



Važna napomena: Za kompletan rad je dovoljna jedna ispitna sveska. Potrebno je podeliti je po oblastima tako što ćete iz sredine sveske da izvučete potreban broj listova za zadatke iz Jave i potreban broj listova za zadatke iz C#-a, a u ostatku sveske pisati odgovore na pitanja (da bi svaka oblast mogla da se pregleda posebno). Na vrhu svakog od listova napisati ime, prezime i broj indeksa. Na kraju predati sve zajedno.

Pitanja

1. Objasniti razliku između jezika sa slabom i jakim tipizacijom.
2. Pokazati na primeru definiciju i poziv metode sa izlaznim parametrom u programskom jeziku C#.
3. Definicija generičke metode u programskom jeziku C#.
4. Pokazati kroz primer sinhronizaciju niti u Javi korišćenjem semafora.
5. Objasniti pojam „čiste funkcije“ u funkcionalnom programiranju.
6. Objasniti pojam dinamičkih web stranica i navesti tehnologije i programske jezike koji se koriste za kreiranje dinamičkih web stranica.

Zadaci Java

7. Na programskom jeziku Java implementirati metod sa jednim celobrojnim argumentom d koji kreira dvodimenzionalno polje sa $2 \times d$ vrsta, takvo da svaka neparna vrsta sadrži podatak o dužini sledeće, a parne vrste imaju slučajnu dužinu između 1 i d i popunjene su proizvoljnim vrednostima. Ne rasipati memorijski prostor.
8. Na programskom jeziku Java implementirati void funkciju koja međusobno razmenjuje vrednosti dvema celobrojnim promenljivama. Implementirati ovu funkciju na sve moguće načine.
9. Da li će sledeći kod raditi bez grešaka/upozorenja, ili će kompajler vratiti upozorenje (ili više njih), ili grešku (ili više njih)? Ukoliko će vratiti upozorenja/greške, navesti koje (nije potrebno napisati doslovno kao kompajler, dovoljno je govornim jezikom opisati upozorenje/grešku).

```
public class Klasa <T> {  
    private T argument;  
    public Klasa(T t) {  
        argument = t;  
    }  
    public Klasa() { }  
    public T getArg() {  
        return argument;  
    }  
}  
  
public class MainClass {  
    public static void main(String[] args) {  
        Klasa k1 = new Klasa();  
        Klasa <Float> k2 = k1;  
    }  
}
```

Zadaci C#

10. Koji je rezultat izvršenja sledećeg programa (tekst zadatka se nastavlja na sledećoj strani):

```
static void Main(string[] args)  
{  
    Prognoza p = Prognoza.Oblaci | Prognoza.Kisa | Prognoza.Vetar;  
    if ((p & Prognoza.Sunce) != 0)  
        Console.WriteLine("Sunčano je.");  
    if ((p & Prognoza.Oblaci) != 0)  
        Console.WriteLine("Oblačno je.");  
    if ((p & Prognoza.Kisa) != 0)  
        Console.WriteLine("Pada kiša.");  
    if ((p & Prognoza.Vetar) != 0)  
        Console.WriteLine("Duva vetar.");  
    if ((p & Prognoza.Sneg) != 0)  
        Console.WriteLine("Pada sneg.");  
}
```

a) ako je enumeracija Prognoza definisana kao:

```
enum Prognoza { NemaPrognoze = 0, Sunce = 1, Oblaci = 2, Kisa = 4, Vetar = 8, Sneg = 16 }
```

b) ako je enumeracija Prognoza definisana kao:

```
enum Prognoza { NemaPrognoze, Sunce, Oblaci, Kisa, Vetar, Sneg }
```

11. Koji je rezultat izvršenja sledećeg programa:

```
class Pravougaoounik
{
    protected float a, b;
    public Pravougaoounik(float a, float b)
    {
        Console.Out.WriteLine("Pravougaoounik "
            + a + "x" + b);
        this.a = a;
        this.b = b;
    }
    public virtual float Povrsina
    {
        get { return a * b; }
    }
}
```

```
class Kvadar : Pravougaoounik
{
    private float c;
    public Kvadar(float a, float b, float c)
        : base(a, b)
    {
        Console.Out.WriteLine("Kvadar " + a
            + "x" + b + "x" + c);
        this.c = c;
    }
    public float Povrsina
    {
        get { return 2 * (a*b + b*c + a*c); }
    }
    public float Zapremina
    {
        get { return Povrsina * c; }
    }
}
```

```
class Program
{
    static void Main(string[] args)
    {
        Pravougaoounik k = new Kvadar(1, 2, 3);
        Console.WriteLine("Površina je: "
            + k.Povrsina);
        Console.WriteLine("Zapremina je "
            + ((Kvadar)k).Zapremina);
    }
}
```

12. Prilikom obračuna plata radnicima u nekoj firmi, podaci iz tabele su pogrešno snimljeni u tekstualni fajl tako da su podaci o nekim radnicima ponovljeni nasumično. Podaci o jednom radniku se mogu ponavljati na više mesta u fajlu, ali jedan zapis o jednom radniku se nalazi uvek u susednim linijama teksta. U svakom zapisu za jednog radnika uvek je snimljen najpre JMBG, zatim ime, prezime pa na kraju plata. Primer teksta:

```
1234567890123
Petar
Petrović
95000.0
9876543210123
Jovana
Jovanović
105000.0
...
1234567890123
Petar
Petrović
95000.0
```

Implementirati klasu Radnik sa potrebnim atributima i u njoj sledeću statičku metodu koja čita podatke iz tog tekstualnog fajla. Metoda vraća instancu generičke klase Dictionary koja sadrži podatke o svakom radniku tačno jednom. Ključ u rečniku je JMBG radnika. Osigurati zatvaranje toka podataka kako god da se završi izvršenje metode (i u slučaju da dođe do izuzetka prilikom rada metode).

```
public static Dictionary<long, Radnik> ProcitajFajl(String putanjaDoFajla)
{
    //dodati potreban kod
}
```