

# Generation

## BRASIL

1. **Namespaces**, são usados para:

- |                            |                   |                            |   |        |
|----------------------------|-------------------|----------------------------|---|--------|
| <input type="checkbox"/> A | Organizar classes | <input type="checkbox"/> B | Definir sistema inteiro                             | 1-(AD) |
| <input type="checkbox"/> C | Corrigir o código | <input type="checkbox"/> D | Facilitar chamada de classes diminuindo sua sintaxe |        |

2. Um arquivo pode conter zero ou vários **namespaces**!.

- |                            |            |                            |       |       |
|----------------------------|------------|----------------------------|-------|-------|
| <input type="checkbox"/> A | Verdadeiro | <input type="checkbox"/> B | Falso | 2-(A) |
|----------------------------|------------|----------------------------|-------|-------|

3. O que pode conter um **namespace**?

- |                            |                   |                            |             |           |
|----------------------------|-------------------|----------------------------|-------------|-----------|
| <input type="checkbox"/> A | Classes           | <input type="checkbox"/> B | Structs     | 3-(ABCDE) |
| <input type="checkbox"/> C | Interfaces        | <input type="checkbox"/> D | Enumerações |           |
| <input type="checkbox"/> E | Outros namespaces |                            |             |           |

4. O método **Main** é o ponto de entrada de um aplicativo C#.(Bibliotecas e serviços não exigem um Main método como um ponto de entrada). Quando o aplicativo é iniciado, Main o método é o primeiro método invocado!.

- |                            |            |                            |       |       |
|----------------------------|------------|----------------------------|-------|-------|
| <input type="checkbox"/> A | Verdadeiro | <input type="checkbox"/> B | Falso | 4-(A) |
|----------------------------|------------|----------------------------|-------|-------|

5. Uma **classe** é como uma planta de construção que define a estrutura para objetos, que são instâncias concretas daquela classe!.

- |                            |       |                            |            |       |
|----------------------------|-------|----------------------------|------------|-------|
| <input type="checkbox"/> A | Falso | <input type="checkbox"/> B | Verdadeiro | 5-(B) |
|----------------------------|-------|----------------------------|------------|-------|

6. Um **objeto** pode ser construído a partir de um construtor da classe!.

- |                            |            |                            |       |       |
|----------------------------|------------|----------------------------|-------|-------|
| <input type="checkbox"/> A | Verdadeiro | <input type="checkbox"/> B | Falso | 6-(A) |
|----------------------------|------------|----------------------------|-------|-------|

7.

```
2 references
3 public class Animal
4 {
5     2 references
6     public string Nome { get; set; }
7     1 reference
8     public string Cor { get; set; }
9     1 reference
10    public string Raca { get; set; }
11
12    0 references
13    public Animal(){}
14
15    1 reference
16    public Animal(string nome, string cor, string raca)
17    {
18        Nome = nome;
19        Cor = cor;
20        Raca = raca;
21    }
22
23    1 reference
24    public void Comer()
25    {
26        System.Console.WriteLine("Comer...");
27    }
28 }
```

7-(A)

O que significa as linhas 5, 6 e 7:

A

São atributos

B

São métodos

C

São construtores

8.

```
2 references
3 public class Animal
4 {
5     2 references
6     public string Nome { get; set; }
7     1 reference
8     public string Cor { get; set; }
9     1 reference
10    public string Raca { get; set; }
11
12    0 references
13    public Animal(){}
14
15    1 reference
16    public Animal(string nome, string cor, string raca)
17    {
18        Nome = nome;
19        Cor = cor;
20        Raca = raca;
21    }
22
23    1 reference
24    public void Comer()
25    {
26        System.Console.WriteLine("Comer...");
27    }
28 }
```

8-(B)

O que significa as linhas 9 e 11:

A

São métodos

B

São construtores

C

São atributos

9.

```
3 2 references
4  public class Animal
5  {
6      2 references
7      public string Nome { get; set; }
8      1 reference
9      public string Cor { get; set; }
10     1 reference
11     public string Raca { get; set; }
12
13     0 references
14     public Animal(){}
15
16     1 reference
17     public Animal(string nome, string cor, string raca)
18     {
19         Nome = nome;
20         Cor = cor;
21         Raca = raca;
22     }
23
24     1 reference
25     public void Comer()
26     {
27         System.Console.WriteLine("Comer...");
28     }
29 }
```

9-(C)

O que significa a linhas 18:

- ☐ A É o método ☐ B São construtores
- ☐ C São atributos
10. **Instância** é o termo dado para dizer que uma classe foi criada e agora possui um objeto nela no sistema, existe 2 formas de fazer isso uma é através do operador **new** e a outra é através de **atribuição**, visto nos exemplos:
- ☐ A **Gato** garfield = **new Gato**(); ☐ B **Gato** garfield = **Gato**();
- ☐ C **Gato** tom = garfield;