

システム	VB基礎 (. Net) ～ クラスについて ～	2010年12月3日 大矢
------	------------------------------	------------------

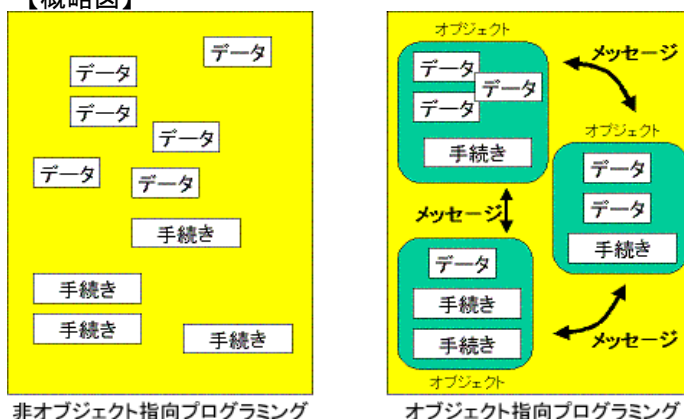
1. 従来のVB/VBAとの違い

- (1) 『. Net』対応となった事が最大の特徴。Winアプリ/Webアプリ/WPF 等いずれも共通のプラットフォーム (Framework) で動作する。
- (2) 使用するGUI部品が全て『Framework』より使用される様になった。
従来のGUIとは全く別物になりました。
- (3) 本格的なオブジェクト指向言語に変わった。(オブジェクト指向プログラミングへの対応)
- (4) 従来と違いオブジェクト指向対応の為、言語使用が高度になり難易度が上がってしまった。
- (5) 『Framework』が搭載されていれば、ランタイムモジュールが不要。
(プログラムの配布簡素化)

要約すれば、非常に高度な開発が可能になった反面、初心者が遊びで使うレベルでは無くなってしまった。

※オブジェクト指向プログラミングでは、データ/機能を一つのオブジェクトにまとめ、独立した部品としてオブジェクトを組合わせて目的の処理を実現する方式。

【概略図】



2. クラスについて (全てのオブジェクトのみなもと)

- (1) 全てのオブジェクトはクラス (設計図) から作成されます。
インスタンスとは、クラス (設計図) を実体化 (オブジェクト) したものを言います。

【車 (レガシー) で例えると】



※設計図があれば同じものを幾つでも作成することが出来る。

【車 1】

【設計図】

【車 2】

- (2) 『. Net』では**全てのものがクラスから生成される**。
- (3) プロパティ: クラスが持つデータを書き換えるインターフェイス。
具体的には、色や形といった様なオブジェクトの表面的な要素や保持するデータ。
- (4) メソッド: 演算や処理といった様な動作や振る舞い。

【車 (レガシー) で例えると】



プロパティ: 色。形状。大きさ。部品の位置。装備品。
メソッド: ライト/エンジンのON/OFF。
アクセル/ブレーキで走行/停止。

2-1. カプセル化

- (1) オブジェクト内部を隠蔽すること。利用者は内部的なアルゴリズムやデータを意識すること無く使用することが出来ます。

【参考例】利用者はアクセルを踏む (INPUT) と走る (OUTPUT) を例えると

①アクセルを踏む → ②スロットルバルブを開く → ③気化燃料供給大 → ④エンジン回転大 → ⑤走行

カプセル化 (隠蔽): 利用者は②～④までを意識せず使用できる

2-2. 継承

- (1) クラスの持つプロパティ/メソッドを受け継いだ別のクラスを作成すること。
元となるクラスを『基底クラス』または『スーパークラス』と呼びます。受継がれたクラスを『派生クラス』または『サブクラス』と呼びます

システム	VB基礎 (. Net) ～ クラスについて ～	2010年12月3日 大矢
------	------------------------------	------------------

2-3. ポリモーフィズム（多態性、多様性）

(1)継承を通じて、同じメッセージに対して異なるオブジェクトが異なる動作をすること。

実際の動作は、それぞれのオブジェクトによって異なるが、外部からは同じインターフェイスで扱えるようになる。

【ポリモーフィズムの例】

```
' %5.3 State tax
Const StateRate As Double = 0.053
' %2.8 City tax
Const CityRate As Double = 0.028
Public Class BaseTax
    Overridable Function CalculateTax(ByVal Amount As Double) As Double
        ' Calculate state tax.
        Return Amount * StateRate
    End Function
End Class

Public Class CityTax
    ' This method calls a method in the base class
    ' and modifies the returned value.
    Inherits BaseTax ← 『BaseTax』を継承
    Private BaseAmount As Double
    Overrides Function CalculateTax(ByVal Amount As Double) As Double
        ' Some cities apply a tax to the total cost of purchases,
        ' including other taxes.
        BaseAmount = MyBase.CalculateTax(Amount)
        Return CityRate * (BaseAmount + Amount) + BaseAmount
    End Function
End Class

Sub TestPoly()
    Dim Item1 As New BaseTax
    Dim Item2 As New CityTax
    ' $22.74 normal purchase.
    ShowTax(Item1, 22.74) → ①
    ' $22.74 city purchase.
    ShowTax(Item2, 22.74) → ②
End Sub

Sub ShowTax(ByVal Item As BaseTax, ByVal SaleAmount As Double)
    ' Item is declared as BaseTax, but you can
    ' pass an item of type CityTax instead.
    Dim TaxAmount As Double
    TaxAmount = Item.CalculateTax(SaleAmount) → ③
```

【説明】

①と②の呼び出し方は同じでも『Item1』と『Item2』は異なるクラスから派生したオブジェクトで有り、振る舞いもそれぞれのクラスに実装されたメソッドが実行される。

最後部『ShowTax』の『Item』はオブジェクトの型（『BaseTax』 or 『CityTax』）が判定されその型に適合したメソッドが呼び出される。

『ShowTax』の第一引数『Item』は『BaseTax』となっていますが、『CityTax』は『BaseTax』を継承して造られたクラスの為、インターフェイスが同じとして扱われます。

その為、エラーとなりません。

但し、『Item1』と『Item2』は派生元のクラスに実装されたメソッドを呼び出します。

カプセル化／継承／ポリモーフィズムはオブジェクト指向プログラミングの3大柱として確立された手法です。

オブジェクト指向プログラミングを理解する上で一番基礎となる重要な概念です。