C#
Guia do Programador

Joel Saade

Novatec

Editor: Rubens Prates
Editoração eletrônica: Camila Kuwabata e Carolina Kuwabata
Revisão gramatical: Gabriela de Andrade Fazioni
Capa: Victor Bittow

ISBN: 978-85-7522-253-9

Histórico de impressões:

Janeiro/2011 Primeira edição

Novatec Editora Ltda.
Rua Luís Antônio dos Santos 110
02460-000 – São Paulo, SP – Brasil
Tel.: +55 11 2959-6529
Fax: +55 11 2950-8869
Email: novatec@novatec.com.br
Site: www.novatec.com.br
Twitter: twitter.com/novateceditora
Facebook: facebook.com/novatec
LinkedIn: linkedin.com/in/novatec

Dados Internacionais de Catalogação na Publicação (CIP)
(Câmara Brasileira do Livro, SP, Brasil)

Saade, Joel
C# : guia do programador / Joel Saade. -- São Paulo : Novatec Editora, 2010.

Bibliografia.
ISBN 978-85-7522-253-9

1. C# (Linguagem de programação para computadores) 2. Microsoft Visual C# I. Título.

10-11327 CDD-005.133

Índices para catálogo sistemático:

1. C# : Linguagem de programação : Computadores : Processamento de dados 005.133

CRM20110117

Introdução

O presente trabalho apresenta a linguagem de programação Microsoft C#.

A proposta é mostrar as características, recursos e o potencial do ambiente de desenvolvimento, próprio de C#, tanto no modo console, como no modo gráfico, abordando desde os tópicos básicos até os mais avançados.

Este livro contém 24 capítulos, dos quais os primeiros 22 abordam os seguintes itens: estrutura básica de um programa C#; tipos de dados; operadores; estruturas de decisão e de iteração; arrays; tipos *value* definidos pelo usuário: enumeração e estrutura; classes, objetos e itens relacionados, como construtor, destrutor, classes static, classes parciais, referência `this`, herança de classes e métodos virtuais; métodos e itens relacionados, como passagem de argumentos, parâmetros *ref*, *out* e *params*, sobrecarga de métodos e métodos recursivos; generics, métodos, classes e estruturas genéricas; interfaces e delegates; sobrecarga de operadores; passagem de argumentos a um programa; exceções; formatação de valores numéricos e de data e hora; namespaces; estruturas do namespace `System`; classes `Environment`, `Math`, `Random` e `String`; objetos `StringBuilder`; modo unsafe (uso de ponteiros); coleções genéricas e não genéricas; DLLs; criação de componentes e manipulação de diretórios, drives e arquivos.

Os exemplos desses itens estão na forma de fragmentos de código e de programas-exemplo completos, como Aplicativos de Console, acompanhados de observações, quando necessárias.

Embora os exemplos tenham sido elaborados como Aplicativos de Console, não quer dizer que não poderiam ser elaborados no modo gráfico também, mas obviamente com ajustes necessários, em função das características próprias da interface gráfica. Como o modo gráfico requer a captura de muitas imagens, assim este livro teria um número elevado de páginas.

O modo gráfico, com mais recursos que o modo console, começa no capítulo 23, "Aplicações gráficas", que aborda os seguintes itens: exibição de message boxes; reprodução de sons; menus; controles ListBox e RadioButton; manipulação de fontes e cores; controle TrackBar; manipulação de linhas e formas geométricas; chamada a programas; exibição de imagens; múltiplos forms; uso do clipboard; barra de ferramentas (controle ToolStrip); impressão de arquivos; DLLs e exibição de páginas Web.

O capítulo 24, "Banco de dados", apresenta conceitos sobre o ADO.NET; principais namespaces relacionados ao ADO.NET; provedores de acesso; objetos DataSet,

DataAdapter e DataReader; Data commands; gravação de arquivos no formato XML e vinculação de dados (data binding). Mostra, por meio de programas-exemplo completos, acesso a bancos de dados SQL Server em aplicações C#.

Este livro, embora não esgote o assunto, é bastante abrangente e muito rico em programas-exemplo completos, com muitas ilustrações e observações. Destina-se a todos que pretendem aprender C# de forma definitiva. Pode ser aplicado nos cursos de informática (técnicos e/ou superiores) em que C# é ministrado como disciplina ou ainda como forma de autoestudo.

Todo o trabalho de programação foi desenvolvido com o Visual C# 2010 Express Edition, sob o sistema operacional Windows XP Professional (Service Pack 3) e o Sistema de Gerenciamento de Banco de Dados SQL Server 2005 Express Edition Service Pack 2.

Os programas-exemplo estão disponíveis para download em:

http://www.novatec.com.br/downloads/C#guiaprogr

Boa leitura e boa programação!

Joel Saade

Novembro/2010

Sobre C#

O Visual C# (ou apenas C#) é uma linguagem de programação da Microsoft projetada para criar aplicações diversas, tanto para Windows, como para a Web, que são executadas no .NET Framework.

É uma linguagem simples, moderna, segura quanto a tipos, orientada a objetos e familiar a programadores C, C++ e Java, pois destas herda várias características. Embora herde características dessas linguagens, C# traz novos recursos e conceitos de programação, tais como indexadores, propriedades e delegates.

O código de C# é compilado como um código gerenciado, isto quer dizer que ele se beneficia dos serviços do Common Language Runtime (CLR), que incluem interoperabilidade de linguagens, garbage collection, segurança e melhor suporte ao controle de versões. O seu ambiente de desenvolvimento é altamente interativo com designers visuais para a criação das aplicações.

Da suite Visual Studio, que contempla também o VB.NET, C# é a sua linguagem principal com um número crescente de usuários.

C# está se posicionando como o paradigma no desenvolvimento de aplicações no ambiente Windows.

CAPÍTULO 1

Primeiros passos

Este capítulo apresenta a estrutura básica de um programa C# por meio da criação de um programa do tipo Aplicativo de Console.

Estrutura básica de um programa C#

Consiste de uma classe e seus métodos, mas podem estar presentes ainda namespaces, estruturas, interfaces, enumerações, eventos e delegates.

Exemplo: Estrutura básica de um programa C# (programa hello.cs)

```
using System;
class Principal {
    public static void Main() {
        Console.WriteLine("Hello C# World!");
    }
}
```

Observações:

- `using`: uso de um namespace, no caso, do namespace `System`.
- `class`: declaração de uma classe, no caso, da classe `Principal`.
- `public static void Main()`: cabeçalho do método `Main()`. Método obrigatório, em que a execução de um programa começa e termina. Deve ser um método `static` e deve estar contido em uma classe ou estrutura.
- `Console.WriteLine(" ... ");`: uso do método `WriteLine()`, da classe `Console`, do namespace `System`. Exibe informações na saída-padrão.
- Classes e métodos são iniciados e terminados por um par de chaves ({ ... }).
- A extensão de programas-fonte C# é `.cs`.

Agora vamos fazer do programa `hello.cs` uma aplicação em C#.

- Inicie o C#. É exibida a página inicial (Figura 1.1).
- Clique no menu **File | New Project**. É exibida a janela `New Project`.
- Selecione `Console Application` e, na caixa de texto `Name`, digite `Hello`. Clique no botão `OK`. É exibida a janela do editor de código com o programa-fonte padrão `Program.cs` (Figura 1.2).

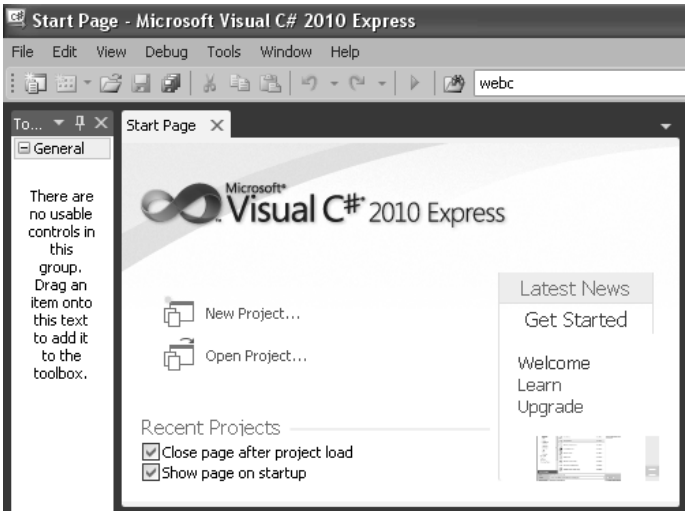

Figura 1.1 – Página inicial de C#.

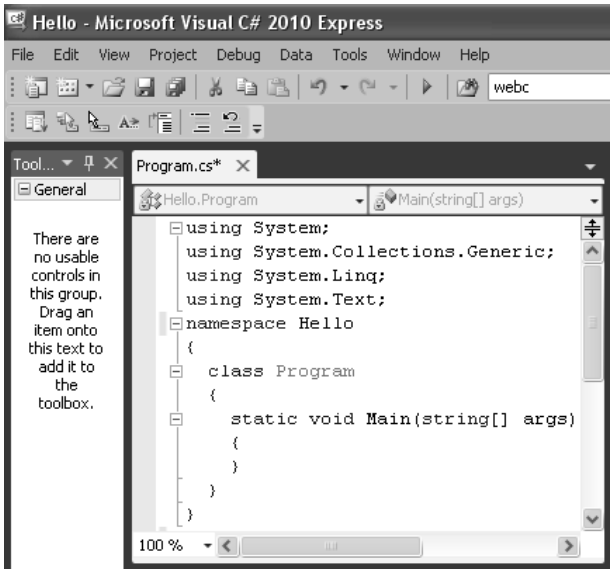

Figura 1.2 – Janela do editor de código: programa-fonte padrão.

- Digite a linha `Console.WriteLine("Hello C# World!");` no método `Main()`. O programa-fonte está conforme a figura 1.3.
- Clique no menu **File|Save All** para salvar o projeto. É exibida a janela `Save Project`.
- Na caixa de texto `Name`, digite `Hello`.

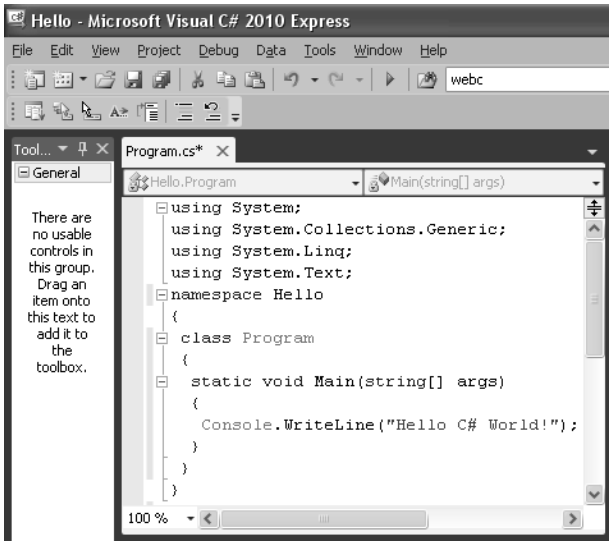

Figura 1.3 – Janela do editor de código: programa Hello.cs.

- Na caixa de texto `Location`, digite `C:\ProgCSharp`. A janela `Save Project` está assim (Figura 1.4):

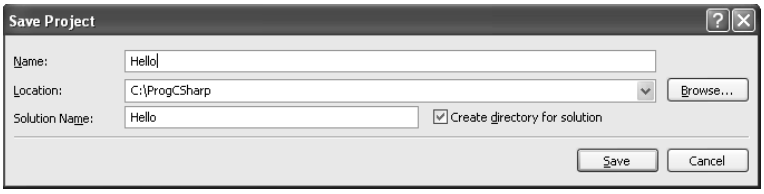

Figura 1.4 – Janela Save Project.

- Clique no botão `Save`. Caso a pasta especificada na caixa de texto `Location` não exista, será criada, uma vez que a opção `Create directory for solution` está marcada.
- Clique no menu **File | Save Program.cs As** para salvar o programa `Program.cs` com outro nome. É exibida a janela `Save File As` (Figura 1.5).
- Na caixa de texto `Nome do objeto`, digite `Hello.cs`. Clique no botão `Salvar`.
- Pressione `F6` ou clique no menu **Debug | Build Solution**. Será criado o programa executável `Hello.exe` no caminho `C:\ProgCSharp\Hello\Hello\bin\Release`.
- Clique no menu **File | Save All**.
- Pressione `Ctrl + F5` para executar a aplicação. A figura 1.6 mostra o resultado.

Embora seja um programa simples, o objetivo foi apresentar a estrutura básica de um programa C# e mostrar como criar uma aplicação do tipo Aplicativo de Console.

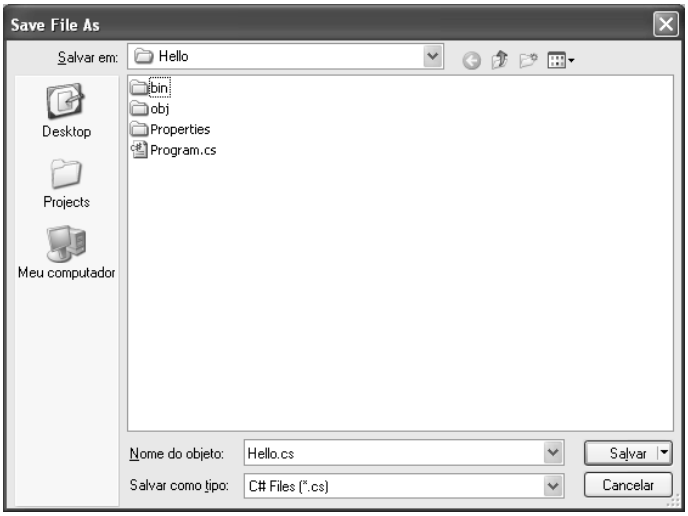

Figura 1.5 – Janela Save File As.

Figura 1.6 – Janela de resultado.

Sobre os programas-exemplo

- Para os programas-exemplo que exigirem a criação de um projeto, será usada a pasta `C:\ProgCSharp`.
- A partir daqui e até o capítulo 22, "Diretórios, drives e arquivos", serão apresentados inúmeros programas-exemplo, como `Aplicativos de Console`, e nestes serão omitidos os namespaces padrões.
- Muitos dos programas-exemplo disponíveis para download, pelo fato de não usarem recursos adicionais do IDE, estão na forma de programas-fonte únicos, não na forma de projetos, como esse primeiro. Para usar tais programas, crie um novo projeto, como Console Application, e na janela do editor de código elimine todas as linhas e copie o programa-fonte na janela do editor de código.