

# 統合開発環境 Visual Studio 2015 と Visual Basic の使い方

総合情報基盤センター 教授 高井正三

Visual Studio 2015 は、Windows や Android, iOS 向けの Application を開発するための、米 Microsoft 社が提供している統合開発環境 (IDE : Integrated Development Environment) で、Visual Basic や C++, C#などのプログラミング言語に対応したソフトウェア開発ツールです。無料版の Visual Studio Community が提供されていますので、本稿では、この IDE の導入・設定の方法と、この IDE 下でのプログラミング言語 Visual Basic の使い方を解説し、利用者の一助となることを期待しています。

## 1. Visual Studio Community の導入と設定

### 1.1 Visual Studio Community の導入と初期設定

検索キーワードに「visual studio」を設定し、下記サイト Visual Studio 画面を表示します (図 1.1 上)。

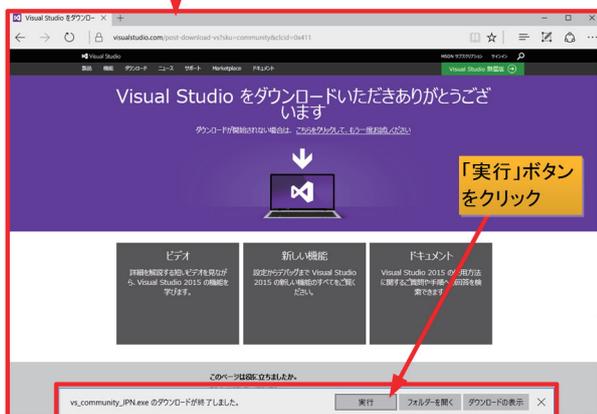
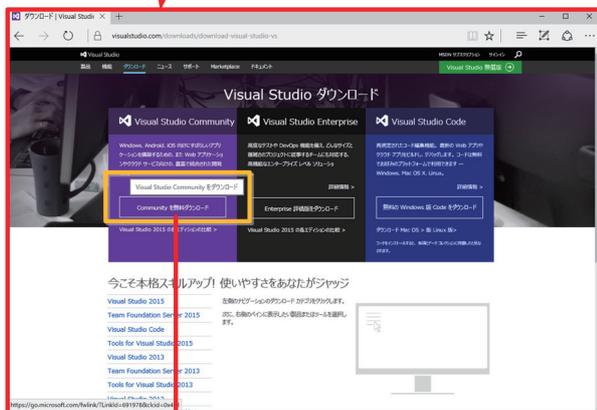
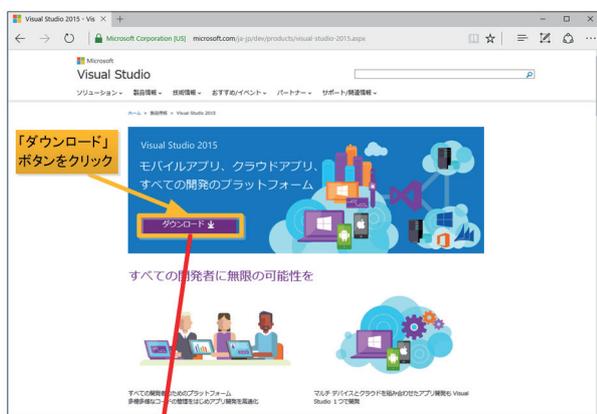


図 1.1 Visual Studio Community の Download 画面

Visual Studio 画面から「ダウンロード」ボタンを押下して「Visual Studio ダウンロード」画面を表示し、「Community 無料ダウンロード」ボタンをクリックします (図 1.1 中)。「vs\_Community\_JPN.exe のダウンロードが終了しました。」ダイアログ・ボックスが表示されるので、「実行」ボタンをクリックします (図 1.1 下)。

また、検索キーワードに「Visual Studio 2015」を設定した場合は、検索結果から図 1.2 の「無料開発ツール - Visual Studio Community 2015」をクリックして、「Visual Studio Community」画面から「ダウンロード」ボタンをクリックします。

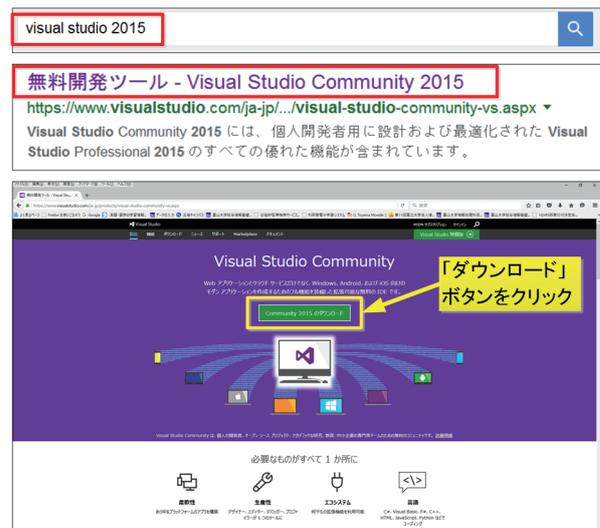


図 1.2 Visual StudioCommunity 画面からダウンロード



図 1.3 vs\_community\_JPN.exe を保存し実行

続いて、ダウンロードした「vs\_community\_JPN.exe」ファイルを保存し、ダウンロード履歴を表示して、該当ファイルを表示し、ダブル・クリックして実行します (図 1.3)。

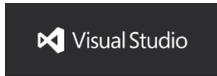


図 1.4 Visual Studio のロゴ

画面上に「Visual Studio」のロゴが表示され (図 1.4)、続いてインストール設定「Visual Studio Community 2015」ダイアログ・ボックスが表示されますので、インストール先、インストールの種類を選択して、「インストール」ボタンをクリックします (図 1.5 左)。

インストレーションには 30 分間程の時間を要しますので、ゆっくり待ちましょう (図 1.5 右)。

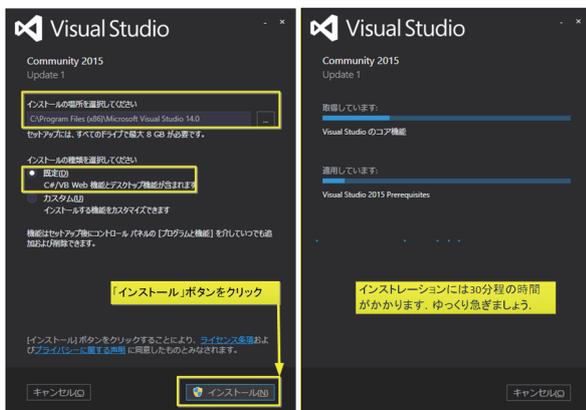


図 1.5 VS Community のインストール設定ダイアログ

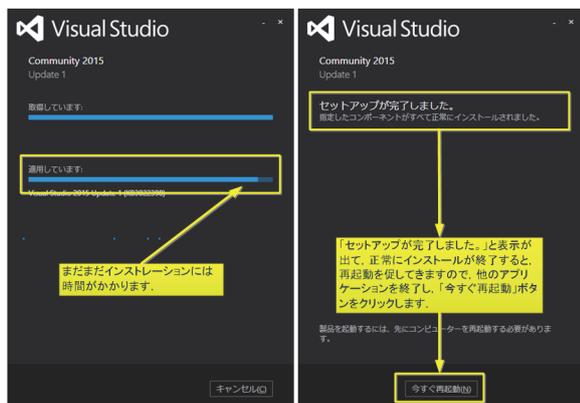


図 1.6 VS Community のインストール中の表示と完了表示

続いて、インストール画面 (図 1.6 左) が続きますので、完了まで待つて下さい。インストールが完了すると、再起動を促す画面が表示されますので (図 1.6 右)、「今すぐ再起動」ボタンをクリックして下さい。

PC を再起動して、Visual Studio 2015 が立ち上がってくれば、Visual Studio の起動時画面 (図 1.7 右) が表示されます。もし正常に再起動しなければ、デスク・トップ画面上のアイコン「Visual Studio 2015」 (図 1.7 左) をダブル・クリックして、再起動して下さい。



図 1.7 Visual Studio 2015 のアイコンと起動時画面

しばらくすると、「ようこそ」画面でサインインを求めてきます (図 1.8 左) ので、Microsoft 社のアカウントとパスワードを入力し、「サインイン」ボタンをクリックし、続いて「続行」ボタンをクリックします (図 1.8 右)。

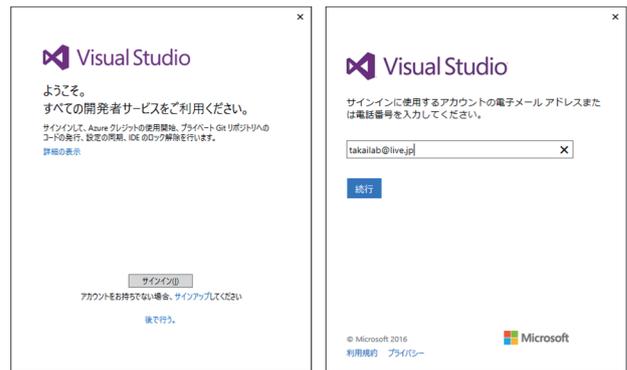


図 1.8 Visual Studio サービス開始とサインイン画面

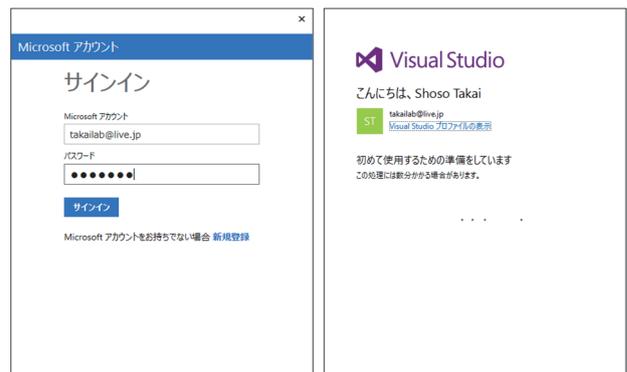


図 1.9 Visual Studio サインイン画面とスタート準備画面  
サインインが終了すると、「こんにちは・・・」というスタート準備画面が現れ、続いて Visual Studio 2015 の初期画面 (図 1.10) が表示されます。

この図 1.10 の初期画面が表示されれば、正常にインストールされ、Visual Basic でのアプリケーション開発ができるようになっています。

では、「終了ボタン」をクリックして一旦終了し、

再度起動するところから初めてみましょう。

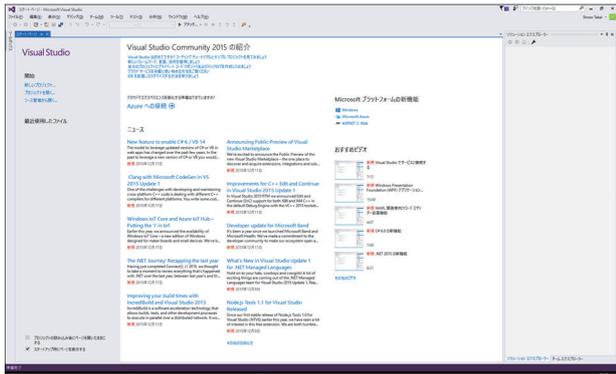


図 1.10 Visual Studio の初期画面

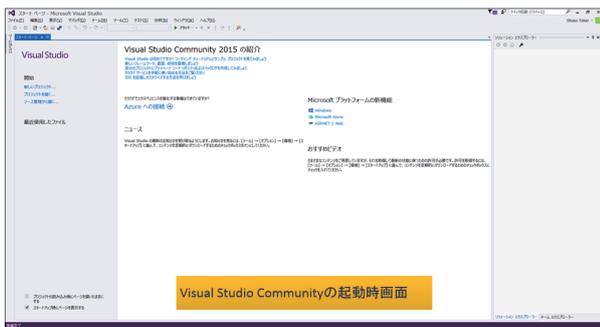
## 1.2 Visual Studio Community の起動

Visual Studio Community 2015 (以下「VS 2015」と略記) を起動するには、図 1.11 のように、以下の 3 通りの方法があります。

- (1) スタート・ボタンから「すべてのアプリ」をクリックして、アプリケーションの先頭「V」からロゴ付き「Visual Studio 2015」をクリック
- (2) スタート・ボタンから「Visual Studio 2015」のタイルをクリック
- (3) デスク・トップ画面から「Visual Studio 2015」のアイコンをダブル・クリック



図 1.11 VS2015 の 3 つの起動方法



VS 2015 が正常に起動すれば、図 1.12 の VS 2015 起動時画面が表示されます。

## 1.3 プロジェクトの保存場所の設定

これから Visual Basic によるアプリケーション・プログラム開発プロジェクトの保存先を、デフォルトの保存先から各自の指定した保存先 (例えばフォルダー「D:\Visual\_Basic」など) へ変更するには、メニュー「ツール」から「オプション」をクリックして、プロジェクトの保存先を変更します (図 1.12)。

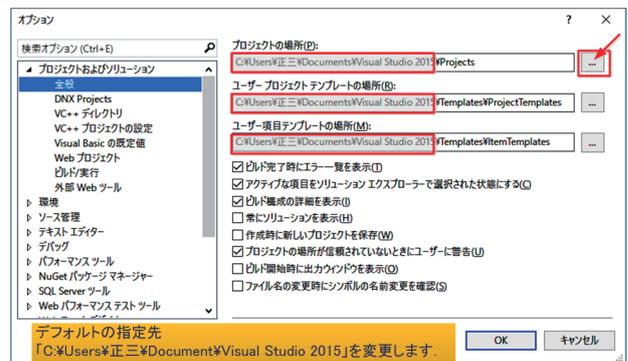
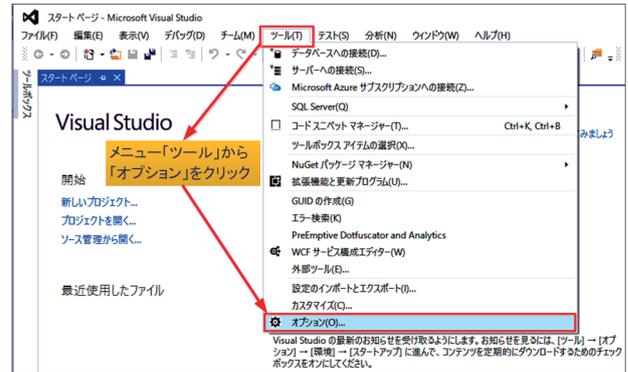


図 1.12 Visual Studio のプロジェクトの保存先を変更

予め Visual Basic 用のプロジェクトの保存場所フォルダー「D:\Visual\_Basic」を作成しておいてから、図 1.12 の操作を実行しましょう。

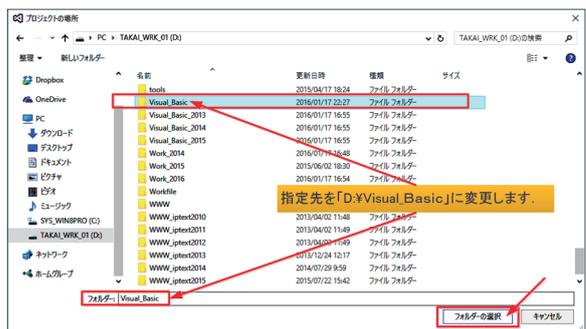


図 1.13 Visual Studio のプロジェクトの保存先を変更

ここでは、既成のフォルダー「D:\Visual\_Basic」を、図 1.13 のように選択し、「フォルダーの選択」ボタンをクリックします。続いて、以下の様に (1) の基礎となるフォルダー部分をコピーして、(2) と (3) のフォルダーも変更します (図 1.14)。

- (1) プロジェクトの場所  
「D:\Visual\_Basic\Projects」

(2) プロジェクト・テンプレートの場所

「D:\Visual\_Basic\Templates\Projects\Templates」

(3) 項目テンプレートの場所

「D:\Visual\_Basic\Templates\Item\Templates」

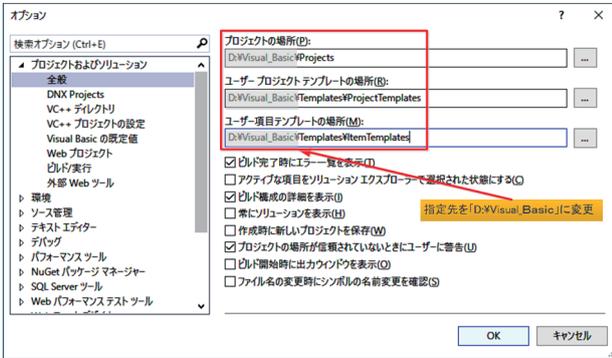


図 1.14 プロジェクトの保存場所等を設定変更

なお、プロジェクト・テンプレートと項目テンプレートを変更した場合、そのフォルダー内に予め用意された言語別の「空」フォルダーが用意されているかを確認しておいて下さい (図 1.15)。

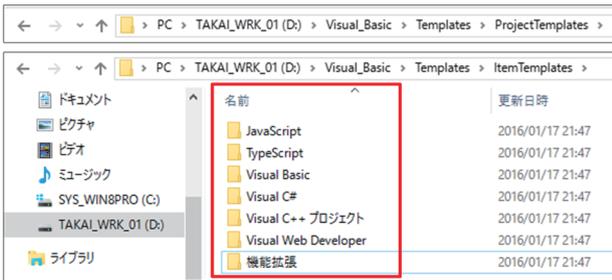


図 1.15 プロジェクトと項目テンプレートの保存場所

## 1.4 ドット・ネット・フレームワーク

Visual Basic (以下「VB」と略記する)、Visual C#, Visual C++などで、一つのアプリケーション・プログラムを開発するために、セキュリティを保証しながら、分散処理させるためのプラットフォームとして「Microsoft .NET Framework」を使用します。プログラムを作成するために、共通言語ランタイムや、この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ある「クラス・ライブラリ」など、多くのファイルが必要となります。これらのファイルは「.NET Framework」が提供してくれます。

VBは Visual Basic .NET(7.0)以降、仕様が変わり、直接 OS 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 (Common Language Runtime : CLR) という「仮想マシン」上で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、セキュリティを向上させる必要があったから、と言われてい

## 1.5 プロジェクトの作成と VB プログラミング

まずは、新しいプロジェクトを作成してみましょう。プロジェクトとは、Windows 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様々な要素をまとめて管理する単位を言います。

VS 2015 の起動時画面が表示されますので、左側のメニューから「新しいプロジェクト...」をクリックします (図 1.16)。

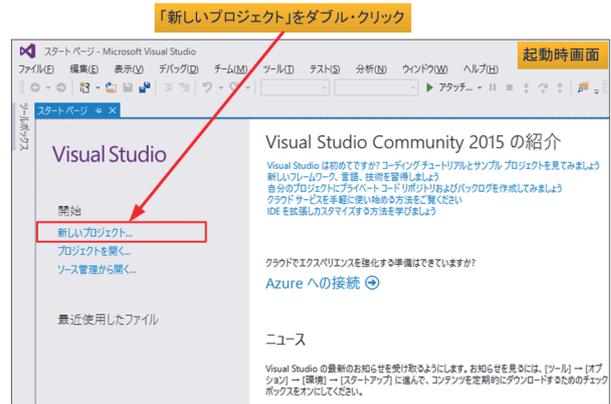


図 1.16 Visual Studio Community 2015 の起動時画面

しばらくすると、「新しいプロジェクト」のダイアログ・ボックスが表示されますので、画面左側のテンプレートで「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。プロジェクトの名前を、ここでは「Hello\_00」と指定しましょう (図 1.17)。



図 1.17 新しいプロジェクトの設定画面

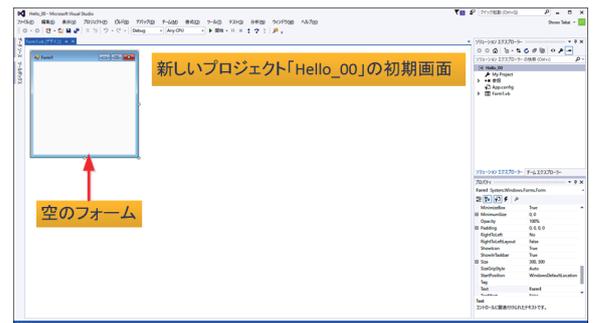


図 1.18 プロジェクト「Hello\_00」画面の「空のフォーム」

表示された新しいプロジェクト「Hello\_00」の画面で「空のフォーム」が表示されます (図 1.18)。

## 1.6 コントロールの利用

フォームに配置するテキスト・ボックス (文字列の入力ボックス) やボタンなどの部品を「コントロール」と呼びます。コントロールの配置は、ツール・ボックスを用いて行います。

ツール・ボックスを表示させるには、「ツール・ボックス」タブをクリックし、プルダウン・メニュー▼から「ドッキング」を選択します (図 1.19)。

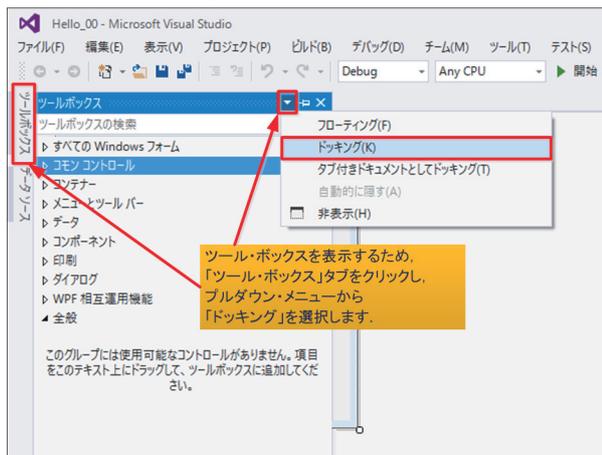


図 1.19 ツール・ボックスの表示

## 1.6 IDE の画面構成

IDE の画面構成は、図 1.20~21 に示す通りです。

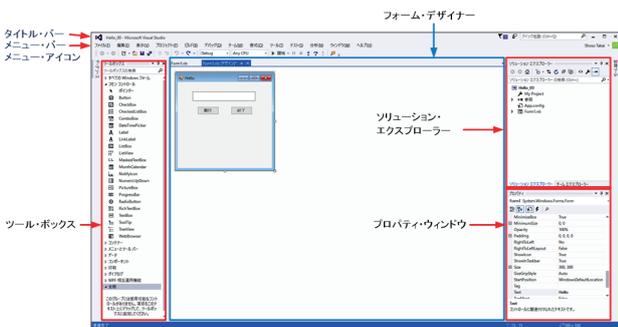


図 1.20 中央がフォーム・デザイナーの表示画面

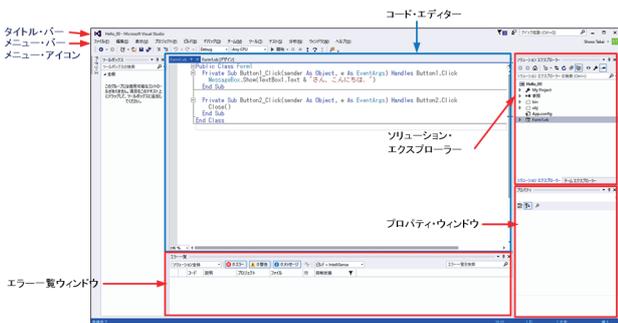


図 1.21 中央がコード・エディターの表示画面

通常の画面構成は、ツール・ボックスを表示した画

面で、中央がドキュメント・ウィンドウで、プログラミングの作業を行い、フォーム・デザイナーやコード・エディター画面が表示されます。また、画面の左右には、作業を能率的に行うための道具 (ツール) が配置され、ツール・ボックスやコンテナー、ソリューション・エクスプローラー、プロパティなどのウィンドウが表示されます (図 1.20~21)。

## 1.7 アプリケーション作成手順

ここでは、挨拶文「〇〇さん、こんにちは。」を表示する Windows フォーム・アプリケーション「Hello」を作成してみましょう。テキスト・ボックスに名前を入力して、「実行」ボタンをクリックすると、「〇〇さん、こんにちは。」というメッセージ・ボックスを表示するプログラムです。プログラムは「終了」ボタンのクリックで終了します (図 1.22)。



図 1.22 挨拶アプリケーション作成手順

作成手順は以下の通りです。

- (1) 新規プロジェクト「Hello\_00」を作成する。
- (2) フォーム・デザイナーを使って、フォームに Textbox と 2 つの Button を配置・設定する。
- (3) コード・エディターを使って VB のコードを記述する。
- (4) デバッグから「デバッグ開始」をクリックして、プログラムが正常に動作するか確認する。
- (5) リリース・ビルト機能を使って、完成したアプリケーションを実行用製品として登録する。

## 1.8 フォームのデザイン

ここでは、Windows フォーム・アプリケーションとして、ツール・ボックスの [COMMON CONTROLS] にある [TextBox] をクリックして、フォーム上にテキスト・ボックスを配置しています (図 1.23)。このテキスト・ボックスは、名前を入力するためのエリアです。

次に、このテキスト・ボックスの下に、「作成」と「終了」のボタンを左右に配置します (図 1.24)。更に、そのボタンの大きさや配置を、メニュー「書式」

の「整列」「同じサイズに整える」等から、整えます (図 1.25)。

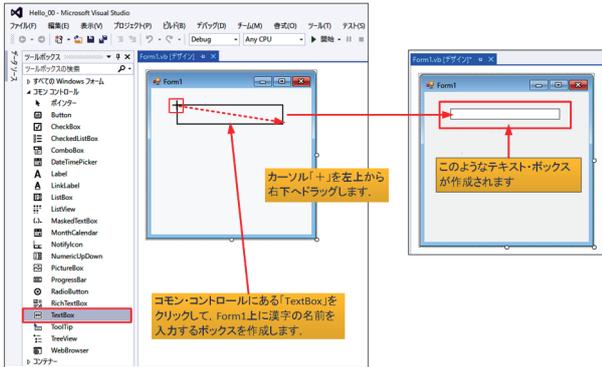


図 1.23 テキスト・ボックスの配置

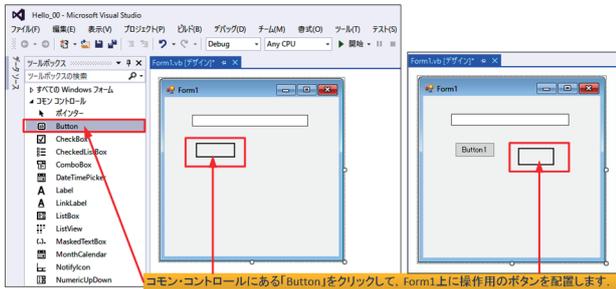


図 1.24 操作ボタン2個の配置

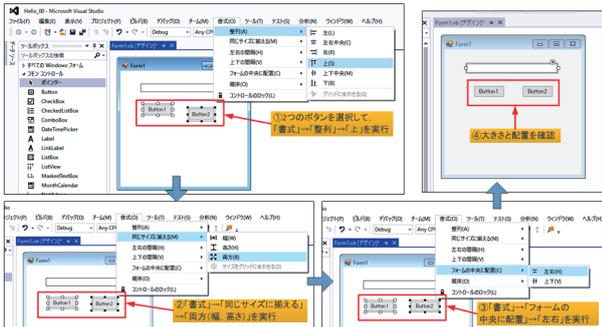


図 1.25 操作ボタンの大きさを同じにして整列

## 1.9 フォームやコントロールのプロパティの確認・設定

テキスト・ボックスの名前を確認するには、フォーム・デザイナーでテキスト・ボックスをクリックして選択し、プロパティ・ウィンドウの[オブジェクト名]ボックスまたは [(Name)] 欄で名前「TextBox1」を確認します (図 1.26)。

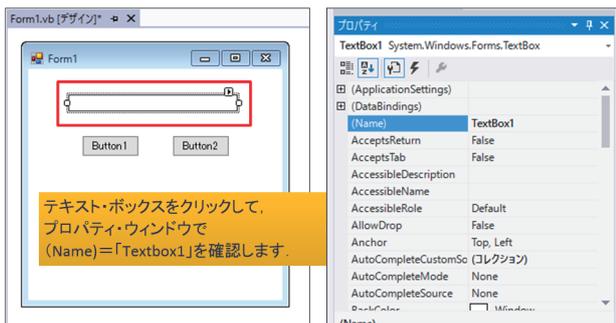


図 1.26 テキスト・ボックスの名前の確認

次にボタンの文字を「Button1」を「実行」、「Button2」を「終了」に変更しましょう。フォーム・デザイナーでボタンをクリックして選択し、プロパティ・ウィンドウの [Text] 欄で、初期値の「Button1」から「実行」に変更、同様に「Button2」を「終了」に変更します (図 1.27)。

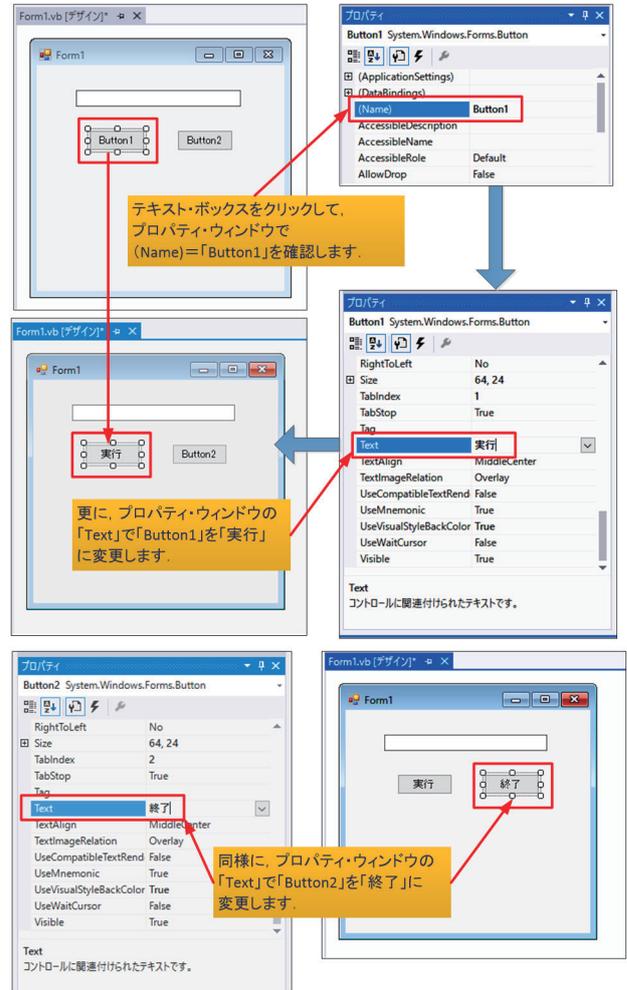


図 1.27 ボタン「Button1」「Button2」の文字を変更

フォームのタイトル変更は、フォーム上の何もない部分をクリックしてフォームを選択し、プロパティ・ウィンドウの [Text] 欄で、初期値の「Form1」から「Hello」に変更します (図 1.28)。

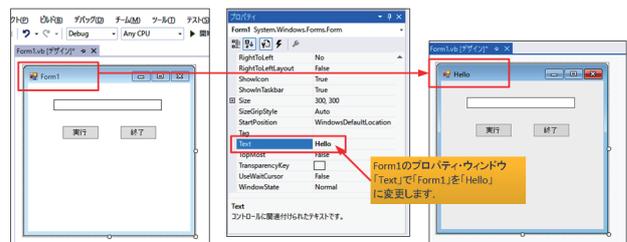


図 1.28 フォームのタイトル変更

再度、テキスト・ボックスに戻って、入力フォント、スタイル、サイズを変更し、入力文字が大きく太字に見えるように変更しましょう (図 1.29)。

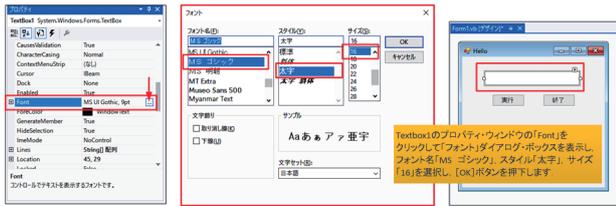


図 1.29 テキスト・ボックスのフォント属性の変更

## 2. Visual Basic プログラミング

### 2.1 コードの記述

VB では、フォームやコントロールに、以下のような何らかの出来事=イベントが発生したときに、処理を実行するようにしましょう。

- 1) 「ボタンがクリックされたとき」
- 2) 「テキスト・ボックスに文字が入力されたとき」
- 3) 「フォームが閉じられたとき」

この出来事を「イベント (Event)」と呼び、イベントが発生したときに実行されるプログラムを「イベント・ハンドラー (Event Handler)」と呼んでいます。

イベント・ハンドラーは、以下の 2 つの方法で作成することができます。

- 1) フォーム・デザイナーでコントロールをダブル・クリックする方法
- 2) コード・エディターでイベントを選択する方法

[実行] ボタン (名前は「Button1」) の Click イベント・ハンドラーを作成するには、[実行] ボタンをダブル・クリックします。

コード・エディター画面が開きますので、カーソルの位置に『「TextBox1」の文字と「さん、こんにちは。」を連結して、メッセージを表示せよ』という意味の命令文を、以下のように記述します (図 2.1)。

```
MessageBox.Show(TextBox1.Text & "さん、こんにちは。")
```

このコードを入力するとき、IDE に備えられている「インテリセンス (IntelliSense)」という入力支援機能を活用して、キーワードなどを効率的に入力することができます (図 2.1 中央部分)。

インテリセンスは、Microsoft 社のソフトウェアに搭載されている入力支援機能で、入力候補の表示によるオートコンプリート機能、タイプミス補正機能 (オートコレクト) などを総称した呼び方です。

該当キーワードを選択すると、キーワードの右側に「パラメーター・ヒント」として、入力している項目の構文がヒントとして表示されます。構文がうろ覚えでも、ヒントを参考にコードを書くことができます。

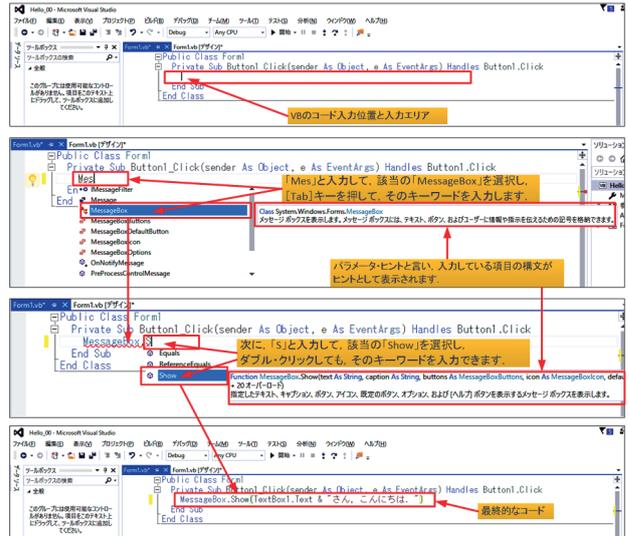


図 2.1 [実行] ボタン「Button1」のイベント・ハンドラー  
次に [終了] ボタン (名前は「Button2」) の Click イベント・ハンドラーを、[実行] ボタンと同じ操作で作成します。

Button2 のイベント・ハンドラーを作成するには、カーソルの位置に、「閉じる」という意味の「Close()」を記述します (図 2.2)。

#### Close ()



図 2.2 [終了] ボタン「Button2」のイベント・ハンドラー

### 2.2 デバッグ (テスト) の実行

イベント・ハンドラーが完成したら、正しく動作するかテストします。このことを、エラーを見つけ出すという意味で「デバッグ (Debug) : エラー=虫 (bug) を取る」と言います。

テストの実行は、ツール・バーの「デバッグ」から [デバッグ開始] ボタンをクリックします (図 2.3)。

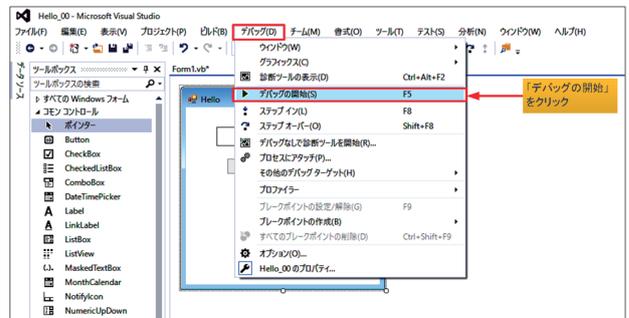


図 2.3 デバックから [デバック開始 (F5)] を実行

この [デバック開始] ボタンをクリックすると、自動的にファイルの保存とビルド (Build) が実行されます。ビルドはプログラムを翻訳して実行可能なファ

イルを作成する作業を指します。

なお、デバッグの開始に際し、スクリプト診断を行うために、インストール先フォルダーのアクセス許可を求めてきますので、アクセス権を与えるために[OK]ボタンを押下します(図2.4)。



図 2.4 フォルダーにアクセス権を許可

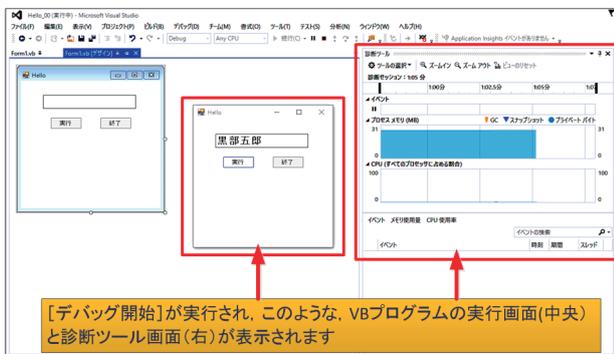


図 2.5 デバッグ画面の表示とテキストの入力

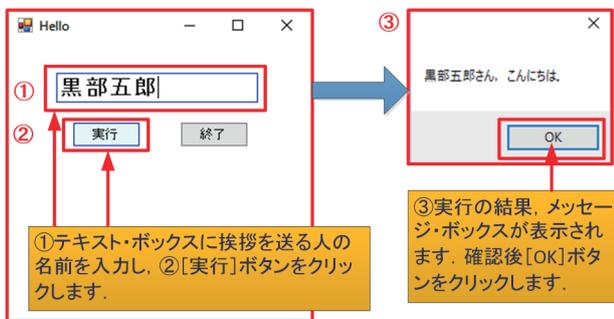


図 2.6 デバッグの実行

[デバッグ開始] ボタンでは「デバッグビルド」といって、デバッグを行うための情報を含んだ実行ファイルが作成されます(図2.5)。

このVBプログラム「Hello\_00」のデバッグでは、テキスト・ボックスに挨拶を送る人の名前を入力し、[実行]ボタンをクリックします。「〇〇さん、こんにちは。」というメッセージ・ボックスが表示されれば、正常に動作しています(図2.6)。

エラーがある場合は、コード入力時に通常のエラーが警告表示されますので、「エラー一覧」から説明を読んで、コード・エディターでコードを正しく修正して下さい(図2.7)。インテリセンス入力支援機能を

使用してキーワード入力していれば、通常のエラーの大部分は防ぐことができます(図2.8の①②)。次に「デバッグ開始」を実行すると、ビルト・エラーが表示されます(図2.8の③)。

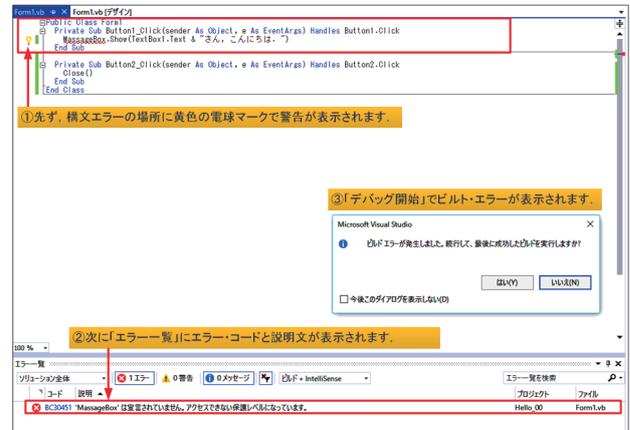


図 2.8 通常のエラー表示とデバッグによるエラー表示

## 2.3 リリース・ビルドとプログラムの実行

テストの結果、動作に問題がなければ、アプリケーション・プログラムを完成版としてビルドする「リリース・ビルド (Release Build)」を実行します。

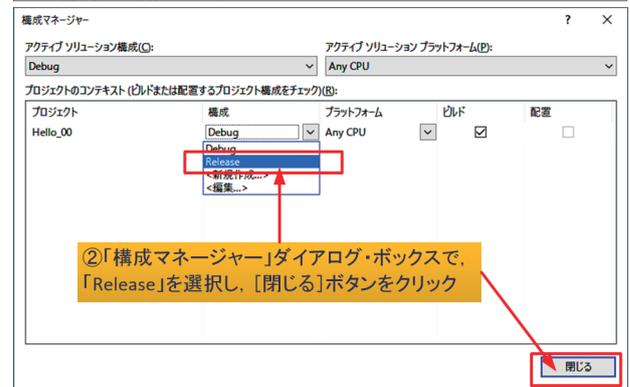
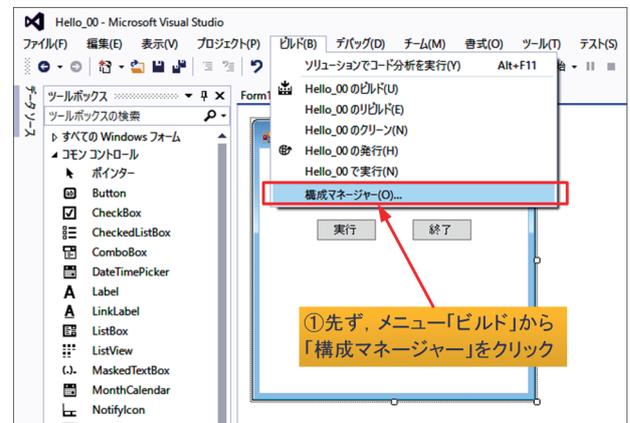


図 2.9 構成マネージャーを「Release」に変更

リリース・ビルドは、ツール・バーの「ビルド」から [構成マネージャー] ボタンをクリックして、プロジェクトの構成を「Debug」から「Release」に変更

して [閉じる] ボタンをクリックします (図 2.9) .  
その後, 再度メニュー「ビルド」から「当該プロジェクトのビルド」, ここでは「Hello\_00 のビルド」をクリックして, 作成します (図 2.10) .

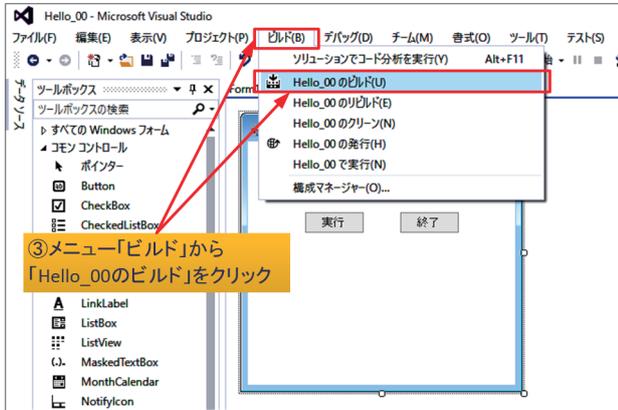


図 2.10 「リリースビルド (Release Build) 」の実行

「Release」されたプログラムは, フォルダ「Visual Basic」の「Hello」→「Hello」→「Bin」→「Release」に, ファイル名「Hello.exe」を確認し (図 2.11) , この実行形式プログラムをダブル・クリックして実行してみてください (図 2.12) .

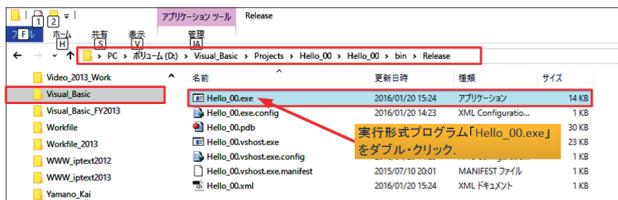


図 2.11 実行形式プログラムの確認と実行



図 2.12 実行形式プログラム「Hello\_00.exe」の実行

## 2.4 プロジェクトの保存

では最後に, プロジェクトの総てを保存しましょう.

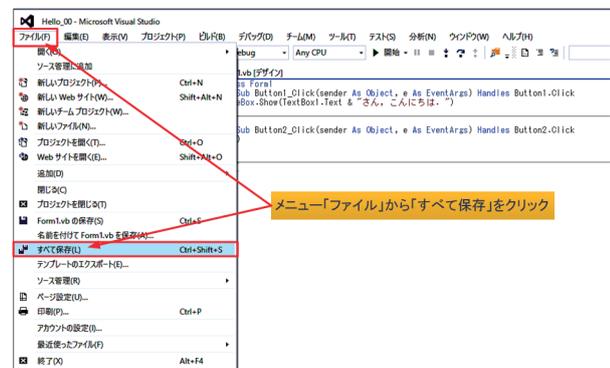


図 2.13 「ファイル」から「すべて保存」をクリック  
メニュー「ファイル」から [すべてを保存] ボタン,

(図 2.13) またはウィンドウの [閉じる] ボタンを押下すると, 「プロジェクトの保存」ダイアログ・ボックスが表示されるので, 名前を確認し, 場所を [参照] ボタンで変更または確認し, [上書き保存] ボタンを押下します (図 2.14) .

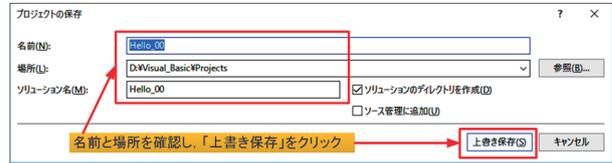


図 2.14 「すべて保存」から「上書き保存」をクリック

## 2.5 Visual Studio 2015 の終了

Visual Studio 2015 を終了するには, メニュー・バー「ファイル」から [終了] ボタンをクリックするか, ウィンドウの [閉じる] ボタンをクリックします.

## 2.6 ファイル構成

プロジェクトは, ソリューション名のフォルダに階層構造で保存されます. 主なファイル構成は図 2.15 の通りです.

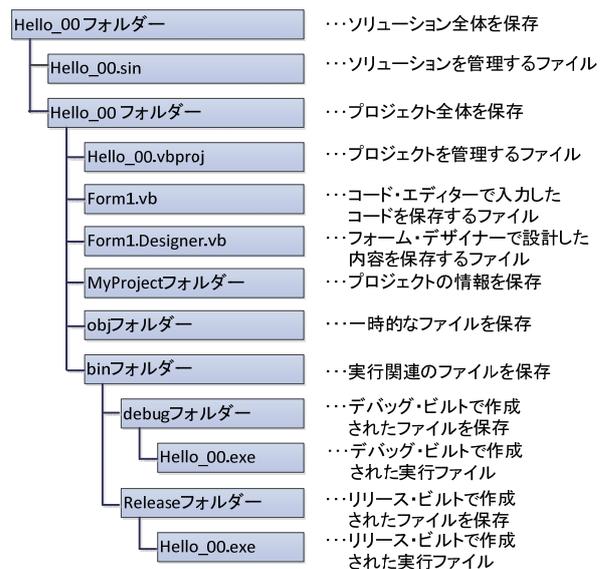


図 2.15 プロジェクト「Hello\_00」の主なファイル構成

## 3. コンソール・アプリケーションの作成

### 3.1 自然対数の底 e の計算プログラム

自然対数の底  $e = 2.718281828459$  は,  $e^x$  の Taylor 展開式で,  $x = 1$  として, 次の級数展開で求めることができます.

$$e^x = 1 + \frac{x}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots, \quad -\infty < x < \infty$$

$$x=1 \text{ として } e = 1 + \frac{1}{1!} + \frac{1}{2!} + \frac{1}{3!} + \dots + \frac{1}{10!}$$

分母の階乗計算を10! 程度でも、小数点以下5桁位の精度で、計算することができですが、ここでは10<sup>-12</sup>の精度で求めてみましょう。また、階乗を計算する関数Factorialも作成し、同時に併記してみます。

自然対数の底 e は Euler's Constant と呼ばれていますので、プロジェクト名を「Euler\_Constant\_00」として、このプログラムを Console Application で作成してみましょう。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。

### 3.2 コンソール・アプリケーションの作成

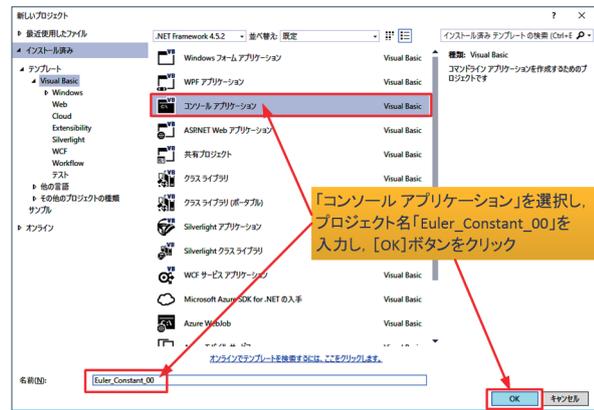


図 3.1 コンソール・アプリケーションの開始

まず、「新しいプロジェクト」ダイアログ・ボックスから「コンソール アプリケーション」を選択し、プロジェクト名「Euler\_Constant\_00」を入力して [OK] ボタンを押します (図3.1)。

画面には VB のソース・コードを入力するエディター画面が表示され、プログラムの単位を表すモジュール (モジュール名: Module1) と、その中にメイン・プロシーチャーを記述するところで、カーソルが位置づけられますので、主 VB プログラムを記述します (図3.2)。

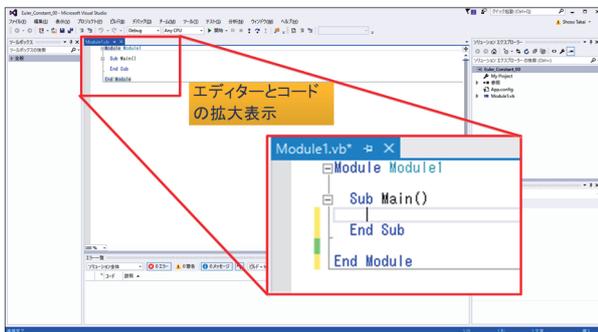


図 3.2 コード・エディターの入力画面

ここでは、この中で使用する Function プロシーチャーの Factorial (再帰呼び出し関数) をメイン・プロシーチャーの後に記述しています (図3.3)。

### 3.3 コンソール・アプリケーションの実行

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。通常、コマンド・プロンプトの画面は、計算結果の表示が終わると、直ぐに閉じてしまいますので、メイン・プロシーチャーの最後には、データの入力待ち状態にして、結果を確認できるように、以下のコードを差し込んでおきます (図3.4)。

```
n = Console.ReadLine()
```

結果の確認が終われば、コマンド・プロンプト画面を閉じて下さい (図3.5)。

```
Function Factorial
  (ByVal n As Integer) As Double
  If n = 0 Then
    Return 1
  Else
    Return n * Factorial(n - 1)
  End If
End Function
```

図 3.3 Function プロシーチャー Factorial のコード



図 3.4 エディターでの VB コード入力

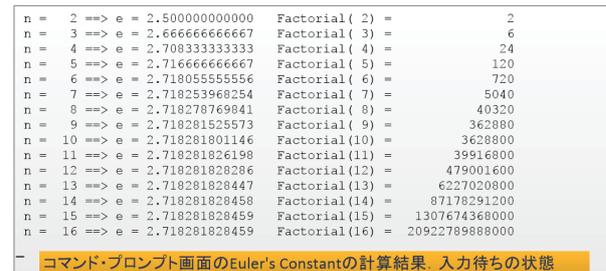
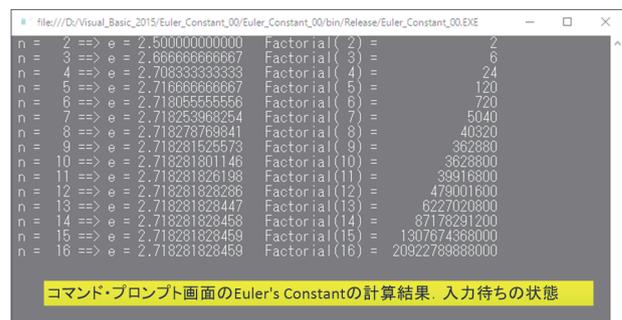


図 3.5 計算結果を表示するコマンド・プロンプト画面

### 3.3 その他のプログラム

以下は、本学の応用情報処理“Visual Basic”で使用した、月単位カレンダー(図 3.6)、年単位カレンダー(図 3.8)の出力結果と、B.W.Kernighan, D.M. Richie の C 言語テキストに出てくる 2 番目の例文を拡張した「摂氏、華氏、絶対温度の換算表」の VB プログラムである[4]。数値を出力する際に出力幅を固定し、右詰めにするなどの、書式を指定する「数値書式指定文字列」の例が少ないので、ここにその例題プログラムを掲載した(図 3.7)。

#### (1) 月単位カレンダー (Monthly Calendar)

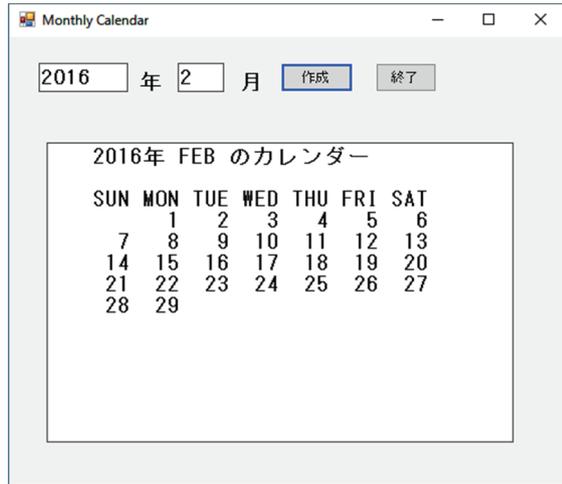


図 3.6 Monthly Calendar の出力例

#### (2) 摂氏、華氏、絶対温度の換算表プログラム

```
Module Module1

Sub Main()
    Dim fahr, celsius, kelvin As Single
    Dim lower, upper, temp As Long
    Dim ipn As String
    ' start here
    Console.WriteLine _
        ("Celsius Fahrenheit Kelvin")

    lower = -40
    upper = 42
    temp = lower
    Do While temp <= upper
        celsius = temp
        fahr = 9 * celsius / 5 + 32
        kelvin = celsius + 273.15
        Console.WriteLine("{0,7:F2}", celsius)
        Console.WriteLine(" ")
        Console.WriteLine("{0,7:F2}", fahr)
        Console.WriteLine(" ")
        Console.WriteLine("{0,7:F2}", kelvin)
        temp += 1
    Loop

    Console.WriteLine("何かを文字を入力して下さい>")
    ipn = Console.ReadLine()

End Sub

End Module
```

摂氏、華氏、絶対温度の換算表VBプログラム

図 3.7 Temperature Convert Table の VB プログラム例

#### (3) 年単位カレンダー (Annual Calendar)

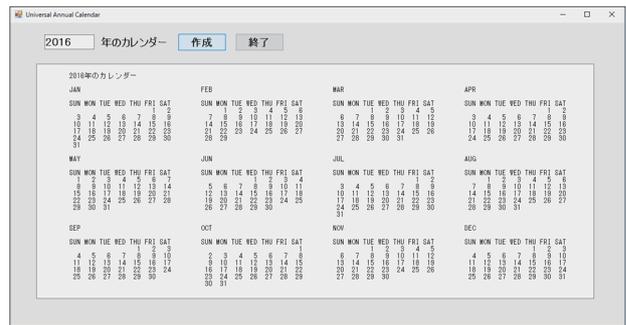


図 3.8 Universal Annual Calendar の出力例

### 3.4 Visual Studio ガイドから

日経ソフトウェアの 2016 年 1 月号に“決定版 Visual Studio ガイド”特集[5]があったので、その記事を紹介し、これからの VS 2015 の活用に大いに参考にして、C#やVBによる Smartphone などのアプリケーション開発に応用して欲しい。

#### 第 1 部：触って覚える Visual Studio の流儀

VS 2015 が、NET アプリケーションだけでなく、Web や iOS/Android アプリケーションを開発できるので、Windows Form ではなく GUI ライブラリの WPF (Windows Presentation Foundation) を利用する例が掲載され、C#6.0 で Todo List を作成している。WPF は表現力豊かなアプリケーションや最新の Windows が備える機能に対応した Desktop アプリに最適としている。

#### 第 2 部：Web API を使う実用アプリを作ってみよう

ここでは、VS 2015 の高度な IntelliSense (入力支援機能) やデバッグ機能を活用し、実用的な WPF アプリとして、簡易的な Wikipedia Viewer を、キーワードに関する Wikipedia の記述を要約して返してくれる「Wikipedia API」を使っているという。

(参照 URL=<http://wikipedia.simpleapi.net>)

#### 第 3 部：UWP アプリの考え方を知る

ここでは、Windows 10 に搭載された新しい仕組み UWP (Universal Windows Platform) 向けの簡単なアプリとして、モバイル画面から 84 インチの大画面をもつ Microsoft の Surface Hub など、同一の画面で幅広いデバイスに対応できる「アダプティブ UI」という機能をもちいた例を挙げている。様々な解像度のプラットフォームで、快適に操作できるようにする「レスポンスな画面設計」を行う必要があるとして、その考え方を、例題を通して説明している。詳しくは [URL=<https://msdn.microsoft.com/en-us/library/dn975273.aspx>] を参照されたい。

## 第4部：VSによるクロス・プラットフォーム開発

VS 2015 で開発できるのはWindows 向けアプリだけでなく、最新の Web プラットフォームや Android /iOS 向けのアプリ開発に対応し、.NET Framework で利用できる C#, Visual Basic, F#, C++といった言語以外に、HTML/JavaScript や TypeScript などの言語向けに支援機能を提供しているという。ここでは「(本体の意味ではない?) クロス・プラットフォーム開発機能」を取り上げている。

[様々なプラットフォームのアプリ開発] とは、

### ・Cordova でハイブリッド・アプリ開発

Apache Cordova とは、ハイブリッド・アプリの開発が行えるフレームワーク。ハイブリッド・アプリとは、ネイティブ・アプリの中に WebView と呼ばれる簡易ブラウザのようなものを表示し、その中に HTML を表示させるアプリのことで、アプリケーション開発に対応した templete をはじめ、Intellisense, デバッガ, 各種コンフィグ・ファイル (config.xml) の編集ウィンドウなどを標準で利用できるようだ。

### ・Xamarin でスマホ・アプリ開発

Xamarin は、Android/iOS 向けのアプリを、.NET を使って開発できる有償のライブラリおよび製品群をいう。C#で開発できるようだ。

### ・C++の開発環境も強化

C++11 言語仕様への対応がほぼ完了し、C++開発環境が大幅に進化したという。

この他、[UWP ブリッジで様々なアプリを開発]、[様々な環境で動作する Visual Studio Code] も紹介されている。

## 4. これからの課題

### 4.1 これからの統合開発環境と応用情報処理科目

このような記事[5]を読むと、大学におけるプログラミングなどの授業が、社会一般のアプリケーション開発に対応してきていないように思われる。

現在、教養教育向けにプログラミングの授業として「応用情報処理科目」が開講され、筆者が「JavaScript Programming」と「Visual Basic Programming」を、准教授の奥村弘氏が「C Programming」と「Cによる数値計算 Programming」を、OS系で布村紀男教授が「UNIX 入門」を、上木佐季子准教授が「HTML と CSS 入門」の授業を開講しているが、受講学生が少なく、プログラミングへの関心がきわめて低い。

昨年8月の夏休み期間に、「小学5・6年生にもわかる、大人に負けない JavaScript プログラミング」

という公開講座を開催したところ、8人の申し込みがあり、実際には6人が受講し、無事終了したが、小学生はみんな「やる気満々」で、目を輝かせて JavaScript プログラミングに挑戦していたのが、印象的であった。

高価な IDE Visual Studio 2010 など、無用の長物で、Eclipse や VS 2015 で十分である。かつて筆者も Eclipse の IDE 下で Java Programming を、CPAD 環境下で C Programming を数年開講したが、受講学生はやはり少なかった。最終的に年単位カレンダーを作成するプログラムを課題にしたが、自分でアルゴリズムやフォーマットを考え、独自にプログラムを開発してきた学生は、ほんの2~3人であった。

最近、「子どもを億万長者にしたいければプログラミングの基礎を教えなさい(松林弘治著, KADOKAWA, 2015.02.20, ISBN978-4-04-067378-3)」など、子ども向けプログラミングを進める本が少しずつ出版されてきている。この本で、オバマ大統領や Steve Jobs 氏がプログラミングを推奨していると言っている。

本学でも、もっと多くの学生が「基礎的なプログラミング」の授業を受けられるように、PC教室の環境を整備するとともに、以前のPCの様に、IDEとして Eclipse や VS 2015 を導入し、Web Browser も IE や Firefox の2つにとどまらず、Google Chrome や Safari、日本版の LunaScape や Sleipnir など、マルチ・プラットフォーム環境でプログラミングが楽しめるように願うばかりである。関係各位の努力に期待したい。

### 4.2 Visual Studio のための参考テキスト

[1] きたみ あきこ著、これからは始める Visual Basic 2010, 秀和システム, 2011.04.01, ISBN978-4-7980-2931-3, ¥2,600+TAX

[2] 中島省吾著、わかりすぎる Visual Basic 2013 の教科書, SCC Books, 2014.03.15, ISBN978-4-88647-219-9, ¥2,000+TAX.

[3] 高橋広樹著、かんたん Visual Basic 2013, 技術評論社, 2010.05.25, ISBN978-4-7741-4231-9, ¥2,580+TAX.

[4] Brian.W.Kernighan, Dennis.M. Richie 著, The C PROGRAMMING LANGUAGE(Second Edition), 1988, 8-13, ISBN0-13-110370-9, \$40+TAX

[5] 五十嵐祐貴 (Microsoft MVP) 著, 決定版 Visual Studio ガイド, 日経ソフトウェア, 日経 BP 社, 2016年1月号, Vol.19, No.1, Ser.230, 10-49, ISSN1347-4685, ¥1,220+TAX4