio ispravan u slučaju da ide Broj pa int, npr:

```
int a = broj1 + 3;
```

85

Šta se ispisuje na ekranu?

```
public class Broj
    {
        public int broj = 3;
        public static int operator +(int a, Broj b)
        {
            return a * b.broj;
        }
        public static int operator +(Broj a, int b)
        {
            return b - a.broj;
        }

        public static void Main()
        {
            Broj broj1 = new Broj();
            int broj2 = 2;
            broj2 += broj1.broj;
            broj2 += broj1;
            Console.WriteLine(broj2);
        }
    }
```

15

Redosled izvršavanja je sledeći: Prvo izvršava “obični” operator + jer se sabiraju dva int-a, pa se dobije da je `broj2` jednak 2+3=5. Zatim se poziva prvi redefinisani operator + i onda je `broj2` jednak 5 * 3 = 15.

86

Šta se ispisuje na ekranu?

```
public class Broj
    {
        public int broj = 3;
        public delegate int Delegat(Broj a, int b);

        public int Dodaj(Broj c, int d)
        {   return c.broj + d;   }

        public int Pomnozi(Broj c, int f)
        {   return c.broj * f;   }
    }

    public class Banka
    {
        public static int Oduzmi(Broj x, int y)
        {   return x.broj - y;   }
        public static void Main(string [] args)
        {
            Broj broj1 = new Broj();
            Broj.Delegat dg;
            dg = broj1.Dodaj;
            dg += broj1.Pomnozi;
            dg = Oduzmi;
            Console.WriteLine(dg(broj1, 1));
        }
    }
```

2

Poslednja metoda (Oduzmi) izbacuje sve ostale metode iz delegata, zato što se koristi = a ne += ili -=. Tako da će se izvršiti 3-1=2.

2

87

Šta se ispisuje na ekranu?

```
public partial class A
    {
        public int broj = 10;
    }
    public partial class A
    {
        public int broj = 5;
    }
    public class B
    {
        public static void Main(string [] args)
        {
            A a = new A();
            Console.WriteLine(a.broj);
        }
    }
```

a) 5
b) 10
c) 0
d) Greška

d) Greška

Sadržaj parcijalnih klasa treba posmatrati kao da je u jednoj klasi. Ne mogu postojati dve promenljive sa istim imenom u klasi, tako da se prijavljuje greška.

88 Šta se ispisuje na ekranu?

```
public class A
    {
        public void Prikazi()
        {   Console.Write("1");    }
    }
    public class B : A
    {
        public new void Prikazi()
        {    Console.Write("2");   }
    }
    public class Primer
    {
        public static void Main(string [] args)
        {
            A a1 = new A();
            a1.Prikazi();
            A a2 = new B();
            a2.Prikazi();
            B b1 = new B();
            b1.Prikazi();
        }
}
```

112

Za objašnjenje, pogledajte nasleđivanje u skritpti za 2 kolokvijum.

89 Šta se ispisuje na ekranu?

```
public class A
    {
        public virtual void Prikazi()
        {   Console.Write("1");      }
    }
    public class B : A
    {
        public override void Prikazi()
        {    Console.Write("2");     }
    }
    public class Primer
    {
        public static void Main(string[] args)
        {
            A a1 = new A();
            a1.Prikazi();
            A a2 = new B();
            a2.Prikazi();
            B b1 = new B();
            b1.Prikazi();
        }
    }
```

122

Za objašnjenje, pogledajte nasleđivanje u skritpti za 2 kolokvijum.

90 Koji su tačni odgovori?

```
public class Primer
    {
        static int broj1 = 5;
        static int? broj2 = 10;
        public static void Main(string[] args)
        {
        }
    }

a) broj1 = broj2;
b) broj2 = broj1;
c) broj1 = null;
d) broj2 = null;
```

b),d)

`broj2` je nullable tip (u ovom slučaju int proširen vrednošću null)

91 Šta se ispisuje na ekranu?

```
try
{
       string naziv = "FON";
       try
       {
             int broj = Convert.ToInt32(naziv);
             Console.Write(1);
       }
       catch (DivideByZeroException) { Console.Write(2); }
       catch (Exception) { Console.Write(3); }
       finally { Console.Write(4); }
       Console.Write(5);
}
catch (FormatException)
{ Console.Write(6); }
finally { Console.Write(7);}
```

3457

Treba samo ispratiti gde se javlja greška, koji je prvi exception koji je uhvati. Finally se uvek izvrsava.

92

Šta se ispisuje?

```
public class A
    {
        public A(int a)
        {
            Console.Write("5");
        }
        public A() : this (4)
        {
            Console.Write("1");
        }
    }
    public class B : A
    {
        public B(): this(1)
        {
            Console.Write("2");
        }

        public B(int i)
        {
            Console.Write("3");
        }
    }
----
public static void Main(string[] args)
        {
            B b = new B();
        }
```

5132

Kod konstruktora poziv se određuje na sledeći način: Ako postoji odgovarajući (isti broj, tip i redosled argumenata) konstruktor nadklase, onda se taj konstruktor prvi poziva. Zatim gledamo da li se poziva neki konstruktor te iste klase. Ako se poziva, onda se on izvršava prvi, pa tek onda „naš“ konstruktor. Ovaj algoritam je rekurzivan.

U ovom slučaju: treba nam `B()`. Pošto klasa B nasleđuje klasu A, gledamo da li postoji konstruktor `A()`. Pošto postoji, sad posmatramo taj konstruktor. Gledamo da li on poziva neki drugi konstruktor svoje klase. Pošto poziva `A(int a)` onda se prvo izvršava kontruktor `A(int a)`. Taj konstruktor ne poziva nijedan drugi, tako da se vraćamo na `A()`. Kad se izvrši on, vraćamo se na `B()`. Da li on poziva neki konstruktor? Da, poziva `B(int i)` . Dakle izvrši se konstruktor `B(int i)` , a nakon njega (pošto on ne poziva nijedan drugi) izvrši se i `B()` .

93

Šta se ispisuje?

```
public class A
    {
        public A()
        {    Console.Write("1");    }
    }
    public class B : A
    {
        public B(): this(1)
        {    Console.Write("2");    }

        public B(int i)
        {
            Console.Write("3");
        }
    }
----
public static void Main(string[] args)
        {
            B b = new B();
        }
```

132

Pogledajte objašnjenje iznad

94

Šta se ispisuje?

```
public class A
    {
        public void Prikazi()
        {    Console.Write("a");     }
    }
    public class B : A
    {
        new public void Prikazi()
        {     Console.Write("b");    }
    }
    public class Primer
    {
        public static void Main(string[] args)
        {
            A a1 = new A();
            a1.Prikazi();
            A a2 = new B();
            a2.Prikazi();
        }
    }
```

aa

Za objašnjenje pročitajte skriptu za drugi kolokvijum.

95 Šta se ispisuje?

```
public class A
    {
        public virtual void Prikazi()
        {    Console.Write("a");     }
    }
    public class B : A
    {
        public override void Prikazi()
        {    Console.Write("b");     }
    }
    public class Primer
    {
        public static void Main(string[] args)
        {
            A a1 = new A();
            a1.Prikazi();
            A a2 = new B();
            a2.Prikazi();
        }
    }
```

ab

Sličan prethodnom. Za objašnjenje pročitajte skriptu za drugi kolokvijum.

2

96 Šta se ispisuje?

```
public class A
    {
        public A(): this("b")
        {     Console.Write("a");     }
        public A(string i)
        {     Console.Write(i);       }
        public virtual void Draw()
        {     Console.Write("d");     }
    }
    public class D : A
    {
        public D()
        {     Console.Write("c");     }
    }

    public class Primer
    {
        public static void Main(string[] args)
        {
            D d = new D();
        }
    }
```

bac

Već je objašnjeno kojim redosledom se konstruktori pozivaju.

97

Šta se ispisuje na ekranu?

```
class Racun
{
    public double stanje;
    public string vlasnik;

    public Racun(double s, string v)
    {
        stanje = s;
        vlasnik = v;
    }
    public string Uplata(double iznos)
    {
        Console.WriteLine(1);
        stanje += iznos;
        return string.Format("Novo stanje je {0}", stanje);
    }
    public string Isplata(double iznos)
    {
        Console.WriteLine(2);
        if (stanje < iznos) return "Nedovoljno sredstava";
        stanje -= iznos;
        return string.Format("Novo stanje je {0}", stanje);
    }

    public delegate string ObradaTransakcije(double broj);
    public static double provizija = 0.05;
    public static string PromeniProviziju(double nova)
    {
        Console.WriteLine(3);
        provizija = nova;
        return   string.Format("Nova   provizija   je   {0}",
provizija);
    }
}

public class Banka
{
    public static void Main(string [] args)
    {
        Racun r = new Racun(400, "Pera");
        Racun.ObradaTransakcije dg;
        dg = r.Uplata;
        Console.WriteLine(dg(200));
        dg = r.Isplata;
        Console.WriteLine(dg(100));
        dg = Racun.PromeniProviziju;
        Console.Write(dg(0.01));
     }
}
```

1

Novo stanje je 600

2

Novo stanje je 500

3

Nova provizija je 0.01

Samo treba pratiti izvršenje transakcija preko delegata, sve je čisto.

98 Šta se ispisuje na ekranu?

```
class Broj
    {
        public int broj;
        public delegate void Delegat(int a);
        public event Delegat Dogadjaj;

        public int Promeni(int x)
        {
            broj = x;
            Obavesti();
            return broj;
        }
        public void Obavesti()
        {
            if(Dogadjaj !=null)
                Dogadjaj(broj);
        }
    }
public class Banka
{
        static int vrednost = 0;

        public static void Dodaj(int x)
        {     vrednost += x;      }
        public static void Oduzmi(int x)
        {     vrednost -= x;      }
        public static void Pomnozi(int x)
        {     vrednost *= x;
      }
        public static void Main(string [] args)
        {   Broj broj1 = new Broj();
            broj1.Dogadjaj += Oduzmi;
            broj1.Dogadjaj += Dodaj;
            broj1.Dogadjaj += Pomnozi;
            broj1.Dogadjaj -= Oduzmi;
            broj1.Promeni(2);
            Console.WriteLine(vrednost);
        }
}
```

4

Redosled ubacivanja izbacivanja u delegat je sledeći:
Dodaje se Oduzmi: `Oduzmi`
Dodaje se Dodaj: `Oduzmi, Dodaj`
Ubacuje se Pomnozi: `Oduzmi, Dodaj, Pomnozi`
Izbacuje se oduzmi: ~~`Oduzmi`~~`, Dodaj, Pomnozi`
Dakle, ostaju samo: `Dodaj, Pomnozi`

99 Koji su tačni odgovori?

```
public class Racun{}
public class TekuciRacun : Racun { }
a) TekuciRacun tr1 = new Racun();
b) Racun r1 = new Racun();
c) Racun r2 = new TekuciRacun();
d) TekuciRacun tr2 = new TekuciRacun();
```

b),c),d)

Primeri sa ispita/kolokvijuma 2012 i 2013

1 Ako je TIndex class, šta će se ispisati:

```
TIndex i1 = new TIndex();
            TIndex i2 = i1;
            TIndex i3 = new TIndex();

            i2.broj = 1;
            i3 = i1;
            i3.broj = 2;
            Console.Write(i2.broj);
```

A. 0 B. 1 C. 2

C. 2

2 Ako je TIndex struct, šta će se ispisati:

```
TIndex i1 = new TIndex();
            TIndex i2 = i1;
            TIndex i3 = new TIndex();

            i2.broj = 1;
            i3 = i1;
            i3.broj = 2;
            Console.Write(i2.broj);
```

A. 0 B. 1 C. 2

B. 1

3 Šta se ispisuje na ekranu?

```
string str1 = "123";
            string str2 = str1;
            string str3 = str2;
            str2 = "321";
            Console.WriteLine(str1[2]);
            Console.WriteLine(str2[2]);
            Console.WriteLine(str3[2]);
```

A. 1 B. 2 C. 3

C.3

4 Šta se ispisuje na ekranu?

```
TIndex i1 = new TIndex();
            i1.broj = 1;
            TIndex i2 = i1;
            i2.broj = 2;
            if (i1 == i2) Console.WriteLine("Isti");
            else Console.WriteLine("Razliciti");
```

A. Isti B. Razliciti C. Prijavice se greska

Isti

5 Koje metode se mogu definisati u navedenoj klasi “Primer”?

```
public class Primer
  {
      public char Proba(double d, string b)
      {
          return 'a';
      }
  }
```

A. `public char Proba(double c, string a) {}`
B. `public char Proba(string b, double d) {}`
C. `public char Proba(ref double d, string b) {}`
D. `public string Proba(double d, string b) {}`
E. `public char Proba(string d, string b) {}`
E. `protected char Proba(double d, string b) {}`

b,c,e

6 Koje metode se mogu pridružiti navednom delegatu?

```
public delegate char Delegat (byte a, Osoba b);
```

A. `public char A (byte b, Osoba a) {}`
B. `public char B (out byte a, Osoba b) {}`
C. `public string C (byte a, Osoba b) {}`
D. `public char D (Osoba a, byte b) {}`
E. `public static char E (byte a, Osoba b) {}`
F. `protected char F (byte a, Osoba b) {}`

a, e, F

7 Šta se ispisuje na ekranu?

```
public partial class A
{
    public A(int broj)
    {
        broj = broj;
    }
}

public partial class A
{
    public int broj = 2;
}

A a = new A(1);
Console.WriteLine(a.broj);
```

A. 0 B. 1 C. 2 D. Prijaviće se greška

c.2

9 Koja je od navedenih naredbi ispravna?

```
char prom1 = 'x';
           char? prom2 = 'y';



           prom2 = prom1;
```

A. `prom1 = null;`
B. `prom2 = null;`
C. `prom1 = prom2;`
D. `prom2 = prom1;`
E. Nijedna
F. Prijaviće se greška

B,D

10 Koja je od navedenih naredbi ispravna?

```
double broj1 = 10.3;
byte broj2 = 3;
```

```
A. broj1 = broj2;
B. broj1 = (double)broj2;
C. broj2 = broj1;
D. broj2 = (byte)broj1;
E. int broj3 = broj2;
F. string broj4 = (string)broj2;
```

a,b,d,e

11 Šta se ispisuje na ekranu?

```
TIndex[] nizIndexa = new TIndex[2];
Console.WriteLine(nizIndexa[0].broj);
```

A. 0 B. 1 C. 2 D. Prijaviće se greška

d.greska

12 Koje metode mora implementirati klasa koja nasleđuje interfejs *I_Y* ?

```
interface I_X
    {
        void C(string podaci);
    }

    interface I_Y : I_X
    {
        bool B(int broj);
        string A();
    }
```

A. A B. B C. C D. Nijednu E. kod nije ispravan

A,B,C (sve)

13 Koja od navedenih metoda se mora nalaziti u klasi *Osoba* da bi navedeni kod bio ispravan?

```
interface I_X
public class Student : Osoba
    {
       public override char Metoda(double d, string b)
       {return 'a'; }
    }
```

A `public new char Metoda(double d, string b) {}`
B. `public sealed char Metoda(double d, string b) {}`
C. `public virtual char Metoda(double d, string b) {}`
D. `public abstract char Metoda(double d, string b) {}`
E. `public static char Metoda(double d, string b) {}`
F. `public char Metoda(double d, string b) {}`

c. virtual

14 Koja od navedenih naredbi je ispravna?

```
    public class CeoBroj { }
    public struct Broj { }
    public class Podatak<T> where T : class { }
```

A. `Podatak <char> p = new  Podatak <char> {};`
B. `Podatak <string> p = new  Podatak <string> {};`
C. `Podatak <Broj> p = new  Podatak <Broj> {};`
D. `Podatak <CeoBroj> p = new  Podatak <CeoBroj> {};`

b, d

15 Koja od navedenih naredbi je ispravna?

```
abstract class Nastavnik
    {
        protected string ime;
        public abstract string Ime
        {
            get;
            internal set;
        }
    }
    abstract class Asistent : Nastavnik
    {
        public string grupa;
    }
```

A. Da B. Ne

a.Ispravna

16 Koja od navedenih klasa je ispravna?

```
class Osoba { }

    sealed class Student : Osoba { }
    class RedovanStudent : Student { };
    abstract class Nastavnik : Osoba { }
    class Asistent : Nastavnik { }
```

A. RedovanStudent B. Asistent

B. Asistent

17 Koja od navedenih naredbi je ispravna?

```
class Osoba { }

    sealed class Student : Osoba { }
    abstract class Nastavnik : Osoba { }
```

A. `Osoba o = new Student();`
B. `Student s = new Osoba();`
C. `Osoba d = new Nastavnik();`
D. `Nastavnik n = new Osoba();`

A.

19 Šta se ispisuje na ekranu?

```
static void Proba(int i)
        {
            i += 1;
        }
        static void Proba(TIndex i)
        {
            i.broj -= 2;
        }

        static void Main(string[] args)
        {
            TIndex index = new TIndex();
            Proba(index.broj);
            Proba(index);
            Console.WriteLine(index.broj);
        }
```

A. -2 B. -1 C. 0 D. 1 E. 2 F. Prijaviće se greška

A. -2

20 Šta se ispisuje na ekranu?

```
static void Proba(ref int i)
        {
            i += 1;
        }
        static void Proba(ref TIndex i)
        {
            i.broj -= 2;
        }

        static void Main(string[] args)
        {
            TIndex index = new TIndex();
            Proba(ref index.broj);
            Proba(ref index);
            Console.WriteLine(index.broj);
        }
```

A. -2 B. -1 C. 0 D. 1 E. 2 F. Prijaviće se greška

b.-1

21 Koje je od sledećih tvrđenja tačno?

```
static void Metoda_1(out int i)
        {
            i = 5;
        }
        static void Metoda_2(out int i)
        {
            i += 5;
        }
static void Main(string[] args)
        {
            int broj = 2;
            poziv ispravne metode

        }
```

A. Metoda_1 je ispravna
B. Metoda_2 je ispravna
C. Nakon poziva ispravne metode promenljiva *broj* ima vrednost 2
D. Nakon poziva ispravne metode promenljiva *broj* ima vrednost 5
E. Nakon poziva ispravne metode promenljiva *broj* ima vrednost 7
F. Nakon poziva ispravne metode promenljiva *broj* nema vrednost

A. metoda_1

D. 5

22 Šta se ispisuje na ekranu?

```
namespace ConsoleApplication9{
public class Broj
    {
        int vrednost = 2;
    }

----
string tekst = @"Data je \t vrednost";
            Broj b = new Broj();

            tekst = string.Format("{1} \t {0}", b, tekst);
            Console.WriteLine(tekst);
```

A. Data je \t vrednost 2
B. Data je vrednost \t Primer.Broj
C. Data je \t vrednost Primer.Broj
D. Data je vrednost \t 2
E. Nije ponuđen tačan odgovor
F. Prijaviće se greška

C.

23 Šta se ispisuje na ekranu?

```
public class A
    {
        public A() : this(5) { Console.Write(6); }
        public A(int broj)   { Console.Write(7); }
    }
    public class B : A
    {
        public B() : this(8) { Console.Write(9); }
        public B(int broj) { Console.Write(broj); }
    }
B b1 = new B();
```

7689

Odgovor napisati na obrascu za odgovore!

24 Kako izgleda niz nakon izvršavanja sledećeg koda?

```
public class Brojevi
    {
        int[] niz = new int[] { 5, 6, 7, 8 };
        public int this[int indeks]
        {
            set { niz[indeks + 1] = value; }
        }
    }

Brojevi b = new Brojevi();
b[1] = 4;
```

5648

Odgovor napisati na obrascu za odgovore!

25 Šta se ispisuje na ekranu?

```
public class A
    {
        public virtual void Prikaz()
        {      Console.Write(4);       }
    }
    public class B : A
    {
        public new void Prikaz()
        {       Console.Write(7);      }
    }

    public class C : A
    {
        public override void Prikaz()
        {       Console.Write(9);      }
    }

A a1 = new A(); a1.Prikaz();
            A a2 = new B(); a2.Prikaz();
            A a3 = new C(); a3.Prikaz();
```

449

Odgovor napisati na obrascu za odgovore!

26 Šta se ispisuje na ekranu?

```
int i=0, j=0, suma = 0;
            do
            {
                suma++;
                while (j < 3)
                {
                    j++;
                    suma++;
                    if (j == 2)
                        break;
                }

                i++;
            } while (i < 2);
            Console.Write(suma++);
```

5

Odgovor napisati na obrascu za odgovore!

27 Šta se ispisuje na ekranu?

```
try
{
        string naziv = "FON";
        Console.Write(1);
        int broj = Convert.ToInt32(naziv);
        try { Console.Write(2); }
        catch (FormatException) { Console.Write(3);     }
        finally { Console.Write(4); }
        Console.Write(5);
}
catch (DivideByZeroException)
{
          Console.Write(6);
}
catch (Exception)
{
          Console.Write(7);
}
finally
{
          Console.Write(8);
}
Console.Write(9);
```

1789

Odgovor napisati na obrascu za odgovore!

28 Šta se ispisuje na ekranu?

```
class Broj
    {
        public int broj = 3;
        public static int operator *(Broj b, int a)
        {
            return a + b.broj;
        }
        public static int operator *(int a, Broj b)
        {
            return a - b.broj;
        }

        static void Main(string[] args)
        {
            Broj broj1 = new Broj();
            int broj2 = 5;
            broj2 *= broj1;
            broj2 *= broj1.broj;
            Console.WriteLine(broj2);
        }
    }
```

6

Odgovor napisati na obrascu za odgovore!

29 Šta se ispisuje na ekranu?

```
public delegate void Delegat(Broj a, ref int b);
    public class Broj
    {
        public int broj = 6;
        public void Dodaj(Broj c, ref int d)
        {   c.broj += d; d++;   }
        public void Pomnozi(Broj e, ref int f)
        {   e.broj *= f; f++;   }
    }
    class Primer
    {
        public static void Oduzmi(Broj x, ref int y)
        {   x.broj -= y; y++;   }

        static void Main(string[] args)
        {
            Broj broj1 = new Broj();
            Delegat dg;
            dg = Oduzmi;
            dg += broj1.Dodaj;
            dg = broj1.Pomnozi;
            dg += broj1.Dodaj;
            dg += Oduzmi;
            dg -= broj1.Dodaj;

            int z = 7;
            dg(broj1, ref z);
            Console.WriteLine(broj1.broj);
        }
```

34

Odgovor napisati na obrascu za odgovore!

30 Šta se ispisuje na ekranu?

```
public delegate void Delegat(int a);
    public class Broj
    {
        public int broj = 8;
        public event Delegat Dogadjaj;
        public void Promeni(ref int x)
        {
            x = broj;
            Obavesti();
        }
        public void Obavesti()
        {
            if (Dogadjaj != null)
                Dogadjaj(broj);
        }
    }

    class Primer
    {
        static int vrednost = 0;
        public static void Dodaj(int x) { vrednost += x; }
        public static void Oduzmi(int x) { vrednost -= x; }
        public static void Pomnozi(int x) { vrednost *= x; }
        static void Main(string[] args)
        {
            Broj broj1 = new Broj();
            broj1.Dogadjaj += Dodaj;
            broj1.Dogadjaj += Oduzmi;
            broj1.Dogadjaj += Pomnozi;
            broj1.Dogadjaj -= Dodaj;
            broj1.Dogadjaj += Oduzmi;

            int z = 7;
            broj1.Promeni(ref z);
            Console.WriteLine(vrednost);
        }
    }
```

-72

Odgovor napisati na obrascu za odgovore!

31 Šta se ispisuje na ekranu?

```
public delegate void Delegat(int a);
public class Broj
{
        public int broj = 4;
        public event Delegat Dogadjaj;

        public void Promeni(ref int x)
        {
            broj = x;
            if (Dogadjaj != null)
                Dogadjaj(broj);
        }
}

class Primer
{
        static int vrednost = 0;
        public static void Dodaj(int x) { vrednost += x; }
        public static void Oduzmi(int x) { vrednost -= x; }
        public static void Pomnozi(int x) { vrednost *= x; }
        static void Main(string[] args)
        {
            Broj broj1 = new Broj();
            broj1.Dogadjaj += Dodaj;
            broj1.Dogadjaj += Oduzmi;
            broj1.Dogadjaj += Pomnozi;
            broj1.Dogadjaj -= Dodaj;
            broj1.Dogadjaj += Oduzmi;

            int z = 5;
            broj1.Promeni(ref z);
            Console.WriteLine(vrednost);
        }
}
```

-30

Odgovor napisati na obrascu za odgovore!

32 Šta se ispisuje na ekranu?

```
class Broj
    {
        public int broj = 5;
        public static int operator *(Broj b, int a)
        {    return a + b.broj;     }
        public static int operator *(int a, Broj b)
        {    return a - b.broj;     }

        static void Main(string[] args)
        {
            Broj broj1 = new Broj();
            int broj2 = 11;
            broj2 *= broj1;
            broj2 *= broj1.broj;
            Console.WriteLine(broj2);
        }
    }
```

30

Odgovor napisati na obrascu za odgovore!

33 Šta se ispisuje na ekranu?

```
public delegate void Delegat(Broj a, ref int b);
public class Broj
    {
        public int broj = 7;
        public void Dodaj(Broj c, ref int d)
        {   c.broj += d; d++;     }
        public void Pomnozi(Broj e, ref int f)
        {   e.broj *= f; f++;     }
    }

class Primer
{
        public static void Oduzmi(Broj x, ref int y)
        {    x.broj -= y; y++;     }

        static void Main(string[] args)
        {
            Broj broj1 = new Broj();
            Delegat dg;
            dg = Oduzmi;
            dg += broj1.Dodaj;
            dg = broj1.Pomnozi;
            dg += broj1.Dodaj;
            dg += Oduzmi;
            dg -= broj1.Dodaj;

            int z = 3;
            dg(broj1, ref z);
            Console.WriteLine(broj1.broj);
        }

}
```

17

Odgovor napisati na obrascu za odgovore!

34 Šta se ispisuje na ekranu?

```
try
{
       string naziv = "FON";
       Console.Write(9);
       int broj = Convert.ToInt32(naziv);
       try { Console.Write(8); }
       catch (FormatException) { Console.Write(7); }
       finally { Console.Write(6); }
       Console.Write(5);
}
catch (DivideByZeroException)
{       Console.Write(4);    }
catch (Exception)
{       Console.Write(3);    }
finally
{       Console.Write(2);    }
Console.Write(1);
```

9321

Odgovor napisati na obrascu za odgovore!

35 Šta se ispisuje na ekranu?

```
public class A
    {
        public virtual void Prikaz()
        {    Console.Write(4);      }
    }
public class B : A
    {
        public new void Prikaz()
        {    Console.Write(3);      }
    }

public class C : A
    {
        public override void Prikaz()
        {    Console.Write(2);      }
    }
public class D : A
    {
        public override void Prikaz()
        {    Console.Write(1);      }
    }

       A a1 = new A(); a1.Prikaz();
       A a2 = new B(); a2.Prikaz();
       A a3 = new C(); a3.Prikaz();
       A a4 = new D(); a4.Prikaz();
```

4421

Odgovor napisati na obrascu za odgovore!

36 Šta se ispisuje na ekranu?

```
namespace Primer{
public class Broj
    {
        int vrednost = 2;
    }

----
string tekst = @"Data je \t vrednost";
Broj b = new Broj();

tekst = string.Format("{1} \t {0}", b, tekst);
Console.WriteLine(tekst);
```

A. Data je \t vrednost 2
B. Data je vrednost \t Primer.Broj
C. Data je \t vrednost Primer.Broj
D. Data je vrednost \t 2
E. Nije ponuđen tačan odgovor
F. Prijaviće se greška

C)

37 Koji konstruktor je moguc?

```
class Osoba { }

    abstract class Student : Osoba { }
    sealed class Nastavnik : Osoba { }
```

a) `Osoba o = new Student();`
b) `Student s = new Osoba();`
c) `Osoba d = new Nastavnik();`
d) `Nastavnik n = new Osoba();`

Osoba d = new Nastavnik();

38 Zaokruzi slova ispred tačnih odgovora (ispravan kod)?

```
public class CeoBroj { }
public struct Broj { }

public class Podatak<T> where T : struct { }
```

a) `Podatak<bool> p = new Podatak<bool>();`
b) `Podatak<string> p3 = new Podatak<string>();`
c) `Podatak<Broj> pd = new Podatak<Broj>();`
d) `Podatak<CeoBroj> ph = new Podatak<CeoBroj>();`

a) i c)

39 Koja od navedenih metoda se mora nalaziti u klasi *Osoba* da bi navedeni kod bio ispravan?

```
interface I_X
public class Student : Osoba
    {
       public override char Metoda(double d, string b)
       {return 'a'; }
    }
```

A `public new char Metoda(double d, string b) {}`
B. `public sealed char Metoda(double d, string b) {}`
C. `public virtual char Metoda(double d, string b) {}`
D. `public abstract char Metoda(double d, string b) {}`
E. `public static char Metoda(double d, string b) {}`
F. `public char Metoda(double d, string b) {}`

c. virtual

40 Šta se ispisuje?

```
TIndex[] nizIndexa = new TIndex[2];
Console.WriteLine(nizIndexa[0].broj);
```

A 0
B. 2
C. 1
D. Greška

d.greska

41

Koje klase su ispravne?

```
interface I_Podaci
{        string Izracunaj();       }

interface I_Prikaz : I_Podaci
{        void Prikazi(string podaci);    }

public class A : I_Podaci
{
     public string Izracunaj()
     {         return "aaa";      }
}

public class B : I_Podaci
{
     public string Izracunaj()
     {      return "aaa";      }

     public void Prikazi(string podaci)
     {      Console.Write(podaci);     }
}

public class C : I_Prikaz
{
     public void Prikazi(string podaci)
     {      Console.Write(podaci);     }
}

public class D : I_Prikaz
{
     public string Izracunaj()
     {      return "aaa";       }

     public void Prikazi(string podaci)
     {      Console.Write(podaci);     }
}
```

Klase A, B i D

42

Šta se ispisuje?

```
public partial class A
    {
        public void Metoda()
        {   broj++;     }
    }

    public partial class A
    {
        public int broj = 0;

        public A()
        {   broj++;    }
    }

A a = new A();
a.Metoda();
Console.WriteLine(a.broj);
```

2

43 Šta se ispisuje?

```
ArrayList a = new ArrayList();
a.Add(new TIndex());
a.Add(new TIndex());
Console.WriteLine(a[0].broj);
```

greska