



Windows フォームで グラフィックソフトを作ろう

Windows フォームへの描画から印刷まで

大澤 文孝 OSAWA, Fumitaka

はじめに

.NET Frameworkでは、Windows フォームを使ってWindowsアプリケーションを作れます。

単純なボタンやテキストボックスを

構成するだけなら、Windowsアプリケーションの構築は簡単です。

しかしウィンドウに絵を描画したり、印刷したり、マウスでドラッグして図形を動かせるようにしたりしたいとなると、とたんに難しくなってきます。

そこで本稿では、簡単なお絵かきソ

フトを作りながら、「ウィンドウに描画するにはどうすればよいのか」「どのように印刷すればよいのか」「マウスでドラッグされた位置をどのように知ればよいのか」といったWindowsフォームにおける基本処理を説明していきます。

本稿で作成するお絵かきソフトは、

Level



Technology Tools

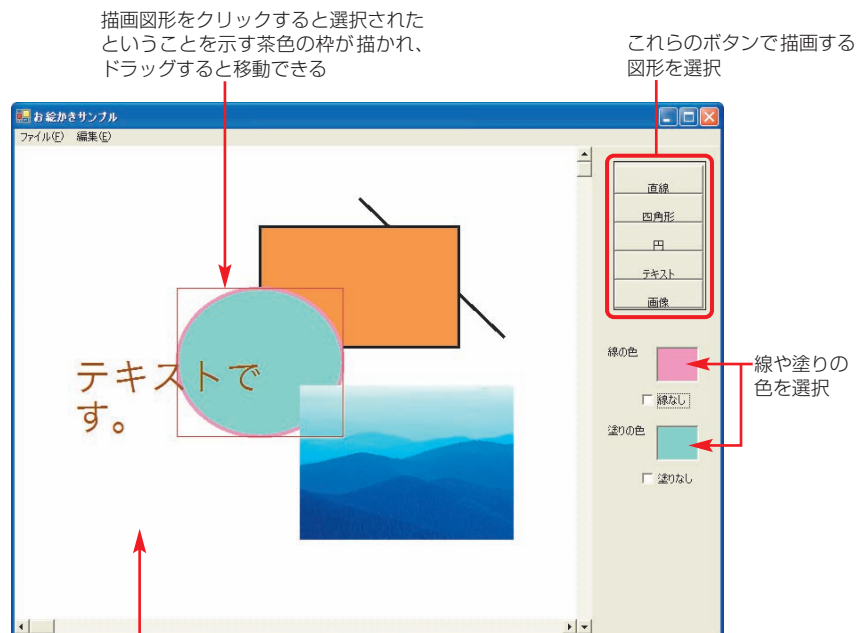
- Visual Basic
- Visual C#
- Visual C++
- SQL Server
- Oracle
- Access
- ASP.NET
- Other:

↓
Visual Studio .NET 2003

Samples

この記事で取り上げたソースコードおよびサンプルプログラムは、
<http://www.shoeisha.com/mag/windev/>
からダウンロード可能です。

図1：本稿で説明するお絵かきソフト



描画できる大きさはA4 サイズ(210mm×297mm)まで。縦横にスクロールできる。ここにはPanelコントロールが貼り付けられており、そのPanelコントロールにはDrawPanelという名前が付けられている



図1に示すように、「直線」「四角形」「円」「テキスト」「画像 (ビットマップ)」の5種類の図形をマウス操作で描画できるようにするというものです。描画した図形は、あとでドラッグして移動することもできます^[註1]。

また、作成した図形をファイルに保存したり、印刷したりする機能も備えています。

なおサンプルプログラムは、少々長く、また泥臭い処理もしているため、本文では、詳細を省き、よりわかりやすくしたリストのみを掲載しています。

全容を知りたい人は、サンプルプログラムに含まれている実際のソースを参照してください。

WYSIWYGを目指す座標系

本稿のアプリケーションでは、画面で描画した図形を、そのまま印刷できるようにします。古い言葉で言えば「WYSIWYG (=What You See Is What You Get)」です。

WYSIWYGを目指すには、単位が「ピクセル(ドット)」ではうまくいきません。なぜなら1ピクセル当たりの大きさは、画面やプリンタの機種によって異なるので、大きさが変化するからです。

そこで、画面上と印刷物との大きさを合わせるために、インチやミリメートルといった、いわゆる世間一般で使われている単位を採用します^[註2]。

今回は、単位にミリメートルを採用することにし、A4サイズ(210mm×297mm)まで描けるようにします^[註3]。

もちろん画面の解像度の問題から、1

ユーザーコントロールを使ってプログラムを分離する

本稿でサンプルとして扱うお絵かきソフトは、「描画領域」と「図形の種類や色を設定するコントロール」とを分離し、それぞれをひとつのユーザーコントロールとして構成しています(図A)。

ユーザーコントロールにしておくと、プログラムが分離されてわかりやすくなり、あとで別のアプリケーションでも再利用できるようになります。

そればかりか、それぞれのコントロールの位置の移動が容易になるというメリットもあります。

たとえば、図形の種類や色を設定するコントロールを別ウィンドウにしたい場合には、別ウィンドウにコントロールを貼り付けるという変更だけで済みます。

図A: ユーザーコントロールで分離する

Form1 フォーム (メインのフォーム)

GraphicsPanel ユーザーコントロール

ToolPanel ユーザーコントロール

画面にA4サイズを収めることはできません。そこで、内部ではA4サイズで図形を保存しておき、その一部をコントロールから覗き見るように構成します(図2)。

覗き見る位置は、スクロールバーで調整できるようにします。ここではスクロールバーの位置を示す座標を、Point F型のscrolloffsetという名前の変数で示すことにします。

```
Dim scrolloffset As PointF = _
    New PointF(0, 0)
```

スクロールバーがクリックされたときには、この変数を増やしたり減らし

注1) 本当は図形の四隅をドラッグすると拡大/縮小できたり、テキストの幅や高さを変更できたりするのですが、サンプルではそこまで実装していません。

注2) 本稿では処理速度については、あまり気にしないことにします。処理速度を求めるなら、座標系はピクセルのままとし、印刷するときだけに他の座標系に変換したほうがよいでしょう。

注3) どのような単位系をとるのは自由です。目的に応じて誤差が少なくなる単位を採用するのがよいといえます。たとえば文字の大きさを厳密に管理しなければならないアプリケーション(DTPソフトなど)では、ポイントやインチを基本単位にするのが望ましいかもしれません。なぜなら、文字サイズをポイントで指定する場合、ミリメートルに換算すると、そこで誤差が生じるためです。