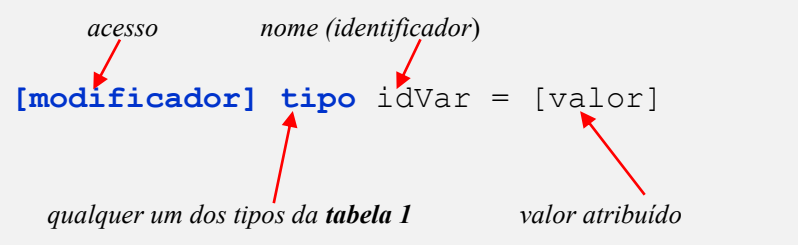
Variáveis Estáticas no C#

Mário Leite

...

Com base nos tipos primitivos de dados ***numérico***, ***caractere*** e ***lógico***, a linguagem **C#** trabalha com vários tipos derivados destes e com alguns novos tipos acrescentados; a **tabela 1** mostra os tipos de dados mais importantes desta linguagem. A declaração/inicialização de uma variável de memória é feita do seguinte modo:



No esquema acima ***modificador*** indica o tipo de acesso, *tipo* representa um dos tipos pré-definidos pela linguagem; neste caso, as variáveis assim declaradas são chamadas "Variáveis Estáticas" ou "Variáveis com Tipagem Estática". Isto quer dizer que uma vez definido o seu tipo (*int*, *float*, *double*, *string*, *char*, *bool* ...) elas não poderão receber valores que não sejam do tipo previamente definido no módulo onde foi declarada.

```
int num; //declarando sem iniciação de valor
int num = 1; //declarando com iniciação de valor
```

(*) **string** é, tecnicamente, um tipo de referência de uma *classe* da qual podem ser criados objetos (*instâncias*) e possui algumas características úteis na codificação.

(**) **void** não é propriamente um tipo de dado; é usado nos métodos que não possuem retorno tal como se comportam as *procedures* das linguagens procedurais tyradicionais.

(***) **var** é um tipo implícito e depende do valor inicial atribuído à variável, pois todas as variáveis declaradas com **var** devem ser iniciadas.

As variáveis do tipo *string* possuem algumas características especiais:

- O tamanho máximo permitido é bastante amplo (cerca de 2048 *bytes*.
- Um tipo *string* não precisa ser iniciado com o operador **new**.
- A atribuição de uma variável do tipo *string* à outra é como se fizesse uma cópia.
- Uma *string* contendo o valor *null* é uma *string* vazia; pode ser usada normalmente.

Entretanto, mesmo sendo uma classe, não se pode criar uma classe derivada de uma *string*; e o seu conteúdo tem que estar entre aspas duplas retas (**" "** não pode ser aspas inglesas **" "**) e o mesmo vale para os apóstrofos nos tipos ***char***: têm que ser apóstrofos (**' '**).

Além dos tipos de mostrados na **tabela 1** o C# também trabalha com outros tipos especiais: ***enum*** (*enumerado*), ***struct*** (*estrutura*), ***array*** (*referência*), ***class*** (*classe*), ***object*** (*instância* de classe), ***interface*** (*tipo definido* pelo usuário) e ***delegate*** (*tipo delegado*). Todos estes tipos extras estendem as funcionalidades da linguagem, melhorando sua performance e aplicabilidade. Entretanto, nas declarações de variáveis do tipo ***float*** é necessário iniciá-las com uma letra correspondente a esse tipo, pois caso contrário serão consideradas como ***double*** (padrão para o tipo flutuante). Então, para variáveis ***float*** deve ser colocado um **'F'** logo após o seu valor; e para variáveis do tipo ***decimal*** coloque **m**. A **figura 1** mostra a saída do programa cujo código é mostrado abaixo, utilizando declaração implícita com **var**. A **figura 2** mostra a saída do programa.

Nota: Postagem baseada no livro: "*Linguagem C#: Com acessos a Bancos de Dados*" publicado pelo autor na "Amazon" e no "Clube de Autores".
https://www.amazon.com.br/Linguagem-Com-Acesso-Bancos-Dados/dp/6587030653

Tipo	Característica	Tamanho	Faixa
sbyte	Inteiro com sinal	1 *byte*	-128 a +127
short	Inteiro com sinal	2 *bytes*	-32768 a +32767
int	Inteiro com sinal	4 *bytes*	- 2147483648 a +2147483647
long	Inteiro com sinal	8 *bytes*	-9223372036854775808 a +9223372036854775807
byte	Inteiro sem sinal	1 *byte*	0 a 255
ushort	Inteiro sem sinal	2 *bytes*	0 a 65535
uint	Inteiro sem sinal	4 *bytes*	0 a 4294967295
ulong	Inteiro sem sinal	8 *bytes*	0 a 18446744073709551615
float	Ponto flutuante	4 *bytes*	1.5×10^{-45} a 3.4×10^{38}
double	Ponto flutuante	8 *bytes*	5.0×10^{-324} a 1.7×10^{308}
decimal	Ponto flutuante alta precisão	16 *bytes*	-79.228162514264337593543950335 +79.228162514264337593543950335
bool	Lógico	1 *byte*	*true/false*
char	Caracteres Unicode	2 *bytes*	Delimitados por apóstrofos: 'x', 'y'
DateTime	Data e Hora (*classe*)		Precisão até 100 ns
string (*)	Cadeia de caracteres	(*)	Delimitada por aspas retas: "Nome"
void (**)	Sem tipo específico		
var (***)	Tipo implícito		Depende do valor iniciado

Tabela 1 - Tipos de dados no C#

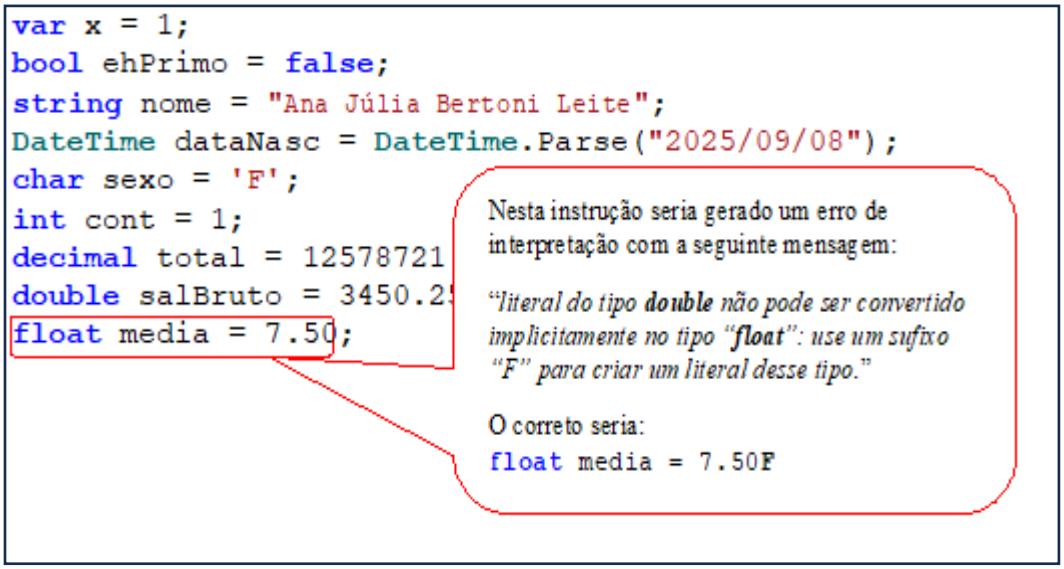


Figura 1 - Delarações válidas e inválidas em C#

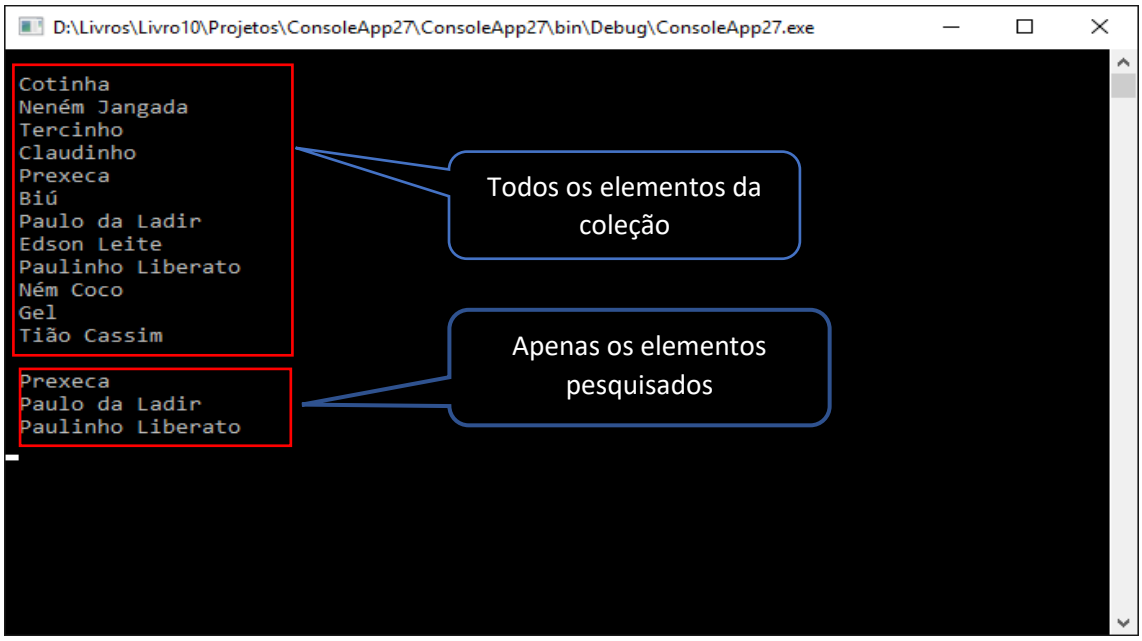


Figura 2 - Saída do programa "PrVarImplicito"

```
internal class Program
{
    static void Main(string[] args)
    {
        string[] VetColegas =
        {
            "Cotinha", "Neném Jangada", "Tercinho", "Claudinho",
            "Prexeca", "Biú", "Paulo da Ladir", "Edson Leite",
            "Paulinho Liberato", "Ném Coco", "Gel", "Tião Cassim"
        };
        /* Usa filtro baseado no LINQ */
        var pesquisa = from colega in VetColegas
              where colega[0] == 'P'
                    select colega;
        Console.WriteLine();
       /* Mostra todos os elementos do vetor "VetColegas" */
        foreach (string item in VetColegas) {
           Console.WriteLine(" " +item);
        }
        Console.WriteLine();
        /* Mostra apenas os elementos pesquisados */
        foreach (var item in pesquisa) {
           Console.WriteLine(" " + item);
        }
        Console.ReadKey();
        }
    }
} //Fim do programa
```