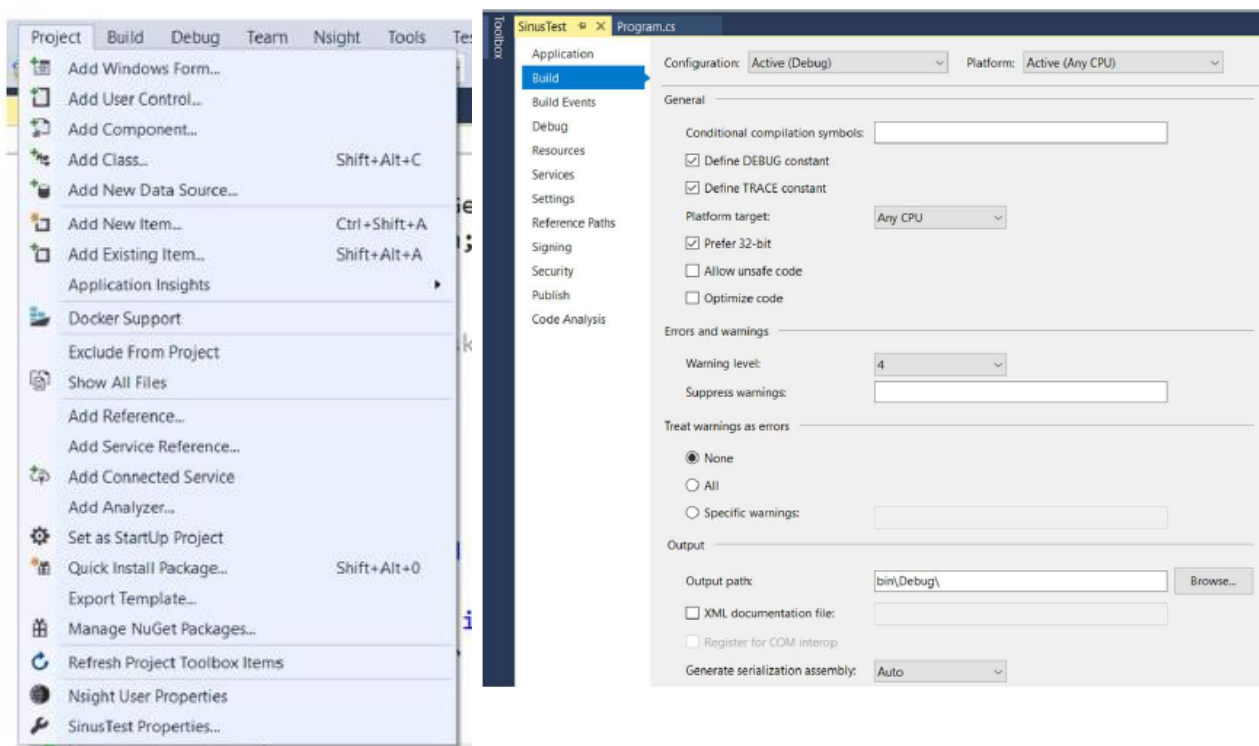


Wskaźnik jest zmienną, której wartość to adres innej zmiennej, np. bezpośredni adres w pamięci. Podobnie do innych zmiennych lub stałych, wskaźnik należy zadeklarować zanim zaczniemy przechowywać w nim adres danej zmiennej.

C# pozwala na używanie wskaźników kiedy bloku kodu jest oznaczony jako **unsafe**. Kod niezabezpieczony lub też kod niekontrolowany to blok kodu używający wskaźników.

Należy jednak pamiętać, aby w **projektancie projektu** włączyć możliwość używania w kodzie słowa kluczowego „unsafe”.



```
class Program
{
    unsafe static void Main(string[] args)
    {
        int* ip;
        // Deklaracja zmiennej całkowitej
        int number = 20;
        // przypisanie do wskaźnika adresu pamięci zmiennej
        // Wyrażenie &nazwa_zmiennej zwraca nam adres pamięci tej zmiennej
        ip = &number;
        Console.WriteLine("Wartość liczby: {0}", number);
        Console.WriteLine("Adres w pamięci: {0}", (int)ip);
        Console.ReadKey();
    }
}
```

Dane przechowywane w lokalizacji wskazywanej przez zmienną wskaźnikową możesz pobrać używając metody ToString().

```
Console.WriteLine("Wartość liczby: {0}", ip->ToString());
```

W języku C#, nazwa tablicy oraz wskaźnik do takiego samego typu danych jak dane w tablicy, **nie jest tym samym typem**.

Dla przykładu **int* p** oraz **int[] p**, nie są tego samego typu. Można zwiększać wartość wskaźnika **p** ponieważ nie jest on stałą wartością w pamięci podczas gdy adres tablicy w pamięci jest wartością stałą i nie może być zwiększony.

W celu uzyskania dostępu do danych tablicy przy użyciu wskaźników, należy ustalić wskaźnik w pamięci używając do tego słowa kluczowego **fixed** (patrz. na poniższy przykład)

```
class Program
{
    unsafe static void Main(string[] args)
    {
        int[] tablica = { 100, 200, 300 }; // deklaracja tablicy
        fixed (int* ptr = tablica) // ustalamy wskaźnik naszej tablicy

            // adres tablicy przechowujemy we wskaźniku
            for (int i = 0; i < 3; i++)
            {
                Console.WriteLine("Adres tablicy[{0}] = {1}", i, (int)(ptr + i));
                Console.WriteLine("Wartość tablicy[{0}] = {1}", i, *(ptr + i));
                Console.WriteLine("Wartość tablicy[{0}] = {0}", (ptr + i)->ToString());
            }
        Console.ReadKey();
    }
}
```

Więcej informacji o wskaźnikach znajdziesz tutaj:

<https://docs.microsoft.com/pl-pl/dotnet/csharp/programming-guide/unsafe-code-pointers/index>

<https://docs.microsoft.com/pl-pl/dotnet/csharp/programming-guide/unsafe-code-pointers/pointer-types>

*	Wykonuje operację wskaźnika pośredniego.
->	Uzyskuje dostęp do elementu członkowskiego struktury za pomocą wskaźnika.
[]	Indeksuje wskaźnik.
&	Uzyskuje adres zmiennej.
++ oraz --	Zwiększa i zmniejsza wartość wskaźnika.
+ oraz -	Wykonuje operacje arytmetyczne na wskaźniku.
==, !=, <, >, <= oraz >=	Porównuje wskaźniki.
stackalloc	Przydziela pamięć na stosie.
<pre>(int* block = stackalloc int[100];)</pre>	
Instrukcja fixed	Tymczasowo ustala zmienną, dzięki czemu można znaleźć ten adres.

Przykład

```
byte[] b = new byte[8];
unsafe
{
    // uzyskanie adresu tablicy - tablica to obiekt zarządzany-aby jej adres się
    // nie zmienił w czasie odwołania przez wskaźnik używamy bloku fixed
    fixed (byte* p = b)
    {
        //rzutowanie adresu liczby typu byte na adres typu int
        int* i = (int*)p;
        //tym sposobem zmieniam 4 kolejne elementy tablicy byte elementy (0-3)
        *i = 1857;
        //a tu zmieniam kolejne 4 elementy tablicy (4-7)
        *(i + 1) = 5678;
        //rzutuję adres do int na adres do long, dzięki temu wyświetlam całą tablicę
        //jako liczbę typu long
        Console.WriteLine(*(long*)i);
    }
}
```

Więcej przykładów znajdziesz tutaj:

https://www.plukasiewicz.net/Csharp_dla_zaaawansowanych/Unsafe_code

Zadanie 1

Napisać program, który wyświetli 25 pierwszych liczb ciągu Fibonacciego z wykorzystaniem wskaźników.

Zadanie 2

Napisać program, który dodaje dwie macierze 25x25 z wykorzystaniem wskaźników.

Zadanie 3

Napisać program, który mnoży dwie macierze 25x25 z wykorzystaniem wskaźników.