

NEW PRODUCTS

Visual Studio.NET/.NET Framework関連ツール新製品レビュー



タイムビューVersion3

多彩な表現力を持ったガントチャートコントロール



せきぐち くみこ
SEKIGUCHI, Kumiko

問
合
先

株式会社ナルボ

TEL : 03-6821-7755

FAX : 03-5475-1881

URL : <http://www.knowlbo.co.jp/>

MAIL : knocx@knowlbo.co.jp

Technology Tools

- ☒ Visual Basic .NET
- ☐ Visual C# .NET
- ☐ SQL Server 2000
- ☐ Oracle 9i
- ☐ Access 2002
- ☐ ASP.NET
- ☐ Internet Information Services
- ☐ Other:

Environment

ターゲットOS
Windows 98/Me/NT4.0/2000/XP/
Server 2003

対応開発環境
Visual Basic 6.0
Visual Studio .NET 2002/2003
Internet Explorer 5.0/6.0
Access 2002

価格
基本パッケージ : 92,400円
(1開発ライセンス、10ランタイムライセンス)

はじめに

システム開発の作業には、受注からリリースまでにさまざまな工程があります。みなさんは作業工程をどのように管理されているでしょうか。おそらく、多くの方が予定や進捗をひと目で把握できるグループウェアやプロジェクト管理ツールを利用されていることと思います。

では、プロジェクト管理ツールのようなアプリケーションを開発することになったら、みなさんはどうしますか。スケジュールの表示や入力インターフェイスに適したコントロールはVisual Basicの標準コントロールには存在しません。複数のコントロールを組み合わせることで実現することもできるでしょうが、作成にかかる工数を考えると得策ではありません。

そこでご紹介したいのが、株式会社ナルボの「タイムビュー Version 3 (以下タイムビュー)」です。

タイムビューとは？

タイムビューは、さまざまな時間スケジュール情報をガントチャート形式

で表示するコントロールです(図1)。スケジュールの変更は、マウスによるドラッグ&ドロップで移動するなど、直感的に操作ができることも大きな特徴といえます。

アプリケーションからプロジェクトやスケジュールを扱うためには、さまざまな機能が必要になります。それらを実現するタイムビューの主要な機能を紹介します。

■タイムスケール

図1の上部の日付が刻まれた部分を「タイムスケール」といいます。タイムスケールの目盛りは、年単位から分単位まで指定可能です。目盛り表示は、大区分、中区分、小区分の3種類があり、それぞれ、目盛りの単位と間隔が設定できます。したがって、年単位の長期にわたるプロジェクトから短期の日程表まで、さまざまなパターンに対応できます。

■ピース

ひとつの仕事や予定は、「ピース」という単位で入力します。ピースの表示形状は、開始日時と終了日時を結ぶバーの部分と、バーの両端それぞれにつ

図1：タイムビューのコントロールイメージ

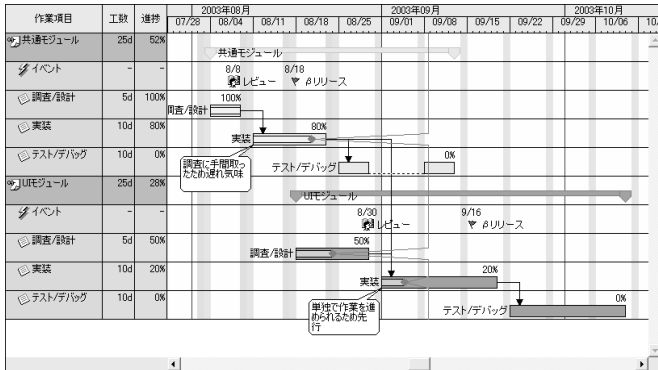
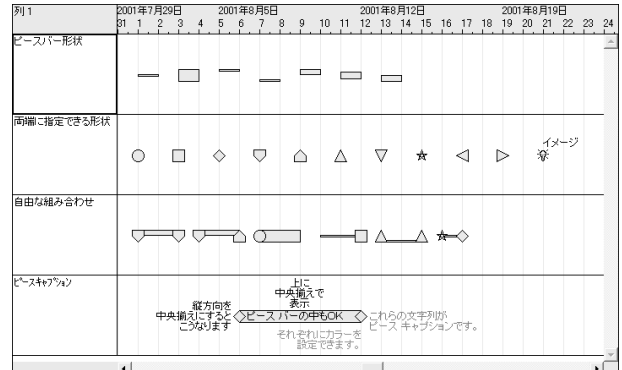


図2：ピースのパターン



いて豊富なパターンが提供されています (図2)。ピースを構成する要素や色を組み合わせて、スケジュールの種類の違いなどを明確に表現することができます。

また、達成率やピース同士の関係線を設定したり、ピースに対するコメントを吹き出しの形式で表示したり、進捗状況を明確に表すための稲妻線を表示したりなど、実用的かつ高い視覚効果を与える機能が充実しています。この他にも、各行に複数のピースを配置したり、ピースを並行に配置することも可能です。

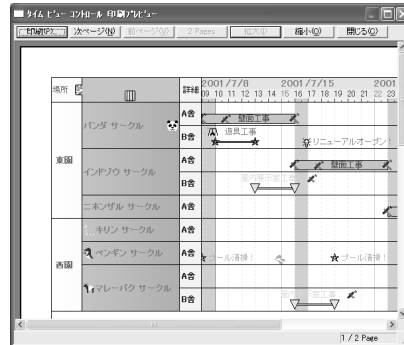
■印刷機能

タイムビューは、専用の印刷および印刷プレビュー機能を提供しています (図3)。この機能を利用すれば、コントロールのみを複数ページに印刷することができます。印刷する行範囲/時間範囲の指定や、余白、倍率、用紙方向、ヘッダ/フッタ領域の高さの指定など、きめ細やかな設定ができるほか、強制改ページの指定や印刷対象日時の範囲をタイムスケールの時間区切りに適合させることも可能です。

■Internet Explorer対応

タイムビューのコントロールコンテンツは、Visual Basic 6.0、Visual Studio

図3：印刷プレビュー画面



.NET 2002/2003、Access 2002だけでなく、Internet Explorer 5.0/6.0でも利用できます (図4)。Webブラウザを利用して、スケジュール管理のように操作性も重視されるアプリケーションを作成するのは容易ではありません。しかし、タイムビューを使えば、Webブラウザを利用したシステムであっても操作性を損なうことなく、機能的なアプリケーションを作成することができます。

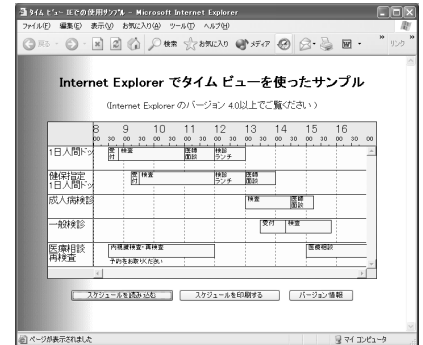
■スケジュール管理アプリを作ろう！

それでは、Visual Basic .NET (以下VB.NET) を使って簡単なスケジュール管理を行なうWindowsアプリケーションを作成してみましょう。

■コントロールの準備

はじめに、タイムビューをVB.NETのIDE上で使用するために、ツールボックスにコンポーネントを追加しまし

図4：Internet Explorerでの実行画面



う。「ツールボックスのカスタマイズ」ダイアログボックスを開き、「COMコンポーネント」タブで、「Knowlbo/OCX Time View」にチェックを付けて [OK] ボタンをクリックすると、ツールボックスにコントロールが追加されます (図5)。

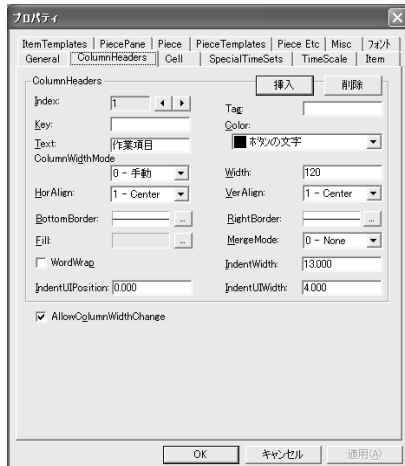
続いて、ツールボックスから、タイムビューとチェックボックス、ボタンをそれぞれひとつずつ配置します。

タイムビューは、コントロールを貼り付けただけでは、見出し (カラムヘッダ) も空白のままですし、タイムスケールもとてもシンプルな状態です。ここから、作成するアプリケーションの用途に応じて、タイムスケールなど

図5：タイムビューコントロール



図6：プロパティページ



を設定してゆかなければなりません。通常ならば、たくさんのプログラムコードを記述しなければならないところですが、タイムビューはプロパティページで設定できる項目が充実しています(図6)。プロパティページを活用して、効率よくプログラミングを行ないましょう。

最終的なフォームのデザインは、図7のようにしました。

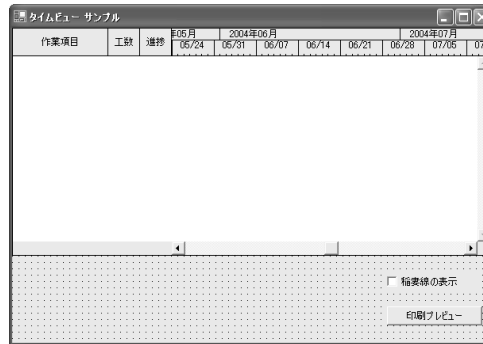
■コーディングのポイント

このサンプルプログラムでは、表示する行とピースをあらかじめフォームのロード時に設定しています。ピースには、稲妻線の達成率ポイントも設定し、[稲妻線の表示] チェックボックス(chkVisibleProgressLine)のチェンジェイventで稲妻線の表示/非表示を切り替えられるようにしました。

[印刷プレビュー] ボタン(btnPreview)では、タイムビュー(tvProject)に表示されているデータのプレビューを行ないます。

また、ピースが追加されたとき(AfterPieceAddイベントが発生)には、既定値のピースではなく、指定した形状のピースを追加するようにしましょう。

図7：フォームのデザイン



フォームロード時に表示する行とピース

まずは、フォームのロードイベントから見てゆきます(リスト1)。少しコード量が多いと思われるかもしれませんが、内容はいたって簡単で、表示したいオブジェクトの追加メソッド(Addメソッド)を呼び出してオブジェクトを追加し、色や文字列、表示位置などを設定しているだけです。

たとえば「稲妻線」であれば、ProgressLinesオブジェクトのAddメソッドを呼び出して、コントロールに稲妻線を追加します。そして、稲妻線を表示する位置の日時を「2004年9月8日」に設定するといった具合です。

なお、稲妻線は、達成率ポイントを追加しなければ垂直な線です。進捗を把握したいピースの数だけProgressPointsオブジェクトを追加し、対象となるピースをProgressPointsオブジェクトのTargetPieceプロパティに設定しておかなければなりません。

稲妻線の表示/非表示

次は、稲妻線の表示チェックボックス(chkVisibleProgressLine)のチェンジェイventです(リスト2)。ここでは、フォームロード時に追加した稲妻線を、チェックボックスの状態に応じて表示したり隠したりしています。

図8：ユーザー操作で追加されるピース



印刷プレビュー

続いて、[印刷プレビュー] ボタン(btnPreview)のクリックイベントです(リスト3)。複雑なコードを記述する必要は一切ありません。ご覧の通り、PrintPreviewメソッドを呼び出すだけという単純さです。

ここでは、何も指定しませんでした。メソッドのパラメータである印刷情報(PrintInfoオブジェクト)には、表1の情報を指定することができます。

ピース追加時の処理

最後は、ユーザーによるピースの追加操作が完了したときに発生する「AfterPieceAddイベント」のコードを記述します(リスト3)。このイベントに何も記述しなければ、タイムビューには既定値のピースが描画されます。追加したピースを指定した形状にしたいときや、開始日時をピースのキャプションとして設定したいときに、このイベントにコードを記述するとよいでしょう。

ピースの形状を変更するときは、既定のピース追加操作をキャンセルしてから、イベントのデータを格納している情報(e)を元に、ピースを追加しておします。今回のサンプルでは、ユーザー操作によって追加されるのは図8のピースにしました。

サンプルプログラムの実行画面は、

リスト1：フォームのロードイベント

```

Private Sub frmTasks_Load(ByVal eventSender As System.Object, _
    ByVal eventArgs As System.EventArgs) Handles MyBase.Load
    Dim pieces(2) As KnTreeViewLib.Piece
    Dim progressLine As KnTreeViewLib.ProgressLine
    Dim itemWork As KnTreeViewLib.Item
    Dim itemRow As KnTreeViewLib.Item
    Dim pieceWork As KnTreeViewLib.Piece
    Dim captionProgress As KnTreeViewLib.PieceCaption
    Dim progressWork As KnTreeViewLib.Progress
    Dim progressPoint As KnTreeViewLib.ProgressPoint

    (略)

    ' 稲妻線の追加
    progressLine = tvProject.ProgressLines.Add()
    progressLine.TargetTime = #9/8/2004 6:00:00 PM#
    chkVisibleProgressLine_CheckStateChanged( _
        chkVisibleProgressLine, New System.EventArgs)

    ' 段落1の作成
    itemWork = tvProject.Items.Add
    With itemWork
        (略)

        ' ピースの追加
        pieceWork = itemWork.Pieces.Add
        With pieceWork
            .Start = #8/4/2004#
            .Finish = #9/13/2004#

            .BarShape.Fill.BackColor = _
                System.Convert.ToUInt32(&H99FF99)
            .BarShape.Line.Color = _
                System.Convert.ToUInt32(&H77DD77)
            .StartShape.Fill.BackColor = _
                System.Convert.ToUInt32(&H99FF99)
            .StartShape.Line.Color = _
                System.Convert.ToUInt32(&H77DD77)
            .FinishShape.Fill.BackColor = _
                System.Convert.ToUInt32(&H99FF99)
            .FinishShape.Line.Color = _
                System.Convert.ToUInt32(&H77DD77)

            .BarShape.Shape = _
                KnTreeViewLib.TivBarShape.tivBarShapeUpperRectangle
            .StartShape.Shape = _
                KnTreeViewLib.TivPointShape.tivPointShapeHomeBase
            .FinishShape.Shape = _
                KnTreeViewLib.TivPointShape.tivPointShapeHomeBase
        End With

        ' キャプションの追加
        captionProgress = pieceWork.Captions.Add
        With captionProgress
            .Text = "標準モジュール"
            .Position = _
                KnTreeViewLib.TivPieceCaptionPosition.tivPieceCenter
            .HorAlign = _
                KnTreeViewLib.TivTextHorAlign.tivTextHorAlignLeft
        End With
    End With

    (略)

    ' 「調査/設計」用の行を追加
    itemRow = tvProject.Items.Add
    With itemRow
        .Cells.Item(1).Value = "調査/設計"
        .Cells.Item(1).IndentLevel = 2
        .Cells.Item(2).Value = "5d"
        .Cells.Item(2).HorAlign = _
            KnTreeViewLib.TivTextHorAlign.tivTextHorAlignRight
        .Cells.Item(3).Value = "95%"
        .Cells.Item(3).HorAlign = _
            KnTreeViewLib.TivTextHorAlign.tivTextHorAlignRight

        .PiecePane.Fill.BackColor = _
            System.Convert.ToUInt32(&HEEEFFF)
        .Cells.Item(1).Fill.BackColor = _
            Convert.ToUInt32(&HEEDDCC)
        .Cells.Item(2).Fill.BackColor = _
            Convert.ToUInt32(&HEEDDDD)
        .Cells.Item(3).Fill.BackColor = _
            Convert.ToUInt32(&HEEDDDD)
    End With

    ' 「調査/設計」のピースを追加
    pieceWork = itemRow.Pieces.Add
    With pieceWork
        .Start = #8/4/2004#
        .Finish = #8/9/2004#
        .BarShape.Fill.BackColor = Convert.ToUInt32(&H99FF99)
    End With

    ' 「調査/設計」の達成率を追加
    progressWork = pieceWork.Progresses.Add
    progressWork.PercentFrom = 0
    progressWork.PercentTo = 0.95
    progressWork.Fill.BackColor = Convert.ToUInt32(&HCCFFCC)

    ' 稲妻線のポイントを追加
    progressPoint = _
        tvProject.ProgressLines.Item(1).ProgressPoints.Add()
    With progressPoint
        .TargetPiece = pieceWork
        .PointShape.Shape = _
            KnTreeViewLib.TivPointShape.tivPointShapeDiamond
        .Height = 10
        .Width = 10
    End With

    (略)
End Sub

```

リスト2：稲妻線の表示チェックボックス

```

Private Sub chkVisibleProgressLine_CheckStateChanged( _
    ByVal eventSender As System.Object, _
    ByVal eventArgs As System.EventArgs _
) Handles chkVisibleProgressLine.CheckStateChanged
    ' チェックされているとき
    If chkVisibleProgressLine.CheckState = _
        System.Windows.Forms.CheckState.Checked Then
        ' 稲妻線を表示し、現在時間の非表示にする
        tvProject.ProgressLines.Item(1).Hidden = False
        tvProject.CurrentTime.Hidden = True
    Else
        ' 稲妻線を非表示にし、現在時間の表示する
        tvProject.ProgressLines.Item(1).Hidden = True
        tvProject.CurrentTime.Hidden = False
    End If
End Sub

```

リスト3：印刷プレビュー

```
Private Sub btnPreview_Click(ByVal sender As System.Object, _
    ByVal e As System.EventArgs) Handles btnPreview.Click
    Dim pi As New KnTVViewLib.PrintInfo

    ' 印刷プレビュー
    tvProject.PrintPreview(pi)
End Sub
```

表1：PrintInfoオブジェクトのプロパティ

プロパティ	内容
AllColumnsAtFirstPage	印刷する時間範囲の先頭時間を含むページにすべての列を印刷するかどうか
BodyBorder	ボディを囲む線の描画属性
ChartBorder	チャート部分の描画を囲む線の描画属性
FinishColumn	印刷する列範囲の終了列のインデックス
FinishIndex	印刷する行範囲の終了行のインデックス
FinishTime	印刷する時間範囲の終了日時
FooterHeight	フッタ領域の高さをセンチメートル単位で示す
HeaderHeight	ヘッダ領域の高さをセンチメートル単位で示す
JustClippingToScale	時間目盛を改ページ時、および右端ページでどのようにクリッピングするか
Magnification	印刷範囲の拡大、または縮小する率を設定
MarginBottom	用紙下側の余白をセンチメートル単位で示す
MarginLeft	用紙左側の余白をセンチメートル単位で示す
MarginRight	用紙右側の余白をセンチメートル単位で示す
MarginTop	用紙上側の余白をセンチメートル単位で示す
MonochromeText	印刷する文字の色を白、または黒に変換して出力するかどうか
Orientation	用紙の方向
PaperSize	用紙サイズ
StartColumn	印刷する行範囲の開始列のインデックス
StartIndex	印刷する行範囲の開始行のインデックス
StartTime	印刷する時間範囲の開始日時
Title	「印刷プレビュー」ウィンドウや「印刷中」ダイアログのウィンドウタイトルに表示する名前
ZoomType	拡大／縮小の方式

図9のようにになります。

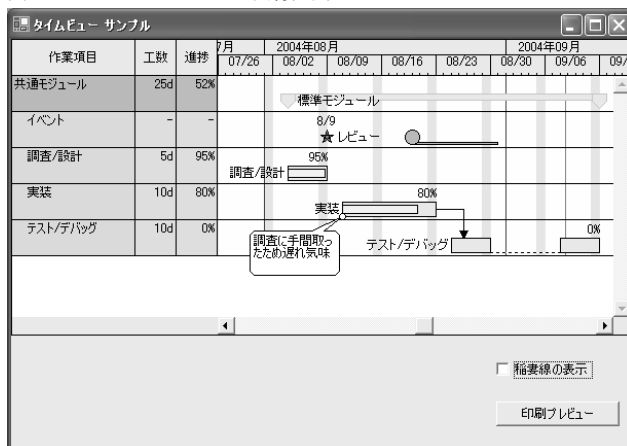
■ おわりに

タイムビューはさまざまなオブジェクトを提供しているため、プロパティの設定が面倒だと思われるかもしれませんが、しかし、前述したとおり、充実したプロパティページも提供されているので、すべてをプログラムコードで記述する必要はありません。

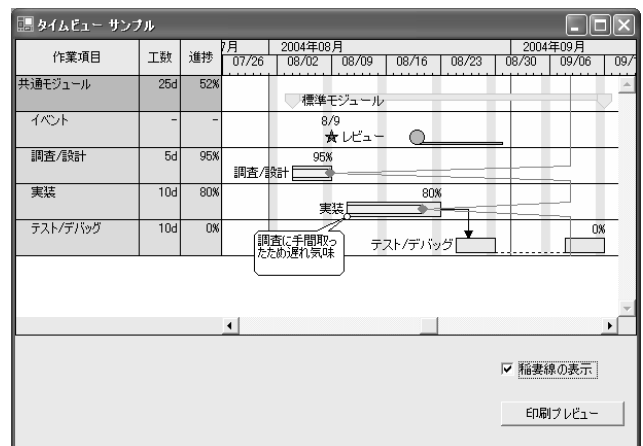
また、タイムビューには、セル結合、タイムスケール、ピース、アイテム移動操作、セルのインデントの変更操作、セルのツリー表現など、基本的に機能単位で、数多くのサンプルプログラムが用意されています。ヘルプファイルを読んだだけではわかりにくい機能でも、サンプルプログラムを動かせば、動作や使い方をより深く理解できるでしょう。

今回は紹介しきれませんでしたでしたが、ピースなどのオブジェクトにあらかじめ用意しておいた雛形を適用する手段や、ピース表示領域の一部の時間帯を任意の塗りつぶしパターンやカラーで描画する「特別時間帯」など、実に多くの機能が提供されています。スケジュール／プロジェクト管理アプリケーションの開発に必要な機能をすべて備

図9：サンプルプログラムの実行画面



稲妻線が非表示の場合



稲妻線を表示した場合

えている、といっても過言ではないでしょう。タイムビューの高い機能と使いやすさを、ぜひ実感してみてください。

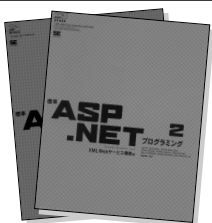
なお、株式会社ナルポでは、ユーザーインターフェイスのコア部分にタイムビューを使用した「Time Scheduler .NET」も開発/販売しています。Time Scheduler .NETは、.NET Framework 対応のWebスケジューラで、「メンバスケジュール」「設備/備品予約」「プロジェクト管理」といった機能を提供しています。また、PCだけでなく、Pocket PC、携帯電話 (i-mode、Ezweb、Vodafone Live!) に対応しているのも大きな魅力です。Webスケジューラの開発を予定されている方は、タイムビューだけでなく、Time Scheduler .NETも検討されるとよいでしょう。

リスト4: ピース追加時の処理

```
Private Sub tvProject_AfterPieceAdd(ByVal sender As Object, _
    ByVal e As AxKnTVViewLib_DKnTVViewEvents_AfterPieceAddEvent _
) Handles tvProject.AfterPieceAdd

    ' デフォルトのピース追加操作を取り消し
    e.Cancel.Value = True

    ' 開始位置が丸く、ピースバーの形状が下線のピースを追加
    Dim pieceWork As KnTVViewLib.Piece
    pieceWork = e.item.Pieces.Add
    With pieceWork
        .Start = e.start
        .Finish = e.finish
        .BarShape.Shape = _
            KnTVViewLib.TivBarShape.tivBarShapeLowerLine
        .StartShape.Shape = _
            KnTVViewLib.TivPointShape.tivPointShapeCircle
        .BarShape.Fill.BackColor = Convert.ToInt32(&HCC99FF)
        .StartShape.Fill.BackColor = Convert.ToInt32(&HCC99FF)
    End With
End Sub
```



標準ASP.NETプログラミング ① ~Webアプリケーション構築編~ 標準ASP.NETプログラミング ② ~XML Webサービス構築編~

SCOTT MITCHELL 他 著 西谷亮 監訳

定価① 4,410円 (本体4,200円+税5%) ② 3,990円 (本体3,800円+税5%) 各 B5変型判 ① 640ページ ② 528ページ ① ISBN4-7981-0125-7 ② ISBN4-7981-0126-5
本書は、中～上級者レベルの開発者を対象に、ASP.NETを使ったさまざまなソリューションの構築について、「Webアプリケーション構築編」と「XML Webサービス構築編」の2巻構成で説明します。実行画面やサンプルコードを豊富に紹介しているので、経験の浅い開発者でもASP.NETの奥深い技術をしっかりと学ぶことができます

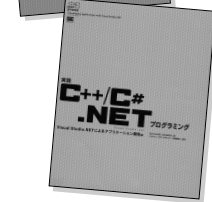


標準VB.NETプログラミング ~.NET環境への移行と開発の基礎~

DAN APPLEMAN 著 グレープシティ株式会社 監訳

定価4,410円 (本体4,200円+税5%) B5変型判 640ページ ISBN4-7981-0216-4

本書は、VB6.0の中～上級ユーザーを対象に、VB6.0からVB.NETへの移行をテーマにしています。VBを知り尽くしているダニエル・アップルマンが、これまでVisual Basicを使い続けてきた開発者がつまづきやすいところを丁寧に解説します。また、豊富なサンプルコードを掲載しているので、迷うことなく.NET対応プログラムを開発することができます。

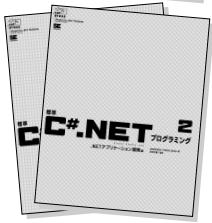


実践C++/C#.NETプログラミング Visual Studio .NETによるアプリケーション構築

RICHARD GRIMES 著 ハラバン・メディアテック 宇野俊夫監訳

定価6,279円 (本体5,980円+税5%) B5変型判 744ページ ISBN4-7981-0348-9

本書は、公式解説書やその他の解説書を卒業したC++/C#プログラマーが、Visual Studio .NETを使って、より実践的、実用的なプログラミングテクニックを身に付けるための書籍です。1～4章では.NET Framework、5～7章ではVisual Studio .NETでアプリケーション開発に利用できるツール類、8～9章ではアプリケーションの開発とデバッグについて取り上げ、著者が独自にIL (中間言語) を逆アセンブルしてハック (解析) した結果をもとにして、様々なテクニックやノウハウが紹介されています。



標準C#.NETプログラミング ① ~C#言語構文編~ 標準C#.NETプログラミング ② ~.NETアプリケーション開発編~

ANDREW TROELSEN 著 矢沢久雄 監訳

定価① 3,360円 (本体3,200円+税5%) ② 3,360円 (本体3,200円+税5%) 各 B5変型判 ① 448ページ ② 608ページ ① ISBN4-7981-0111-7 ② ISBN4-7981-0112-5
本書は、中～上級者レベルの開発者を対象に、C#言語の詳細および.NET対応プログラムの開発技法を解説した書籍です。「C#言語を知りたい、.NETも知りたい」という開発者にお勧めです。

株式会社翔泳社 東京都新宿区舟町5 〒160-0006
出版局営業部: TEL.03-5362-3810 FAX.03-5362-3817 巻末の振替用紙で直接ご購入いただけます。

プロフェッショナルSEの知的探求心を満足させる日本初のITセレクトショップ
<http://www.seshop.com/>

SEshop.com

SE
SHOEISHA