

わかりすぎる



# Visual Basic 2013

## の教科書



サンプルゲームを動かしながら  
VBプログラムの基礎が学べる！

中島省吾 著

SCC

わかりすぎる

Visual  
Basic 2013  
の教科書

中島省吾 著

SCC

※本書の使用(本書のとおりに操作を行う場合を含む)により、万一損害が発生しても、出版社、著作権者は一切の責任を負いかねますので、あらかじめご了承ください。

※Microsoft、Windows は米国 Microsoft Corporation の米国およびその他の国における登録商標です。

※その他、本書に記載されている会社名・製品名などは、一般にそれぞれ各社の商号・登録商標または商標です。

※本書では™および®の記載は省略しました。



# はじめに

Visual Basic が米国で始めて発売されたのが 1991 年のことですから、Visual Basic は既に 20 年以上の歴史があることとなります。1991 年といえば、Microsoft が Windows 3.0 を発売した翌年です。それまで Windows 用のアプリケーションを開発するには、C 言語と呼ばれるプログラミング言語を使い高度なスキルが要求されました。そのため、プログラミング初心者が気軽に Windows アプリケーションを作れるような状況ではありませんでした。

しかし、「現状のままでは Windows の普及は進まない」と考えたマイクロソフトは、MS-DOS 向けに発売していた初心者向けプログラミング言語「BASIC」を基に、新しい Windows アプリケーション開発言語を作りました。それが Visual Basic です。

Visual Basic は、当時としては画期的なプログラミング言語でした。それまで Windows プログラミングといえば、ウィンドウを 1 つ表示させるだけで数十行のコードが必要でしたが、Visual Basic では、自動生成されたコードに 1 行コードを追加するだけでウィンドウに文字を表示することができました。ボタンやテキストボックスも、フォームと呼ばれる土台にマウスで貼り付けるだけで終了です。ついに、初心者でも Windows アプリケーションが開発できるようになったのです。

ところが 10 年ほど前から、インターネットの普及とともに「Java」のような新しい開発言語が現れ、Web アプリケーションと呼ばれる形態が台頭するようになります。Web アプリケーションは、ブラウザさえあればアプリケーションの実行ができ、OS の種類を選びません。

そこで、Microsoft が打ち出したのが、「.NET 構想」と呼ばれる考え方です。これは、アプリケーションをオブジェクト単位の組み合わせで開発し、.NET プラットフォーム上で動作させることで、インターネット時代の新しい開発スタイルに対応しようとしたものです。それに合わせて、Visual Basic も VB.NET というオブジェクト指向言語に生まれ変わりました。この VB.NET が、現在の Visual Basic 2013 の直接の祖先となります。

そして、本書は、この Visual Basic 2013 の入門書です。前述したように、当初の Visual Basic は初心者向けの言語でしたが、現在は本格的なオブジェクト指向プログラミング言語になっています。そこで、初心者でも無理なく Visual Basic 2013 の学習が進められるように、サンプルプログラムとしてゲームを用意し、次のような構成で進めることにしました。



## はじめに

---

1. まず、これから学ぶ文法の概要を説明します。
2. 次に、できるだけ簡単なサンプルを用いて、プログラムを動かしながら文法を理解します。
3. 最後に、登場した文法を利用したゲームプログラムを紹介し、復習と確認のため簡単な問題を解きます。

このように、単に文法を理解するだけではなく、実際のプログラムでどのように使われているのか、まさに「わかりすぎる」Visual Basic の入門書を目指しました。しかも、最後に登場するゲームは「ルーレットマシン」です。是非、この最終ゲームを楽しみに Visual Basic 2013 の学習を進めていただければと思います。

著 者

---

# 目次

はじめに

<b>第1章 Visual Basic 2013 の概要</b> .....	1
1.1 Visual Basic 2013 とは .....	2
1.2 Microsoft.NET .....	3
1.3 .NET Framework と VB2013.....	4
1.4 オブジェクト指向 .....	6
1.5 Visual Studio 2013.....	7
1.6 Visual Studio Express 2013 for Windows Desktop のインストール.....	8
1.7 プログラムの作成と実行 .....	13
1.7.1 プロジェクト作成.....	13
1.7.2 コントロールの利用.....	15
1.7.3 プロパティの変更.....	17
1.7.4 プロジェクトの保存、終了、再読み込み.....	18
1.8 コンソールアプリケーション .....	19
1.8.1 コンソールアプリケーションのプロジェクト.....	20
1.9 サンプルプログラムのダウンロード .....	23
1.10 第1章のまとめと練習問題 .....	24
1.10.1 第1章のまとめ.....	24
1.10.2 練習問題.....	25
<b>第2章 基本文法</b> .....	27
2.1 変数 .....	28
2.1.1 変数とは.....	28
2.1.2 変数の正体.....	29
2.1.3 変数名.....	31

---

2.1.4	変数の初期化.....	34
2.2	データ型.....	36
2.2.1	型の種類.....	36
2.2.2	プール型.....	37
2.2.3	整数型.....	38
2.2.4	浮動小数点型.....	39
2.2.5	文字型.....	40
2.2.6	文字列型.....	40
2.2.7	10進数型.....	41
2.3	リテラル.....	42
2.4	演算子.....	45
2.4.1	算術演算子.....	45
2.4.2	文字列連結演算子.....	46
2.4.3	代入演算子.....	46
2.4.4	ビットシフト演算子.....	47
2.5	キャスト.....	48
2.5.1	暗黙のキャスト.....	48
2.5.2	明示的なキャスト.....	49
2.6	定数.....	50
2.6.1	定数とは.....	50
2.6.2	定数の宣言.....	51
2.7	配列.....	53
2.7.1	配列とは.....	53
2.7.2	配列の宣言.....	54
2.7.3	配列の値へアクセスする.....	55
2.7.4	配列の初期化.....	56
2.7.5	配列の推論.....	58
2.7.6	配列の要素数を変更する.....	59
2.7.7	多次元配列.....	61



2.8	おみくじゲームを作る .....	63
2.8.1	プロジェクトの作成.....	63
2.8.2	ツールボックスからコントロールを追加.....	64
2.8.3	コードエディターでコードを追加する.....	68
2.8.4	コードの解説.....	70
2.9	第2章のまとめと練習問題 .....	73
2.9.1	第2章のまとめ.....	73
2.9.2	練習問題.....	74
<b>第3章</b>	<b>制御構文 .....</b>	<b>77</b>
3.1	分岐処理 .....	78
3.1.1	分岐処理とは.....	78
3.1.2	If 文 .....	78
3.1.3	Else 文.....	79
3.1.4	If 文の演習 .....	80
3.1.5	Select Case 文 .....	84
3.1.6	Select Case 文の利点と欠点 .....	86
3.1.7	Select Case 文の演習 .....	86
3.2	比較演算子 .....	88
3.2.1	比較演算子の種類.....	88
3.2.2	比較演算子の演習.....	89
3.3	論理演算子 .....	91
3.3.1	論理演算子の種類.....	91
3.3.2	論理演算子の演習.....	92
3.3.3	その他の論理演算子.....	94
3.4	繰り返し処理 .....	95
3.4.1	繰り返し処理とは.....	95
3.4.2	Do...Loop 文.....	96
3.4.3	Do...Loop 文の演習.....	97

---

3.4.4	For...Next 文.....	99
3.4.5	For...Next 文の演習.....	100
3.4.6	For Each 文.....	102
3.4.7	Continue 文.....	104
3.5	プロシージャ.....	105
3.5.1	プロシージャとは.....	105
3.5.2	プロシージャの宣言.....	106
3.5.3	Return 文.....	108
3.5.4	パラメータ.....	110
3.5.5	複数のパラメータ.....	113
3.5.6	Function プロシージャ.....	114
3.6	じゃんけんゲームを作る.....	117
3.6.1	作成するじゃんけんゲーム.....	117
3.6.2	プロジェクトの作成.....	118
3.6.3	ツールボックスからコントロールを追加.....	119
3.6.4	コードエディタでコードを追加する.....	122
3.6.5	コードの解説.....	126
3.7	第3章のまとめと練習問題.....	131
3.7.1	第3章のまとめ.....	131
3.7.2	練習問題.....	132
<b>第4章</b>	<b>オブジェクト指向.....</b>	<b>135</b>
4.1	クラス.....	136
4.1.1	オブジェクトとは.....	136
4.1.2	クラスとは.....	137
4.1.3	なぜクラスが必要なのか.....	138
4.1.4	プログラムの世界のクラス.....	139
4.1.5	VB2013 でクラスを宣言.....	140
4.2	インスタンス.....	142

4.2.1	インスタンスとは.....	142
4.2.2	New 演算子.....	142
4.2.3	オリジナルクラスのインスタンス化とアクセス.....	144
4.2.4	なぜオブジェクト指向で考えるのか.....	145
4.3	コンストラクタ .....	147
4.3.1	コンストラクタとは.....	147
4.3.2	なぜコンストラクタというものがあるのか.....	149
4.4	アクセス修飾子 .....	152
4.4.1	アクセス修飾子の種類.....	152
4.4.2	アクセス修飾子の演習.....	154
4.5	クラスをオブジェクトとして使う .....	157
4.5.1	インスタンスメンバ.....	157
4.5.2	インスタンスメンバの生成と利用.....	157
4.5.3	クラスメンバ.....	161
4.5.4	クラスメンバの生成と利用.....	162
4.6	カプセル化 .....	165
4.6.1	カプセル化とは.....	165
4.6.2	ブロック変数.....	166
4.6.3	Public と Private.....	168
4.7	プロパティ .....	170
4.7.1	プロパティ宣言.....	170
4.7.2	プロパティの演習.....	171
4.7.3	読み出し専用・書き込み専用プロパティ.....	173
4.7.4	インデクサ.....	174
4.8	名前空間 .....	178
4.8.1	名前空間とは.....	178
4.8.2	.NET Framework の名前空間 .....	178
4.8.3	Imports .....	179
4.8.4	Namespace .....	183



---

4.9	タイピングゲーム.....	184
4.9.1	タイピングゲームの紹介.....	184
4.9.2	プロジェクトの作成.....	185
4.9.3	ツールボックスからコントロールを追加.....	185
4.9.4	出題問題を作る.....	188
4.9.5	コードエディタでコードを追加する.....	189
4.9.6	コードの解説.....	192
4.10	第4章のまとめと練習問題.....	200
4.10.1	第4章のまとめ.....	200
4.10.2	練習問題.....	202
<b>第5章 高度なオブジェクト指向.....</b>		<b>205</b>
5.1	継承.....	206
5.1.1	継承とは.....	206
5.1.2	継承の宣言.....	208
5.1.3	派生クラスから基本クラスを利用する.....	209
5.1.4	基本クラス型の変数.....	212
5.1.5	暗黙に継承される Object クラス.....	214
5.2	基本クラスのコンストラクタ.....	216
5.2.1	コンストラクタは継承されない.....	216
5.2.2	パラメータ付きコンストラクタの呼び出し.....	219
5.2.3	MyBase.....	222
5.2.4	NotInheritable 修飾子.....	223
5.3	オーバーロード.....	224
5.3.1	オーバーロードとは.....	224
5.3.2	派生クラスのオーバーロード.....	226
5.4	オーバーライド.....	230
5.4.1	オーバーライドとは.....	230
5.5	ポリモフィズム.....	233

5.5.1	ポリモフィズムとは.....	233
5.5.2	なぜポリモフィズムが必要か.....	234
5.6	抽象クラス.....	235
5.6.1	抽象クラスとは.....	235
5.6.2	抽象クラスの宣言.....	236
5.6.3	抽象メソッド.....	237
5.6.4	抽象メソッドの宣言.....	237
5.6.5	抽象メソッドによるポリモフィズムの保証.....	240
5.7	インターフェイス.....	244
5.7.1	多重継承.....	244
5.7.2	インターフェイスとは.....	246
5.7.3	インターフェイスの宣言.....	246
5.7.4	インターフェイスの実装.....	247
5.7.5	インターフェイスの演習.....	249
5.7.6	USBの機能にインターフェイスを使う理由.....	250
5.7.7	インターフェイスを継承する.....	253
5.8	デリゲート.....	254
5.8.1	デリゲートとは.....	254
5.8.2	デリゲート宣言.....	254
5.8.3	マルチキャストデリゲート.....	257
5.9	イベント.....	259
5.9.1	イベントとは.....	259
5.9.2	イベントの宣言とイベントの発生.....	260
5.9.3	イベントハンドラの追加と削除.....	262
5.9.4	WithEvents と Handles.....	264
5.10	ダンスゲームを作ろう.....	266
5.10.1	ダンスゲームの紹介.....	266
5.10.2	ダンスゲームのシナリオ.....	267
5.10.3	プロジェクトの作成.....	267

---

5.10.4	ツールボックスからコントロールを追加.....	268
5.10.5	コードエディタでコードを追加する.....	271
5.10.6	音声ファイルと画像ファイルをコピーする.....	274
5.10.7	コードの解説.....	274
5.11	第5章のまとめと練習問題.....	282
5.11.1	第5章のまとめ.....	282
5.11.2	練習問題.....	284
<b>第6章</b>	<b>さまざまな言語仕様.....</b>	<b>291</b>
6.1	例外処理.....	292
6.1.1	エラー処理.....	292
6.1.2	戻り値によるエラー処理の問題点.....	296
6.1.3	例外処理とは.....	298
6.1.4	例外処理の構文.....	299
6.1.5	例外処理の基本.....	300
6.1.6	例外処理の応用.....	305
6.1.7	Finally ブロック.....	307
6.1.8	オリジナルの例外をスローする.....	308
6.2	構造体.....	311
6.2.1	参照型と値型.....	311
6.2.2	構造体.....	313
6.3	列挙体.....	316
6.3.1	列挙体とは.....	316
6.3.2	列挙体の演習.....	316
6.4	Me.....	319
6.4.1	オブジェクト指向と参照値.....	319
6.4.2	Me.....	320
6.5	拡張メソッド.....	325
6.5.1	継承を使ってクラスを拡張する.....	325



6.5.2	拡張メソッドで機能を拡張する.....	326
6.5.3	.NET Framework のクラスを拡張する.....	328
6.6	ルーレットゲームを作ろう.....	329
6.6.1	ルーレットゲームの紹介.....	329
6.6.2	ルーレットゲームのシナリオ.....	330
6.6.3	プロジェクトの作成.....	330
6.6.4	コードエディタでソースコードを追加する.....	331
6.6.5	画像ファイルをコピーする.....	334
6.6.6	コードの解説.....	334
6.7	第6章のまとめと練習問題.....	346
6.7.1	第6章のまとめ.....	346
6.7.2	練習問題.....	348
練習問題 解答・解説.....		351
索引.....		365

---

---

# 第 1 章

---

---

## Visual Basic 2013の 概要

この章では、Visual Basic 2013の概要とインストール方法を紹介し、プロジェクトの作成と保存について学びます。

# 1.1 Visual Basic 2013 とは

Visual Basic 2013(以降 VB2013)は、Microsoft 社が開発した、オブジェクト指向プログラミング言語です。

Visual Basic の特徴は、アプリケーションを作るためのツールがすべて用意されていることです。このように 1 つの開発環境でソースコードの入力(プログラムを記述したテキストをソースコードといいます)からデバッグ(プログラムの不都合を修正する作業)、実行までできるツールを「統合開発環境 (IDE: Integrated Development Environment)」と呼んでいます。

発売当初の Visual Basic では、プログラムの実行方式に「インタプリタ」を採用していました。インタプリタは、プログラムの命令を 1 行単位で解釈しながら、その都度 CPU が実行できるマシン語(ネイティブコード)に変換して実行していきます。そのため、どうしても実行速度が遅くなるという欠点がありました。

そこで Visual Basic 5.0 以降、あらかじめソースコードをマシン語に変換(このプログラムを「コンパイル」と呼びます)してから実行する「コンパイル」方式が採用されました。これにより、Visual Basic でも高速なプログラムが作れるようになりました。

その後、Visual Basic は Visual Basic .NET(7.0)となり、大きく仕様を変えることとなります。実は 6.0 までの Visual Basic は、不完全なオブジェクト指向(オブジェクト指向については後に説明があります)と呼ばれていました。しかし 7.0 では、これまでの言語仕様を捨て、C++や Java といったオブジェクト指向言語と同等の言語仕様に生まれ変わりました。また 7.0 では、直接 OS の上でプログラムを動作させるのではなく、共通言語ランタイム (CLR: Common Language Runtime)という「仮想マシン」上で、プログラムを動作させるようになりました。これは、インターネットが普及したため、セキュリティを向上させる必要があったからです。

そして 2013 年、Visual Basic のバージョンは「12」になりました。その間に、データベース連携、Web 連携、クラウド連携、Modern UI(Windows 8) 対応といった、数多くの機能を取り込み、今なお Visual Basic は進化を続けています。

## 1.2 Microsoft.NET

VB2013 は、Visual Basic の最新バージョンであると共に、「Microsoft .NET」を構成する言語の 1 つでもあります。

Microsoft .NET とは、Microsoft が 2000 年 6 月に発表したインターネット技術や戦略の総称のことです。当時はインターネットが猛烈な勢いで普及していました。そのため、IT 業界には「インターネットに対応したアプリケーションを安く早く開発する」という要求が多く出されていました。その要求に Microsoft が出した答えが Microsoft .NET だったのです。

Microsoft .NET は、インターネット上の様々なコンピュータで、セキュリティを保障しながら分散処理させることを目的としています。そのため、異なるコンピュータ上に OS の代わりとなるプラットフォームが必要でした。そのプラットフォームとして登場したのが「.NET Framework」です。

.NET Framework は、開発ツールとして「クラスライブラリ」を持ち、実行環境には「共通言語ランタイム」を利用します。「クラスライブラリ」とは、共通言語ランタイム上で動作するプログラムを作るための部品の集まりです。したがって、.NET Framework のクラスライブラリを扱うことができるのであれば、プログラム言語に制限はありません。現在、.NET Framework のプログラムを開発できる言語には、Visual Basic をはじめとして、Visual C++、Visual C# などがあります。

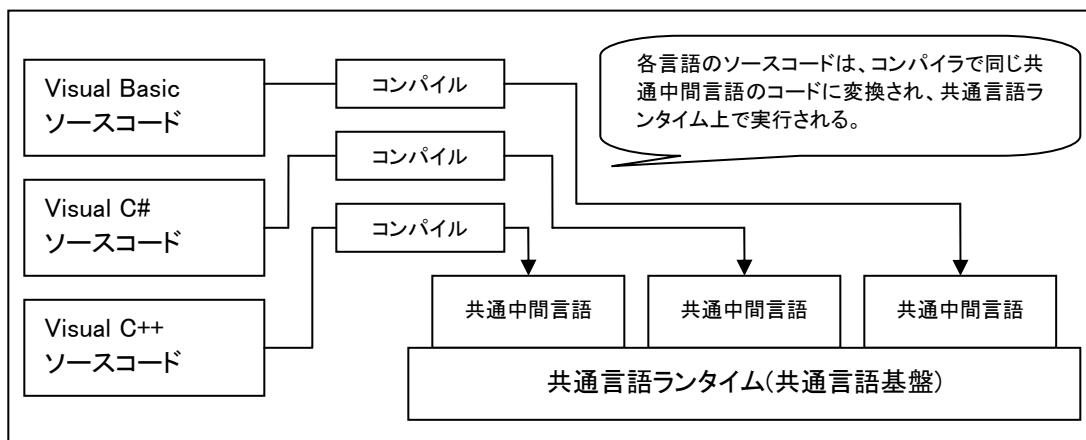
.NET Framework は、すでに多数のバージョンが存在します。2002 年に最初のリリースである「.NET Framework 1.0」が発表されましたが、その後 2.0、3.0、4.0 を経て、2013 年には「.NET Framework 4.5.1」へとバージョンアップが繰り返されています。

## 1.3 .NET Framework と VB2013

前述したように、VB2013 で作ったプログラムは、「.NET Framework」と呼ばれるプラットフォームにある「共通言語ランタイム」という仮想マシン上で動作します。

この「共通言語ランタイム」は「共通言語基盤(CLI [Common Language Infrastructure])」と呼ばれる仕様に基づいて実装されています。

共通言語基盤は、共通中間言語(CIL [Common Intermediate Language])と呼ばれる仕様に基づいて生成された「CLI コード (Microsoft は、「マネージコード」と呼んでいます)」を実行します。そのため、CLI コードが生成できればプログラム言語は問いません。



このような仕様により、VB2013 のプログラムは共通言語基盤の仕様に基づいたプラットフォームさえ用意できれば動作します。したがって OS も Windows である必要はありません。

**参考** 現在、2013 年 12 月現在、CLI コードを実行できる環境は、「.NET Framework」以外には、「Mono」と呼ぶオープンソースの実装があります。Mono は、Linux 上で動作する .NET Framework 互換の環境で、現在は Xamarin 社が開発、販売、サポート業務を行っています。

### JIT(Just In Time)コンパイラ

共通言語ランタイム上で CLI コードが動作するといっても、実際には OS 上で動作するコード(ネイティブコード)に変換(コンパイル)しなければいけません。この変換プログラムを「JIT(Just In Time)コンパイラ」といいます。

JIT コンパイラの特徴は、ネイティブコードが必要になるまで、コンパイルしないことと、最初の変換時にのみコンパイルすることです。コンパイル後はコンパイル済みのコードを再利用するため、CLI コードの動作は、ネイティブコードと遜色ない速さで動作します。

### ガベージコレクション

共通言語ランタイムには、ガベージコレクションという機能があり、プログラムが確保したメモリー領域を自動的に開放してくれます。

C や C++といったプログラミング言語では、プログラムの実行中にメモリーを確保して利用することができますが、確保した領域を開放し忘れると「メモリーリーク」という問題が発生します。メモリーリークとは、解放し忘れた領域で利用可能なメモリーが埋め尽くされてしまい、他のプログラムがメモリーを利用しようとしてもできなくなる状態をいいます。しかし共通言語ランタイムの場合、メモリーリークがおきることはほとんどありません。それはメモリー管理に、「ガベージコレクション」という機能を採用しているためです。

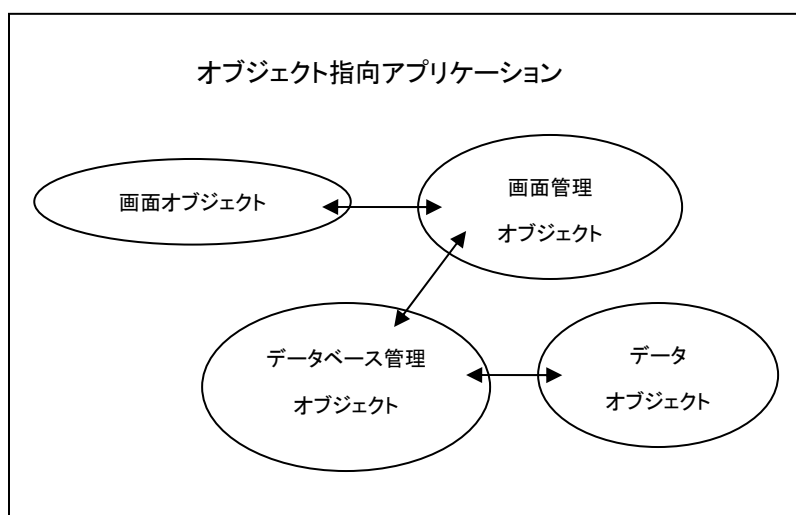
ガベージコレクションは、定期的にプログラムから参照されているメモリー領域を調べ、利用されていないメモリー領域があれば、その領域を解放して再利用します。またプログラムが、メモリーの確保と解放を繰り返すと、利用できるメモリー範囲が歯抜け状態となり、一度に大きな領域を確保できなくなります。そこでガベージコレクションは、プログラムが確保しているメモリー領域を自動的に移動することもあります。このガベージコレクションによって、プログラマはメモリー管理をする必要が無くなり、プログラムの安全性が高まります。



## 1.4 オブジェクト指向

VB2013 は、完全な「オブジェクト指向言語」です。

ここでいう「オブジェクト」とは特定の役割をもったプログラムのことです。オブジェクト指向言語は、多くのオブジェクト同士が連携して 1 つのアプリケーションを構成するという特徴を持ちます。



たとえば C 言語などでは「構造化」という手法でプログラムの構成を考えます。「構造化」とは、複雑なプログラムを簡単な処理単位に分解してから組み合わせるという手法です。

しかし構造化には、プログラムが利用する「データ」と「その操作方法」を、まとめて管理する決まりはありませんでした。そのためデータと処理は、プログラマが独自に決めたルールに従って強く結び付けられるため、他のプログラマが「処理だけ」を再利用するのが難しいという問題がありました。

そこでオブジェクト指向では、データとその操作方法を「オブジェクト」という 1 つの単位で管理するようにしています。そのため、オブジェクト単位での再利用がしやすくなり、プログラムの再利用性を高めることができるようになっていきます。

## 1.5 Visual Studio 2013

VB2013 は、「Visual Studio 2013」と呼ばれる統合開発環境から利用します。この Visual Studio 2013 には、プログラムの用途に合わせて、以下のエディションが用意されています。

Visual Studio Ultimate 2013

Visual Studio Premium 2013

Visual Studio Professional 2013

Visual Studio Test Professional 2013

Visual Studio Team Foundation Server 2013

エディションの最上位に位置するのが「Ultimate」です。Ultimate エディションは、Visual Studio のすべての機能が利用できます。次の「Premium」エディションでは、チーム開発に必要なツールが統合され、「Professional」エディションでは、Windows、Windows Phone、Office、Web、クラウドアプリケーションの作成と配置を簡略化するツール等が統合されています。

その他、プログラムのテスト担当者向け製品である「Test Professional」エディションや、大規模なチーム開発向けのサーバー構築が可能な、「Team Foundation」と呼ばれるエディションもあります。

また、Visual Studio には、学習用として「Visual Studio Express 2013」も用意されています。「Express」エディションは、プロフェッショナル機能に制限はありますが、基本機能はすべて利用できる、無償で提供されるエディションです。

そこで本書では、Visual Basic でデスクトップアプリケーションの開発が可能な、「Visual Studio Express 2013 for Windows Desktop」を使用することにします。

また、Visual Studio Express 2013 for Windows Desktop を使用する関係で、サポートされる OS は「Windows 7 SP1」以降、「Windows 8/8.1」以降、「Windows Server 2008 R2 SP1」以降、「Windows Server 2012/R2」以降になるので注意してください。

## 1.6 Visual Studio Express 2013 for Windows Desktop のインストール

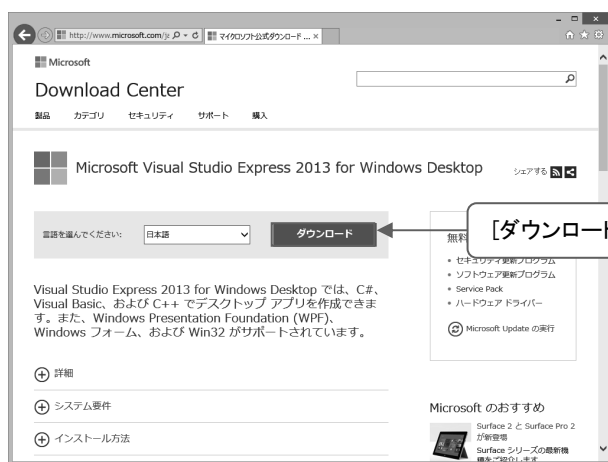
それでは、Visual Studio Express 2013 for Windows Desktop をインストールしてみましょう。ここでは、OS に Windows 8.1、ブラウザに Internet Explorer 11 を使ったインストール方法を紹介します。

※インストール画面やインストール方法は、他の OS やブラウザでもほとんど同じです。

Visual Studio Express 2013 for Windows Desktop をインストールするには、まず、以下のサイトへアクセスします。

Visual Studio Express 2013 for Windows Desktop 日本語版

<http://www.microsoft.com/ja-jp/download/details.aspx?id=40787>



上記のサイトにアクセスすると、Download Center - Microsoft Visual Studio Express 2013 for Windows Desktop ページが開くので、[ダウンロード]ボタンをクリックします。

**注意** ダウンロード先のアドレスやダウンロード方法は、変更される場合があります。目的のページが見つからない場合は、検索サイトを使い「Microsoft Visual Studio Express 2013 for Windows Desktop」で検索してください。



「ダウンロードするプログラムを選んでください」ページに切り替わるので「wdexpress\_full.exe」をクリックしてチェックを入れ、右下にある[次へ]ボタンをクリックします。

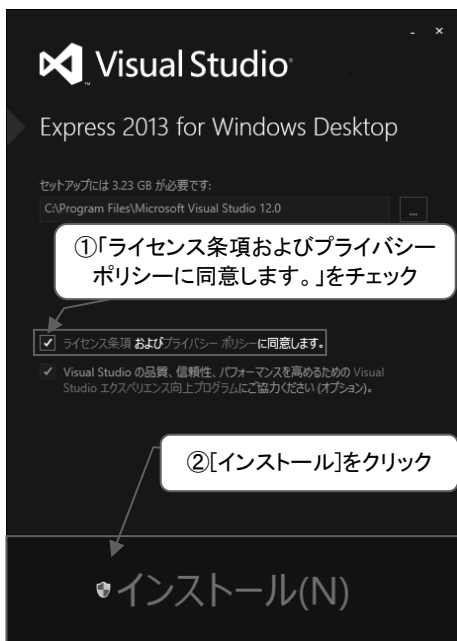
※複数の PC にインストールする場合は、「VS2013\_RTM\_DskExp\_JPN.iso」をダウンロードして、DVD-R に焼いたりマウントしたりすることで、繰り返しインストールできます。

**注意** 「www.microsoft.com」からのポップアップがブロックされました」ダイアログが表示された場合は、[このサイトのオプション]ドロップダウンボタンをクリックして、「常に許可」を選択してください。



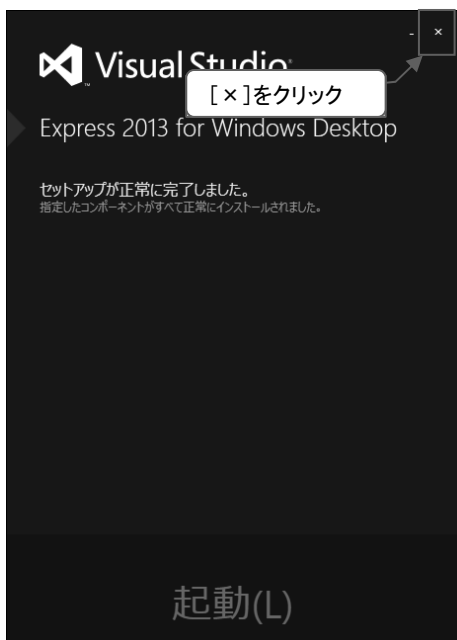
「download.microsoft.com から wdexpress\_full.exe (0.99MB) を実行または保存しますか？」ダイアログが表示されるので、[実行]ボタンをクリックします。

**注意** 「ユーザーアカウント制御」ダイアログが表示された場合は、[はい]ボタンをクリックしてください。



インストーラーが起動するので、「ライセンス条項およびプライバシー ポリシーに同意します。」をクリックしてチェックを入れ、現れた[インストール]ボタンをクリックします。

**注意** デフォルトでは、「C:\Program Files\Microsoft Visual Studio 12.0」にインストールされます。C ドライブに 3.23GB 以上の空き領域がない場合は、[...]ボタンをクリックして、他のドライブとフォルダーを指定してください。



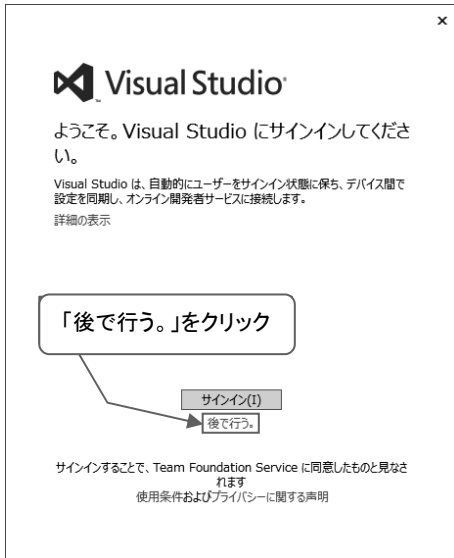
インストールが完了すると、「セットアップが正常に完了しました。」と表示されるので、[×](クローズ)ボタンをクリックして閉じます。



Visual Studio Express 2013 for Windows Desktop を起動するには、Windows 8/8.1 では「スタート画面」でアプリ一覧を表示して、Visual Studio 2013 グループの[VS Express 2013...]タイトルをクリックします。

※Windows 7 などでは、スタートメニューに登録された「VS Express for Desktop」をクリックします。

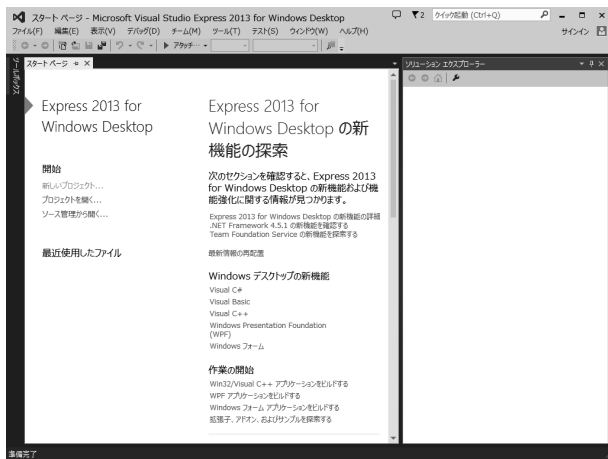
**注意** Visual Studio Express 2013 for Windows Desktop を起動すると、互換性に問題があるというメッセージが表示される場合があります。その場合は、Update を促されるので、指示に従って更新プログラムをインストールしてください。



最初の起動時に、サインインを求められますが、「後で行う。」リンクをクリックしてダイアログを閉じてください。

**注意** 継続利用する場合は、30 日以内に Microsoft アカウントでサインインする必要があります。Microsoft アカウントの登録を済ませたら Visual Studio Express 2013 for Windows Desktop を起動して、メニューから「ヘルプ」→「製品の登録」を選択してダイアログを表示させ、[サインイン]ボタンをクリックし、Microsoft アカウントでサインインしてください。

## 第 1 章 Visual Basic 2013 の概要



Visual Studio Express 2013 for Windows Desktop が起動します。

最初に表示されている画面左側「スタートページ」タブの部分で、「スタートページ」と呼びます。Visual Studio Express 2013 for Windows Desktop を起動すると表示されるので、覚えておきましょう。

Visual Studio Express 2013 for Windows Desktop をクローズする場合は、一番右上にある[×]ボタンをクリックするか、メニューから「ファイル」→「終了」を選択します。

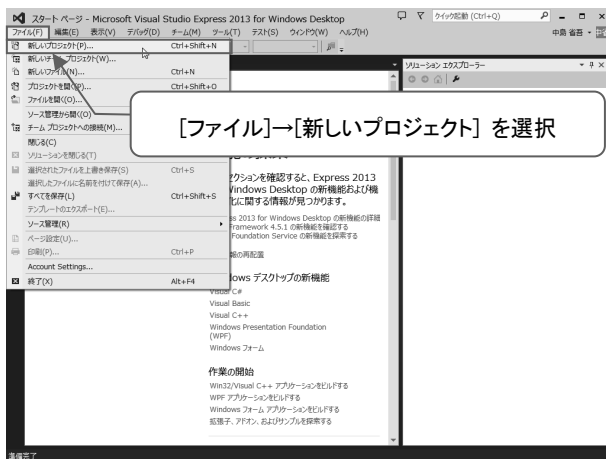
## 1.7 プログラムの作成と実行

Visual Studio Express 2013 for Windows Desktop が起動できたら、簡単なプログラムを作ってみましょう。

### 1.7.1 プロジェクト作成

まず、Window アプリケーションのプロジェクトを作成します。

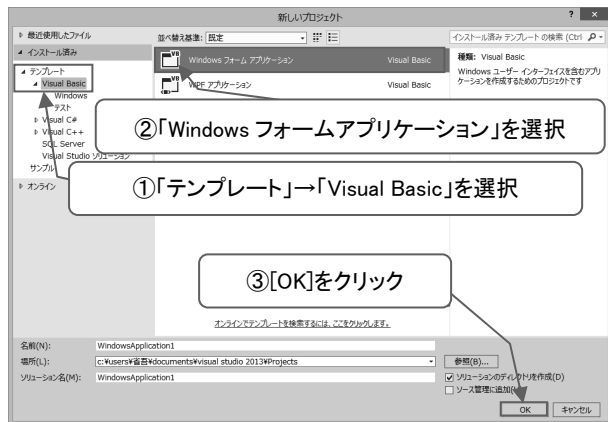
プロジェクトとは、ウィンドウ画面のデザイン、画像、データ、ソースコードなど、プログラムを構成するさまざまな要素をまとめて管理する単位です。さらにその上には「ソリューション」という単位が存在し、複数のプロジェクトで構成されたアプリケーションを開発することもできます。



Visual Studio Express 2013 for Windows Desktop のメニューから [ファイル]→[新しいプロジェクト] を選択します。

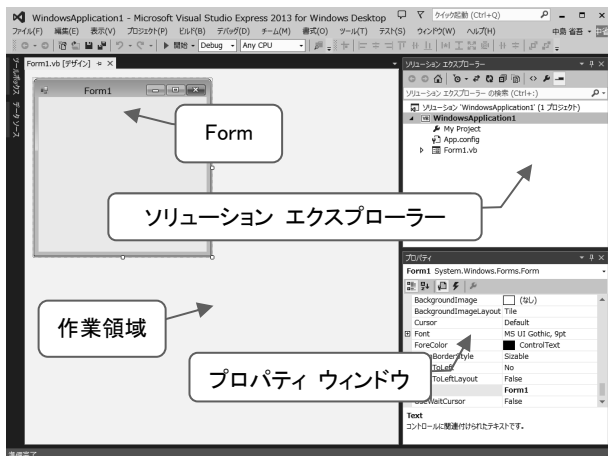


## 第1章 Visual Basic 2013 の概要



「新しいプロジェクト」ダイアログボックスが表示されます。

左側の「テンプレート」→「Visual Basic」を選択し、中央の領域から「Windows フォームアプリケーション」を選択します。プロジェクト名は変更することもできますが、今回は「名前」「ソリューション名」共に [WindowsApplication1] のままにして、[OK] ボタンをクリックします。



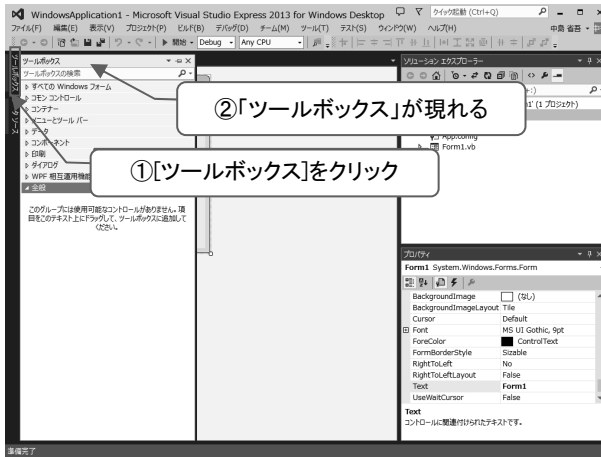
Windows アプリケーションを作成するための画面に変化します。

「ソリューション エクスプローラー」に追加された「Form1.vb」というファイルは、新しく作成された Form (フォーム) です。この Form が、実行時にウィンドウとして表示されます。

Form は、作業領域が[デザイン]になっているとき変更することができます。この状態で「ボタン」や「ラベル」など、アプリケーションの部品(コントロール)を Form に追加することで、プログラムを構築していきます。

右下にある「プロパティ ウィンドウ」は、オブジェクトのプロパティを一覧表示して、各プロパティに設定されている「プロパティ値」を表示します。また、値を編集することもできます。「プロパティ」とは、オブジェクトの「属性」を表します。VB2013 では Form やボタンのような機能をもったものを「オブジェクト」と考え、これらオブジェクトには表示する文字列や大きさ、色などの「属性」があり、その属性を変更することで表示する文字列や大きさなどを変更します。

## 1.7.2 コントロールの利用



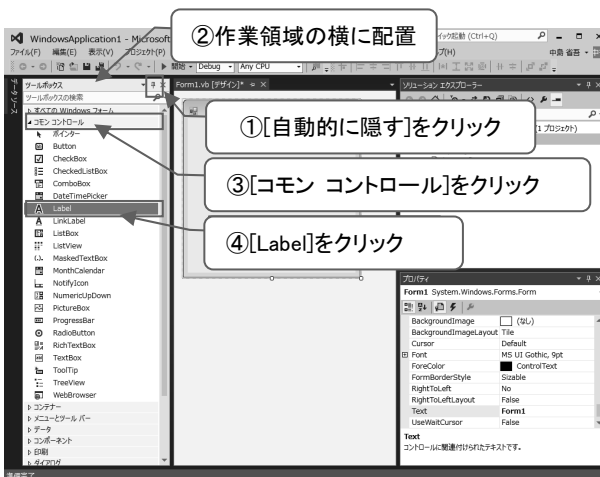
この状態でプログラムを実行しても、Form と同じウィンドウが表示されるだけです。そこで、Form の上に「コントロール」を追加して、文字を表示させてみましょう。

「コントロール」とは、ウィンドウを構成する部品のことです。

コントロールを Form に追加するには、「ツールボックス」を使います。

ツールボックスを表示させるには、作業領域の左端にある[ツールボックス]をクリックするか、もしくは、メニューの [表示]→ [ツールボックス] を選択します。

※ツールボックスが非表示の場合は、メニューから「表示」→「ツールボックス」を選択すると表示されます。



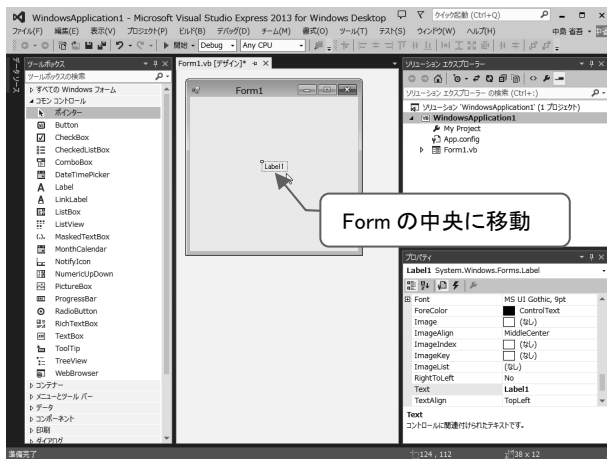
ツールボックスを表示したら「自動的に隠す」ボタンをクリックすると、ツールボックスが作業領域の横に配置されます。

ここで、「コモン コントロール」をクリックして展開し「Label」をクリックします。

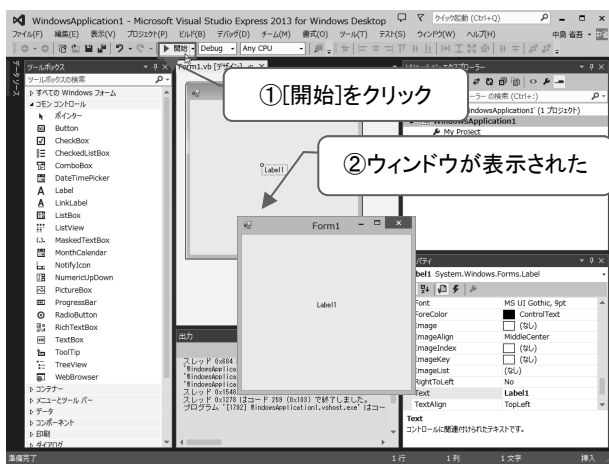
この Label(ラベル) が、コントロールです。

Form の適当な場所でクリックすると、Form に「Label1」という表示が現れます。

## 第1章 Visual Basic 2013 の概要



Label1 の位置は、マウスでドラッグすることで移動することができます。Form の中央に移動させます。



プログラムを起動してみましょう。プログラムを起動するには、ツールバーの[開始]ボタンをクリックします。

※[開始]ボタンをクリックすると、コンパイルが自動的に行われ、実行ファイルが作成されます。この実行ファイルを作成する作業を「ビルド」と呼びます。

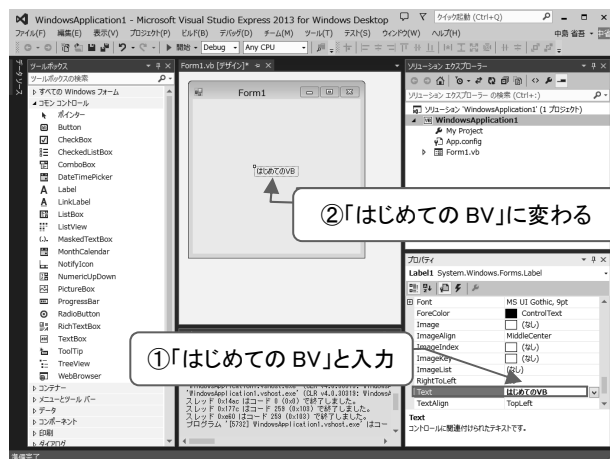
プログラムが起動すると、画面中央に Label1 と表示されたウィンドウが表示されます。このプログラムを終了させるには、表示されたウィンドウの[×]ボタンをクリックします。

## 1.7.3 プロパティの変更

ウィンドウに文字を表示させるプログラムは完成しましたが、表示する文字が Label1 では面白くありません。そこで Label1 に表示される文字を変更してみましょう

Label コントロールの表示文字列を保持しているのが、「Text」プロパティです。この Text プロパティの内容を変更するには、プロパティウィンドウを使います。

まず、Form1 の Label1 をクリックします。すると、枠が表示されます。この枠のある状態を「アクティブ」といいます。



次に、プロパティウィンドウを見ます。プロパティウィンドウは、アクティブなコントロールのプロパティを表示するので、Label1 のプロパティが表示されています。

このプロパティの中から、「Text」を探すと、右横に「Label1」と表示されています。

ここを「はじめのBV」と上書き入力して変更したら[Enter]キーを押します。

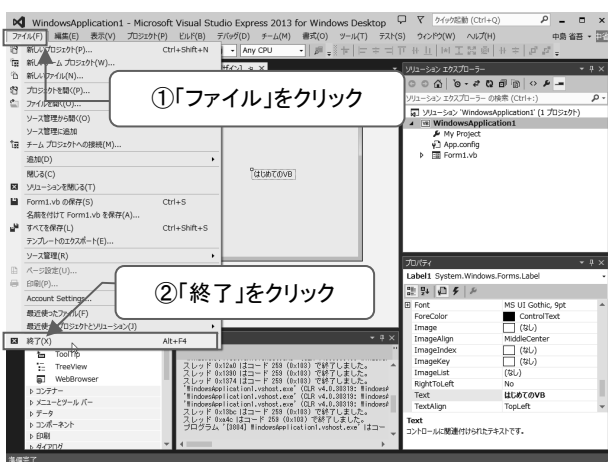
すると、Form 上の「Label1」という表示も「はじめのVB」に変わります。



Label コントロールをドラッグして位置を調整したら[開始]ボタンをクリックして実行します。

**注意** プログラムを終了するには、[×]ボタンをクリックします。

## 1.7.4 プロジェクトの保存、終了、再読み込み



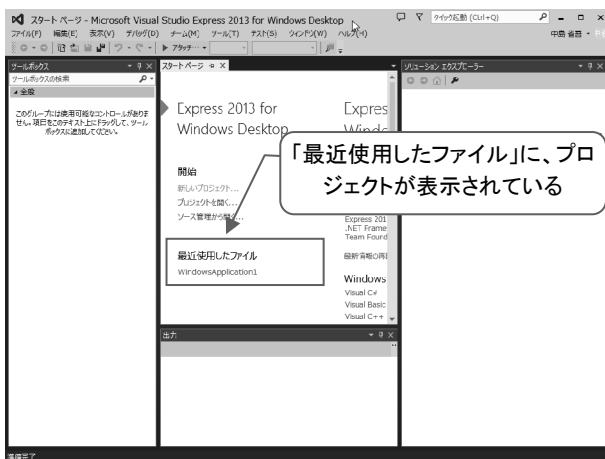
プログラムの作成をやめる場合は、プロジェクトを保存します。Visual Studio Express 2013 for Windows Desktop を終了すれば、プロジェクトは自動的に保存されます。

Visual Studio Express 2013 for Windows Desktop を終了させるには、[×]クローズボタンをクリックするか、メニューから「ファイル」→「終了」を選択します。

プロジェクトが存在する場合に終了しようとすると、変更があったファイルを一覧表示

するダイアログが表示される場合があります。その場合は[はい]をクリックすると、ファイルは上書き保存されます。

プログラムがまだ動作している状態で終了しようとした場合は、[デバックを中止しますか? ]というメッセージボックスが表示されます。その場合、[はい]をクリックすると、プログラムを終了して Visual Studio Express 2013 for Windows Desktop も終了します。



再び Visual Studio Express 2013 for Windows Desktop を起動すると、スタートページの「最近使用したファイル」の中に、保存したプロジェクトの名前が表示されています。

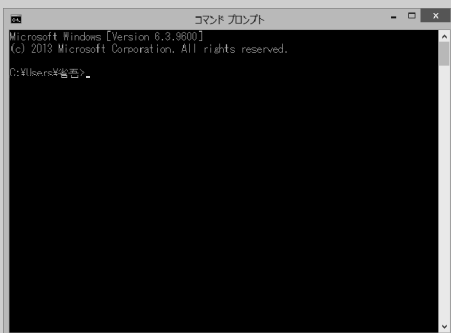
開きたいプロジェクトをクリックすれば、作業を再開できます。

## 1.8 コンソールアプリケーション

VB2013 を使えば、プログラムを入力しなくても簡単なウィンドウを表示するプログラムが作成できることはわかりました。それでは早速 Visual Basic の文法を勉強しましょう・・・と言いたいところですが、ウィンドウを表示するプログラムは、実は簡単に見えてかなり複雑なコードが自動生成されて動いています。そのためウィンドウを表示するプログラムで Visual Basic の勉強をはじめると、習っていない概念やコードが次々に出てきてしまい、文法に集中できません。

そこで、文法を学ぶためにコンソールアプリケーションの作り方も知っておくと、文法のチェックがやりやすくなります。

### コンソールアプリケーション

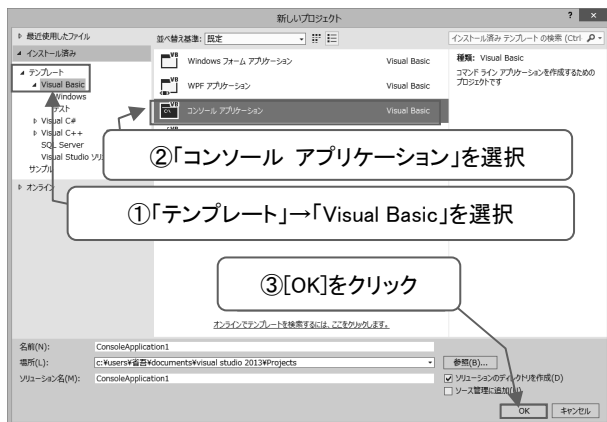


「コンソール」とは、Windows のコマンドプロンプト画面を指します。Windows 8.1 の場合、「スタート」ボタンで右クリックして[コマンド プロンプト]を選択すると、表示することができます。

Windows 7 などでは、「スタート」メニューの「アクセサリ」の中に「コマンド プロンプト」という項目があるので選択すると、コマンドプロンプトが現れます。

## 1.8.1 コンソールアプリケーションのプロジェクト

コンソールアプリケーションを作るには、コンソールアプリケーション専用のプロジェクトを作ることから始めます。

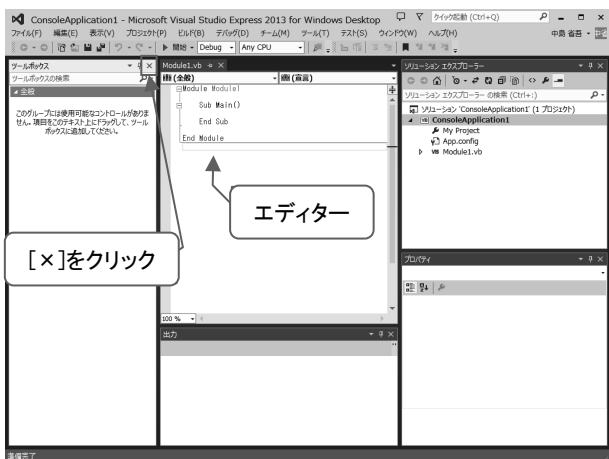


Visual Studio Express 2013 for Windows Desktop を起動したら、メニューから「ファイル」→「新しいプロジェクト」を選択します。

「新しいプロジェクト」ダイアログボックスが表示されたら、「コンソール アプリケーション」を選択します。

プロジェクト名は「ConsoleApplication1」のままでかまいません。

[OK] ボタンを押すと新たなプロジェクトが作成されます。



Windows アプリケーションと違い、Form は表示されません。その代わりに、ソースコードを入力するエディターが作業領域に表示されます。

※ツールボックスは使用しないので、[×]ボタンで閉じてかまいません。

エディターは、プログラムコードをキーボードから入力するために使用します。

表示されるエディター画面には、VB2013 によって自動的に生成されたコードが表示されています。

```
Module Module1
    Sub Main()
        End Sub
End Module
```

Sub Main と、End Sub の間に、以下のコードを追加します。

```
Module Module1
    Sub Main()
        Console.WriteLine("はじめての VB")
    End Sub
End Module
```

※上記から、スペースの関係で改行だけの行は削除します。エディターでコードが見にくい場合は、改行を入れてください。

**注意** 「はじめてのVB」以外のコードは、全て半角文字で入力します。入力を始めると、カーソルの下に入力候補の一覧が現れます。これは、「インテリセンス」と呼ばれる入力補完機能です。この一覧の中から選択して入力することも可能ですが、慣れないと間違いやすいので今回は使用しません。全てキーボードから入力してください。

Console.WriteLine というコードは、コマンドプロンプトに一行文字列を表示させる命令です。このような命令を「メソッド」と言います。

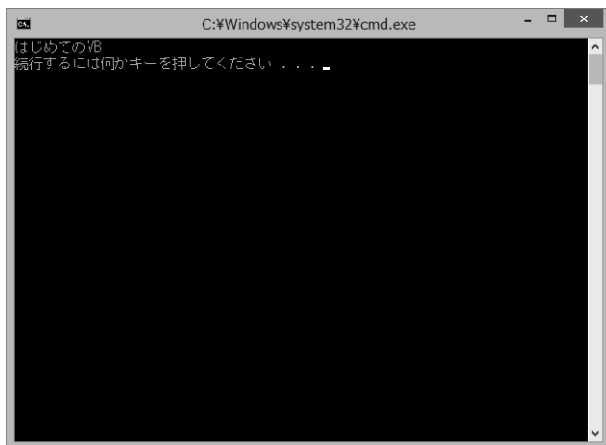
プログラムをビルドして実行してみましょう。コンソールアプリケーションのビルドと実行は、[Ctrl]キーを押しながら[F5]キーを押します。



## 第 1 章 Visual Basic 2013 の概要

---

※[開始]ボタンでもビルドと実行を行うことはできますが、表示されたコマンドプロンプトはすぐに閉じてしまいます。



コマンドプロンプトが起動し、「はじめてのVB」と表示されます。

また、その下に「続行するには何かキーを押してください . . .」と表示され、入力待ちの状態になります。

この表示と入力待ちは、[Ctrl]+[F5]キーを押して実行したとき自動的に挿入される処理で、プログラムの処理とは関係ありません。

プログラムを終了するには、何かキーを押します。すると、コマンドプロンプトのウィンドウが閉じます。

プロジェクト全体を終了する方法は、Windows アプリケーションのときと同じです。Visual Studio Express 2013 for Windows Desktop の[×]クローズボタンをクリックするか、メニューで終了します。

これが、コンソールアプリケーションの作り方です。Windows アプリケーションと見た目は違いますが、どちらも同じ Visual Basic のプログラムです。この2つのプロジェクトを使い分けて、効率よくVB2013を学習していくことにしましょう。

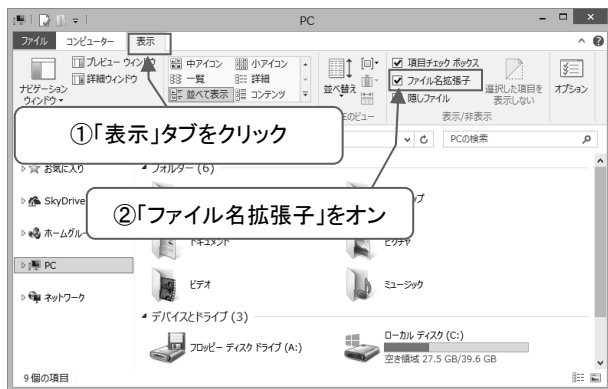
## 1.9 サンプルプログラムのダウンロード

最後に、第2章以降で利用するサンプルコードを以下の Web サイトからダウンロードしておきましょう。

<http://www.scc-kk.co.jp/scc-books/support/B-374/support.html>

ダウンロードするファイルは Projects.zip です。ダウンロード先は、適当なフォルダーでかまいません。ダウンロードに成功したら解凍しておきます。解凍先もどこでもかまいません。

たとえば C ドライブのルートに MyVB フォルダーを新規作成してそこに解凍する場合は、Projects.zip をダブルクリックして開き、表示される Projects フォルダーを MyVB フォルダーにコピーするだけで解凍されます。



**注意** Windows のデフォルト仕様では、ファイルの拡張子は表示されません。プログラム作成において、拡張子が表示されないのは不便なので、表示するように変更しておきましょう。

Windows 8/8.1 の場合は、エクスプローラーの「表示」タブを選択すると「表示/非表示」グループに、「ファイル名拡張子」のチェックボックスがあるので、クリックしてチェックします。

Windows 7 などの場合は、エクスプローラーを開きコマンドバーにある[整理]メニューから[フォルダーと検索のオプション]を選択します。フォルダオプションダイアログが表示されるので、[表示]タブを選択し[登録されている拡張子は表示しない]のチェックを外します。

## 1.10 第 1 章のまとめと練習問題

第 1 章で説明してきた内容のまとめと、正しく理解しているか簡単な三択のテストをしてみましょう。

### 1.10.1 第 1 章のまとめ

1. VB2013 は、Microsoft 社が開発した、オブジェクト指向プログラミング言語です。
2. コード入力から実行、デバッグまでできるツールを「統合開発環境」といいます。
3. VB2013 の実行ファイルは、共通言語ランタイム (CLR[Common Language Runtime])で動作します。
4. 「共通言語ランタイム」は「共通言語基盤(CLI [Common Language Infrastructure])」に基づいて実装されています。
5. 共通言語基盤は、共通中間言語(CIL [Common Intermediate Language])に基づいて生成された「CLI コード」を実行します。
6. オブジェクト指向は、オブジェクト同士が連携してアプリケーションを構成します。
7. オブジェクト指向は、データとその操作方法を「オブジェクト」単位で管理します。
8. VB2013 でアプリケーションを作る場合、最初に「プロジェクト」を作成します。
9. ウィンドウを表示するプログラムは「Windows フォーム アプリケーション」のプロジェクトを作ります。
10. VB2013 では、ウィンドウは Form で作ります。
11. ウィンドウに文字を表示するには、Label コントロールを利用します。
12. Form や Label などのオブジェクトには表示する文字や形を決める「プロパティ」という属性があります。
13. プロパティは VB2013 のプロパティウィンドウで編集できます。
14. コマンドプロンプトで動作するプログラムは「コンソール アプリケーション」のプロジェクトを作成します。

## 1.10.2 練習問題

問題は、三択です。解答・解説は第 6 章の次にあります。ぜひトライして頭の中を整理しましょう。

### ■ 問題 1

Visual Studio Express 2013 for Windows Desktop のように、コードの入力から、デバック、実行までを行える開発環境を、なんと言いますか。

1. 統合データベース
2. 統合開発環境
3. 統合開発言語

### ■ 問題 2

プログラムの命令を 1 行単位で解釈しながら実行していく方式は、次のうちどれですか。

1. インターネット
2. インターポール
3. インタープリタ

### ■ 問題 3

プログラムの命令をあらかじめマシン語に変換しておいて実行する方式は次のうちどれですか。

1. コンパイル
2. コマンドライン
3. ランタイム

### ■ 問題 4

Microsoft .NET で、OS の代わりとなるプラットフォームは、次のうちどれですか。

1. Java
2. NET Framework
3. Windows

### ■ 問題 5

.NET Framework の共通言語ランタイムは、どの仕様に基づいて実装されていますか。

1. 共通中間言語
2. CLI コード
3. 共通言語基盤

### ■ 問題 6

VB2013 で Windows アプリケーションを作るとき、Form に追加することができる部品オブジェクトをなんと呼びますか。

1. コンシューマ
2. コントロール
3. フレーム

### ■ 問題 7

VB2013 の Form や Label などのオブジェクトにある属性は、次のうちどれに相当しますか。

1. プロパティ
2. プロファイル
3. プロジェクト