

Оператори и искази код показивача

оператор / исказ	употреба
*	добивање променљиве на коју указује показивач
&	враћа адресу променљиве
->	приступа члану структуре преко показивача
[]	индексирање показивача
++, --	инкрементира и декрементира показивач
+, -	изводи адресну аритметику
==, !=, >, >=, <, <=	упоређује показиваче
stackalloc	приступ меморији на стеку
fixed	привремено фиксира променљиву да би се добила њена адреса

Пример:

```
using System;
namespace Proba
{
    internal class Program
    {
        static void Main(string[] args)
        {
            unsafe
            {
                int broj = 27;
                int* pok = &broj;
                Console.WriteLine($"Vrednost promenjive: {broj}");
                Console.WriteLine($"Adresa promenjive:
                    {(long)pok:X}");
            }
        }
    }
}
```

Даје:

Vrednost promenjive: 27

Adresa promenjive: EFEFF8

Операнд оператора & мора бити фиксна променљива (променљива у меморији на коју не делује сакупљач ђубрета и која се налази на стеку).

Променљиве које нису фиксне променљиве се називају покретне (movable) променљиве.

Поља објеката и елементи низова су пример покретних променљивих.

Адреса покретне променљиве се може добити са фиксним исказом.

Пример:

```
using System;
namespace Proba
{
    internal class Program
    {
        static void Main(string[] args)
        {
            unsafe
            {
                char slovo = 'A';
                char* pok = &slovo;
                Console.WriteLine($"Vrednost promenjive slovo:
                                {slovo}");
                Console.WriteLine($"Adresa promenjive slovo:
                                {(long)pok:X}");
                *pok = 'Z';
                Console.WriteLine($"Vrednost promenjive slovo posle
                                promene: {slovo}");
            }
        }
    }
}
```

Показивач на низ

За израз са р показивачем, показивач на елемент је у форми $p[n]$ и он се евалуира као $*(p + n)$, где је n типа `int`, `uint`, `long` или `ulong`.

Пример:

```
using System;
namespace Proba
{
    internal class Program
    {
        static void Main(string[] args)
        {
            unsafe
            {
                char* pok = stackalloc char[123];
                for (int i = 65; i < 123; i++)
                {
                    pok[i] = (char)i;
                }
                Console.WriteLine("Velika slova: ");
                for (int i = 65; i < 91; i++)
                {
                    Console.Write(pok[i]);
                }
            }
        }
    }
}
```

```

    }
    Console.WriteLine("\n");
}
}
}
}
}

```

Дaje:

Velika slova: ABCDEFGHIJKLMNOPQRSTUVWXYZ

Израз `stackalloc` приступа блоку меморије на стеку.

Блок меморије алоциран на стеку који је креиран током извршења методе се аутоматски одбацује када се метода врати.

Помоћу `stackalloc` се не може експлицитно ослободити меморија.

Пример:

```

using System;
namespace Proba
{
    internal class Program
    {
        static void Main(string[] args)
        {
            unsafe
            {
                int duzina = 3;
                int* brojevi = stackalloc int[duzina];
                for (var i = 0; i < duzina; i++)
                {
                    brojevi[i] = i;
                    Console.WriteLine(brojevi[i]);
                }
            }
        }
    }
}

```