

01. Da li je kod ispravan?

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

odgovor: Greška

komentar: Promenljiva broj je nedostupna metodi klase B usled neodgovarajućeg modifikatora pristupa.

Build-ovanje prolazi sa protected .

// protected int broj;

02. Šta ispisuje sledeći kod:

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

odgovor: 0

komentar: Ocena ima default vrednost a to je 0, kao i svaka podrazumevana vrednost podatka tipa int. Kako vršimo prenos preko vrednosti, novoj promenljivoj ocena tipa int dodeljujemo vrednost atributa ocena objekta klase Student, a to je 0.

//Da je u klasi Student stajalo public int ocena = 30; ispisalo bi se 30.

03. Šta ispisuje sledeći kod:

```
class Primer{
    static void Metoda(int x)
    {
        x--;
    }
    static void Main()
    {
        int broj = 1;
        Metoda(broj);
        Console.WriteLine(broj);
    }
}
```

odgovor: 1

komentar: Vršiti se prenos preko vrednosti. Napravi se nova kopija promenljive broj i Metoda() radi sa tom kopijom, a vrednost promenljive broj ostane netaknuta kada se program vrati u Main metodu i zato se ispiše 1.

// kada bi u kodu pisalo

static void Metoda(ref int x) i

...

Metoda(ref broj);

Tada bi se izvršio prenos preko reference i broj bi se umanjio za 1 i ispisala na kraju 0.

04. Šta ispisuje sledeći kod:

```
class Program
{
    public void Izmeni(int a)
    { a = a + 10; }
    public void Izmeni1(int a)
    { a = a - 10; }

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

odgovor: 20

komentar: Vršimo prenos parametra preko vrednosti. Metode Izmeni() i Izmeni1() rade sa kopijom parametra ulaz i zbog toga kada se program vrati u Main() metodu, vrednost te promenljive ostaje nepromenjena. Da bi smo zaobišli prenos parametra preko vrednosti potrebno je izvršiti prenos parametra preko reference putem ključne reči ref.

05. Šta ispisuje sledeći kod:

```
class Student
{
public int brojIndeksa = 112;
}
...
Student s = new Student();
Student s1 = s;
s.brojIndeksa = 100;
Console.WriteLine(s1.brojIndeksa);
```

odgovor: 100

komentar: Klasa je referentni tip podataka. Kada napišemo Student s1 = s; , to znači da sadržaj iz promenljive s upiši u s1, a to je adresa. Nakon toga situacija je da reference s i s1 pokazuju na objekat tipa student na heap-u i kada promenimo vrednost promenljive brojIndeksa, ona se menja i za s i za s1.

06. Šta ispisuje sledeći kod ?

```
class Broj
{public int vred;}
class primer
{static void Main(){
Broj br1 = new Broj();
    Broj br2 = br1;
    br1.vred = 50;
    br2.vred = 100;
Console.WriteLine(br1.vred);
}}}
```

odgovor: 100

komentar: Klasa Broj je referentni tip podataka, što znači da na stacku sadrži adresu objekta koji se nalazi u heapu(dinamičkoj memoriji). Napravimo prvu instancu klase Broj br1, nakon toga napravimo drugu instancu klase Broj, u koju se upisuje sadržaj prve instance, odnosno adresa, tako da br1 i br2 referenciraju na istu memorisku lokaciju. Sa promenom vrednosti u jednoj instanci, menja se i vrednost u drugoj, tako da kada ispišemo vrednost uvek se posmatra poslednja promena i ona će biti ispisana.

07. Šta ispisuje sledeći kod:

```
class A
{ public void Ispisi()
{Console.WriteLine("X");}
}
class B:A{}
...
B b = new B();
b.Ispisi();
```

odgovor: X

komentar: Nasleđena klasa nasleđuje polja i metode bazne klase, tako da može da ih koristi. Pripaziti na označeni modifikator, može da dođe promena.

08. Šta ispisuje sledeći kod ?

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

odgovor: 3

komentar: Vršimo eksplicitnu konverziju ili tzv. cast-ovanje. Kod cast-ovanja, vrši se odsecanje bitova koje prouzrokuje gubitak određenih informacija. Kako smo odsekli 32 bita od podatka tipa double(koji inače ima 64 bita), da bi mogli da ga smestimo u

podatak tipa int(32 bita), tako je došlo i do odsecanja decimale, pa je rezultat 3. **Da smo iskoristili metodu Convert.ToInt32(); izvršili bi zaokruživanje po matematičkom pravilu i rezultat bi bio 4**

09. Da li je sledeći kod ispravan:

```
int i = 100;  
const int broj = i;
```

odgovor: nije

komentar: Konstanti se može pridružiti samo konstanta. Dok se vrednost konstante može upisati u broj. Mora stajati `const int i = 100;`

10. Da li je kod ispravan?

```
class Krug{  
Public int precnik; }  
class Primer {  
static void Main(){  
Krug k; ---  
k.precnik=50;  
Console.WriteLine(k.precnik) }}
```

odgovor: Greška

komentar: Nismo napravili objekat (instancu) klase Krug preko operatora new i poziva konstruktora klase Krug (već samo referencu), a nakon toga smo pokušali da pristupimo njegovoj instanci. **Potrebno je izmeniti kod tako da piše.**

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

11. Da li je kod ispravan?

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

odgovor: Greška

komentar: Nismo napravili objekat (instancu) klase Broj preko operatora new i poziva konstruktora klase Broj (već samo referencu), a nakon toga smo pokušali da pristupimo njegovoj instanci. **Potrebno je izmeniti kod tako da piše.**

// `Broj br = new Broj();`

12. Šta se ispisuje na ekranu?

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

odgovor: PMF

komentar: Preko switch-a skačemo na odgovarajući case. Kako je promenljiva u okviru switch-a dobila vrednost PMF, skočili smo na odgovarajući case za PMF i izvršili pridruženu naredbu. **Da je kojim slučajem umesto PMF u promenljivoj kod pisalo ETF, takođe bi se izvršila odgovarajuća komanda za PMF, jer case ETF nema u sebi ni jednu naredbu, pa bi bilo dozvoljeno napredovanje ka narednom case-u a to je PMF.**

13. Šta ispisuje sledeći kod:

```
string s = "Ne";
switch(s)
{
case "Ne": Console.WriteLine("Ne");
            break;
case "Da": Console.WriteLine("Da");
            break;
}
```

odgovor: Ne

komentar: preko switch-a skačemo na odgovarajući case, a to je Ne i izvršavamo odgovarajuću naredbu.

14. Šta ispisuje sledeći kod:

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

odgovor: 2 3 4 5

komentar: Kod for petlje prvo se inicijalizuje vrednost, proveriti uslov, izvrši telo petlje, pa zatim se uveća iterator. U prvom krugu u promenljivu i se upiše 0, u promenljivu a se upiše takođe 0. Zatim se ulazi u while petlju koja se izvršava dokle god je uslov ispunjen. Uslov je da $a < i + 2$ i ispunjen je 2 iteracije. U prvoj iteraciji u a dobije vrednost 1, dok u drugoj dobije vrednost 2, nakon toga uslov prestaje da važi i iskače se iz petlje i ispisuje poslednja vrednost promenljive a, a to je 2. Sada se iterator poveća za 1, odnosno i postane 1 i proveriti se opet uslov for petlje i pošto je ispunjen izvršava se telo petlje. I postupak se ponavlja dokle god promenljiva i ne dobije vrednost 4, kada se iskače iz for petlje.

15. Koliko puta će se izvršiti sledeća petlja:

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

odgovor: 4 puta

komentar: do-while prvo odradi telo petlje pa zatim proveriti uslov.

16. Šta je rezultat izvršenja sledećeg koda ?

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

odgovor: 1 3 5 5

17. Šta ispisuje sledeći kod:

```
string str1 = "AAA";
string str2 = str1;
```

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

odgovor: AAA
CCC

komentar: Iako je string referentni tip podataka, **kod njega se uvek pravi nova kopija u memoriji**. Posle linije koda `string str2 = str1`; pravi se nova kopija sa istim sadržajem u memoriji, tako da promenljive `str1` i `str2` sadrže u sebi različite memorijske adrese, ali njihov sadržaj je isti. Sa promenom sadržaja jedne ne menja se druga promenljiva. Isto se odnosi na promenljivu `str3`. `str3` ostaje da pokazuje na niz AAA, dok `str2` pokazuje na niz CCC.

18. Vrednost str2 nakon izvršavanja sledećeg koda je:

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

odgovor: Greska

komentar: Substring je metoda klase String i ona ima 2 preopterećenja. Drugo preopterećenje prima kao prvi parametar početni indeks tipa int, a kao drugi parametar dužinu podniza takođe tipa int. Kako je u `str1` na drugom mestu C, jer indeksiranje kreće od 0, a željena dužina podniza 2, to bi značilo da ukupna dužina niza koji treba da se iseče treba da bude 4, što kod nas nije slučaj i zato iskače izuzetak za prekoračenje niza.

```
// da je stajalo str1 = "ABCD";
    ispisalo bi se: CD
//da je stajalo str1.Substring(2,1)
    ispisalo bi se: C
```

19. Šta ispisuje sledeći kod:

```
string str1 = "ABC";
string str2 = str1.Insert(3,"D");
Console.WriteLine(str2);
```

odgovor: ABCD

komentar: Insert je metoda klase String i ona prima 2 parametara. Prvi parametar je mesto gde želimo da ubacimo string i ono je tipa int (indeksiranje kreće od 0), drugi parametar je string koji želimo da ubacimo. Dakle, želimo na treće mesto da ubacimo D, a to je baš kraj našeg predašnjeg stringa i s'toga je rezultat upravo ABCD.

```
// da je stajalo string str2 = str1.Insert(2,"D");
    ispisalo bi se ABDC
```

20. Šta ispisuje sledeći kod:

```
string str1 = "C#";
string str2 = "C#";
if(str1 == str2)
{
    Console.WriteLine("Isti");
}
else
{
    Console.WriteLine("Razliciti");
}
```

odgovor: Isti

komentar: Iako se pravi nova kopija stringa u memoriji, preko operatora `==` se ispituje sadržaj u if. Kako im je sadržaj isti ispisuje se odgovarajuća poruka u konzoli.

21. Šta ispisuje sledeći kod:

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

odgovor: g

komentar: Kada radimo sa nizovima uvek se pravi odgovarajuća kopija u memoriji. Tako da nakon druge linije koda promenljiva b pokazuje na string Beograd, koji je se nalazi na drugoj lokaciji u odnosu na string Beograd na koji a pokazuje. Kada promenimo da a pokazuje na Novi Sad, b ostaje da pokazuje na Beograd. Nakon toga vršimo ispis člana niza sa indeksom 3, a to je zapravo četvrto slovo od Beograd.

22. Da li je sledeći kod ispravan:

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

odgovor: jeste – može da se build-uje

23. Vrednost konstante Zeleno je:

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

odgovor: 2

komentar: Enumeracija (nabrajanje) je korisnički definisani vrednosni tip podataka, koji u svojoj osnovi uvek ima pozadinski tip koji je podrazumevano int. Svaka vrednost (konstanta) smeštena u enumeraciju, čuva se kao vrednost pozadinskog tipa. Ukoliko vrednostima enumeracije nisu dodeljene pozadinske vrednosti, one počinju od 0 i svaka je veća od prethodne za 1. Pozadinske vrednosti enumeracije Semafor počinju od 0, tako da Crveno ima vrednost 0, a Zuto vrednost za 1 veću odnosno 1, a Zeleno za 1 veću u odnosu na Zuto. **Da bi se ispisala vrednost koja ima Zeleno, potrebno je izvršiti eksplicitnu konverziju vrednosti Zeleno enumeracije Semafor, putem Console.WriteLine((int)Semafor.Zeleno);**

24. Šta ispisuje sledeći kod:

```
class Primer {
    enum Pol
    {
        zenski,
        muski = 5
    }
    static void Main(string[] args)
    {
        Pol p = Pol.muski;
        Console.WriteLine(p - 5);
    }
}
```

odgovor: zenski

komentar: Pozadinska vrednost zenski je 0, kada od 5 oduzmemo 5 dobijemo 0, tako da odgovarajuću vrednost pridruženu 0 ispisemo.

Da je stajalo Console.WriteLine(p - 4);

Prikazao bi 1, jer u enumeraciji Pol ne postoji ni jedna vrednost koja ima pozadinsku vrednost 1. Dodati u enumeraciji Pol ispod zenski liniju koda dete., Sada probati Console.WriteLine(p - 4);

25. Koja je vrednost konstante "Crvena"?

```
public enum Semafor
{
    zelena,
    zuta = 2,
    crvena
}
```

odgovor: 3

komentar: Vrednost crvena je za 1 veća nego vrednost zuta.

26. Koje metode se mogu pridružiti definisanom delegatu?

```
class Osoba
{
    public delegate void Delegat(Broj a, int b);
}
```

Ponudjeno:

- A. **public int A(Broj c, int d){}**
- B. **public static int B(Broj c, int d){}**
- C. **public string C(Broj a, int b){}**
- D. public int D(int a, Broj b){}

Odgovor: A, B, C. (mora samo potpis da bude isti)

27. Koje od sledećih metoda mogu da stoje zajedno sa metodom

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

- a) public int Metoda(int c, string d){}
- b) public bool Metoda(int a, string b){}
- c) private int Metoda(int a, string b){}
- d) **public int Metoda(string a, int b){}**
- e) **public int Metoda(int a, int b){}**
- f) **public int Metoda(int a){}**
- g) **public int Metoda(ref int a, string b){}---??**
- h) **public int Metoda(out int a, string b){}**

odgovor: d, e, f, g – za g nisam siguran, ali tako je po svesci

28. Data je sledeća klasa:

```
public class Klasa
{
    public double Metoda(double d, bool b) {return 10;}
}
```

Koja od sledećih metoda se ne mogu definisati u okviru ovoj klasi:

1. public double Metoda(bool b, double d) {return 10;}
2. public double Metoda(ref double d, bool b) {return 10;}??
3. **private double Metoda(double d, bool b) {return 10;}**
4. **protected double Metoda(double d, bool b) {return 10;}**
5. public double Metoda(double d, double b) {return 10;}
6. **public long Metoda(double d, bool b) {return 10;}**
7. public double Metoda(double d) {return 10;}
8. public double Metoda1(double d, bool b){return 10;}

odgovor: 3, 4, 6 i 2 – za 2 nisam siguran, ali tako je po svesci

komentar: Ne mogu se definisati jer imaju isti potpis. Da bi se mogle definisati, moraju imati različit potpis, odnosno ime metode kao i tip, redosled i broj ulaznih parametara.

29. Koje od sledećih metoda imaju isti potpis?

- a) void Metoda(double a){...}
- b) void Metoda(double b){...}
- c) int Metoda(double a){...}
- d) void Metoda(out double a){...}
- e) void Metoda(double a, int b){...}

odgovor: a, b, c

komentar: Da bi metode imale isti potpis one moraju da imaju isto ime, isti broj, tip i redosled parametara.

30. Šta se ispisuje na ekranu ?

```
class Program
{
    public void Izmeni(int a)
    {
        a = a + 10;
    }
}
```

```

}

public void Izmeni1(ref int a)
{
    a = a - 10;
}

static void Main(string[] args)
{
    Program pro = new Program();

    int ulaz = 20;
    pro.Izmeni1(ref ulaz);
    Console.WriteLine(ulaz);
}

```

odgovor: 10

komentar: Vršni se prenos parametara preko reference, tako da se u metodi Izmeni1() promeni vrednost tog parametra i kasnije ispiše.

31. Šta se ispisuje na ekranu ?

```

class Program
{
    public void Izmeni(int a)
    {
        a = a + 10;
    }

    public void Izmeni1(out int a)
    {
        a = 15;
    }

    static void Main(string[] args)
    {
        Program pro = new Program();

        int ulaz = 20;
        pro.Izmeni1(out ulaz);
        Console.WriteLine(ulaz);
    }
}

```

odgovor: 15

komentar: parametru a je dodeljena vrednost 15 pre nego što napusti metodu Izmeni1() i ta metoda služi samo za pribavljanje informacija.

32. Šta ispisuje sledeći kod:

```

class Primer{
static void Metoda(out int x)
{x--;}
static void Main()
{
int broj = 1;
Metoda(out broj);
Console.WriteLine(broj);
}
}

```

odgovor: Greška

komentar: Parametru x uz koga stoji out mora biti dodeljena vrednost pre nego što napusti metodu Metoda(); Nasa Metoda() služi samo za pribavljanje informacija od nje, jer uz njen parametar stoji out.

da je pisalo x=5; umesto x--;, kod bi se izvršio i pisalo bi 5

33. Šta ispisuje sledeći kod?


```

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);
    }
}

```

odgovor: Greška

komentar: Greška nastaje jer parametar p, koji je označen sa out mora da dobije vrednost (a pošto je za sada samo referenca, mora da se instancira) pre nego što izađe iz metode Povecaj().

Ispisati u metodi Povecaj(), prvu liniju koda: `p = new Proizvod();` sada program ispisuje 150.

34. Šta ispisuje sledeći kod?

```

class Program
{
    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);
    }
}

```

odgovor: 20

komentar: Promenljiva broj kao parametar ulazi u metodu A, ali usled toga što se vrši prenos parametra preko vrednosti, nakon izlaska iz metode A promenljiva broj i dalje ima vrednost 30, jer metoda A radi sa kopijom. Nakon toga promenljiva broj sa vrednošću 30 ulazi u metodu B i vrši se prenos preko reference. Metoda B umanjuje promenljivu za 10 i nakon toga kada se izvrši ispis, njena vrednost je 20.

35. Šta ispisuje sledeći kod:

```

class Automobil
{ public int regBroj;

public Automobil(int regBroj)
{regBroj = regBroj;}
}
...
Automobil a = new Automobil(222);
Console.WriteLine(a.regBroj);

```

odgovor: 0

komentar: U konstruktoru klase Automobil parametar se dodeli samom tom parametru, a ne polju klase. I kada izvršimo ispis tog atributa, prikaže se njegova default vrednost ,odnosno 0. **Potrebno je iskoristiti this, da bi kod radio. Dakle, u konstruktoru izmeniti kod da bude**

`this.regBroj = regBroj;`

ispisaće se 222

36. Šta se ispisuje?

```
class Krug {
    public int precnik;
    public Krug(int precnik) {
        precnik = precnik;
    }
}

...
Krug k = new Krug(2);
Console.WriteLine(k.precnik);
```

odgovor: 0

komentar: isti kao u prethodnom primeru

37. Šta ispisuje sledeći kod ?

```
class Trougao{
    Public int visina;
    public Trougao(int visina)
    {
        visina=100;
    } ...
    Trougao t= new Trougao(50);
    Console.Writeline(t.visina);
```

odgovor: 0

komentar: U klasi Trougao atribut visina nije inicijalizovan. U konstruktoru samom ulaznom parametru dodeljujemo vrednost 100, ali time samo menjamo njegovu vrednost u okviru konstruktora. Kada se napusti konstruktor, ta promena se ne vidi, jer smo izašli iz oblasti važenja te promenljive i nismo tu promenu sačuvali u atributu klase. **Potrebno je da izmenimo kod u `this.visina = visina;` da bi se ta promena videla u atributu visina klase Trougao.**

38. Šta ispisuje sledeći kod:

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

class B:A
{ publicB():base(5){Console.WriteLine("2");}
  B b = new b();}
```

odgovor: 1 2

komentar: Klasa B nasleđuje klasu A. Konstruktor klase B poziva konstruktor bazne klase (klase A) i prosleđuje mu parametar. Izvršava se naredba konstruktora bazne klase i nakon toga naredba konstruktora izvedene klase.

39. Šta ispisuje sledeći kod:

```
class A
{
    public A():this("b")
    {Console.WriteLine("c");}
}
public A(string i)
{Console.WriteLine(i);}
}
class B:A
{
    public B()
    {Console.WriteLine("a");}
}
....
B b = new B();
//A a = new A();
//A a = new B();
//A a = new B("M");
```

odgovor: bca //bc bca M

komentar: pravimo objekat izvedene klase B i pozivamo njegov konstruktor, on poziva neparаметarski konstruktor glavne klase, a on poziva parametarski konstruktor preko this i prosleđuje parametar, tako da se prvo izvršava naredba parametarskog konstruktora klase A, naredba neparаметarskog konstruktora klase A i na kraju naredba neparаметarskog konstruktora klase B. Provezhati ostale slučajeve.

40. Šta ispisuje sledeći kod:

```
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();
    }
}
```

odgovor: 132

komentar: Pročitati prethodni primer.

bez : this(1) u kodu ispisalo bi se 1 2

41. Šta se ispisuje?

```
class A {public A(int broj){ConsoleWrite("A");}}
class B:A { public B(){ConsoleWrite("B");}}
...
B b = new B();
```

odgovor: Greška

komentar: Nastaje greška usled toga što glavna klasa nema neparаметarski konstruktor koji konstruktor izvedene klase želi da pozove. Potrebno je u klasu A dodati

```
public A()
{
    Console.WriteLine("C");
}
ili ako želimo da pozovemo parametarski konstruktor koji postoji

public A():this(4)
{
    Console.WriteLine("C");
}
```

42. Da li je ispravan kod ?

```
class A
{
```

```

public A():base()
{
}
}

```

odgovor: da – može da se build-uje

43. Šta je rezultat izvršenja sledećeg koda ?

```

class A{
    protected void Ispis(){Console.WriteLine("Tekst");}
}
class B:A{}
...
B b = new B();
b.Ispis();

```

odgovor: ispisuje tekst, vi ste pogriješili :-P

komentar: Metoda Ispis() glavne klase ima modifikator pristupa protected, pa je nedostupna objektu izvedene klase.

44. Šta ispisuje sledeći kod:

```

class A
{
public static int broj;
}
...
A a = new A();
a.broj = 5;

```

odgovor: Greška

komentar: Atribut klase A broj je static, što znači da mu se pristupa preko klase, a ne preko instance. Tako da zbog toga javlja grešku.

Ispravno je: `A.broj = 5;`

45. Šta ispisuje sledeći kod?

```

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

```

odgovor: Greška

komentar: Metoda Kvadriraj() je static metoda i ne može se pozivati preko objekta, već samo preko klase.

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

46. Koja su tvrđenja tačna:

- a) Staticki članovi se mogu pozivati i preko klase i preko instance.
- b) Staticki članovi se mogu pozivati iako prethodnog nije kreirana instanca date klase
- c) Kada se statickom polju jednom dodeli vrednost ta vrednost se ne može više menjati
- d) Staticka polja se čuvaju u memoriji samo jednom bez obzira na to koliko ima instanci date klase.

47. Sta se ispisuje na ekranu?

```

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

```

```

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

```

odgovor: Greška

komentar: Metoda string.Format kao prvi parametar prima 2 član promenljive id, ali pošto je ta promenljiva sa istim nazivom imenovana u datom redu, prevodilac će prijaviti grešku.

Umesto dva poslednja reda ubaciti:

```

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

```

48. Koja je od navedenih naredbi ispravna?

```

abstract class Osoba
{
    public string ime;
    public string prezime;

    static void Main()
    {
    }
}

```

```

class Student : Osoba
{
}

```

Ponudjeno:

A. Osoba o = new Osoba();

B. Osoba o = new Student();

C. Student o = new Osoba();

odgovor: B

komentar: Nije moguće napraviti instancu apstraktne klase Osoba, a Student je klasa koja je nasledila klasu Osoba (izvedena je iz klase Osoba) tako da je Student Osoba i zbog toga je odgovor pod B.

49. Da li je sledeci kod ispravan?

```

abstract class Osoba
{
    protected string ime;
    protected string prezime;

    public abstract string DajPodatke();
}

class Student : Osoba
{
    string brojIndeksa;
}

```

odgovor: ne

komentar: u klasi Student nije reimplementirana (override) apstraktna metoda DajPodatke() ili nije ostavljena kao apstraktna.

50. Da li je sledeci kod ispravan?

```

sealed class Osoba
{
    public string ime;
    public string prezime;
}

class Student : Osoba

```

```
{  
    public string brojIndeksa;  
}
```

odgovor: ne

komentar: zapečaćena (sealed) klase se ne može naseđivati

51. Koji od sledeća dva koda je ispravan ?

1) class A{
 class B:A{
 A a = new B();

2) class A{
 class B:A{
 B b = new A();

odgovor: ispravan je 1 kod

komentar: Klasa B izvedena iz klase A, tako da B jeste tipa A i to može da se iskomplajlira – isti princip važi za Račun i TekućiRačun.

52. Šta ispisuje sledeći kod:

```
Try  
{  
    string a = "AAA";  
    try  
    {  
        int c = Convert.ToInt32(a);  
        Console.Write("F");  
    }  
    catch(DivideByZeroException dbze)  
    {  
        Console.WriteLine("dbze");  
    }  
    finally  
    {  
        Console.WriteLine("d");  
    }  
    Console.WriteLine("c");  
}  
catch(FormatException fe)  
{  
    Console.WriteLine("b");  
}  
finally  
{  
    Console.WriteLine("a");  
}
```

```
try  
{  
    string naziv = "FON";  
    try  
    {  
        int broj = Convert.ToInt32(naziv);  
        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);  
}
```

odgovor: dba

komentar: Prvo ulazimo u 1 try blok i u njegov try blok, gde pokušavamo da string konvertujemo u int. Tada nastaje izuzetak i odmah se skače na prvi catch blok, kako on ne može da obradi ovaj izuzetak koji je tipa FormatException, prelazi se na blok finally i ispisuje odgovarajuća poruka. Nakon toga se prelazi u sledeći catch blok i pošto on može da obradi izuzetak ispisuje se odgovarajuća poruka, nakon toga se prelazi u finally blok i ispisuje odgovarajuća poruka. Finally se uvek izvršava bez obzira da li je nastala greška i služi za oslobađanje resursa. Svaki try blok može imati samo 1 finally blok, a može imati više catch blokova, ali oni ne mogu hvatati istu vrstu izuzetaka. Uvek se traži najbliži catch blok koji može da obradi izuzetak. Treba poštovati redosled od posebnih ka opštijim izuzecima, npr.

1. DividedByZero

2. Exception

53. Sta se ispisuje na ekranu?

```
try
{
    string naziv = "FON";
    try
    {
        int broj = Convert.ToInt32(naziv);
        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);
}
```

odgovor: 5431

komentar: Ulazi se u prvi try blok i u njegov try blok. Pokušavamo da string konvertujemo u int, tako da tu nastaje izuzetak. Skače se na catch blok koji može da obradi svaki izuzetak i ispisuje se 5, izvršava se finally blok i ispisuje se 4 i nakon toga se ispisuje 3 i izvršava se finally blok koji odgovara prvom try-u i ispisuje se 1. Voditi računa o naredbama ispod ugnježenog try bloka, kao što je slučaj sa **Console.Write(3);**

54. Šta ispisuje sledeći kod:

```
class Broj
{
    public int vrednost = 1;
}
...
Broj[] niz = new Broj[3];
Console.WriteLine(niz[0].vrednost);
```

odgovor: Greška

komentar: Promenljiva niz je referencira na niz objekata tipa Broj. Kako je Broj referentnog tipa, njegova podrazumevana vrednost je null, tako da je niz u suštini niz sa 3 prazne reference. Sada kada želimo da pristupimo 1 članu tog niza, sa indeksom 0, javiće se null reference exception. Da bi mogli da radimo sa bilo kojim nizom koji sadrži referentne tipove potrebno je da se za svaku referencu napravi objekat. Kako vršimo ispis samo prvog člana ovog niza, potrebno je ubaciti deo koda.

```
niz[0] = new Broj();
```

55. Šta se ispisuje na ekranu?

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

odgovor: 1

komentar: Napravili smo matricu (pravougaoni niz – Rectangular Array) dimenzija 2 x 2. Nakon toga tržimo ispis elementa koji se nalazi u 2 redu i 2 koloni, jer se indeksiranje vrši od 0.

56. Šta ispisuje sledeći kod:

```
class A  
{  
    public virtual void Prikazi(){Console.WriteLine("a");}  
}  
class B:A  
{  
    public override void Prikazi() {Console.WriteLine("b");}  
}  
....  
A a = new A();  
a.Prikazi();  
a = new B();  
a.Prikazi();
```

odgovor: a b

komentar: Klasa A ima virtuelnu metodu, koju klasa B reimplementira(redefiniše) preko ključne reči override. Kako je B izvedeno iz A, prevodilac će prilikom prevođenja odrediti odgovarajući tip objekta i pozvati u skladu sa tim odgovarajuću metodu. On će izvršiti takozvano kasno povezivanje *late binding*. Preko operatora new i poziva konstruktora B(); prevodilac će znati da pozove metodu Prikazi() klase B.

57a. Šta ispisuje sledeći kod:

```
class A  
{  
    public void Prikaz() { Console.WriteLine("A");}  
}  
  
class B:A  
{  
    public void Prikaz(){Console.WriteLine("B"); }  
}  
  
A a = new A();  
a.Prikaz();  
  
A a1 = new B();  
a1.Prikaz();  
  
B b1 = new B();  
b1.Prikaz();  
  
//B b2 = new A();  
//b2.Prikaz();
```

odgovor: aab

komentar: Prilikom poziva nevirtuelne metode tip reference određuje koje metode mogu da se pozovu. Ukoliko komentar iz koda uklonimo, program neće moći da se izvrši jer je klasa B izvedena iz A.

57. Šta ispisuje sledeći kod:

```
class A{  
    public void Prikazi()
```



```
{Console.WriteLine("a");}
}
```

```
class B:A{
new public void Prikazi()
{ Console.WriteLine("b");}
}
```

....

```
A a = new A();
a.prikazi();
a = new B();
a.Prikazi();
B b = new B();
b.Prikazi();
```

odgovor: aab

komentar: Sa new mi definišemo potpuno novu metodu Prikazi() u klasi B i naglašavamo prevodiocu da ta nova metoda nije nastala redefinisanjem metode Prikazi() klase A. Prilikom poziva metode tip reference određuje koje metode mogu da se pozovu, dakle kao i kod nevirtuelnih metoda.

58. Šta ispisuje sledeći kod?

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

odgovor: 3 1

komentar: Poziv konstruktora iza operatora new određuje metode koje to klase će biti pozvane.

59. Da li je sledeći kod ispravan?

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

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

odgovor: ne

komentar: Operator + se ne može primeniti na u ovom slucaju jer redosled nije dobar. Prvo treba tipa int pa onda tipa broj.

60. Sta se ispisuje na ekranu?

```
class Broj
{
```

```

public int broj = 3;
public static int operator +(int a, Broj b)
{
    return a * b.broj; 5*3=15
}

public static int operator +(Broj b, int a)
{
    return a - b.broj;
}

static void Main()
{
    Broj broj1 = new Broj();
    int broj2 = 2;
    broj2 += broj1.broj; 2+3=5
    broj2 += broj1;
    Console.WriteLine(broj2);
}
}

```

odgovor: 15 (broj2=2 se sabira sa broj=3 => broj2=5, pa iz pretposlednjeg reda imamo int pa Broj, idemo na 1. operator gde je a=broj2=5, b=broj1=3, pa je broj2=15)

61. Šta će se ispisati 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);
    }
}

class Banka
{
    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.WriteLine(dg(0.01));
}
}

```

odgovor:

1

Novo stanje je 600

2

Novo stanje je 500

3

Nova provizija je 0.01

62. Sta se ispisuje na ekranu?

```

class Broj
{
    public int broj = 4;
    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);
    }
}

```

```

class Program
{
    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()
    {
        Broj broj1 = new Broj();
        broj1.Dogadjaj += Oduzmi;
        broj1.Dogadjaj += Dodaj;
        broj1.Dogadjaj += Pomnozi;
        broj1.Dogadjaj -= Oduzmi;
        broj1.Promeni(2);
        Console.WriteLine(vrednost);
    }
}

```

odgovor: 4 (x=2, idu redom operacije sa +=, a sa -= ne ide i ponistava, dakle samo "Dodaj" i "Pomnozi")

63. Sta se ispisuje na ekranu?

```
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 e, int f)
    {
        return e.broj * f;
    }
}

class Program
{
    public static int Oduzmi(Broj x, int y)
    {
        return x.broj - y;
    }

    static void Main()
    {
        Broj broj1 = new Broj();
        Broj.Delegat dg;
        dg = broj1.Dodaj;
        dg += broj1.Pomnozi;
        dg = Oduzmi;
        Console.WriteLine(dg(broj1, 1));
    }
}
```

odgovor: 2 (poslednji zapamcen dg je za metodu "Oduzmi" 3-1=2)

64. Sta se ispisuje na ekranu?

```
string tekst = "Naziv je:\t FON. ";
Console.Write(tekst);
tekst = @"Naziv je:\t FON. ";
Console.Write(tekst);
```

Ponudjeno:

- A. Naziv je: FON. Naziv je:\t FON.
- B. Naziv je:\t FON. Naziv je: FON.
- C. Naziv je:\t FON. Naziv je:\t FON.
- D. Naziv je: FON. Naziv je: FON.
- E. Prijavice se greska.

odgovor: A

komentar: koristimo verbatim (@) stringove, da bi prevodilac zanemario takozvane izlazne sekvence(escape sequence – počinju sa \ , najvažnije su \n – novi red, \t – horizontalni tabulator, \\ - obrnuta kosa crta, \r – prelazak na početak reda, \v – vertikalni tabulator).

primer:

```
string s = "c:\\Program Files\\Microsoft Office\\Office";
string s = @"c:\Program Files\Microsoft Office\Office";
```

Ovo je nepotpuno objašnjeno, ali je dovoljno da se razumes za test.

4. Sta se ispisuje na ekranu?

```

public partial class A
{
public int broj = 10;
}
public partial class A
{
public int broj = 5;
}
A a = new A();
Console.WriteLine(a.broj);

```

- a. 5
 - b. 10
 - c. ne vidim sta pise
- Trebalo bi da prijavi **gresku** jer ne moze dva broja!

7. Sta se ispisuje na ekranu?

```

public class Primer
{
static void Proba(TAdresa a)
{
a.broj +=10;
}
static void Proba(Ref TAdresa a)
{
a.broj -= 5;
}
static void Main(){
TAdresa adresa = new TAdresa();
Proba(adresa);
Proba(ref adresa);
Console.WriteLine(adresa.broj);
}
}

```

- a. 10
- b. 5**
- c. 0
- d. -10
- e. -5
- f. prijavice se greska

8. Sta se ispisuje na ekranu?

```

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

```

- a. 10
- b. 5
- c. 0

d. -10

e. -5

f. prijavice se greska

--nije nigde definisan a, pa ne znam da li je greska ili nije, mislim da je greska zbog out.

9. Sta se ispisuje na ekranu?

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

a. 10

b. 5

c. 0

d. -10

e. -5

f. prijavice se greska

-5 odgovor

10. Sta se ispisuje na ekranu?

```
public class A
{
    public void Prikazi() {
        Console.Write("1");
    }
}
public class B : A
{
    public new void Prikazi(){ Console.Write("2");
    }
}
A a1 = new A();
    a1.Prikazi();
A a2 = new B();
    a2.Prikazi();
B b1 = new B();
    b1.Prikazi();
```

a. 111

b. 112

c. 122

d. 121

e. 222

f. prijavice gresku

public class TAdresa

```
{
    public string adresa;
    public int broj;
```

```

}

class Program
{
    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.WriteLine("Isti"); //<-----
        }
        else
        {
            Console.WriteLine("Razliciti");
        }
    }
}

```

namespace nasledjivanja1

```

{
    public class A
    {
        public virtual void Prikazi()
        {
            Console.WriteLine("1");
        }
    }

    public class B : A
    {
        public new void Prikazi()
        {
            Console.WriteLine("2");
        }
    }

    class Program
    {
        static void Main(string[] args)
        {
            A a1 = new A();
            a1.Prikazi(); //1
            A a2 = new B();
            a2.Prikazi(); //1
            B b = new B();
            b.Prikazi(); //2
        }
    }
}

```

namespace nasledjivanje2

```

{
    public class A
    {
        public virtual void Prikazi()
        {
            Console.WriteLine("1");
        }
    }
}

```

```

public class B : A
{
    public override void Prikazi()
    {
        Console.WriteLine("2");
    }
}
class Program
{
    static void Main(string[] args)
    {
        A a1 = new A();
        a1.Prikazi(); //1
        A a2 = new B();
        a2.Prikazi(); //2
        B b = new B();
        b.Prikazi(); //2
    }
}
}

```

namespace nasledjivanje3

```

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

    public class B : A
    {
        public new void Prikazi()
        {
            Console.WriteLine("2");
        }
    }

    class Program
    {
        static void Main(string[] args)
        {
            A a1 = new A();
            a1.Prikazi(); //1
            A a2 = new B();
            a2.Prikazi(); //1
            B b = new B();
            b.Prikazi(); //2
        }
    }
}

```

namespace nullable

```

{
    class Program
    {
        static int broj1 = 5;
        static Nullable<int> broj2 = 10;
        static void Main(string[] args)
        {
            //broj1 = broj2; ne moze

```



```

    broj2 = broj1; //moze
    //broj1 = null; ne moze
    broj2 = null; //moze
}
}
}

```

Sta 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

Da li je sledeci kod ispravan?

```

Class Broj {public int Broj=3;
Public static int operator (broj a,int b) {
Return a
broj*b; }}
...
Broj Broj 1=new Broj();
Int a=5+Broj 1: }

```

NIJE

Zaokruziti tacno

```

Public class Racun{ }
TekuciRacun:Racun{ }

```

- a) TekuciRacun tr=new Racun();
- b) **Racun r=new Racun ();**
- c) **Racun r=new TekuciRacun();**

Imas u pitanjima

E SAD TI PRICAM DODATKE KOJE JE MENI RADOVAN ISPRICAO, MISLIM DA JE PRVO PITANJE, ONO SA ADRESAMA TACAN ODGOVOR 25, PITANJE GDE PISE PARTIAL JE TACAN ODGOVOR-NE,A PITANJA NA PRVOJ STRANI SA REF, MISLIM DA SU 5, 6,7 ZAOKRUZI SVE -5, POSTO JE NA DVA MESTA -5. A NA JEDNOM 10, I OBAVEZNO REFERENTNE TIPOVE NAUCI 😊

DODATAK:

```
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();
    }
}
```

Odgovor: 132

bez : this(1) odgovor: 12

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

class B:A
{
    new public void prikazi()
    {
        Console.WriteLine("b");
    }

    static void Main(string[] args)
    {
        A a = new A();
        a.prikazi();
        B b = new B();
        a.prikazi();
    }
}
```

Odgovor: aa

```

-----
class A
{
    public virtual void prikazi()
    {
        Console.Write("a");
    }
}
class B : A
{
    public override void prikazi()
    {
        Console.Write("b");
    }

    static void Main(string[] args)
    {
        A a = new A();
        a.prikazi();
        a = new B();
        a.prikazi();
    }
}

```

Odgovor: ab

Sa foruma:

1. Apstraktne metode se mogu deklarirati u okviru:

a) obične klase, b) apstraktne klase, c) interfejsa, d) sva tri navedena
(trebalo bi da je b i c)

2. Sta se dobija izvršavanjem sledeceg koda:

```

public class Usluga
{
    public int cena=200;
}
class Class
{
    static void Uplati(out Usluga u, int iznos)
    {
        u.cena -= iznos;
    }
    static void Main(string[] args)
    {
        Usluga u = new Usluga();
        Uplati(out u,50);
        Console.WriteLine(u.cena);
    }
}

```

a) ispisuje se 200, b) ispisuje se 150, c) prijaviće se greška
(definitivno greška, tj. c)

3. Sta se ispisuje izvršavanjem sledeceg koda:

```

public class A

```

```

{
    public A():this("b")
    {
        Console.WriteLine("a");
    }
    public A(string i)
    {
        Console.WriteLine(i);
    }
    public virtual void Draw()
    {
        Console.WriteLine("Drawing A!");
    }
}

```

```

public class D:A
{
    public D()
    {
        Console.WriteLine("c");
    }

    static void Main()
    {
        D d = new D();
    }
}

```

a)bac, b)ba, c)c, d)prijavljuje gresku
(ispisuje a)bac

13.Referentni se cuva?

Resenje: HEAP

13b.Vrednosni se cuva?

Resenje: STACK

14.{int i = 0}

{int i =20;}

Resenje: NE

15.Kojim resosledom se ispisuje u V Studio?

1.Class 2.Namespace 3.Method

Resnje: 2-1-3

16.Rezultat sledeceg koda je: 1) 50 2) 100

```

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

```

}

Resenje: 100

17.Private :

1.samo u okviru te klase 2. U okviru te i nadredjene klase 3. Pristup nije ogranicen

Resenja: 1

18.Koja su tvrdjenja tacna:

a) Staticki clanovi se mogu pozivati i preko klase i preko instance.

b) Staticki clanovi se mogu pozivati iako prethodno nije kreirana instanca date klase

c) Kada se statickom polju jednom dodeli vrednost ta vrednost se ne moze vise menjati

d) Staticka polja se cuvaju u memoriji samo jednom bez obzira na to koliko ima instanci date klase.

Resenje: b) i c)

19.Metoda koja vraca vrednost necega mora imati rezervisanu rec return:

1.ne mora navesti 2.mora tacno jednom 3.bar jednom

Resenje: 3

21.Kod je:

Enum Podne

```
{ dobroJutro, Podne,
```

```
Vece}
```

```
0 – 1 – 2 => Podne = 1;
```

Resenje je 1

22.Da li je kod ispravan:

```
Double broj1 = 1.3;
```

```
Double broj2 = 2.4;
```

```
Int zbir = broj1 + broj2;
```

Resenje: NE

23.Struktura se moze naslediti?

Resenje: NE

24a.Vrednost konstante :

1)U vreme kompajliranja 2)U vreme izvrsavanja

Resenje: 1

24b. Vrednost promenljive :

1)U vreme kompajliranja 2)U vreme izvrsavanja

Resenje: 2

26.Sta uzima vrednost u C#?

1.Klasa 2.Metoda 3.Konstruktor 4.Struktura

Resenje: 4

27.Nizovi gde su ponudjena resenja ISTI i RAZLICITI treba da bude ovo drugo.

28.Virtual ---- override Resenje: 24

29.Kada je drug public onda je resenje 22

.

31. Suma = 6

```
32.string str1="java";
string str2=str1;
string str3=str2;
str2="c#";
Console.WriteLine(str3);
```

Resenje: Java

```
35.
string s= "ne";
switch (s) {
case "ne":
case "da":
Console.WriteLine("da");
break;}
-----
```

Resenje: Da

Bilo je nullable

Bilo je sa @ tj. verbatim stringovi

Bile su dve klase gde jedna nasledjuje drugu, a prva ima this(int i), tako nesto

Bilo je koje metode mogu da stoje zajedno (preklapanje metoda)

Bilo je teorijsko pitanje: Koju su referentni tipovi podataka, pa naborjano

Bilo je teorijsko u vezi statica, da moze da se koristi bez instanci

Bilo je vezano za pravljenje objekata klasa, pa je u kodu a1=a2, pa promene a2.vrednost, mislim da su adrese bile u pitanju, pa je pitanje da li su isti ili razliciti (isti su jer pokazuju na istu lokaciju)

Bilo je partial, dve klase sa istim imenom, obe partial, i obe imaju isto polje definisano, pitanje je dal je ispravno

Bila je konverzija sa castom double d=2.7; a=int(d); koliko je a?

Bilo je sa konstantama

Bilo je sa enumeracijama, enum Semafor {crveno, zuto = 2, zeleno} pitanje je koliko je zeleno

Sa excetionom je bilo Convert.ToInt32("FON")

Bio je niz= new int [,] {{4,1} {2,1}} napisati niz[1,1]

Bili su preklopljeni operatori

Bio je jedan poduzi kod iz delegata i jedan isto poduzi iz dogadjaja (mislim da je identican taj kod bio i u junu)

Bilo je nesto u kodu int a = 2+s.vrednost; sto je sabiranje intigera

Pjp – pismeni u septembru

1. Definisan je dvonizni niz, napunjen vrednostima, odstampaj element na poziciji [1,1]
2. Virtual/ovveride
3. Virtual metoda/new metoda
4. Obicna metoda/new metoda i ispis

5. Referentni tipovi da se prepoznaju (neki su predefinisani a neki korisnicki definisani)
6. Apstraktna klasa, obicna je nasledjuje i istanciranje. Trazi se da se ispise sta je ispravno (aa, ba, ab) 1)aklasa o = new aklasa() 2)oklasa a = new aklasa() 3) aklasa a = new oklasa()
7. Data je obicna klasa i jos jedna koja je nasledjuje trazi se isto kao i predhodno da se napise sta je ispravno
8. Da li je ispravan kod: public class A {...}; public A():base(){ } – nije, ne secam se zbog cega, mislim da je zato sto nisu pozvane metode bazne klase
9. Da li moze sealed klasa da se nasledi – ne moze
10. Public class A{ public A(int i)} console.wl (...) public class B{public B(); } – mislim da je u pitanju da li su iste ili tako nesto – pitati didi.
11. Try {string “FON” convert.toint c.wl(7)
Catch devideByZero... (6)
Catch (exception) (5)
Finally (4)
Catch FormatException (3)
Finally (2) C.wl(...) } - priavice gresku posto je prvo postavljen specificniji izuzetak, a posle opstiji
12. Definisan je niz od 2 elementa, nisu popunjene vrednosti, pozvan je element na poziciji [0];
13. Da li je dobro definisan operator...
14. Sta je rezultat izvršenja operatora...
15. Partial class A..., partial class B – nemam pojma sta u vezi toga, da li je bio mozda neki ispis ili neko nasledjivanje
16. Internal protected – gde se moze pozvati (praktican primer)
17. Koje metode su identicne/iste kao i metoda public int A(char a, string b)
18. 18-22 vredonosni i referentni tipovi, prenosenje vrednosti (string, int, objekti)
23. 23-25 delegati i event-ovi
26. this (2) c.wl(“3”) else 1 – ima u igorovim beleskama sto ih je slikao. Rezultat mislim – 231
27. enum
28. sta je tacno u vezi static metode – videti igorovu sliku
29. konstanti je dodeljena vrednost. Kasnije se postavlja druga vrednost – greska
30. string a =”fon”;
Try(swich a){case “fon” c.wl(“fon”)
Case “etf”:
Case “pmf” c.wl(“etf”);
31. double i = 3.7; (int)i =? (ponudjeno 0, 3, 4, greska)
- 32.dupla petlja
While (i<3, i++){ suma ++ if (j==2) break } suma ++} c.wl(suma); - mislim da je 7, proveriti
- 33.out parametar – preko metode - otprilike isto kao i do sada
- 34.ref parametak preko metode. Oprilike isto kao i do sada
35. koja od klasa moze naslediti mislim partial A(stringa, int b) – pa su ponudjenepublic, protected, pa klasa koja ne nasledjuje parametre,...
36. nisam siguran - nasledjuje se bazna klasa ili apstraktna klasa, a ne nasledjuju se apstraktne metode ili se nasledjuju privatne metode - greska