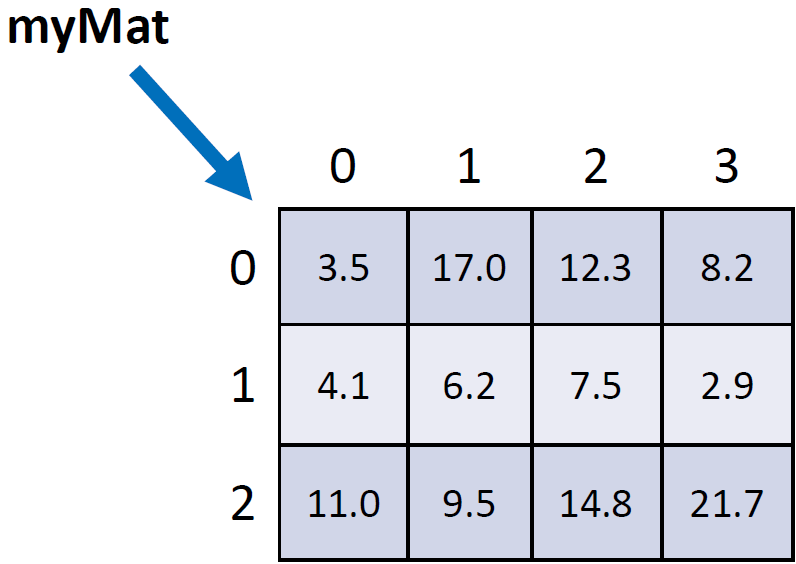
# Matrizes 04/12/2020

## Em programação, "matriz" é o nome dado a arranjos bidimensionais

Arranjo é uma estrutura de dados:

* Homogênea (dados do mesmo tipo)
* Ordenada (elementos acessados por meio de posições)
* Alocada de uma vez só, em um bloco contíguo de memória



Vantagens:

* Acesso imediato aos elementos pela sua posição

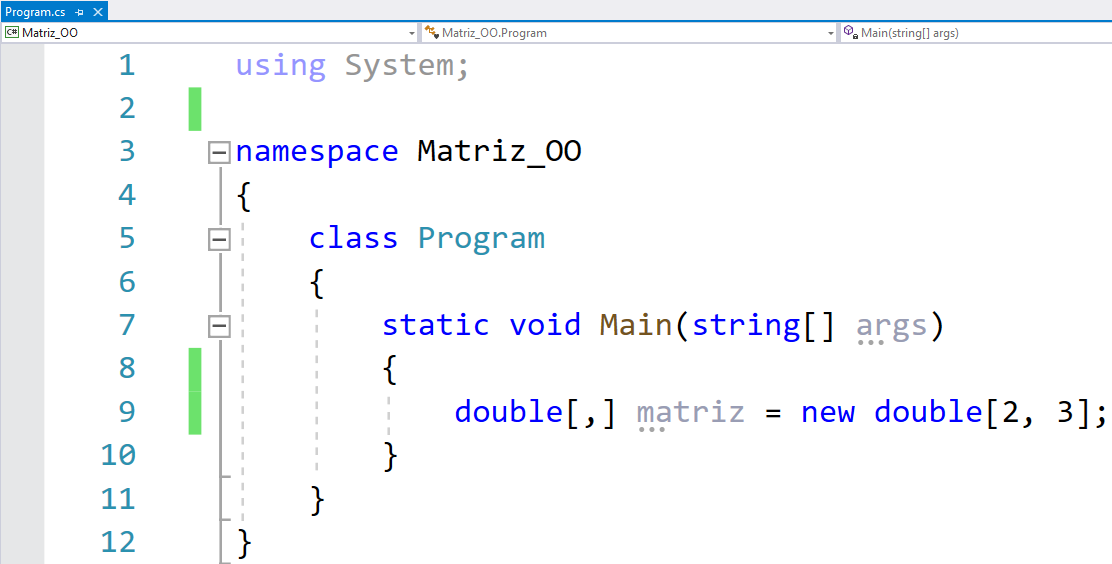
Desvantagens:

* Tamanho fixo
* Dificuldade para se realizar inserções e deleções

# Sintaxe

tipo[ , ] nomeDaMatriz = new tipo[ linhas, colulas ];

Exemplo:



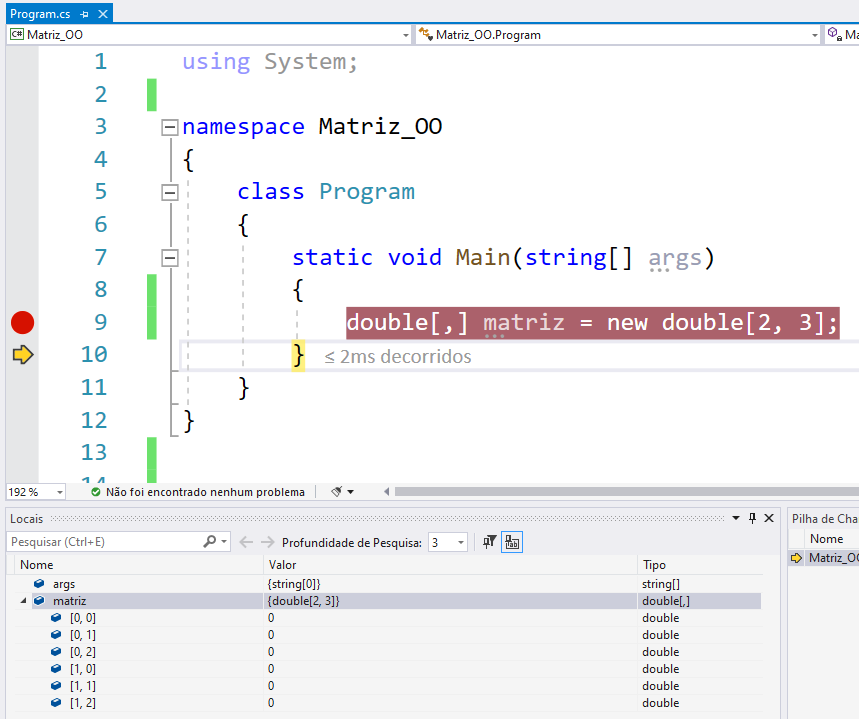
Para mostrar melhor, vamos debugar o código.

Vou criar breakpoint aqui na linha 9 (tecla F9)

Executando em modo debug com a tecla F5

Executando o próximo passo com a tecla F10

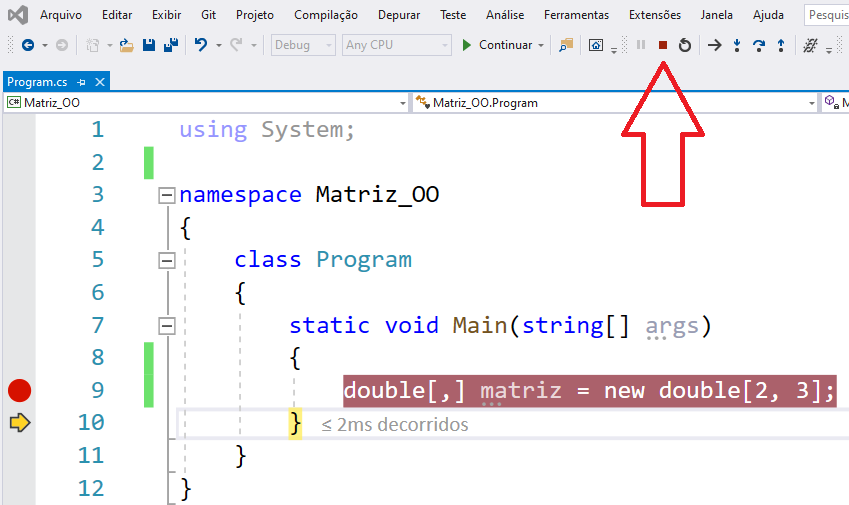
Expandindo a matriz



Repare que a matriz foi criada.

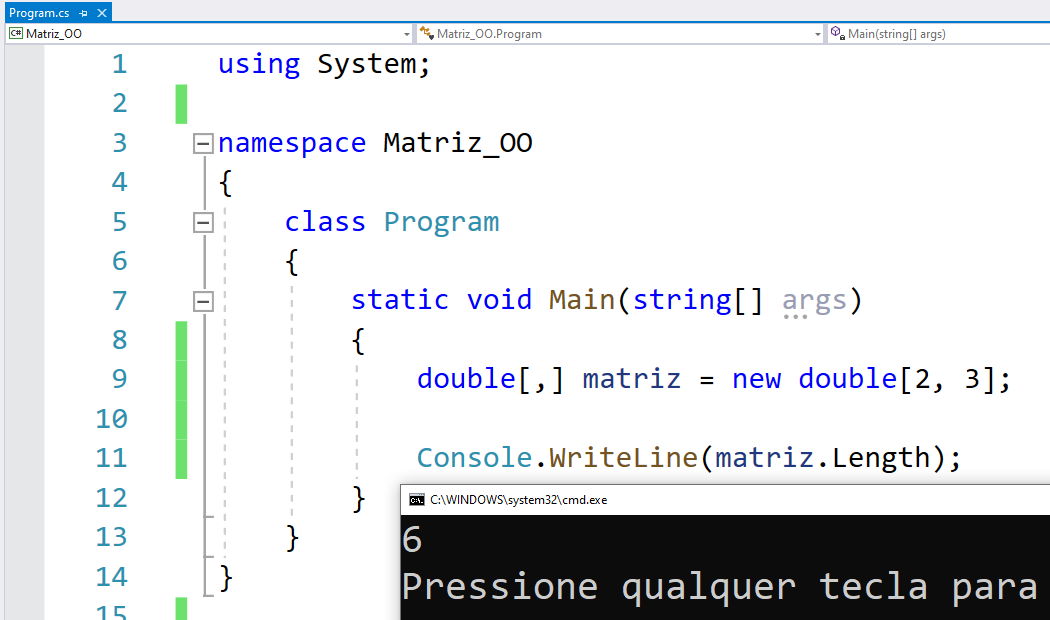
Desfazendo o Debug

Pressione o Stop



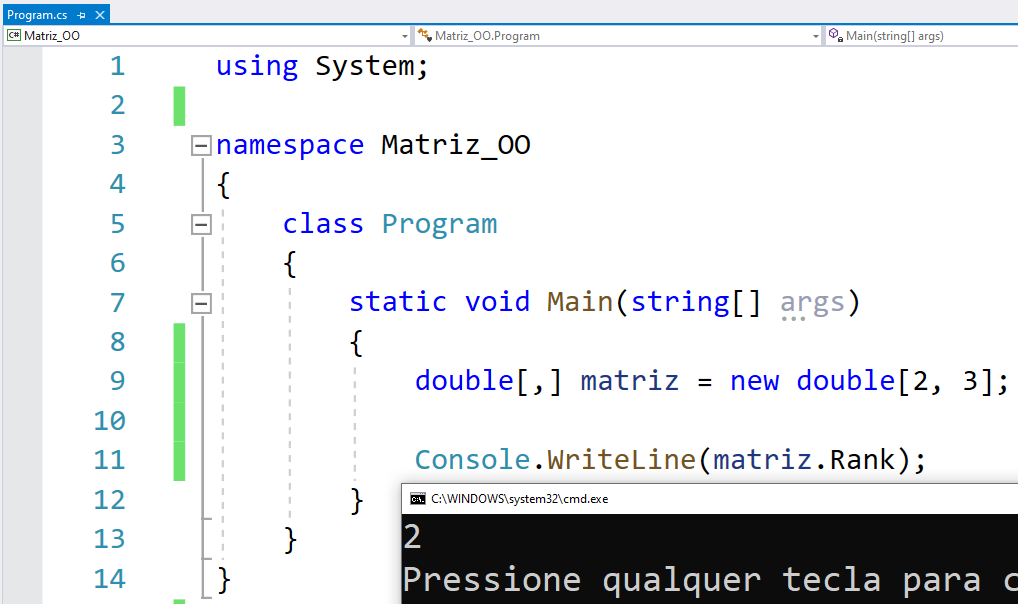
Coloque o cursor na linha 9 e pressione F9 para desmarcar o breakpoint.

# Length – A propriedade length



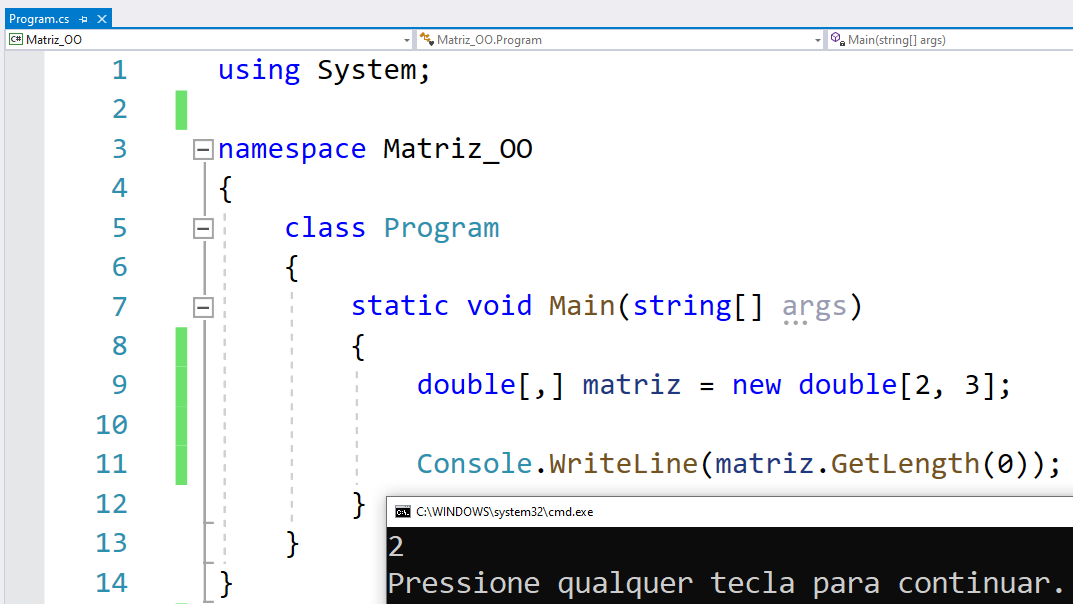
A propriedade Length diz quantos elementos a matriz tem no total.

# Rank – A propriedade Rank

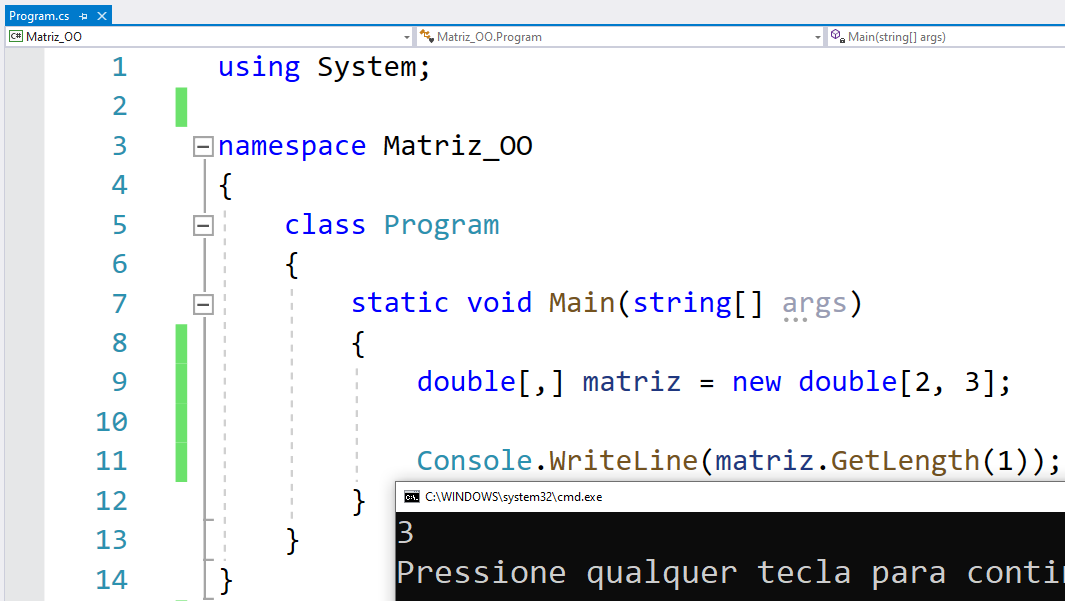


A propriedade Rank traz a primeira dimensão da matriz. Ou seja, a quantidade de linhas.

GetLength() –

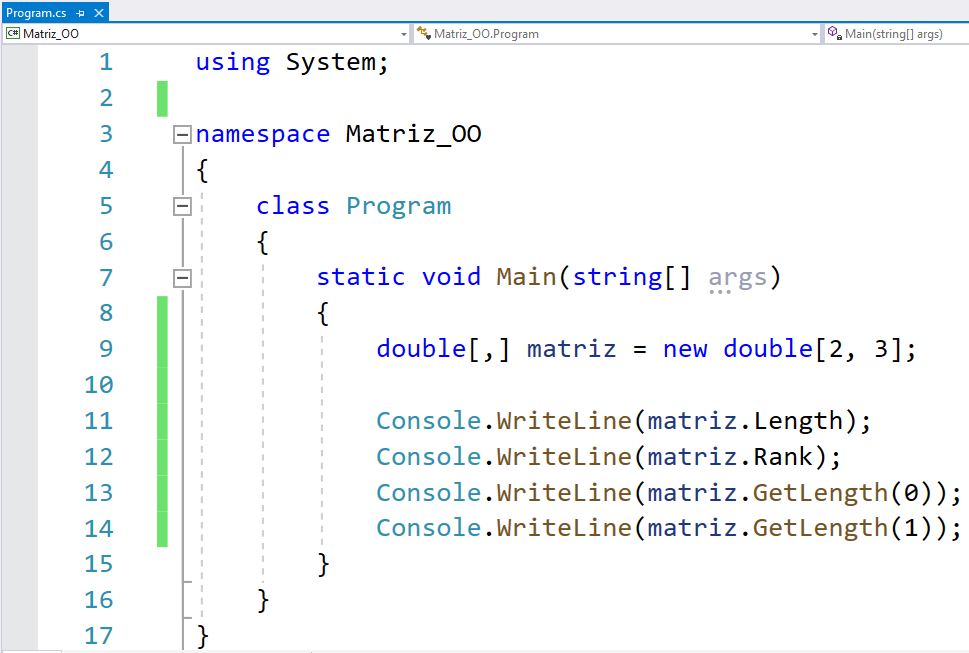


A propriedade GetLength traz o tamanho da dimensão especificada. No caso a quantidade de linhas da matriz.



Nesse exemplo, vimos a que a segunda dimensão da matriz tem o tamanho 3. Ou seja, a quantidade de colunas da matriz.

# Todas as demonstrações:



# Código das Demonstrações:

double[,] mat = new double[2, 3];

Console.WriteLine(mat.Length);

Console.WriteLine(mat.Rank);

Console.WriteLine(mat.GetLength(0));

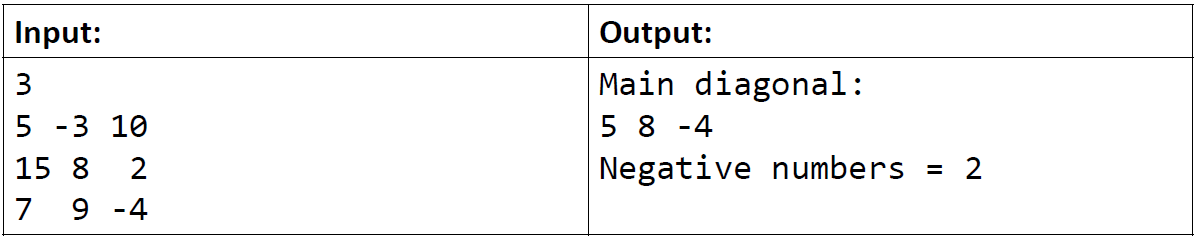
Console.WriteLine(mat.GetLength(1));

# Exercício:

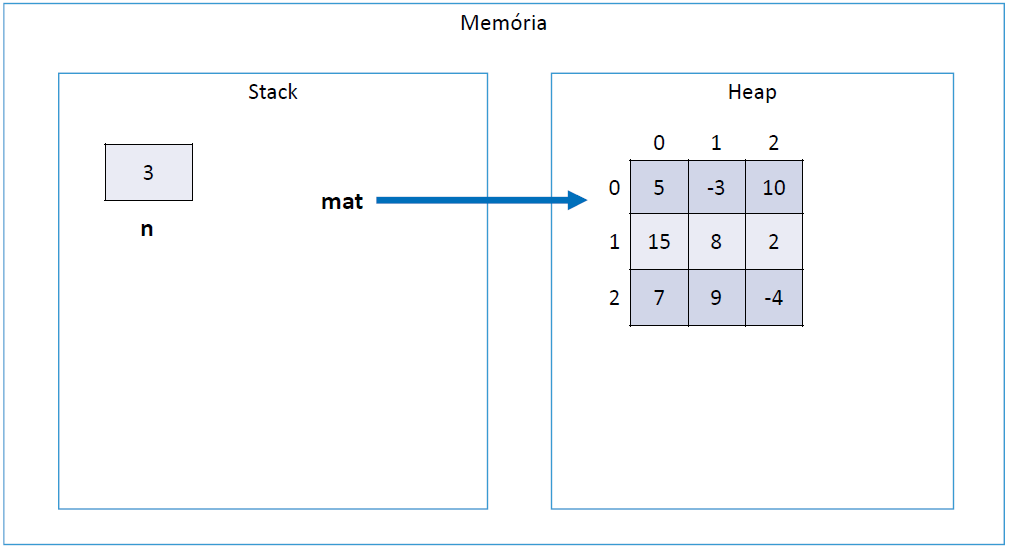
Fazer um programa para ler um número inteiro N e uma matriz de ordem N (uma matriz quadrada, com a quantidade igual entre linhas e colunas) contendo números inteiros. Em seguida, mostrar a diagonal principal e a quantidade de valores negativos da matriz.

Exemplo:

No exemplo abaixo, a diagonal principal é composta pelos números 5 8 2

****

Como a memória se comporta.

****

Código do exercício exemplo:

using System;

namespace MatizExercicioFixação

{

class Program

{

static void Main(string[] args)

{

int n = int.Parse(Console.ReadLine());

int[,] matriz = new int[n, n];

for (int i = 0; i < n; i ++)

{

string[] valores = Console.ReadLine().Split(' ');

for (int j = 0; j < n; j++)

{

matriz[i, j] = int.Parse(valores[j]);

}

}

Console.WriteLine("Diagonal principal:");

for ( int i = 0; i < n; i++ )

{

Console.Write(matriz[i,i] + " ");

}

Console.WriteLine();

int negativos = 0;

for ( int i = 0; i < n; i++ )

{

for (int j = 0; j < n; j++ ) {

if (matriz[i,j] < 0)

{

negativos++;

}

}

}

Console.Write("Números negativos = " + negativos);

Console.WriteLine();

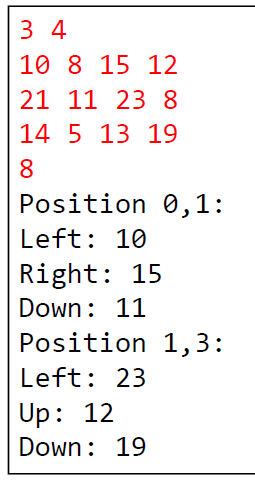
}

}

}

# Exercício de fixação

Fazer um programa para ler dois números inteiros M e N, e depois ler uma matriz de M linhas por N colunas contendo números inteiros, podendo haver repetições. Em seguida, ler um número inteiro X que pertence à matriz. Para cada ocorrência de X, mostrar os valores à esquerda, acima, à direita e abaixo de X, quando houver, conforme exemplo.



# Referência:

Docs Microsoft – Matrizes multidimensionais (guia de programação C #) – <https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/arrays/multidimensional-arrays > Acessado dia 03 de dezembro de 2020.