

Generation

BRASIL

1. Um **método** é um bloco de código que contém uma série de instruções. Um programa faz com que as instruções sejam executadas chamando o método e especificando os argumentos de método necessários.

☐ A Falso ☒ B Verdadeiro

2. Onde é possível fazer a **declaração** de um método?

☒ A Dentro de uma classe ☐ B Dentro de um Record
☐ C Dentro de um Struct ☐ D Dentro de uma Função

3. A assinatura de um método consiste respectivamente de:

<input type="checkbox"/> A	Parametros Nível de acesso Modificadores opcionais Tipo de retorno Nome	<input type="checkbox"/> B	Nome Nível de acesso Parametros Modificadores opcionais Tipo de retorno
<input checked="" type="checkbox"/> C	Nível de acesso Modificadores opcionais Tipo de retorno Nome Parametros		

4. `public static int Dividir(int a, int b);`
O que pode ser falado sobre o método acima:

<input type="checkbox"/> A	Método publico, de instância, que aguarda retorno do tipo inteiro, chamado Dividir, que espera 2 parametros a e b	<input type="checkbox"/> B	Método publico, estatico, que aguarda retorno do tipo string, chamado Dividir, que espera 2 parametros a e b
<input checked="" type="checkbox"/> C	Método publico, estatico, que aguarda retorno do tipo inteiro, chamado Dividir, que espera 2 parametros a e b		

5. `public static void Dividir(int a, int b);`

O que pode ser falado sobre o método acima:

- ☒ A Método publico, estatico, que nao aguarda retorno, chamado Dividir, que espera 2 parametros a e b
- ☐ B Método publico, estatico, que aguarda retorno do tipo string, chamado Dividir, que espera 2 parametros a e b
- ☐ C Método publico, estatico, que aguarda retorno do tipo inteiro, chamado Dividir, que espera 2 parametros a e b

6. Qual a diferença de um método de `instância` e um `estático`?

- ☒ A O método de instância precisa ser chamado pela sua instância, ja o estatico pode se chamado pelo seu nome
- ☐ B O método de instância precisa ser chamado pela seu nome, ja o estatico apenas pelo seu objeto
- ☐ C Não há nenhuma diferença

7. `Funções locais` são métodos privados de um tipo que estão aninhados em outro membro. Eles só podem ser chamados do membro que os contém.

- ☐ A Falso
- ☒ B Verdadeiro

8. `Funções locais` podem ser declaradas apartir de?

- ☒ A Métodos
- ☐ B Construtores
- ☐ C Outras funções locais

9.

```
// Isso é um Método
private static string Dividir(int a, int b)
{
    if (eDividendoMenorIgualZero(a))
    {
        return "Dividendo menor ou igual a zero";
    }

    return string.Format("{a} / {b} = {a / b}");

    //Isso é uma Função interna
    bool eDividendoMenorIgualZero(int a)
    {
        if (a <= 0)
        {
            return true;
        }
        return false;
    }
}
```

A responsabilidade do `método` da figura é dividir e a `função` é garantir que um dividendo não passe se for menor ou igual a zero!.

- ☒ A Verdadeiro
- ☐ B Falso

