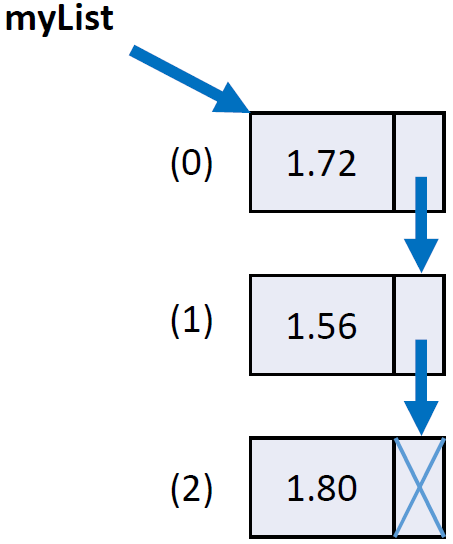
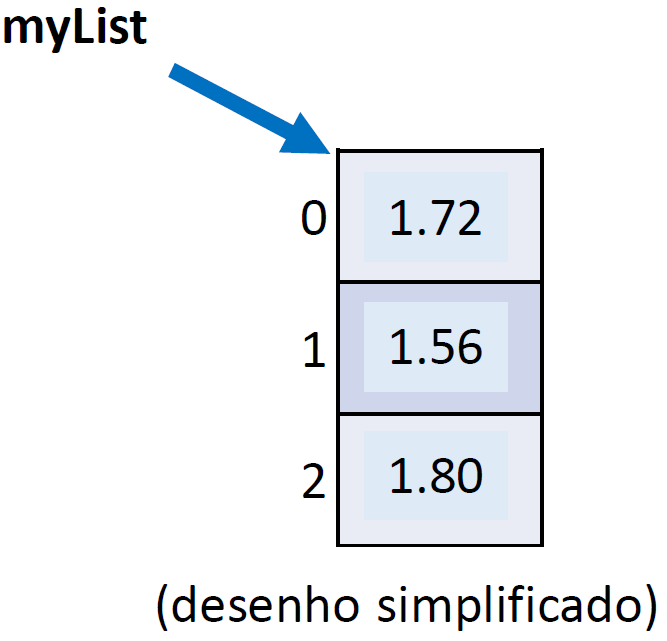
# Listas – list 30/11/2020

## Lista é uma estrutura de dados:

* Homogênea (dados do mesmo tipo).
* Ordenada (elementos acessados por meio de posições).
* Inicia vazia, e seus elementos são alocados sob demanda.
  + Diferente do vetor, que ao instanciar, já precisa informar o tamanho.
* Cada elemento ocupa um "nó" (ou nodo) da lista.



A seta é uma referência para os nodos relacionados. O ultimo nodo não aponta para ninguém.



# Classe: List

Para utilizar, precisamos adicionar uma biblioteca.

Namespace: System.Collections.Generic

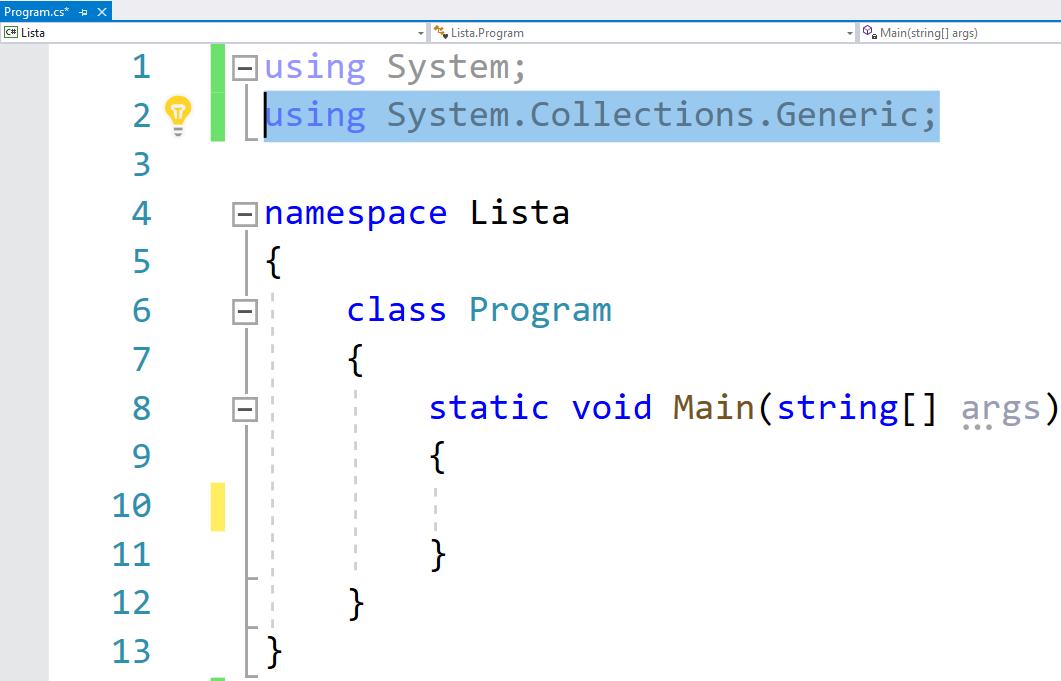
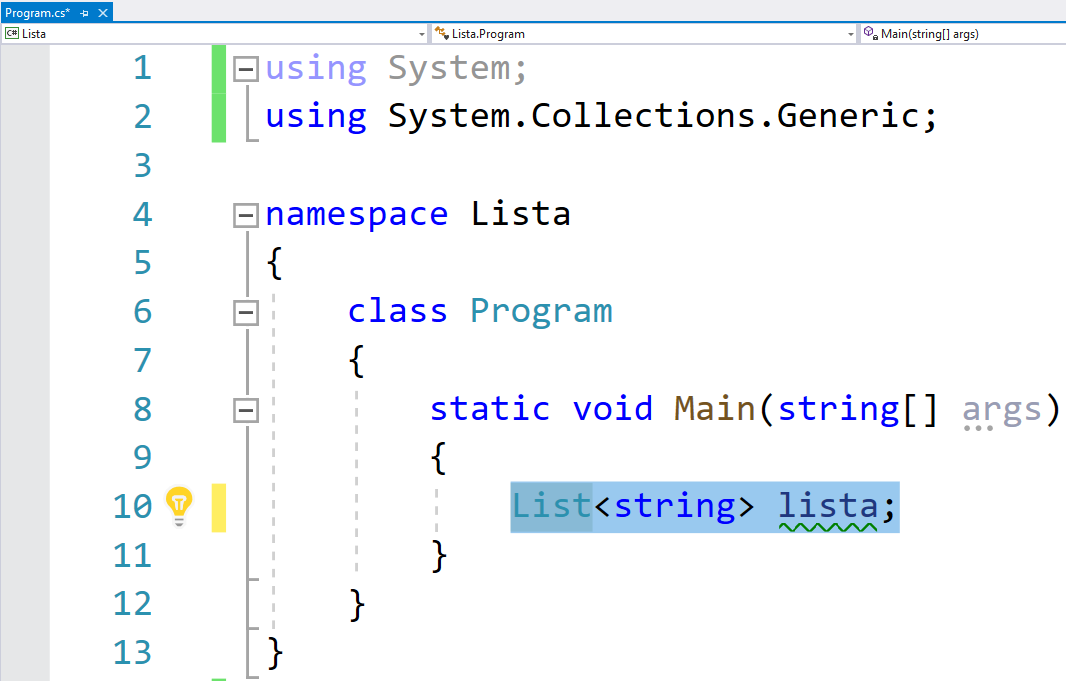
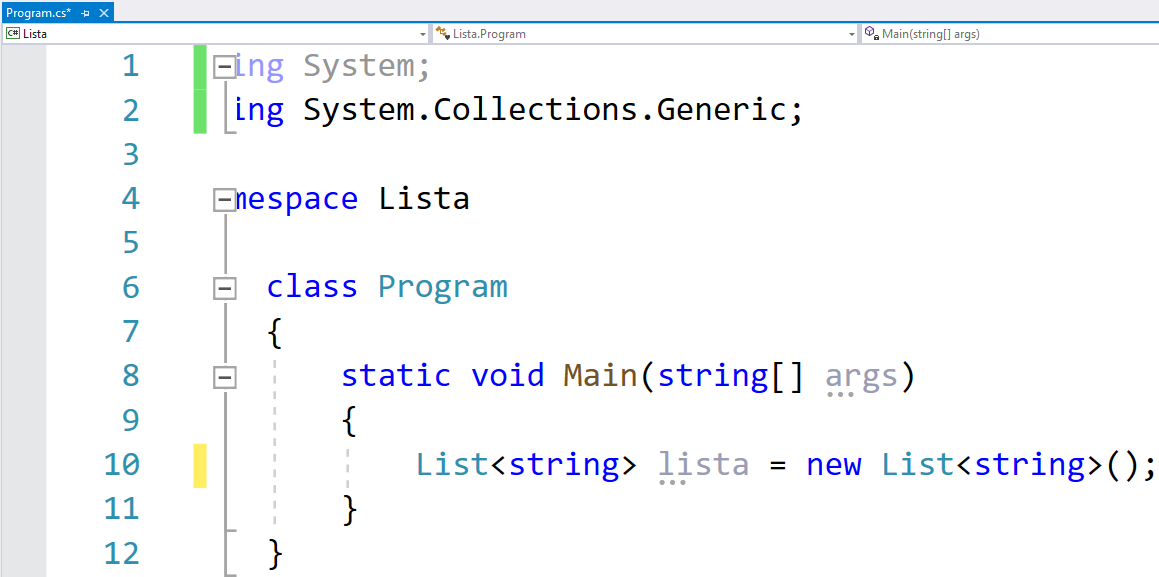


Figura 1. Biblioteca adicionando.

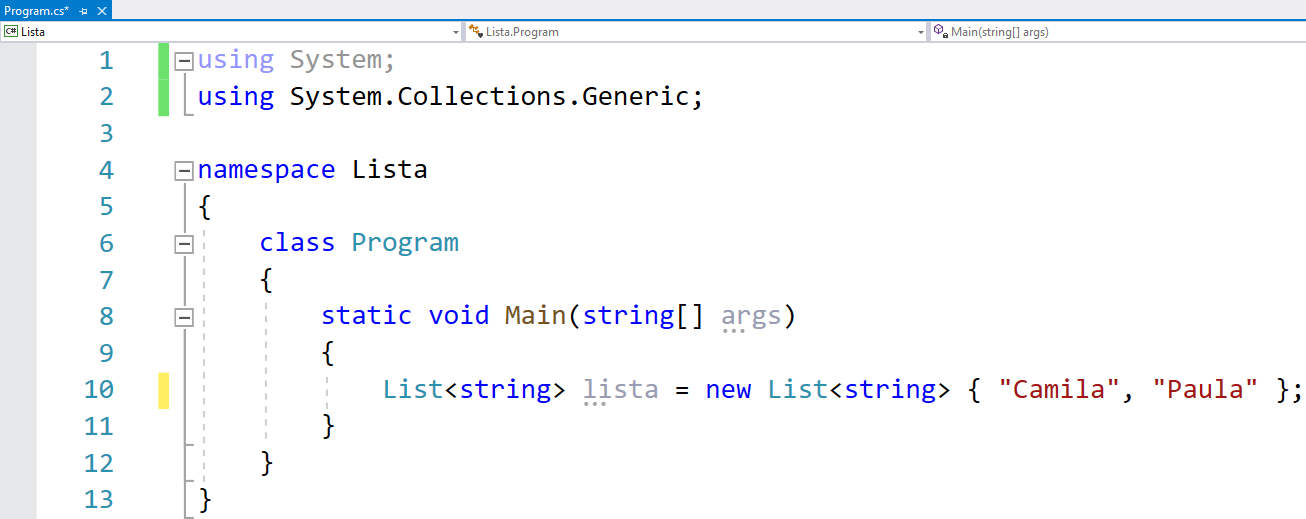
Criando uma lista



Podemos criar e instanciar a lista ao mesmo tempo.



Podemos também criar a lista já adicionando elementos.



Vantagens:

* Tamanho variável.
* Facilidade para se realizar inserções e deleções.

Desvantagens:

* Acesso sequencial aos elementos \* (Precisa percorrer toda a lista para acessar um elemento, o que perde performance).
  + Ressalva para listas otimizadas, que possuem um acesso razoavelmente rápido.

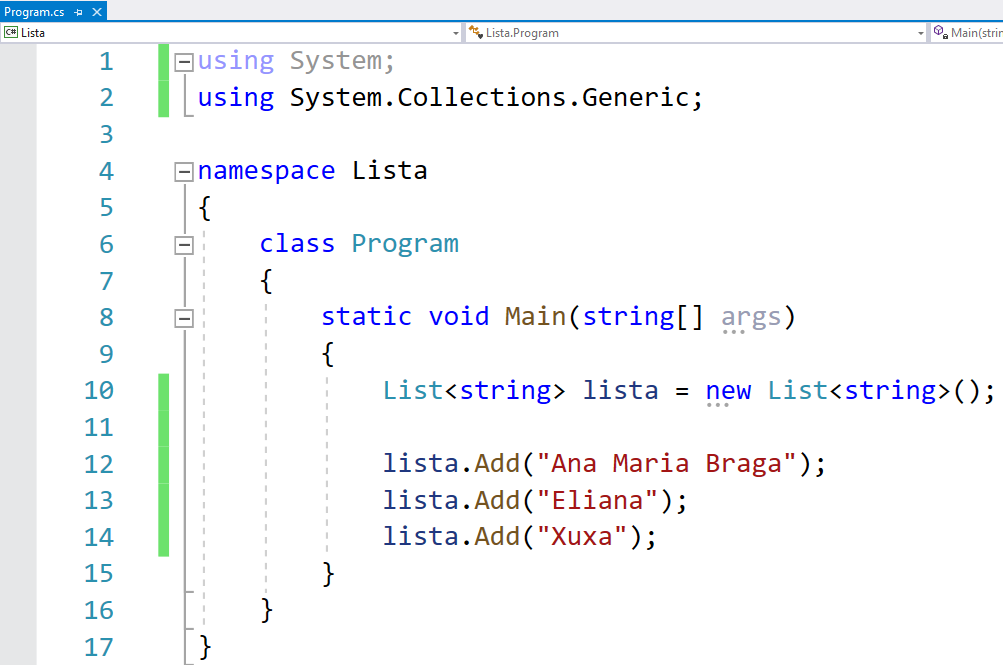
Ou seja, as desvantagens do vetor são corrigidas na lista. Porém, a desvantagem da lista não existe no vetor.

# Inserindo elementos na lista: Add, Insert

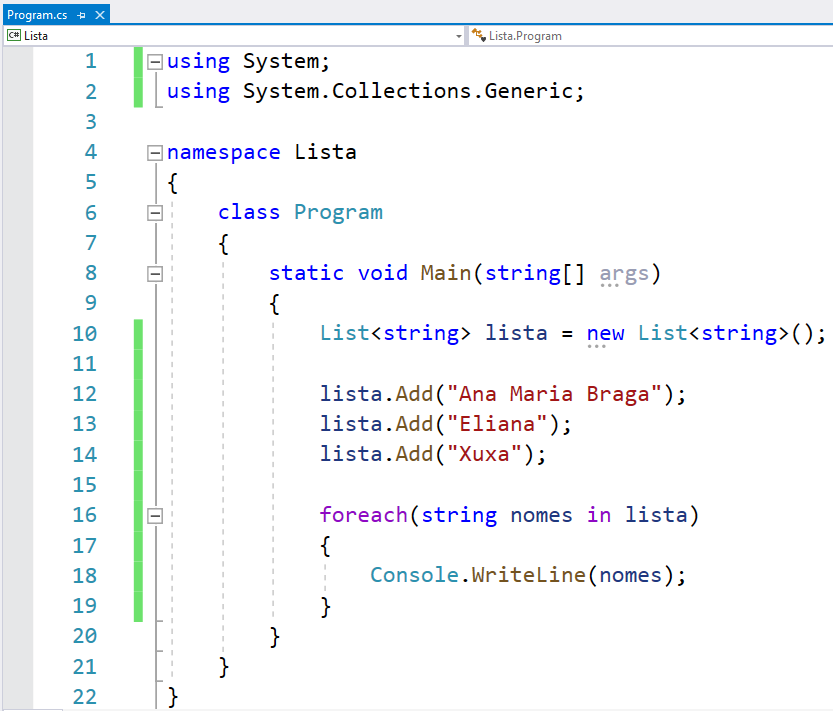
O método Add adiciona na mesma ordem inserida.

Sintaxe:

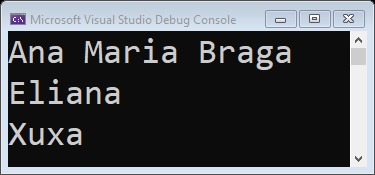
nomeDaLista.Add(“valor”);



Vamos testar com um foreach



Ok:



# Método Insert.

Com esse método, é possível inserir o elemento na posição desejada.



A primeira instrução é a posição, e a segunda o valor.

# Referência:

Docs Microsoft – List <T> Class – <https://docs.microsoft.com/en-us/dotnet/api/system.collections.generic.list-1?redirectedfrom=MSDN&view=net-5.0> Acessado dia 27 de novembro de 2020.