

C# Essencial



**Classes e Métodos :
Exercícios Teóricos**

Exercícios

1- O que é uma classe e qual a sua composição ?

Uma classe é um modelo para criar objetos (uma estrutura de dados específica), fornecer valores iniciais para estado (variáveis ou atributos de membro) e implementações de comportamento (funções ou métodos de membro).

2- O que é um objeto e como pode ser criado ?

Um objeto é uma instância de classe que pode ser usada para acessar métodos de classe. A palavra-chave "New" pode ser usada para construir um objeto.

3- Qual a diferença entre classe e objeto ?

A classe é o modelo que descreve os objetos e suas características e comportamentos. Um objeto é o modelo personalizado com valores e estado e podem ser categorizados e agrupados. O objeto materializa a classe.

4- O que é um método e como pode ser criado e chamado ?

Um método é um bloco de código que contém uma série de instruções. Um programa executa as instruções chamando o método e especificando argumentos. Para chamar um método temos que criar uma instância da classe.

5- O que é um parâmetro de método ?

Um parâmetro é um valor, proveniente de uma variável ou de uma expressão mais complexa que representa um valor que o método espera receber quando for chamado. Um parâmetro possui um nome.

6- O que é um argumento ?

Um argumento representa o valor que você passa para um parâmetro de método quando você chama o método. O código de chamada fornece os argumentos quando chama o método. Um argumento não tem nome.

Exercícios

7- O que é um construtor e qual o seu objetivo ?

Um construtor é um método especial com o mesmo nome de sua classe. O construtor é invocado automaticamente quando um objeto é criado. Enquanto a classe está sendo inicializada, ela constrói todos os valores dos membros de dados.

8- Quais os usos da palavra chave this ?

A palavra-chave this refere-se à instância atual da classe e pode ser usada também para passar um objeto da instância atual para outros métodos, invocar outro construtor da mesma classe ou da classe base.

9- O que é um método com retorno ?

Os métodos com um tipo de retorno não void devem usar a palavra-chave return para retornar um valor. A palavra-chave return também interrompe a execução do método.

10- O que é sobrecarga de método ?

A sobrecarga de métodos é o processo de gerar muitos métodos na mesma classe com o mesmo nome, mas assinaturas distintas. O compilador utiliza resolução de sobrecarga para identificar qual método invocar quando compilamos.

11- Como é feita a passagem de parâmetros por valor ?

O método que tem um parâmetro de um tipo por valor, ao ser executado, recebe uma cópia do argumento (valor do parâmetro passado na chamada do método). O valor original da variável é alterado.

12- Como é feita a passagem de parâmetros por referência ?

O método que tem um parâmetro de um tipo por referência, ao ser executado, recebe uma referência do argumento, ou seja o endereço de memória da variável. O valor original da variável não é alterado.

Exercícios

13- Qual a diferença entre os parâmetros out e o parâmetro ref ?

Quando um argumento é passado como ref, ele deve ser inicializado antes de poder ser passado para o método. Um parâmetro out, por outro lado, não precisa ser inicializado antes de passar para um método.

14- O que são argumentos nomeados ?

Os argumentos nomeados permitem que você especifique um argumento para um parâmetro correspondendo o argumento com seu nome em vez de com sua posição na lista de parâmetros.

15- O que são parâmetros opcionais ?

Parâmetros opcionais são parâmetros que podem ser omitidos na chamada do método. Os parâmetros opcionais são definidos no final da lista de parâmetros, após quaisquer parâmetros obrigatórios

16- Qual o efeito da palavra-chave static ?

A palavra-chave static significa *relativo ao próprio tipo, em vez de uma instância do tipo*. Ao usar static podemos acessar um membro estático usando o nome do tipo em vez de uma referência ou um valor.

17- O que é método estático e qual o seu comportamento ?

Um método estático é um método que mantém apenas uma cópia do método no nível do tipo, não no nível do objeto. Isso significa que todas as instâncias da classe compartilham a mesma cópia do método e seus dados.

18- O que é um campo estático e qual o seu comportamento ?

Um campo estático é a declaração de uma variável que será mantida em comum por todas as instâncias de uma classe

Exercícios

19- O que são propriedades ?

As propriedades se comportam como campos quando são acessadas. No entanto, elas são implementadas com acessadores(*get* e *set*) que definem as instruções executadas quando uma propriedade é acessada ou atribuída. As propriedades permitem encapsular os atributos da classe.

20- Qual a diferença entre campos e propriedades ?

As propriedades expõem os campos. Os campos devem (quase sempre) ser mantidos privados para uma classe e acessados por meio das propriedades *get* e *set*. As propriedades fornecem um nível de abstração que permite alterar os campos sem afetar a maneira externa como eles são acessados por quem for utilizar a classe.

21- O que é uma struct ?

Um tipo struct é um tipo de valor que pode encapsular dados e funcionalidades relacionadas possuindo praticamente o mesmo comportamento de uma classe.

22- Qual a diferença entre uma struct e uma classe ?

Uma struct é um tipo de valor e uma classe é um tipo de referência. Assim os objetos criados a partir de uma struct são alocados na Stack e os objetos da classe são alocados na Heap.

23- O que é uma Enum ?

Um tipo de enumeração (ou tipo enum) é um tipo de valor definido por um conjunto de constantes nomeadas do tipo numérico integral subjacente.

Exercícios

24- Quais são e qual o escopo dos modificadores de acesso ?

Os modificadores de acesso C# são -

- Private - Um atributo ou método privado é aquele que só pode ser acessado de dentro da classe. -
- Public - Quando um atributo ou método é declarado público, ele pode ser acessado de qualquer lugar no código.
- internal - Quando uma propriedade ou método é definido como interno, ele só pode ser acessado a partir do assembly atual dessa classe.
- Protected - Quando um usuário declara um método ou atributo como protegido, ele só pode ser acessado por membros dessa classe e por aqueles que a herdam.

25- Qual a regra geral para usar os modificadores de acesso ?

Limitar a visibilidade de seus tipos e/ou membros restringindo-os a apenas a quem precisa realmente acessar o tipo ou membro. Quanto menos visibilidade houver, menor será a probabilidade de ocorrer uma alteração indesejada ou seja, menor será a chance de outras partes do sistema mudar quando você fizer uma atualização.

Exercícios

26- Defina partial class ?

Uma partial class (classe parcial) na linguagem C# é uma forma de dividir a definição de uma classe em múltiplos arquivos. Uma classe parcial permite que você declare a implementação de uma classe em partes separadas, em arquivos distintos, mas que, quando compilados, são tratados como uma única classe pelo compilador.