

Показивачи

Показивач (pointer) је промењива чија вредност је адреса неке друге промењиве тј адреса меморијске локације.

Као код било које промењиве или константе, и показивач се мора декларисати пре него се користи за смештање адресе неке промењиве.

Општа синтакса декларације показивача:

```
tip* imePokazivaca;
```

Пример:

```
int* ip;    //pokazivac na promenjivu sa celobrojnomo vrednosti
double* dp; //pokazivac na promenjivu sa double vrednosti
float* fp;  //pokazivac na promenjivu sa float vrednosti
char* cp;   //pokazivac na promenjivu sa char vrednosti
int** p;    //pokazivac na celobrojni pokazivac sa celobrojnomo vrednosti
int*[] p;   //1D niz pokazivaca na celobrojne vrednosti
```

Овде се користи оператор дереференцирања (\*), док се оператор референцирања (&) користи за добијање меморијске адресе на коју указује промењива:

```
tip* imePokazivaca = &identifikatorPromenjive;
```

Тип података постављен пре \* се назива референтни тип.

Пример:

```
using System;
namespace Proba
{
    internal class Program
    {
        static unsafe void Main(string[] args)
        {
            int x = 20;
            int* xp = &x;
            Console.WriteLine($"Podatak je: {x}");
            Console.WriteLine($"Adresa je: {(int)xp}");
            Console.WriteLine($"Podatak je i: {xp->ToString()}");
        }
    }
}
```

Даје:

Podatak je: 20

Adresa je: 17821928

Podatak je i: 20

Декларисање више показивача истовремено: `int* p1, p2, p3;`

Небезбедан контекст кода

Да би се могао стартовати unsafe код под VS: двоструки клик на ставку Properties са листе компоненти под Solution Explorer.

Тако се отвара екран са картицом под називом активног пројекта; клик на Build ставку.

Клик на опцију Allow unsafe code.

Најчешће су кодови у C# верификовано безбедни што значи да .NET алати гарантују сигурност написаног кода.

То значи да безбедан код не користи показиваче (не приступа меморији директно) већ користи објекте.

C# подржава небезбедан контекст унутар којег се може писати неверификовани код, што значи да се могу користити показивачи, приступити слободним блоковима меморије и позивати методе коришћењем показивача.

Пример:

```
using System;
namespace Proba
{
    internal class Program
    {
        static void Main(string[] args)
        {
            int[] a = new int[5] { 10, 20, 30, 40, 50 };
            unsafe
            {
                fixed (int* p = &a[0])
                {
                    int* p2 = p;
                    Console.WriteLine(*p2);
                    p2 += 1;
                    Console.WriteLine(*p2);
                    p2 += 1;
                    Console.WriteLine(*p2);
                    Console.WriteLine("-----");
                    Console.WriteLine(*p);
                    *p += 1;
                    Console.WriteLine(*p);
                    *p += 1;
                    Console.WriteLine(*p);
                }
            }
            Console.WriteLine("-----");
            Console.WriteLine(a[0]);
        }
    }
}
```

Даје:

10

20

30

-----

10

11

12

-----

12

У примеру се користе `unsafe` службена реч и `fixed` исказ, и показују како се инкрементира показивач.

Прво се са `fixed (int* p = &a[0])` фиксира објекат у меморији као и сам показивач на први елемент у низу `a`.

Пошто је показивач `p` фиксиран, мора се креирати други показивач (`p2`) који ће показати инкрементацију.

Са линијом `int* p2 = p;` се обезбеђује да показивач `p2` указује на садржај који се налази у показивачу `p`, а то је меморијска локација првог елемента у низу `a`.

Инкрементацином показивача `p2` (`p2 += 1;`), указује се на другу меморијску локацију која је за 4 бајта удаљена од претходне, пошто су у меморијским локацијама смештени подаци типа `int` који заузимају 4В.

Инкрементацијом `*p += 1;` се повећава садржај на који указује `*p` а то је вредност која се налази у првом елементу низа `a` (са 10 на 11).