

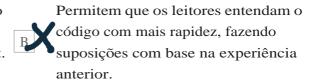
1.	Como é feita a entrada e saida de dados n	o C#?			
A	leia() e escreva()	В	input() e print()		
X	Console.ReadLine(); e Console.WriteLine();				
2.	O que signifíca uma linguagem ser fortemente tipada?				
A	Não possue tipos para definir suas variaveis	X	Variáveis e constantes têm um tipo, assim como cada expressão que é avaliada como um valor		
С	Somente algumas variaveis possuem atributos tipados				
3.	C# possui 4 tipos de variaveis, sendo elas literais e tipos genéricos.	: Tipos	s internos, tipos personalizados, tipos		
X	Verdadeiro	В	Falso		
4.	A definição: Representam inteiros, valor caracteres de texto, valores decimais e ou variavel?	-	·		
X	Tipos Internos	В	Tipos Personalizados		
С	Tipos Literais	D	Tipos genéricos		
5.	A definição: Utilizado para criar seus prop de variavel?	prios ti	pos personalizados. É aplicada a qual tipo		
A	Tipos Internos	X	Tipos Personalizados		
С	Tipos Literais	D	Tipos genéricos		

6.	A definição: Você pode especificar como um literal numérico deve ser digitado anexando uma letra ao final do número. É aplicada a qual tipo de variavel?			
A	Tipos Internos	В	Tipos Personalizados	
X	Tipos Literais	D	Tipos genéricos	
7.	A definição: Um tipo pode ser declarado como um espaço reservado para o tipo r variavel?			
A	Tipos Internos	В	Tipos Personalizados	
С	Tipos Literais	X	Tipos genéricos	
8.	É possivel declarar variáveis de diferentes declarando corretamente a variavel nome		as no C#, quais das alternativas abaixo esta	
A	string nome;	×	string nome = "Gustavo Boaz";	
С	<pre>string nome = Console.ReadLine();</pre>	X	string nome = "Gustavo" + " " + "Boaz";	
9.	Os Operadores (+, -, *, /, %), são operadores:			
A	Lógico	В	Relação	
С	Atribuição	X	Aritiméticos	
10.	Os Operadores (=, +=, -=, *=, /=, %=), são operadores:			
A	Lógico	В	Relação	
X	Atribuição	D	Aritiméticos	
11.	Os Operadores (==, !=, <, <=, >, >=), são operadores:			
A	Lógico	×	Relação	
С	Atribuição	D	Aritiméticos	
12.	Os Operadores (&&,), são operadores:			
×	Lógico	В	Relação	
С	Atribuição	D	Aritiméticos	

13. Quais as finalidades de escrever uma boa nomenclatura ao desenvolver um código?



Criam uma aparência consistente para o código, para que os leitores possam se concentrar no conteúdo e não no layout.





Facilitam a cópia, a alteração e a manutenção do código.



Demonstram as práticas recomendadas do C#.

14. O que esta acontecendo no trecho de código: float preco = (float) Convert.ToDouble(Console.ReadLine());

Esta convertendo de string para double, e depois convertendo para float

Esta convertendo de float para double, e depois fazendo um cast para string

Esta convertendo de string para double, e depois fazendo um cast para float