

Regras da linguagem

APRESENTAR AS REGRAS DA LINGUAGEM C#

AUTOR(A): PROF. APARECIDA DE FATIMA CASTELLO ROSA

Introdução às regras da Linguagem C#

Assim como, quando vamos aprender um novo idioma devemos conhecer suas regras, tais como a sintaxe do idioma, a semântica, acentuação, etc., o mesmo acontece em programação.

Quando vamos aprender uma nova linguagem de programação devemos conhecer suas regras.

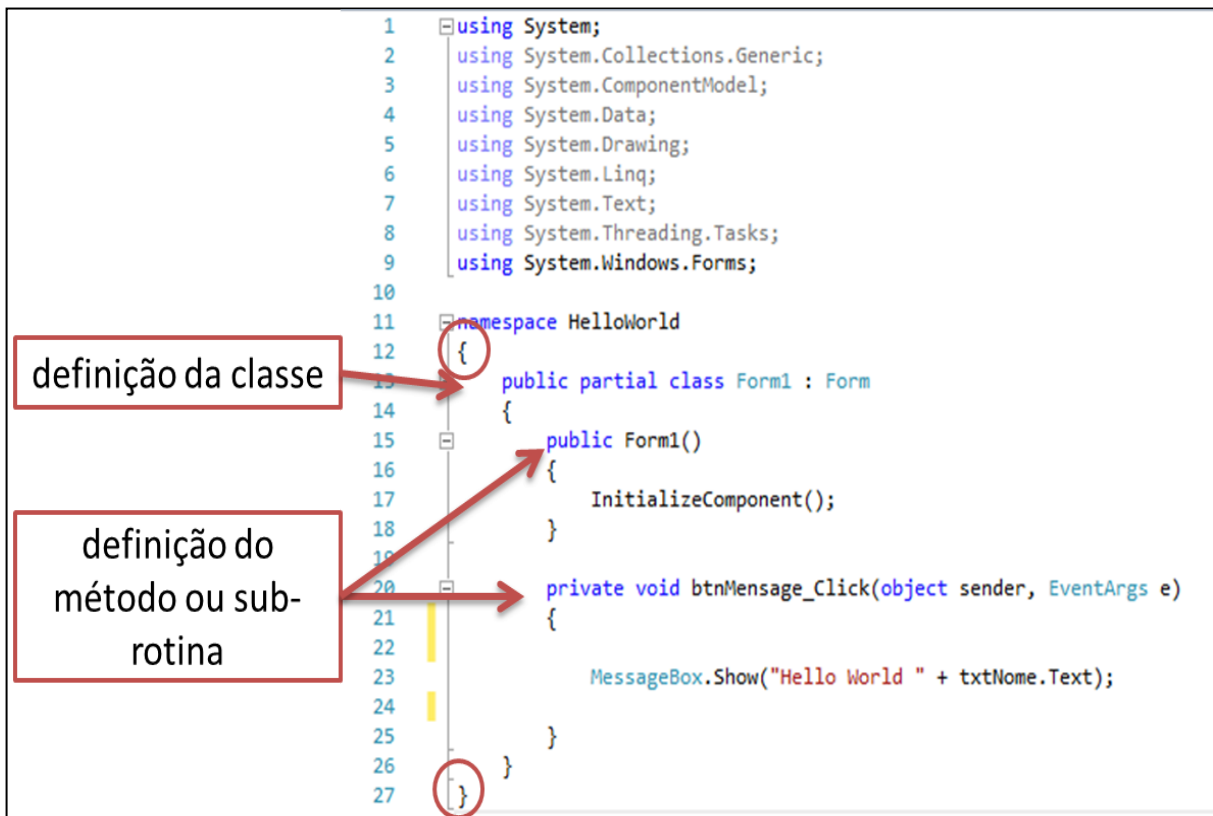
C# é uma linguagem de programação Orientada a Objeto, e mesmo que você escreva um simples programa para apresentar uma mensagem, a codificação do programa deverá estar dentro de uma classe.

Os aplicativos em C# contém uma ou mais classes e cada classe normalmente contém um ou mais métodos, sendo que uma das classes precisa do método `Main` para iniciar o programa.

Os métodos são blocos de códigos que podem realizar tarefas e retornar informações quando essas tarefas são concluídas, porém, também existem métodos que não retornam nenhum tipo de informação.

Na programação procedural, que é a nossa abordagem, os métodos serão as nossas funções e sub-rotinas.

Veja a figura a seguir.



A linha 23 contém o código implementado pelo programador.

A maior parte dessa estrutura é gerada automaticamente pelo IDE do Visual Studio.

Então vamos ver as regras da linguagem C#.

Regras da Linguagem C#

O C# é case-sensitive, ou seja, o C# faz distinção entre letras maiúsculas e minúsculas, portanto, Total é diferente de total, que é diferente de toTal.

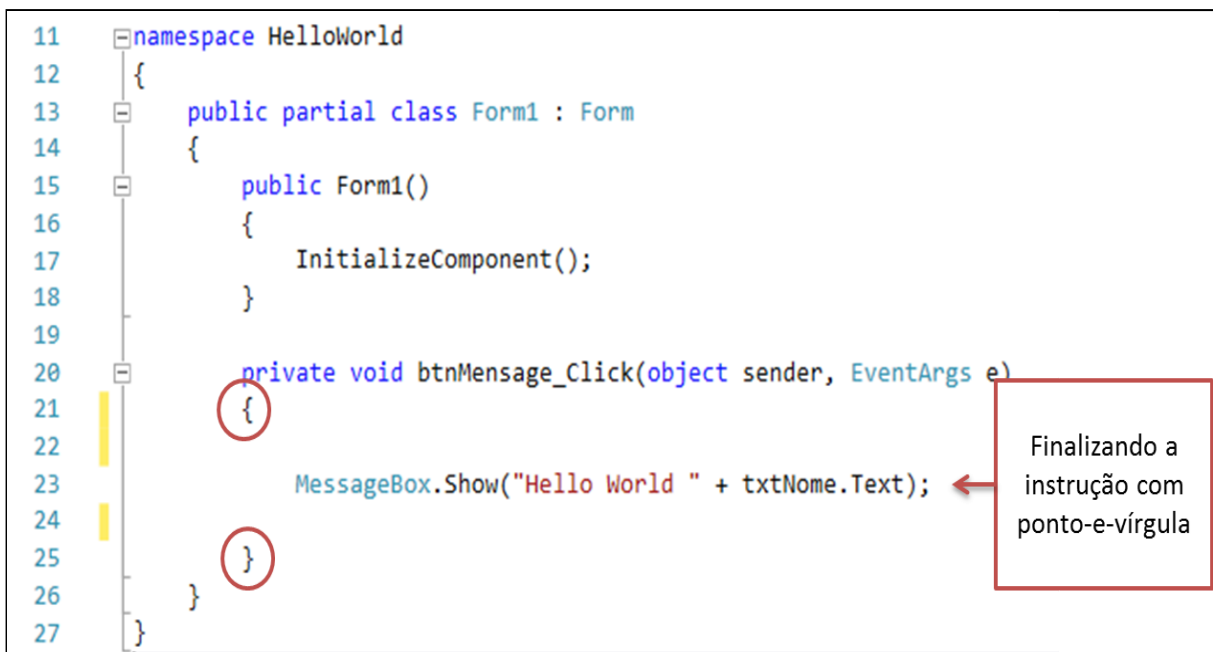
O compilador do C# ignora os espaços em branco adicionais no código, quebra de linhas ou tabulações. Você poderá utilizar os espaços e quebras de linha para facilitar a organização e clareza do código.

As instruções (comandos, declarações) devem terminar com *ponto-e-vírgula* (;). Omitir o *ponto-e-vírgula* ao final da instrução gera um erro de sintaxe. Veja o exemplo de um comando que utiliza um `MessageBox` (`MessageBox` é uma classe que contém o método `Show` que fornece uma caixa de diálogo) para apresentar uma saudação.

```
1. MessageBox.Show("Hello, World");
```

C# é uma linguagem estruturada em blocos. Um *par de chaves* ({ }) define um bloco de código que pode ser utilizado em uma classe, método ou uma estrutura.

Uma chave de abertura ({) inicia o corpo de toda definição de classe ou método. Uma chave de fechamento (}) correspondente finaliza cada definição de classe ou método. Se as chaves não ocorrerem em pares, o compilador indicará um erro. Observe na figura a seguir.



Comentários em C#

Os comentários são utilizados para você inserir notas no código fonte. São muito utilizados para documentar e melhorar a clareza do código. Eles também ajudam as outras pessoas a lerem e entenderem seus programas. Os comentários podem ser colocados em qualquer lugar do programa e são tratados pelo compilador como espaços em branco.

No IDE do Visual Studio .NET, todo texto de comentário aparece na cor verde.

Você pode comentar somente uma linha ou um bloco de código. Para comentar uma linha utilize // e para comentarmos um bloco de código utilize /* */.

Veja o exemplo de um trecho de código que apresenta na linha 23 um comentário de linha.

```
20 private void btnExibir_Click(object sender, EventArgs e)
21 {
22     //Exibi uma caixa de mensagem na tela
23     MessageBox.Show("Hello, World");
24 }
```

Agora, veja o mesmo trecho de código utilizando comentário em bloco. O comentário inicia na linha 22 até a linha 26.

```
20 private void btnExibir_Click(object sender, EventArgs e)
21 {
22     /*
23     A instrução a seguir chama o método Show( ) do objeto MessageBox
24     * Esse objeto é uma caixa de mensagem que aparece na tela.
25     * Ela irá exibir a mensagem Hello, World
26     */
27     MessageBox.Show("Hello, World");
28 }
```

Linhas em branco, espaços e tabulações são ignorados pelo compilador e utilizados pelos programadores para melhorar a legibilidade do programa.

Erros em C#

Quando não seguimos as regras de uma linguagem de programação alguns erros podem acontecer. Frequentemente o IDE do Visual Studio sinaliza quando alguma coisa no código está errada.

Erro de Sintaxe

Quando o compilador não reconhece uma instrução, dizemos que temos um erro de sintaxe. Normalmente o compilador fornece uma mensagem de erro para nos ajudar a localizar e corrigir o erro. Erros de sintaxe são violações das regras da linguagem.

Erro lógico

Erros lógicos são erros que impedem o programa de fazer o que você pretendia fazer. Seu código pode ser compilado e executado sem erros, mas o resultado de uma operação pode produzir um resultado não esperado, ou seja, houve um erro de lógica.

Erro de tempo de execução

Erros em tempo de execução são erros que ocorrem enquanto o programa é executado. Eles normalmente ocorrem quando o programa tenta realizar uma operação que é impossível de ser executar. Por exemplo, divisão por zero.

Erro de compilação

Erros de compilação, também conhecido como erros de compilador, são erros que impedem o programa de ser executado.

Nomeação de Identificadores

Como você já deve ter visto, a nomeação dos identificadores devem seguir estas regras:

- Somente letras (maiúsculas e minúsculas), dígitos (números), e o caractere underline “_” (sublinhado) são permitidos.
- Um identificador pode ter um ou mais caracteres.
- Um identificador deve começar com uma letra. O primeiro caractere não pode ser em hipótese alguma um número.
- Não pode ter espaços em branco.
- identificadores devem ser únicos dentro de um determinado escopo.
- Identificadores são case-sensitive, ou seja, diferenciam letras maiúsculas de minúsculas.
- Palavras reservadas da linguagem não podem ser utilizadas para identificadores, bem como o nome dos controles.
- Não utilizar acentuação e nem “ç”.

Utilize as mesmas regras para nomear os controles visuais da Linguagem C#.

Algumas convenções que devem ser seguidas para definir os identificadores:

- Identificadores devem ser legíveis.
- Não utilize abreviações.
- Identificadores devem transmitir o significado, ou seja, dê nomes significativos que expressam o que o identificador representa.

Convenções de Nomenclatura

As convenções adotadas em C# não são obrigatórias, mas criaram um padrão para a codificação que é largamente utilizado.

As convenções de nomenclatura recomendadas sugerem a utilização dos seguintes padrões:

Notação	Definição	Para nomear	Exemplos
camel casing	primeira letra de cada palavra em minúsculo	variáveis e parâmetros	nome, nomeCompleto, nomeDoLivro
Pascal casing	cada palavra inicia com letra maiúscula	métodos e outros identificadores	CalcularMedia()

Para nomear os controles visuais, utilize o prefixo apropriado para cada elemento de interface.

A tabela a seguir apresenta uma sugestão para nomeação dos controles.

Controle	Mnemônico	Exemplo
Label	lbl	lblNome
TextBox	txt	txtNome
Button	btn	btnSalvar
Form	frm	frmPrincipal
ComboBox	cbo	cboUF
ListBox	lst	lstDetalhes
RadioButton	rdb	rdbAtivo
CheckBox	chk	chkLazer
DataGrid	dtg	dtgAlunos
Image	img	imgLogotipo
Panel	pnl	pnlDados
Table	tbl	tblCliente

Escopo de Variáveis

O escopo de uma variável refere-se ao bloco de código dentro do qual é possível usar essa variável sem qualificação.

Por exemplo, uma variável de escopo local é uma variável declarada dentro de um método. Essa variável somente estará disponível dentro de bloco de código desse método, depois de ter sido declarada. Quando o método finalizar, a variável local será destruída, ou seja, não será mais possível acessar essa variável.

Além de variáveis locais, C# tem variáveis do tipo campo e de parâmetro. No entanto, C# não tem variáveis de escopo global, como por exemplo na linguagem C++.

Palavras-Chave - Keywords

Palavras-chave são palavras predefinidas, que tem um significado especial para o compilador. São palavras reservadas da linguagem que não podem ser utilizadas para nomear identificadores.

Veja as palavras-chave na tabela a seguir.

abstract (https://msdn.microsoft.com/en-us/library/sf985hc5.aspx)	as (https://msdn.microsoft.com/en-us/library/cscsdfbt.aspx)	base (https://msdn.microsoft.com/en-us/library/hfw7t1ce.aspx)	bool (https://msdn.microsoft.com/en-us/library/5bdc6693.aspx)
break (https://msdn.microsoft.com/en-us/library/adbctzc4.aspx)	byte (https://msdn.microsoft.com/en-us/library/5bdc6693.aspx)	case (https://msdn.microsoft.com/en-us/library/06tc147t.aspx)	catch (https://msdn.microsoft.com/en-us/library/06tc147t.aspx)
char (https://msdn.microsoft.com/en-us/library/x9h8tsay.aspx)	checked (https://msdn.microsoft.com/en-us/library/74b4xzyw.aspx)	class (https://msdn.microsoft.com/en-us/library/0b0thckt.aspx)	const (https://msdn.microsoft.com/en-us/library/0b0thckt.aspx)
continue (https://msdn.microsoft.com/en-us/library/923ahwt1.aspx)	decimal (https://msdn.microsoft.com/en-us/library/364x0z75.aspx)	default (https://msdn.microsoft.com/en-us/library/25tdedf5.aspx)	delegate (https://msdn.microsoft.com/en-us/library/25tdedf5.aspx)
do (https://msdn.microsoft.com/en-us/library/370s1zax.aspx)	double (https://msdn.microsoft.com/en-us/library/678hzkk9.aspx)	else (https://msdn.microsoft.com/en-us/library/5011f09h.aspx)	enum (https://msdn.microsoft.com/en-us/library/5011f09h.aspx)
event (https://msdn.microsoft.com/en-us/library/8627sbea.aspx)	explicit (https://msdn.microsoft.com/en-us/library/xhbhez4.aspx)	extern (https://msdn.microsoft.com/en-us/library/e59b22c5.aspx)	false (https://msdn.microsoft.com/en-us/library/e59b22c5.aspx)
finally (https://msdn.microsoft.com/en-us/library/zwc8s4fz.aspx)	fixed (https://msdn.microsoft.com/en-us/library/f58wzh21.aspx)	float (https://msdn.microsoft.com/en-us/library/b1e65aza.aspx)	for (https://msdn.microsoft.com/en-us/library/b1e65aza.aspx)

ADD (https://msdn.microsoft.com/pt-br/library/cc713648.aspx)	alias (https://msdn.microsoft.com/pt-br/library/ms173212.aspx)	ascending (https://msdn.microsoft.com/pt-br/library/cc713606.aspx)
async (https://msdn.microsoft.com/pt-br/library/hh156513.aspx)	await (https://msdn.microsoft.com/pt-br/library/hh156528.aspx)	descending (https://msdn.microsoft.com/pt-br/library/cc713622.aspx)
dynamic (https://msdn.microsoft.com/pt-br/library/dd264741.aspx)	from (https://msdn.microsoft.com/pt-br/library/bb383978.aspx)	get (https://msdn.microsoft.com/pt-br/library/ms228503.aspx)
global (https://msdn.microsoft.com/pt-br/library/cc713620.aspx)	group (https://msdn.microsoft.com/pt-br/library/bb384063.aspx)	into (https://msdn.microsoft.com/pt-br/library/bb311045.aspx)
join (https://msdn.microsoft.com/pt-br/library/bb311040.aspx)	let (https://msdn.microsoft.com/pt-br/library/bb383976.aspx)	orderby (https://msdn.microsoft.com/pt-br/library/bb383982.aspx)
partial (tipo) (https://msdn.microsoft.com/pt-br/library/wbx7zzdd.aspx)	partial (método) (https://msdn.microsoft.com/pt-br/library/6b0scde8.aspx)	remove (https://msdn.microsoft.com/pt-br/library/cc713642.aspx)
select (https://msdn.microsoft.com/pt-br/library/bb384087.aspx)	Set (https://msdn.microsoft.com/pt-br/library/ms228368.aspx)	Valor (https://msdn.microsoft.com/pt-br/library/a1khb4f8.aspx)
var (https://msdn.microsoft.com/pt-br/library/bb383973.aspx)	where (restrição de tipo genérico) (https://msdn.microsoft.com/pt-br/library/bb384067.aspx)	where (cláusula de consulta) (https://msdn.microsoft.com/pt-br/library/bb311043.aspx)
yield (https://msdn.microsoft.com/pt-br/library/9k7k7cf0.aspx)		

Fonte: <https://msdn.microsoft.com/pt-br/library/x53a06bb.aspx> (<https://msdn.microsoft.com/pt-br/library/x53a06bb.aspx>)

ATIVIDADE

Analise as afirmações a seguir referente a palavras-chaves da linguagem C#.

- I. Os programadores podem nomear identificadores utilizando palavras-chaves.
- II. As palavras-chaves da linguagem C# são definidas pelo programador.
- III. Palavras-chaves são palavras reservadas da linguagem que não podem ser utilizadas para nomear identificadores.
- IV. Palavras-chaves são palavras pré-definidas, que tem um significado especial para o compilador.

É correto apenas o que se afirma em

- A. I e II
- B. II e III
- C. III e IV
- D. II e IV

ATIVIDADE

Erros em tempo de execução são erros que ocorrem enquanto o programa é executado. Eles normalmente ocorrem quando o programa tenta realizar uma operação que é impossível executar.

Selecione a alternativa correta que gera um erro em tempo de execução.

- A. Quando o resultado final de uma determinada operação não é o esperado.
- B. Não fechar o par de chaves de um bloco de código.
- C. Não finalizar uma instrução com ponto-e-vírgula.
- D. Dividir um número por zero.

Dicas

Você poderá baixar toda a especificação da Linguagem C# no link da Microsoft <<https://www.microsoft.com/en-us/download/details.aspx?id=7029> (<https://www.microsoft.com/en-us/download/details.aspx?id=7029>)>

Guia de Programação em C# <<https://msdn.microsoft.com/pt-br/library/67ef8sbd.aspx> (<https://msdn.microsoft.com/pt-br/library/67ef8sbd.aspx>) >

Referência de C# <<https://msdn.microsoft.com/pt-br/library/618ayhy6.aspx> (<https://msdn.microsoft.com/pt-br/library/618ayhy6.aspx>) >

REFERÊNCIA

DEITEL, H. M. *et al.* C# Como Programar. São Paulo: Pearson Makron Books, 2003.

OLSSON, Mikael. *C# Quick Syntax Reference*. Berkeley, CA: Apress, 2013.

SHARP, John. *Microsoft Visual C# 2013 Step by Step*. Microsoft Press, 2013.

SITES

Microsoft <<https://msdn.microsoft.com/pt-br/library/x53a06bb.aspx> (<https://msdn.microsoft.com/pt-br/library/x53a06bb.aspx>)>. Acesso em 20/09/2015

Microsoft <<https://msdn.microsoft.com/en-us/library/x53a06bb.aspx> (<https://msdn.microsoft.com/en-us/library/x53a06bb.aspx>)>. Acesso em 20/09/2015

Guia de Programação em C# <<https://msdn.microsoft.com/pt-br/library/67ef8sbd.aspx> (<https://msdn.microsoft.com/pt-br/library/67ef8sbd.aspx>) >. Acesso em 20/09/2015

Referência de C# <<https://msdn.microsoft.com/pt-br/library/618ayhy6.aspx> (<https://msdn.microsoft.com/pt-br/library/618ayhy6.aspx>) >. Acesso em 20/09/2015

