

Visual Basic入門

多川, 孝央
九州大学情報基盤センター

<https://doi.org/10.15017/1470506>

出版情報：九州大学情報基盤センター広報：学内共同利用版. 2 (3), pp.160-166, 2002-11. 九州大学
情報基盤センター
バージョン：
権利関係：

Visual Basic 入門

多川孝央 (九州大学情報基盤センター)

tagawat@cc.kyushu-u.ac.jp

1. はじめに

情報基盤センターの教育用システムでは、プログラム開発用の環境として Microsoft Visual Studio 6.0 を利用者用パーソナルコンピュータにインストールしています。Visual Studio には Visual Basic、Visual C++、Visual J が含まれていますが、本稿ではこのうち Visual Basic の使いかたについて紹介します。

2. Visual Basic の操作とサンプルプログラム

Visual Basic は Windows で動作するアプリケーションプログラムを開発するための言語環境です。Visual Basic の特徴は、マウス操作を中心としたグラフィカルな開発環境のもとで、高度な知識がなくても簡単に Windows のプログラムを作成することが可能なことです。ここでは、Visual Basic の利用に関連する初歩的な知識や概念を、簡単なサンプルプログラムの作成の様子を交えながら紹介します。

Visual Basic の起動

Visual Basic を起動するには、スタートメニューから「プログラム(P)」 「Microsoft Visual Studio 6.0」 「Microsoft Visual Basic 6.0」と順を追って選択してください。起動すると、最初に「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。普通の Windows アプリケーションを作成するというので、今回は「標準.EXE」を選択してください。

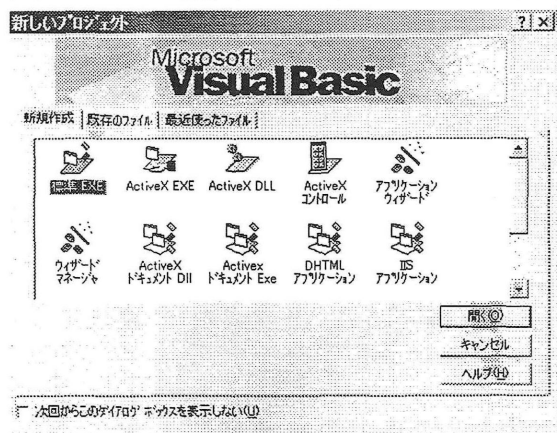


図 1 「新しいプロジェクト」 選択画面

Visual Basic のウィンドウが表示されます。

ここで、画面の左側にある、いくつものアイコンが並んでいる領域を「ツールボックス」、画面の右側の上にある樹形状の表示領域を「プロジェクトエクスプローラ」、その下にある表示領域を「プロパティウインドウ」といいます。

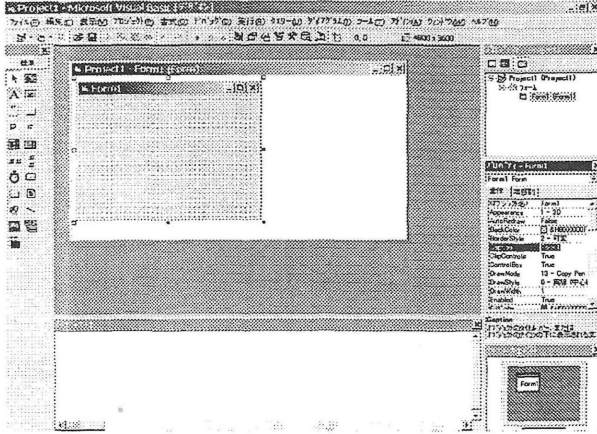


図 2 Visual Basic6.0 の画面

中央に表示されているのは「フォームレイアウトウインドウ」といいます。フォームレイアウトウインドウの中には、「フォーム」と呼ばれる四角形の領域が配置されています。このフォームは、作成するプログラムの外見の雛形になるものです。

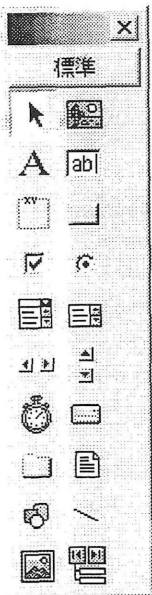


図 3 ツールボックス

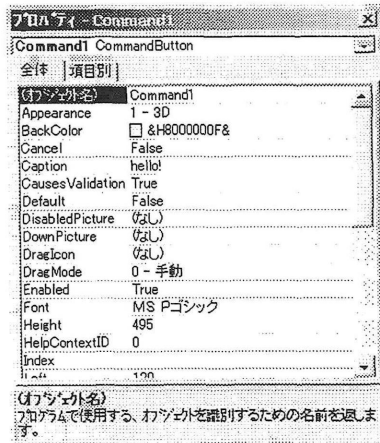


図 4 プロパティウインドウ

ツールボックスに並んでいるアイコンを「コントロール」といいます。各コントロールは様々な機能を持っていて、プログラマはこれをフォームの上に配置することによってプログラムのユーザインタフェースをデザインすることになります。

Visual Basic のプログラミングは、基本的にはフォームにコントロールを配置し、各コントロールに関する設定をプロパティウインドウで行い、その後ユーザがコントロールに対して操作を行ったときに実行する処理を記述し、最後にコンパイルすることによって実行可能なアプリケーションを作成するという手順になります。

プロジェクト

プロジェクトとは、Visual Basic におけるプログラミングの単位で、一つのプロジェクトで一個のアプリケーションプログラムを作成することになります。プロジェクトは、プロジェクトエクスプローラでは最上部、樹形状の幹にあたる部分に存在し、その下に一個ないし複数のフォームが配置されます。プロジェクトエクスプローラ上でプロジェクトをクリックすると、プロパティウインドウにはプロジェクトの名前だけが表示されます。「全体」と名前をついたタブの中の「(オブジェクト名)」という項目にカーソルを合わせて文字を入力すると、プロジェクトの名前が変化します。最終的に作成されるアプリケーションの名前はこのオブジェクト名になります。この例では、作成するプログラムの内容を示す `helloworld` という名前にします。

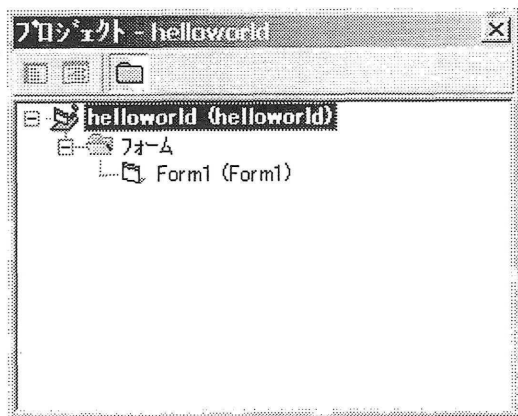


図 5 プロジェクトエクスプローラ

コントロールの配置

コントロールとは Visual Basic のプログラムの部品で、Visual Basic のデザインは、このコントロールの組み合わせによって行われます。コントロールには「ボタン」「テキストボックス」「チェックボックス」などがあります。ツールボックスのアイコンをクリックし、次にマウスのカーソルをフォームの上でドラッグすると、フォーム上にコントロールが配置されることになります。一旦配置したコントロールは、マウスでの操作によってフォームの上での位置を変更したり、大きさを変えたりすることができます。フォーム上に配置

したコントロールをクリックすると、プロパティウインドウにそのコントロールのプロパティが表示されます。このプロパティの値を設定することで、フォーム上の各コントロールの見かけや細かい動作を変更することができます。この例では、テキストボックスとコマンドボタン、ラベルを利用しますので、それらを図のようにフォーム上に置きます。

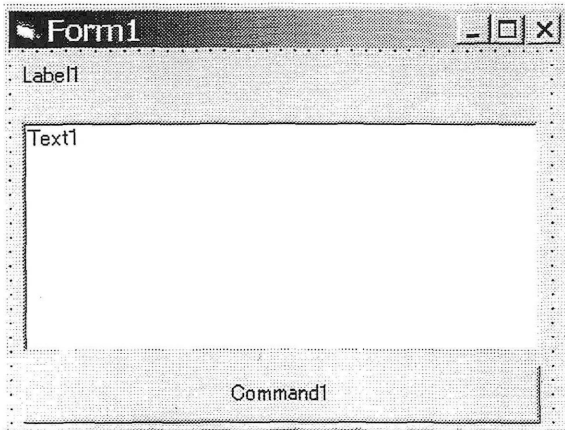


図 6 フォーム上にコントロールを配置した例

ここで、外見を整えるためにフォーム、テキストボックス、コマンドボタン、ラベルのプロパティを適切に設定します。フォームをクリックし、プロパティウインドウ上の「Caption」という項目を「Sample Program」と変更します。すると、フォームの上部の青い部分に表示される文字が「Form1」から「Sample Program」に変わります。同様に、配置したコマンドボタンの「Caption」を「Command1」から「hello!」に、テキストボックスの「Text」という項目を「メッセージはここに表示されます」に、ラベルの「Caption」を「Label1」から「テスト」に変更すると、フォームの外見は下のようになります。これで、ひとまずプログラムの外見は完成しました。

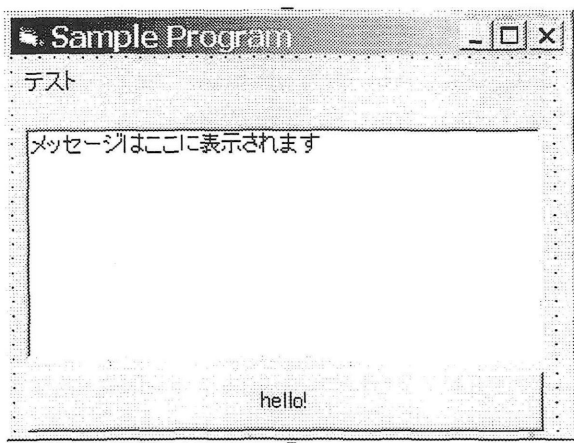


図 7 プロパティの設定でコントロールの表示内容を変更した例

コードの記述

コントロールを配置することでアプリケーションの外見を作ることができました。次は、アプリケーションの動作を決定する必要があります。これは、フォーム自身やそれぞれのコントロールに対してユーザが特定の操作を行った場合に（これを「イベント」と呼びます）どのように反応するかを記述することによって行われます。

フォーム上のコントロールをダブルクリックすると、「コードエディタウインドウ」が開きます。Visual Basic では、このウインドウに Basic 言語で処理を書くことでアプリケーションの動作を指示します。このサンプルプログラムでは、ユーザの操作はコマンドボタンをクリックするだけですので、フォームレイアウトウインドウ上のコマンドボタンをダブルクリックします。すると、下のようにコードエディタウインドウが表示されます。

詳しい説明はここでは省略しますが、「Private Sub Command1_Click()」という行と、「End Sub」という行の間に、Basic 言語で操作を記述することによって、ボタンがクリックされた際のプログラムの動作を設定することになります。このサンプルプログラムでは、
`Text1.Text = "Hello,world!"`

とだけ記述します。これは、「Text1.Text と変数に、"Hello,world!"という値を代入する」という操作になります。

ところで、このプログラムにおいては、Text1.Text という変数は、フォーム上に配置されたテキストボックス Text1 のテキストボックスの表示内容を指します。すなわち、この記述内容は、テキストボックスの表示内容を"Hello,world!"に変更することを意味しています。この一行をボタンがクリックされた際の動作として記述したことによって、「ボタンをクリックしたら、テキストボックスに"Hello,world!"と表示される」という機能が実現されたことになります。

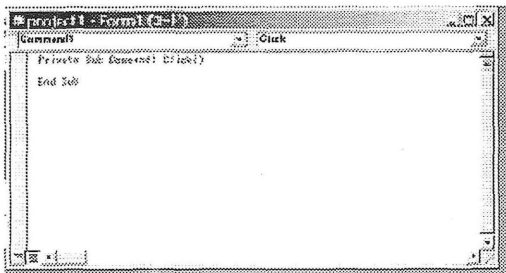


図 8 コードエディタウインドウ

動作テスト・デバッグ

フォームとコントロールを配置し、コードを記述したらひとまずプログラムは完成です。この後は出来上がったプログラムを実行可能なアプリケーションの形にする作業が残っていますが、その前に、プログラムが自分の意図した通りに動作するかをチェックする必要があります。

では、プログラムをテスト的に実行してみます。メニューから「実行(R)」>「開始(S) F5」を選択するか、キーボードの上段の「F5」と書いてあるキーを押す、あるいは、ツールバー上で右向きの三角のアイコンをクリックすると、プログラムが実行され、フォームが表示されます。この時、フォームの上に配置されたコントロールの外見は、プロパティウインドウで設定した通りになっています。

ここでコマンドボタンを押すと、テキストボックスの内容が、「メッセージはここに表示されます」から、「Hello, world!」に変化します。意図した通りの動作をしていますので、これでテストは終了です。メニューから「実行(R)」>「終了(E)」を選択するか、ツールバー上で停止を示す四角形のアイコンをクリックすることで、テスト的な実行を終了させます。

プログラムが正常に実行できないことがシステムにとって明らかな場合には、エラーの種類を示す番号とともにエラー内容を示すメッセージウインドウが表示されます。ここで、ウインドウの中の「デバッグ」と書いてあるボタンをクリックすると、コードエディタウインドウ中のエラーの発生した箇所が黄色く表示されます。ここを修正し、もういちどプログラムテストを行ってください。

実行形式ファイルの作成

プログラムが自分の意図の通りに動作することが確認できたら、実行形式のファイルを作成します。「ファイル(F)」から「helloworld.exe の作成」を選択してください。保存場所を選ぶ画面が表示されます。プログラムを、現在利用している開発環境なしで実行可能な形にしたものが、この画面の操作によって保存されることとなります。ここで保存されるプログラムファイルの名称は、現在のプロジェクトの名称にあわせたものになります。Visual Basic を起動したときそのままプロジェクト名を変更していない場合は、`project1.exe` になります。

Visual Basic 6.0 で作成したアプリケーションプログラムは、必ずしも単体で完全に動作するとは限りません。多くの場合、Visual Basic 用のランタイムと呼ばれるファイル群が実行するパーソナルコンピュータにインストールされている必要があります。Visual Basic がインストールされているパーソナルコンピュータでは、ランタイムを追加してインストールする必要はありません。

利用の終了

Visual Basic での作業を終了する時には、メニューから「ファイル(F)」>「Microsoft Visual Basic の終了(X) Alt + Q」と選択してください。プロジェクトのファイルとフォームのファイル（それぞれ拡張子が `vbp` と `frm` になります）に、プロジェクトの内容とフォームの内容が保存されます。これまでに行った作業の続きを再開したい時には、Visual Basic 起動時の画面で既存のファイルを開くことを選択し、適切なプロジェクトファイルを選択すれば、前回に保存した時の状態から作業を継続することが可能です。

3. おわりに

本稿ではサンプルプログラムの作成例を交えながら Visual Basic の操作に関する初歩的な事柄を紹介しました。ただし、紙幅の都合から Basic 言語によるコーディングやデバッグツールの使用法などは省略してあります。また、コントロールやイベントについても特定のものしか扱っていません。

Visual Basic では、コントロールとイベント、プロパティの設定の組み合わせやコードの記述によって、様々な機能のアプリケーションを簡単に作成することが可能です。その方法は、本稿に示した手順と基本的には同じものです。本稿が Visual Basic の利用の入門としてお役に立てば幸いです。