

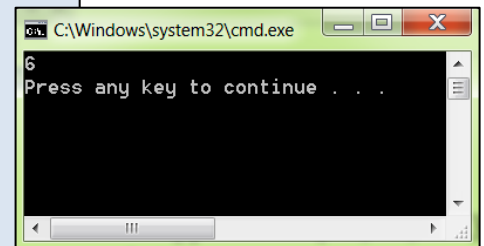
ИТЕРАТИВНА НАРЕДБА FOREACH

foreach је итеративна наредба, слична наредби **for** и увек се користи када је потребна итерација кроз колекцију, нпр. све елементе низа.

1. **Задатак:** Написати програм за израчунавање суме елемената низа, преко наредбе **foreach**.

```
using System;

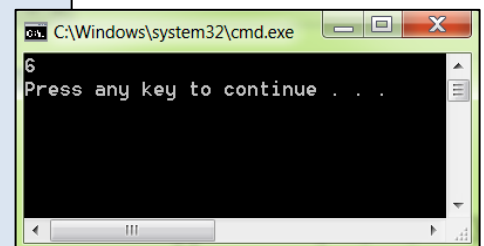
class Program
{
    static void Main()
    {
        int[] niz = { 1, 2, 3 };
        int suma = 0;
        foreach (int x in niz) suma += x;
        Console.WriteLine(suma);
    }
}
```



2. **Задатак:** Написати програм за израчунавање суме елемената низа, преко наредбе **for**.

```
using System;

class Program
{
    static void Main()
    {
        int[] niz = { 1, 2, 3 };
        int suma = 0;
        for (int i = 0; i < niz.Length; i++) suma += niz[i];
        Console.WriteLine(suma);
    }
}
```



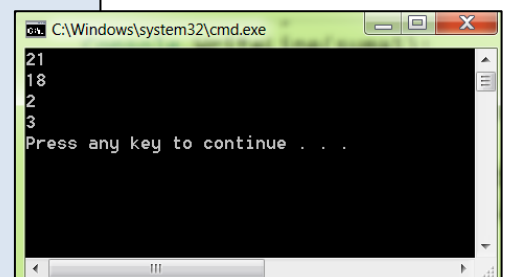
3. **Задатак:** Написати програм за израчунавање суме елемената дводимензионалног низа, преко наредбе **for** и **foreach**.

```
using System;

class Program
{
    static void Main()
    {
        //matrica 2 vrste 3 kolone
        int[,] a = new int[2,3]{{1,2,3},{4,5,6}};
        int suma1 = 0;
        foreach (int x in a)
            suma1 += x;
        Console.WriteLine(suma1);

        int[,] b = new int[2,2] { { 3, 4 }, { 5, 6 } };
        int suma2 = 0;
        for (int i = 0; i < b.GetLength(0); i++)
            for (int j = 0; j < b.GetLength(1); j++)
                suma2 += b[i, j];
        Console.WriteLine(suma2);

        //objasenje za GetLength
        Console.WriteLine(a.GetLength(0)); // broj vrsta
        Console.WriteLine(a.GetLength(1)); // broj kolona
    }
}
```



4. Написати програм који исписује колико смо у програму креирали објеката тј. фигура (правоугаоник и круг) и које су то фигуре.

```
using System;

public abstract class Figura
{
    public static int BrojFigura = 0;
    public Figura() { BrojFigura++; }
}

public class Pravougaonik: Figura{}
public class Krug : Figura {}

class Program
{
    static void Main()
    {
        Pravougaonik pr1 = new Pravougaonik();
        Pravougaonik pr2 = new Pravougaonik();
        Krug kr1 = new Krug();

        Console.WriteLine("Broj figura=" + Figura.BrojFigura + "\n");

        Figura[] niz = new Figura[Figura.BrojFigura];
        niz[0] = pr1; niz[1] = pr2; niz[2] = kr1;

        Console.WriteLine("Prikaz figura: 1 nacin");
        foreach (Figura el in niz) Console.WriteLine(el);

        Console.WriteLine("\nPrikaz figura: 2 nacin");
        for (int i = 0; i < niz.Length; i++) Console.WriteLine(niz[i]);

        // foreach (el in niz) Console.WriteLine(el);
        //foreach (int i in niz) Console.WriteLine(niz[i]); //nije int vec Figura
    }
}
```

```
C:\Windows\system32\cmd.exe
Broj figura=3

Prikaz figura: 1 nacin
Pravougaonik
Pravougaonik
Krug

Prikaz figura: 2 nacin
Pravougaonik
Pravougaonik
Krug
Press any key to continue . . .
```

АНАЛИЗИРАТИ ПРИМЕРЕ И УПИСАТИ ТАЧНЕ ОДГОВОРЕ:

224. Дати су делови кода у програмском језику С# који треба да рачунају збир елемената матрице **a**, декларисане на следећи начин: `int[,] a = new int[10, 10]`. Анализирајте дате кодове и процените који од предлога је тачан. Заокружити бројеве испред очекиваних одговора:

1. `int sum = 0;`
`for (int i = 0; i < b.Length; i++)`
`for (int j = 0; j < b[i].Length; j++)`
`sum3 += b[i][j];`
2. `int sum = 0;`
`foreach (int x in a) sum1 += x;`
3. `int sum = 0;`
`for (int i = 0; i < a.GetLength(0); i++)`
`for(int j=0; j<a.GetLength(1); j++)`
`sum2 += a[i,j];`
4. `int sum = 0;`
`foreach (int[] vrsta in b)`
`foreach (int el in vrsta)`
`sum4 += el;`

219. Дат је код програма у програмском језику C# који декларише два објекта типа **Pravougaonik** и један објекат типа **Krug**. Класе **Pravougaonik** и **Krug** наслеђују апстрактну класу **Figura** и обе класе имају методу **ToString**. Код програма треба допунити наредбом која ће исписивати низ **Figura[]** декларисан у коду програма.

```

1. public abstract class Figura {
2.     protected static int brojFigura=0;
3.     public static int BrojFigura {get{return brojFigura;}}
4.     public Figura() { brojFigura++; }
5.     public abstract double površinaFigure();
6.     public override string ToString(){return "Figura ";}
7. }
...
1. Pravougaonik pr1 = new Pravougaonik();
2. Pravougaonik pr2 = new Pravougaonik(3, 2);
3. Krug krl = new Krug(2);
4.
5. Console.WriteLine("Broj figura="+Figura.BrojFigura+"\n");
6. Figura[] niz = new Figura[Figura.BrojFigura];
7. niz[0] = pr1; niz[1] = pr2; niz[2] = krl;
8.
9. Console.WriteLine("Prikaz figura:");
10. _____

```

Заокружити бројеве испред одговора који представљају могуће начине приказа низа фигура:

- ```

1. foreach(Figura el in niz) Console.WriteLine(el);
2. for(int i=0; i<niz.Length; i++) Console.WriteLine(niz[i]);
3. foreach (el in niz) Console.WriteLine(el);
4. foreach (int i in niz) Console.WriteLine(niz[i]);

```

2

Треба уочити да и не знамо начин функционисања задатка можемо одредити тачне одговоре. Одговори који нису тачни су под 3 и 4. Зашто?

**Одговор под 3:** помоћна променљива нема тип. У овом примеру то је **Figura**.

**Одговор под 4:** **niz** је низ објеката типа **Figura**, онда је логично да променљива **i** може бити типа **Figura** а не **int** као што је наведено.