クラス同士が呼び出す仕組み
インターフェイスとイベントを理解する

大澤 文孝
OSAWA, Fumitaka

Technology Tools

- ☑ Visual Basic .NET
- ❑ Visual C# .NET
- ❑ SQL Server 2000
- ❑ Oracle 9i
- ❑ Access 2002
- ❑ ASP.NET
- ❑ Internet Information Services
- ❑ Other:

Level

★ ★ ★ ★ ★

Samples

はじめに

インターフェイスやイベントは、.NET Frameworkのクラスライブラリで日常的に使われるものです。そのため、意識しなくても、使わざるを得ないから使っているという状況にあるとも言えます。

しかしインターフェイスやイベントを使いこなすと、オブジェクト間のメソッドの呼び出しを簡素化できるというメリットがあります。

そこで本稿では、インターフェイスやイベントの仕組みと、それを使う意義について説明します。

異なるオブジェクトを配列で管理する

インターフェイスを考察するにあたって、簡単なグラフィックツールを考えてみます。

グラフィックツールでは、「四角形」「円」「多角形」「ビットマップ」の4種類の図形を描画できるものとします。

そうした場合、四角形、円、多角形、ビットマップの位置や大きさ、色などをオブジェクトとして管理できるのが理想でしょう。

それぞれのオブジェクトは、種類によって管理するデータが違ってきます。

たとえば、四角形、円の場合には、「左上座標」「右下座標」「色」「線幅」が必要になります[注1]。それに対して、多角形の場合には、複数の点で構成されるため、「左上座標」「右下座標」ではなく、多角形の個々の点の座標が必要となるでしょう。

またビットマップの場合には、「左上座標」と描画する画像のファイル名だけが必要で、色と線幅の概念はないはずです。

これらのことを考えると、四角形、円、多角形、ビットマップは、図1のようにクラス定義できます（リスト1）[注2]。

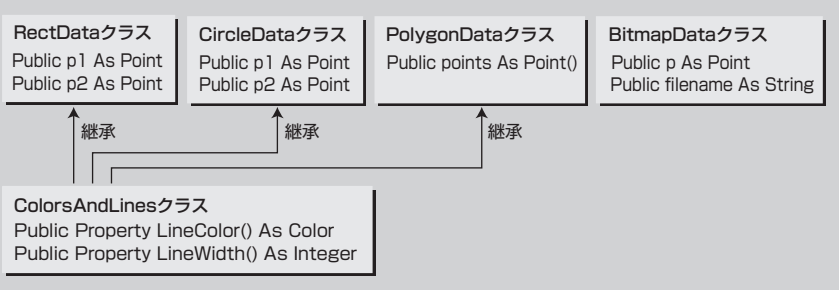
ここでは、四角形、円、多角形には、色と線幅が共通に存在するので、これらを構成するColorsAndLinesクラスを

注1）もちろん実際には、塗りの色や線の種類（波線や点線など）といった要素もありますが、ここでは話を簡単にするため、内部を塗らず、枠線だけを実線で描画するものとします。

作成し、そこから継承させて作成しています。

これに対して、ビットマップの場合には、色と線幅は存在しないので、ColorsAndLinesクラスから継承させていません[注3]。

図1：図形を管理するクラス群



注2）図1は、UMLの図ではありません。UMLの図で示すならば、矢印の矢の向きが逆になり、また矢が白抜きになります。UMLで示しても良いのですが、矢印の向きが感覚的なものと逆になり、UMLを知らない人にとってわかりにくいので、本稿ではUMLの図では記さないことにします。

注3）四角形も円も左上座標、右下座標を共通して利用するので、これらも何か適当なクラスを用意して、そこから継承させるという手もあります。しかしここでは、あまりたくさん継承させると話が複雑になるので、そこまではしていません。そもそもクラス設計は、「同じものがあればまとめれば良い」というものでもないので、まとめるメリットがあるかどうかはケースバイケースと言えます。図1およびリスト1は、あくまでも一例にすぎません。

リスト1：図1の実装

```
Imports System.Drawing.Graphics

Public Class ColorsAndLines
  ' 色と線幅
  Private _linecolor As Color
  Public Property LineColor() As Color
    Get
      Return _linecolor
    End Get
    Set(ByVal Value As Color)
      _linecolor = Value
    End Set
  End Property

  Private _linewidth As Integer
  Public Property LineWidth() As Integer
    Get
      Return _linewidth
    End Get
    Set(ByVal Value As Integer)
      _linewidth = Value
    End Set
  End Property
End Class

Public Class RectData
  Inherits ColorsAndLines
  ' 四角形のデータ
  Public p1 As Point   ' 左上の座標
  Public p2 As Point   ' 右下の座標

  Sub New(ByVal p1 As Point, ByVal p2 As Point, _
   ByVal linecolor As Color, ByVal linewidth As Integer)
    Me.p1 = p1
    Me.p2 = p2
    Me.LineColor = linecolor
    Me.LineWidth = linewidth
  End Sub
End Class

Public Class CircleData
  Inherits ColorsAndLines
  ' 円のデータ
  Public p1 As Point   ' 左上の座標
  Public p2 As Point   ' 右下の座標

  Sub New(ByVal p1 As Point, ByVal p2 As Point, _
   ByVal linecolor As Color, ByVal linewidth As Integer)
    Me.p1 = p1
    Me.p2 = p2
    Me.LineColor = linecolor
    Me.LineWidth = linewidth
  End Sub
End Class

Public Class PolygonData
  Inherits ColorsAndLines
  ' 多角形のデータ
  Public points As Point()

  Sub New(ByVal p As Point(), _
   ByVal linecolor As Color, ByVal linewidth As Integer)
    Me.points = p
    Me.LineColor = linecolor
    Me.LineWidth = linewidth
  End Sub
End Class

Public Class BitmapData
  ' ビットマップのデータ
  Public p As Point
  Public filename As String

  Sub New(ByVal p As Point, ByVal filename As String)
    Me.p = p
    Me.filename = filename
  End Sub
End Class
```