ПРОСЛЕЂИВАЊЕ АРГУМЕНАТА ПО ВРЕДНОСТИ И РЕФЕРЕНЦИ

Аргументи или **стварни параметри** су конкретне вредности које се задају код позивања методе. У наведеном примеру то је број **8**.

Параметри или **формални параметри** немају конкретну вредност већ указују на тип вредности. У наведеном примеру **int x**

```
class Program
{
        static void Main(string[] args)
        {
            unos(8);                    // 8 je argument

        }

        static void unos(int x) {       // x je parametar
            ...
        }
}
```

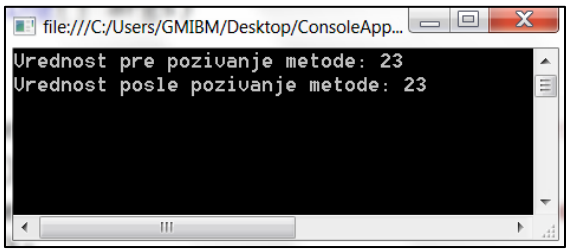
Прослеђивање аргумената по вредности

Када се променљива вредносног типа (int, float, double, struct, bool …) преноси методи, у ствари се преноси **копија променљиве**.

```
using System;

class Program
    {
        static void Main(string[] args)
        {
            int i = 23;
            Console.WriteLine("Vrednost pre pozivanje metode: {0}", i);
            inkrementacija(i);  // pravi kopiju i
            Console.WriteLine("Vrednost posle pozivanje metode: {0}", i);
            Console.ReadKey();
        }

        static int inkrementacija(int i) {
            i = i + 1;
            return i;
        }
    }
```



У примеру, променљива **i** не мења садржај, пошто је променљива **i** у методи **Main** и методи **inkrementacija** на различитим локацијама у меморији.

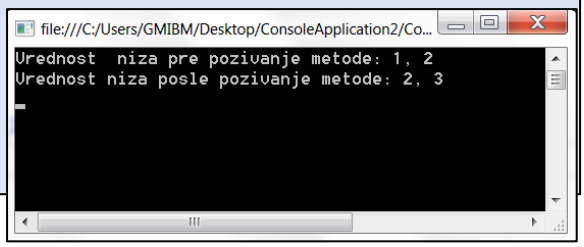
Прослеђивање аргумената по референци

Променљива референтног типа (низови, стрингови, класе, …)не садржи податке директно, већ садржи референцу на податке. Када се аргумент референтног типа преноси методи, могуће је променити податке на које показује референца.

```
using System;

class Program
    {
        static void Main(string[] args)
        {
            int[] niz = new int[] { 1, 2 };
            Console.WriteLine("Vrednost  niza pre pozivanje metode: {0}, {1}", niz[0], niz[1]);
            inkrementacija(niz);
            Console.WriteLine("Vrednost niza posle pozivanje metode: {0}, {1}", niz[0], niz[1]);
            Console.ReadKey();
        }

        static void inkrementacija(int [] niz) {
            niz[0] = niz[0] + 1;
            niz[1] = niz[1] + 1;
        }
    }
```



ПРОМЕНЉИВЕ УНУТАР МЕТОДЕ

Променљиве декларисане унутар једне методе су локалне променљиве и не могу се користити у телу неке друге методе. Дакле, променљиве унутар једне методе су независне од променљивих које су декларисане унутар других метода, чак иако имају иста имена.

150.	Дати су типови променљивих у програмском језику C#. Како се назива променљива која је дефинисана унутар неког метода: 1. Глобална променљива 2. Статичка променљива 3. Блоковска променљива 4. Локална променљива	**1**

За наведене примере дати тачан одговор:

154.	Дат је део кода који је написан на C# програмском језику. Одредити шта ће бити на излазу.
	`classTest`
	`{`
	`publicstaticvoid Main(string[] args){`
	`Test p = newTest();`
	`p.start();`
	`}`
	`void start() {`
	`bool b1 = false;`
	`bool b2 = fix(b1);`
	`Console.WriteLine(b1 + " " + b2);`
	`}`
	`bool fix(bool b1){`
	`b1 = true;`
	`return b1;`
	`}`
	`}`
	1. true true
	2. false true
	3. true false
	4. false false
		1

НАПОМЕНА: Логички типови података су вредносни типови података.

158.	У програмском језику C# дефинисана је метода са аргументом низовног типа. Одредити шта се тачно преноси том методом: 1. Копија датог низа 2. Копија првог елемента датог низа 3. Референца на дати низ 4. Дужина датог низа	**1**

НАПОМЕНА: Метод са аргументима вредносног типа преноси копију аргумената, док метод са аргументима референтног типа преносе референцу аргумената.

НАПОМЕНА: 168. Питање, већ сте преписали и **нема потребе поново да преписујете**. Знамо да је одговор под 2, зато што метод fun() није дефинисан да буде статички. Ако претпоставимо да је метод fun() статички, **шта ће програм да прикаже**?

168.	Дат је код програма у програмском језику C#. Анализирати дати код и проценити његову тачност:
	
	`namespace TestPrimer {`
	`classProgram {`
	`staticvoid Main(string[] args) {`
	`int n = 2;`
	`fun(n);`
	`Console.WriteLine("n je " + n);`
	`}`
	`void fun(int n) { n++; }`
	`}`
	`}`
	
	1. Програм има грешку, јер метод fun() не враћа ниједну вредност.
	2. Програм има грешку, јер метод fun() није дефинисан да буде статички.
	3. Програм приказује 1 на екрану.
	4. Програм приказује 2 на екрану.
	5. Програм приказује 3 на екрану.
		1

182.	Дат је код програма у програмском језику C#. Анализирати програм и одредити шта се приказује на екрану као резултат његовог извршавања.
	
	`namespace TestPrimer {`
	`classProgram {`
	`staticvoid Main(string[] args){`
	`int[] x = { 0, 1, 2, 3, 4 };`
	`Inc(x);`
	`int[] y = { 0, 1, 2, 3, 4 };`
	`Inc(y[0]);`
	`Console.WriteLine(x[0] + " " + y[0]);`
	`}`
	`publicstaticvoid Inc(int[] a){`
	`for (int i = 0; i < a.Length; i++) a[i]++;`
	`}`
	`publicstaticvoid Inc(int n) { n++; }`
	`}`
	`}`
	
	1. Порука о грешци
	2. 1 0
	3. 2 2
	4. 2 1
	5. 1 1
		2

201. Дат је део кода који је написан на C# програмском језику. Анализирати и одредити шта ће се приказати на излазу извршавањем овог кода:

```
classPassA{
publicstaticvoid Main(string[] args){
PassA p = newPassA();
        p.start();
    }
publicvoid start() {
long[] a1 = {3,4,5};
long[] a2 = fix(a1);
Console.Write(a1[0] + a1[1] + a1[2] + " ");
Console.WriteLine(a2[0] + a2[1] + a2[2]);
    }
long[] fix(long[] a3){
        a3[1] = 7;
return a3;
    }
}
```

1. 12 15
2. 15 15
3. 345 375
4. 375 375

2

203. Дат је део кода који је написан на C# програмском језику. Анализирати код и одредити шта ће бити на излазу:

```
classPassS{
publicstaticvoid Main(String[] args){
PassS p = newPassS();
        p.start();
    }
void start(){
String s1 = "slip";
String s2 = fix(s1);
Console.WriteLine(s1 + " " + s2);
    }
String fix(String s1){
        s1 = s1 + "stream";
Console.Write(s1 + " ");
return"stream";
    }
}
```

1. slip stream
2. slipstream stream
3. stream slip stream
4. slipstream slip stream

2