

C#

Chapter6_Windowsアプリケーション 作りの基礎

この章からはWindowsアプリケーション作成を行う。

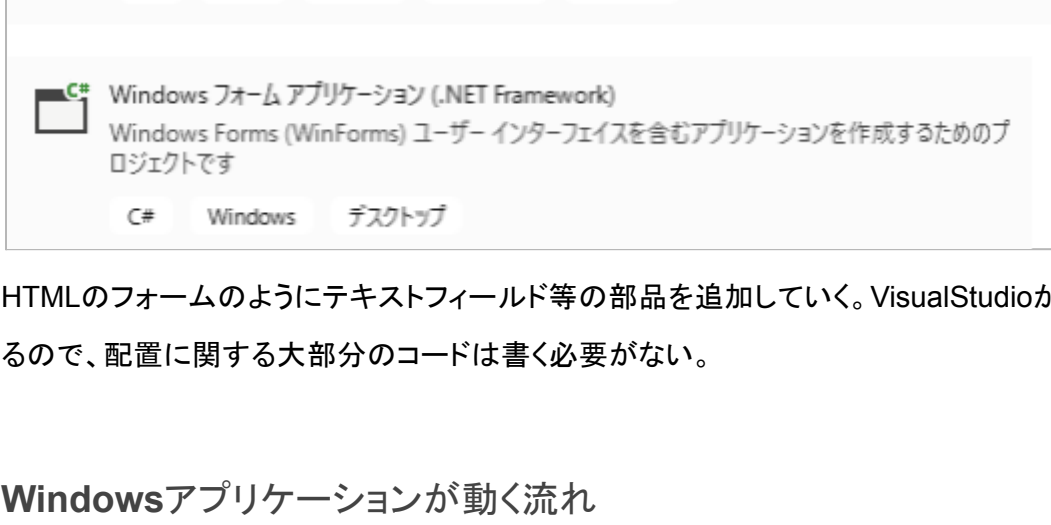
6-1 Windowsアプリ作りの概要

Windowsアプリケーション作成の概要を学ぶ。

Windowsアプリケーションを作るには

これまでの「コンソールアプリケーション」ではなく、「Windowsフォームアプリケーション」を使う。

図6-1 「コンソールアプリケーション」と「Windowsフォームアプリケーション」



HTMLのフォームのようにテキストフィールド等の部品を追加していく。VisualStudioから部品をドラッグ & ドロップで配置するので、配置に関する大部分のコードは書く必要がない。

Windowsアプリケーションが動く流れ

アプリケーション起動後の動作の流れは次の通り。



6-2 ハローワールド～最初のWindowsアプリケーション～

Windowsアプリケーション作成の基本を学ぶ。

Windowsアプリケーションの概要

Windowsアプリケーション作成前に用語をいくつか覚える必要がある。

表6-1 Windowsアプリケーション作成における用語

用語	説明
フォーム	Windowsアプリケーションの土台になるもので、HTMLのフォームとほぼ同じイメージ。
コントロール	フォームに配置する部品。ラベル・ボタン・テキストボックス等。
イベント	キーボードやマウス等からの入力。
イベントハンドラ	イベントに対応して動作するメソッド。
イベントドリブン	イベントハンドラを実行する仕組み。イベントに対応した処理のみを行う方式。

Windowsアプリケーションの作成手順

Windowsアプリケーションの作成手順は以下のステップを踏む。

- コントロールをフォームに配置する
- コントロールにイベントハンドラを追加する
- 入力に応じた処理をイベントハンドラに書く

以下手順は後ほど詳細説明するので、ここでは説明を省略する。

- ◆手順1 コントロールをフォームに配置する
- ◆手順2 コントロールにイベントハンドラを追加する
- ◆手順3 入力に応じた処理をイベントハンドラに書く

🔧 イベントの種類

コントロールには多彩なイベントがあり、全てを紹介するのは現実的ではない。
以下のリファレンスから全イベントを確認できる。

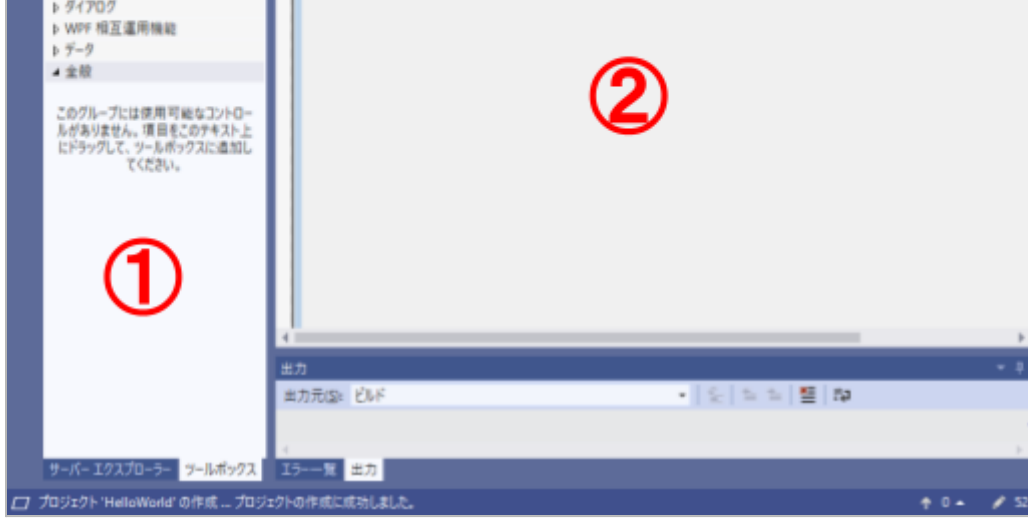
[Control クラス \(System.Windows.Forms\)](#)

Step01 プロジェクトを作成しよう

Windowsアプリケーションを作成するため、プロジェクトをどのように用意すればよいのかを学ぶ。

Windowsアプリケーションを作成するには、プロジェクト作成時に「コンソールアプリケーション」ではなく、「Windowsフォームアプリケーション」を使う。

図6-2 Windowsフォームアプリケーション



- ①ツールボックス
- ②Windowsフォームデザイナー
- ③プロパティウィンドウ

◆Windowsフォームデザイナー

アプリケーションの画面設計をこのウィンドウで行う。
ツールボックスからコントロールをドラッグ & ドロップして配置していく。

◆ツールボックス

テキストボックス、ラジオボタン等の各種コントロールが用意されている。

◆プロパティウィンドウ

フォームと配置したコントロールの設定を行う。
イベントとの関連付けもこのウィンドウで行う。

🔧 ツールボックスがない？

初回起動時はツールボックスが最小化されており、画面左側にある「ツールボックス」をクリックすれば表示される。

Step02 コントロールを配置しよう

コントロールの配置方法を学ぶ。

◆ボタンを配置する

ツールボックスからButtonをフォームにドラッグ & ドロップで配置する。

◆ボタンの表示を変更する

配置したボタンをクリックすることで選択状態にできる。その上でプロパティウィンドウのプロパティ「Text」の値を変更すれば、ボタンのテキスト表示が変更される。

◆ボタンの名前を変更する

プロパティ「Name」の値を変更することでボタンの名前を変更することができる。
この値はプログラム中で変数名として利用するので、使いやすい名前にしておくこと。

◆ラベルを配置する

ボタンを配置したときと同様の手順でラベルを配置する。

◆ラベルの表示を変更する

ボタンの表示を変更したときと同様の手順でラベルの表示を変更する。

◆ラベルの名前を変更する

ボタンの名前を変更したときと同じ手順でラベルの名前を変更する。

解説 コントロールとは何か

コントロールはプログラマ的にはControlクラスを継承したクラスである。
それぞれのクラスがプロパティを持ち、プロパティを変更することで挙動や表示を変更することができる。
[Control クラス \(System.Windows.Forms\)](#)

Step03 ボタンにイベントハンドラを設定しよう

コントロールにイベントハンドラを設定する方法を学ぶ。
方法は対象のコントロールを選択状態にして、プロパティウィンドウのイベントタブで「Click」等のイベントのプロパティに名前を付ければ良い。
付けた名前はイベントハンドラの名前としてコード上にイベントハンドラが自動生成される。

解説 イベントハンドラの追加

プロパティウィンドウのイベントタブでは設定可能なイベントが列挙されており、それぞれ値を設定することでイベントハンドラがコード上に自動生成される。

🔧 ダブルクリックでもイベントハンドラを追加できる

フォーム上に配置されたコントロールをダブルクリックすることでイベントハンドラを追加することができる。何のイベントを追加するかはVisual Studio任せなので、任意のイベントハンドラを追加したい場合はプロパティウィンドウから追加を行ったほうが良い。

Step04 イベントハンドラの中身を書こう

イベントハンドラの中身を実装する方法を学ぶ。
自動生成されたイベントハンドラに処理を書く。

解説 イベントハンドラでラベルの文字を変更する

フォーム上に配置したコントロールはプロパティで設定した名前(Name)を変数として使うことで参照できる。
フォームに配置したラベル名が「helloLabel」ならばhelloLabel.Text = "Hello, World!"; とすることで、ラベルのプロパティを変更することができる。

まとめ

この章では以下の点を学んだ。

- Windowsフォームアプリケーションのプロジェクト作成方法
- コントロールの配置
- コントロールのプロパティを変更
- イベントハンドラを設定
- イベントハンドラを実装