クラス： wxPrintout, wxPrinter, wxPrintPreview, wxPrinterDC, wxPrintDialog, wxPrintData, wxPrintDialogData, wxPageSetupDialog, wxPageSetupDialogData

' このページを印刷 '、' ドキュメントの中にこのページが存在するか？ ' などといった、特別な要求に応じることができるメンバ関数を備えたクラスを提供するために、印刷フレームワークは、アプリケーションを使用する。この方法は、プレビューページを回転させる、印刷ダイアログボックスを呼ぶ、印刷デバイスコンテキストを作成する、などを監視するための義務を wxWidgets に与える。：アプリケーションは、デバイスコンテキストへの情報の描写のみに集中することができる。

ドキュメント／ビューフレームワークは、デフォルトでビューごとに wxPrintout オブジェクトを作成し、あらかじめ組み込まれた印刷／プレビュー機能を構築するために、wxView::OnDraw を呼ぶ。

ドキュメントの印刷能力は、アプリケーションの中で、wxPrintout クラスの派生により表現される。このクラスは、要求に応じてページを印刷したり、実際にドキュメントを印刷するために wxPrinter オブジェクトの Print 関数に渡されたり、プレビューの初期化のために wxPrintPreview オブジェクトに渡される。以下のコード（印刷サンプルより抜粋）は、一旦 wxPrintout 機能が定義されてしまえば、印刷、プレビュー、印刷設定ダイアログの初期化がいかに簡単かということを示している。注目すべきは、印刷とプレビューの両方で、MyPrintout が使用されるということである。全てのプレビューユーザインタフェース機能は、wxWidgets により管理されている。MyPrintout がどのように定義されているかは、印刷サンプルのコードを参照のこと。

```
case WXPRINT_PRINT:
{
  wxPrinter printer;
  MyPrintout printout("My printout");
  printer.Print(this, &printout, true);
  break;
}
case WXPRINT_PREVIEW:
{
  // プレビューと印刷のために、２つの printout オブジェクトを渡す。
  wxPrintPreview *preview = new wxPrintPreview(new MyPrintout, new MyPrintout);
  wxPreviewFrame *frame = new wxPreviewFrame(
                                preview, this, "Demo Print Preview",
                                wxPoint(100, 100), wxSize(600, 650));
  frame->Centre(wxBOTH);
  frame->Initialize();
  frame->Show(true);
  break;
}
```