



入門編

～誰もそこまでは教えてくれなかった～

クラスの考え方と作り
方を「使い方から学
ぶ」テキスト C#



丸山プログラミング塾

内容

クラスを作りたい	4
クラスのイメージ.....	4
利用側から見るクラスイメージ.....	4
クラスイメージの解説.....	6
製作側（実装）から見るクラスのイメージ	8
ポイント.....	9
オブジェクト指向概念と実装手段.....	10
カプセル化	10
概念.....	10
隠蔽.....	10
実装.....	10
インヘリタンス（継承）	11
概念.....	11
実装.....	11
ポリモーフィズム	12
実装.....	12
オーバーライド	12
実装.....	12
静的化（STATIC）	13
静的化は慎重に	13
インターフェイス(C#)	14
クラスで継承するインターフェイス	14

実際のクラスで検証 (STRING クラス)	15
STRING クラスで検証 (利用側 : 公開編)	15
STRING クラスで検証 (開発側 : 実装編)	16
コンストラクタ	16
公開プロパティ、公開メソッド.....	17
.NET ソースコードのダウンロードとインストール	17
クラス内の実装設計	19
クラス内の実装はメンバーをグループ単位にまとめる (参考例)	19
命名規則 (名前付けルール)	20
1 クラス 1 ファイルの原則.....	21
全体を一望できる図表を作成.....	21
デモプログラム (テストプログラム) を用意.....	21
特殊なクラス	21
利用者任せは「クラス製作」とは言えない	21
クラスに関する C# 言語仕様補足事項	22
自分自身は[THIS]、親は[BASE]	22
親の公開メンバーを隠す NEW キーワード	22
ガベージコレクションとは	22
クラス構成図	23
ソリューション構成図	23
特定商取引法に基づく表示	24
最後に	25

ご注意

本書では、「.NET Framework Class Library」を「Class Library」と呼びます。

VB ユーザーの方へ

Class Library のほとんどは、C#で記述されています。Visual Basic アセンブリは VB で記述されていますが、一部は C#で記述されているほど C#が中心になっています。WPF や XNA のサンプルも C#が中心です。Microsoft の Library やサンプルコードで内部を深く学ぶためには C#は欠かせない言語となっています。これを機に C#言語も併用で学習することをおすすめします。

クラスを作りたい

この解説は、一通りC#言語の文法を学習し、プログラミングの経験がある方を前提にしています。

クラスを作りたい・・・と思っても、具体的にどう考え、どう組み立てていけばいいのでしょうか？

プログラムという観点から考えるクラスは、文法からでは見えてきません。オブジェクト指向という概念を勉強しても同じです。

具体的なイメージを描くことから始めてみましょう。

クラスのイメージ

概念や文法を考える前に、クラスを使う観点でクラスをイメージしてみましょう。

利用者側から見るクラスイメージ

こんなストーリーを考えてみました。眠れる森の美女がクラスです。とある王子は、眠れる森の美女に秘められた力が欲しいと考えていたのです。クラスの特徴を擬人化してイメージしていきましょう。



1
眠れる森の美女よ、
私のキスで目覚めておくれ

2
このドレスに着替えて

3
何かあったら、
連絡はコチラからするからね

4
もし、状況が変わったら
コチラへ連絡してくれ



君の力はおじいさんの力だね
君の家系図が知りたい

5

君は隠し事がうまいね
見せたくないモノと見せたいモノを
うまく使い分けているようだ

6

君のデータを配列で見せてくれ
君のデータを順番に見せてくれ

7

お休み、君に会いたくなったら、
また君にキスをするよ。

8

君は、じぶんの事をよく知っているね
僕は何も考えず君に安心して頼めるよ

オブジェクトは、対象とか概念も含めた存在
物質的なモノと考えると分からなくなる

それぞれ、クラスの特性を表していますが、何を指しているか分かりますか？

少し、考えて見て下さい。

メモ

オブジェクト指向のオブジェクトは「対象」と考えます。モノだけでなく、数式もデータ操作も概念もオブジェクトとなりうることを理解することが重要だと考えています。したがって、オブジェクトを「モノ」として解説する情報源には注意すべきです。

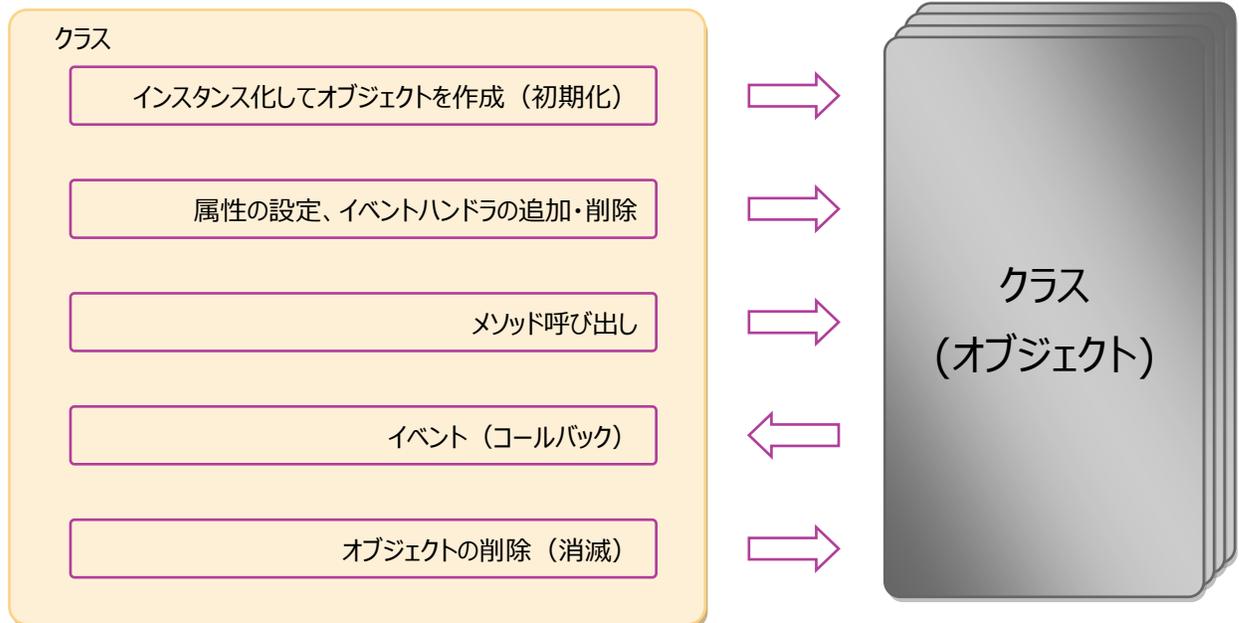
クラスイメージの解説

クラスの「イメージー用語ー解説」表

No.	イメージとクラス特性		解説
1	眠れる森の美女よ、 私のキスで目覚めておくれ		コンパイルされたクラスは、準備状態になっているだけでそのままでは実行することはできません。そのためには new でインスタンス化して実行できるようにします。これをオブジェクトと呼びます。※static 属性のメンバーはこの適用から除外され、(唯一のインスタンスとして) いつでも使用することができます。
	インスタンス化	オブジェクト	
2	このドレスに着替えて		初期値とは、コンストラクタに渡す引数です。引数があれば、値で Class 内を初期化し、引数がなければ、初期値で Class 内を初期化します。Class 内の変数の初期設定：外部からその状況は想定することはできないし、想定する必要もありません。 属性とは、プロパティのことです。プロパティによりクラスの性質を調整したりします。クラスの特性を取得することもできます。
	コンストラクタ 初期化	プロパティ 属性	
3	何かあったら、連絡はコチラからする からね		クラスがクラスとして実行するためには、メソッドやプロパティを呼び出してもらう必要があります。 呼び出されたメソッドは、内部のメソッドを呼び出したり、他のクラスを使用したりして処理を遂行します。 プロパティは、外部から見たら「変数」。通常の変数とは異なり、設定不可や参照不可、設定値のチェック、状況に応じた値を返すといった柔軟性を持っている。
	メソッド	プロパティ	
4	もし、状況が変わったらコチラへ連絡してくれ		事前に、コールバックするメソッドを登録しておく、ある事象が発生したとき呼び出してくれる機能。呼び出し不要になったときは解除することができる。
	イベント		
5	君の力はおじいさんの力だね。君の家系図が知りたい		クラスは必ず親（ベース）クラスを継承します。指定しなければ、object クラスが親クラスになります。継承すると、親クラスと親クラスが継承している全クラスを継承します。
	継承		
6	君は隠し事がうまいね。見せたくないモノと見せたいモノをうまく使い分けているようだ		クラス間インターフェイス クラスで重要な事は、データと機能や処理を一つにまとめることと、外部に対して何を見せるかです。それ以外は、すべて隠します。
	カプセル化	情報隠蔽	
7	君のデータを配列で見せてくれ 君のデータを順番に見せてくれ		Indexer とは、クラス内で持つ同一型のデータを配列のように見せる機能です。 Collection とは、クラス内で持つ同一型のデータを先頭から順に返す機能です。
	Indexer	Collection	

			能です。 <pre>String[] Datas = new String({"1","2","3"}); forerch (string in Datas) System.Console.WriteLine(string);</pre>
8	お休み、君に会いたくなったら、 また君にキスをするよ		
	デストラクタ	Dispose	

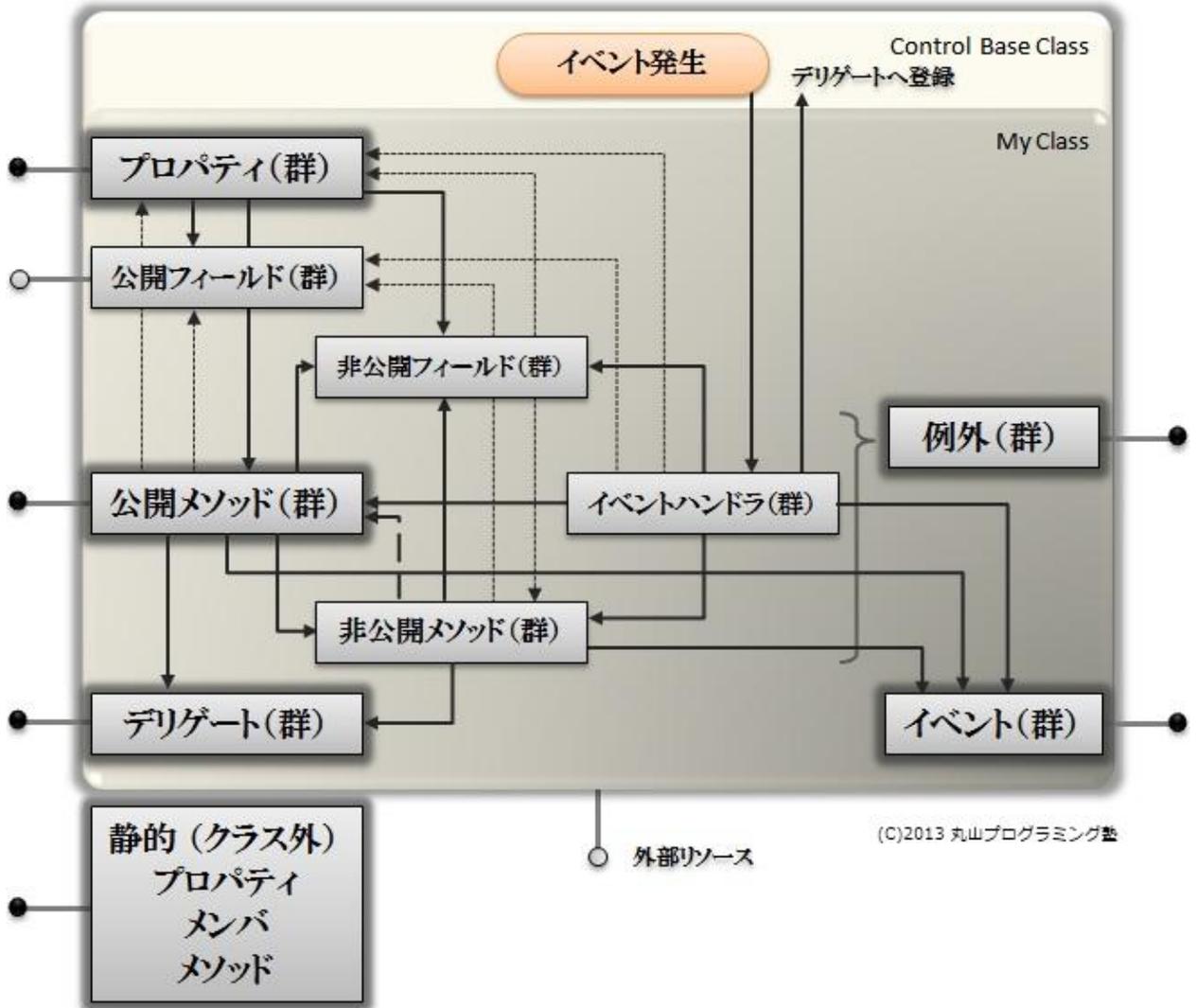
これ以外に、仲間のクラスがあります。クラス内にあるオープンな内部クラス、公開されているクラスを使用したデータの受け渡しです。



製作側（実装）から見るクラスのイメージ

クラス構造（メンバーと関連機能）の概念図

● 名前空間	● クラス	● メンバー
● フィールド	● メソッド（関数）	● コンストラクタ
● デストラクタ	● インターフェイス	● インデクサー
● デリゲート	● イベント	● 反復子



このクラス概念図は Control クラスを継承したクラスを想定しています。