

C# プログラミング入門 2

この入門書では「C# プログラミング入門 1」に引き続き C# 言語によるプログラミングについて説明します。プログラムを書いた経験のない方を対象に、サンプルを交えながら C# 言語の構文規則やプログラミングの際に知っておく必要がある事柄についてテーマごとに説明しています。

本入門では開発ツールとして Visual Studio を使用します。Visual Studio には入門者向け、プロフェッショナル プログラマ向け、企業や団体向けなどの、いくつかのエディションがあります。入門者向けのエディションである「Visual C# 2005 Express Edition 日本語版」は以下の URL のサイトから無料でダウンロードできます。

<http://www.microsoft.com/japan/msdn/vstudio/express/vcsharp/>

* Microsoft、Windows、Windows Vista、Visual Studio、Visual C#、Visual Basic は、米国 Microsoft Corporation の米国及びその他の国における登録商標または商標です。

* Windows の正式名称は、Microsoft Windows Operating System です。

© 2007 Microsoft Corporation. All rights reserved.

■ 文書情報

タイトル	ねらい	章題	キーワード
32. 論理演算子 【サンプル プログラム有り】	● bool 型に対する論理演算子の使い方について理解します。	1. 論理演算子 2. & 演算子、&& 演算子 3. 演算子、 演算子 4. ^ 演算子 5. ! 演算子 6. 論理演算子の優先順位 演習課題	ショートサーキット
33. キャスト演算子	● キャスト演算子の使い方について理解します。	1. キャスト演算子 2. 演算子の優先順位	キャスト演算子
34. 多次元配列 【サンプル プログラム有り】	● 多次元配列の使い方について理解します。	1. 多次元配列 2. 多次元配列の宣言と作成 3. 多次元配列の使用 演習課題	多次元配列 ジャグ配列

タイトル	ねらい	章題	キーワード
35. ジャグ配列 【サンプル プログラム有り】	<ul style="list-style-type: none"> ● ジャグ配列の使い方について理解します。 	<ol style="list-style-type: none"> 1. ジャグ配列 2. ジャグ配列の宣言と作成 3. ジャグ配列の使用 演習課題	ジャグ配列
36. メンバ	<ul style="list-style-type: none"> ● クラスのメンバについて理解します。 ● メンバのアクセス修飾子および、静的メンバの使い方について理解します。 	<ol style="list-style-type: none"> 1. メンバの種類 2. メンバの定義 3. 静的メンバ 4. メンバの利用 	メンバ アクセス修飾子 静的メンバ 静的フィールド 静的メソッド インスタンス メンバ インスタンス フィールド インスタンス メソッド
37. フィールド	<ul style="list-style-type: none"> ● フィールドと定数の定義の仕方について理解します。 	<ol style="list-style-type: none"> 1. フィールド 2. 定数 3. 読み取り専用フィールド 	フィールド 定数
38. メソッド	<ul style="list-style-type: none"> ● メソッドの定義の仕方について、特にオーバーロードについて理解します。 	<ol style="list-style-type: none"> 1. メソッド 2. オーバーロード 	オーバーロード
39. コンストラクタ 【サンプル プログラム有り】	<ul style="list-style-type: none"> ● コンストラクタとインスタンス生成について理解します。 ● null リテラルについて理解します。 	<ol style="list-style-type: none"> 1. コンストラクタとは 2. null リテラル 	コンストラクタ null
40. デストラクタ	<ul style="list-style-type: none"> ● デストラクタの使い方について理解します。 	<ol style="list-style-type: none"> 1. デストラクタ 	デストラクタ
41. 例外処理 【サンプル プログラム有り】	<ul style="list-style-type: none"> ● try/catch による例外処理の仕方について理解します。 	<ol style="list-style-type: none"> 1. 例外 2. 例外処理 3. 例外のキャッチ 4. finally ブロック 5. 例外のスロー 	例外処理
42. 付録 便利なクラス 2 【サンプル プログラム有り】	<ul style="list-style-type: none"> ● 標準のクラス ライブラリの中でよく使われるいくつかのクラスについて紹介します。 	<ol style="list-style-type: none"> 1. StringBuilder クラス (System.Text.StringBuilder) 2. Random クラス (System.Random) 3. TimeSpan 構造体 (System.TimeSpan) 4. Directory クラス (System.IO.Directory) 	
43. 付録 演算子の優先順位	<ul style="list-style-type: none"> ● 今回のシリーズで取り上げた演算子の優先順位についてまとめて紹介します。 	<ol style="list-style-type: none"> 1. 演算子の優先順位 	

■ キーワード索引

キーワード	章題
英	
.NET Framework	29. 名前空間とライブラリ
C# 言語	1. 初めてのプログラミング
case ステートメント	21. 条件分岐
Console クラス	5. 標準出力
Convert クラス	13. 文字列型
CPU	2. プログラムが動くしくみ
do ループ	22. ループ
false	12. 基本型
for ループ	22. ループ
foreach ループ	22. ループ
if ステートメント	21. 条件分岐
Main メソッド	3. Main メソッド
null	39. コンストラクタ
OS	2. プログラムが動くしくみ
Parse メソッド	13. 文字列型
Read メソッド	6. 標準出力
ReadLine メソッド	6. 標準出力
switch ステートメント	21. 条件分岐
ToString メソッド	13. 文字列型
true	12. 基本型
Unicode 文字	12. 基本型
using ディレクティブ	29. 名前空間とライブラリ
while ループ	22. ループ
あ	
アクセス修飾子	36. メンバ
値型	11. データ型
暗黙の型変換	12. 基本型
インスタンス	28. クラスの概要
インスタンス メンバ	36. メンバ
インスタンス フィールド	36. メンバ
インスタンス メソッド	36. メンバ
インデックス	20. 配列
エスケープ シーケンス	12. 基本型 13. 文字列型

キーワード	章題
演算子	14. 算術演算子
オーバーフロー	14. 算術演算子
オーバーロード	38. メソッド
オブジェクト指向言語	31. 付録 C# 言語の特徴
か	
ガベージ コレクション	31. 付録 C# 言語の特徴
完全修飾名	29. 名前空間とライブラリ
キャスト演算子	33. キャスト演算子
キーボード	2. プログラムが動くしくみ
組み込み型	11. データ型
クラス	28. クラスの概要
共通言語ランタイム	29. 名前空間とライブラリ
クラス ライブラリ	29. 名前空間とライブラリ
桁落ち	15. 演算精度
構造体	27. 構造体
コード エディタ	1. 初めてのプログラミング
コマンド プロンプト	1. 初めてのプログラミング
コメント	4. コメント
コンストラクタ	39. コンストラクタ
コンソール アプリケーション	1. 初めてのプログラミング
コンパイル エラー	23. エラーと警告
コンポーネント指向	31. 付録 C# 言語の特徴
さ	
参照	7. 変数
参照型	11. データ型
実行	1. 初めてのプログラミング
実行形式のファイル	2. プログラムが動くしくみ
実行時例外	23. エラーと警告
実数値リテラル	12. 基本型
ジャグ配列	34. 多次元配列 35. ジャグ配列
修飾子	24. 関数 (メソッド)
順次制御	21. 条件分岐
情報欠落	15. 演算精度
書式情報	5. 標準出力

キーワード	章題
ショートサーキット	32. 論理演算子
ステートメント	21. 条件分岐
整数リテラル	12. 基本型
静的メンバ	36. メンバ
静的フィールド	36. メンバ
静的メソッド	36. メンバ
ソース コード	2. プログラムが動くしくみ
ソース ファイル	1. 初めてのプログラミング 2. プログラムが動くしくみ
た	
代入	7. 変数
タイプ セーフ	31. 付録 C# 言語の特徴
多次元配列	34. 多次元配列
単純型	11. データ型
定数	37. フィールド
ディスプレイ	2. プログラムが動くしくみ
デストラクタ	40. デストラクタ
統合開発環境	1. 初めてのプログラミング
な	
名前空間	3. Main メソッド 29. 名前空間とライブラリ
は	
ハードディスク	2. プログラムが動くしくみ
配列	20. 配列
配列の要素	20. 配列
バグ	23. エラーと警告
パラメータ	24. 関数 (メソッド)
引数	24. 関数 (メソッド)
引数の型	24. 関数 (メソッド)

キーワード	章題
標準入出力	5. 標準出力
ビルド	2. プログラムが動くしくみ
フィールド	27. 構造体 37. フィールド
プログラム	1. 初めてのプログラミング
変数	7. 変数
変数名	7. 変数
ま	
マウス	2. プログラムが動くしくみ
丸め誤差	15. 演算精度
明示的な型変換	12. 基本型
メソッド	3. Main メソッド 24. 関数 (メソッド)
メソッド名	24. 関数 (メソッド)
メモリ	2. プログラムが動くしくみ
メンバ	28. クラスの概要 36. メンバ
文字列リテラル	13. 文字列型
戻り値	24. 関数 (メソッド)
戻り値の型	24. 関数 (メソッド)
や	
ユーザー定義型	11. データ型
ら	
リテラル	7. 変数とリテラル
例外処理	31. 付録 C# 言語の特徴 41. 例外処理
列挙型	26. 列挙型
列挙子	26. 列挙型