

DelphiによるWindowsプログラミング

正代, 隆義
九州大学大学院システム情報科学研究科情報理学専攻

<https://doi.org/10.15017/6770309>

出版情報：情報処理教育広報. 20, pp.3-10, 1997. Educational Center For Information Processing, Kyushu University

バージョン：

権利関係：



Delphi による Windows プログラミング

正代 隆義

九州大学大学院システム情報科学研究科情報理学専攻

shoudai@i.kyushu-u.ac.jp

平成9年 9月 11日

1 はじめに

Delphi は Pascal を基本とした完全なオブジェクト指向言語 Object Pascal の Windows 上で動作する処理系です。言語 Delphi¹は、手続きやデータなどが完全にパッケージ化されており、コンポーネント (部品) を組み合わせるだけで、ソフトを開発することができます。Pascal というと、情報処理教育のための言語というイメージが先行してしまいがちですが、Delphi は、Pascal の文法を基礎にしたきれいなプログラムが書けることに加え、Windows 上の先進的な機能を使用できる全く新しい言語です。Pascal を基礎に置くことでアマチュアからプロまで広い範囲のユーザを対象にしています。実際、Delphi は Delphi 自身で書かれているそうです。

Delphi 1.0J が登場したのは、1995 年 10 月です。その翌年、九州大学情報処理教育センターでは、Windows95 をメインに新しい計算機の導入が行われました。また、それと同時に Windows95 上で使えるワープロやプログラミングツールなどが公開されました。その中の 1 つが Delphi 1.0J²です。Borland 社は Delphi を以前にヒットした Turbo Pascal の後継に位置付けています。Windows プログラムをするのであれば、Borland C++ や Visual C++ などの C++ 言語や Visual BASIC など候補にあげられるでしょう。しかし、C++ はちょっと敷居が高い、Visual BASIC は実行速度が遅い、ランタイムライブラリが必要であるなどの不満が多く聞かれます。Delphi はそれらの点を十分考慮して設計された言語として、今日注目が集まっています。また、多くの解説本が出版されています [1, 4, 5]。

Delphi のもととなる Pascal についての解説は、1970 年に Wirth が設計して以来、様々な分野の人達が数多く書いています。我々も六本松地区の情報処理演習のためのホームページを公開しています [2]。今年 4 月、我々は、六本松地区での情報処理教育に携わって来たノウハウを、テキスト「12 回で学ぶ情報処理」[3]として出版しました。このテキストは、Pascal の文法の基礎から Delphi の統合環境を使った初歩的な Windows プログラミングまで、日頃使用している OHP 資料を使って、ちょうど 12 回で学べるよう解説しています。

本稿では、Delphi を使って Windows プログラミングをしようとする方のために、Delphi が Delphi らしい Windows プログラミングの部分、オブジェクト指向言語に不慣れな方のために拙著 [3] を参考に簡単に紹介します。

¹Delphi は製品化されている Object Pascal のほとんど唯一の処理系であるため、Delphi で言語自身を参照することが多いようです。

²現在、Delphi 3 が販売されています。コンパイラの実行速度は 3~4 倍になり、内部では並列処理までしているという噂です。

2 Delphi

2.1 オブジェクト

Delphiはオブジェクト指向型言語です。オブジェクト指向型言語の詳しい説明は他の文献に委ねることにしますが、とりあえず次のように思うことにします。

Windows95 を使っていて現れるボタン、メニュー、ウィンドウ、文字、チェックボックス、その他あらゆるものは全てオブジェクトである。

Windows95 には大きなボタン、かわいいボタン、四角いボタンなど様々なボタンが現れます。我々は、これらのボタンをひっくるめて、その概念を「ボタン」と呼んでいます。この「ボタン」がオブジェクトです。ひとつひとつのボタンのことは「ボタン」のインスタンスと呼びます。オブジェクトのそれぞれのインスタンスは色、形、大きさなど様々な性質を持っています。これをプロパティといいます。プロパティ自身もプロパティを持っていることがあります。

オブジェクト、インスタンス、プロパティ、これだけを知っていれば Delphi はなんとかなりそうです。

2.2 Delphi の起動

Delphi の起動は、Windows95 の [スタート]³ メニューの中のサブメニュー [プログラム] から行います。[Delphi] や [Netscape] などの項目が現れますので、その中の [Delphi] メニューから [Delphi] を選びます。Delphi が起動し終わると、図 1 のような画面になります。これが、Delphi の統合環境です。Delphi を初めて体験した人は、この画面を見てまず圧倒されるようです⁴。

Delphi による Windows プログラミングは次の繰り返しによって行われます。

あらかじめ用意されたフィールドに、ボタンなどの部品を配置し、その部品に行うであろうマウスクリックなどの操作に対する受け答えを Pascal の文法で記述する。

このプログラミングの方式をイベントドリブン方式によるプログラミングということがあります。Delphi の起動画面は、以下の 6 つの部分から成ります。

メインメニュー Windows95 の普通のアプリケーションに見られる一般的なメニューです。[ファイル], [編集] などの他、コンパイラを起動しプログラムを実行するための [実行] メニューがあります。プログラムをコンパイル、実行するときは、この [実行] メニューから [実行] を選びます。また、[ヘルプ] メニューの、[キーワードで検索] は関数または手続きの機能を知りたいときに便利です (図 2)。

スピードバー メインメニューのよく使う機能をアイコンで表示したものです (図 3)。

コンポーネントパレット Delphi のコンポーネント (ボタン (Button), チェックボックス (CheckBox) などの部品) を並べたライブラリです。コンポーネントは全てオブジェクトです。使いたいコンポーネントの詳しい説明が欲しいときは、そのコンポーネントをマウスでクリックした後、**F1** キーを押して下さい (図 4)。

³マウスでクリックできるメニューやボタンのキーワードを [] で囲むことにします。

⁴Delphi は迷宮という意味です。初めて Delphi を使う人にこれを話すと必ず納得してもらえます。

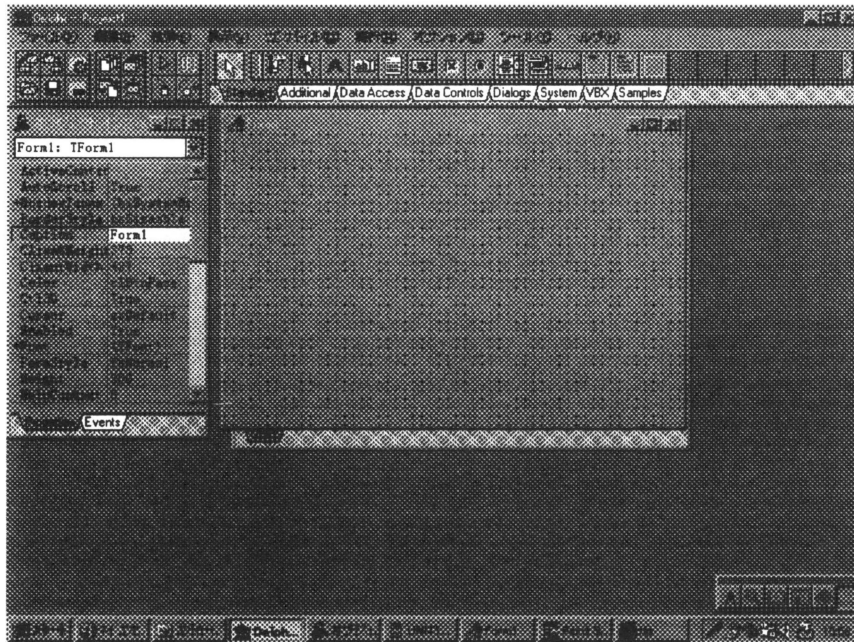


図 1: Delphi の起動画面



図 2: メインメニュー



図 3: スピードバー



図 4: コンポーネントパレット

フォーム Delphi の起動画面の中央に現れるウィンドウです。実は、フォームは Form というオブジェクトのインスタンスです。名前を Form1 といいます。プログラムが出来上がったときには、このウィンドウが実行画面になります (図 5)。

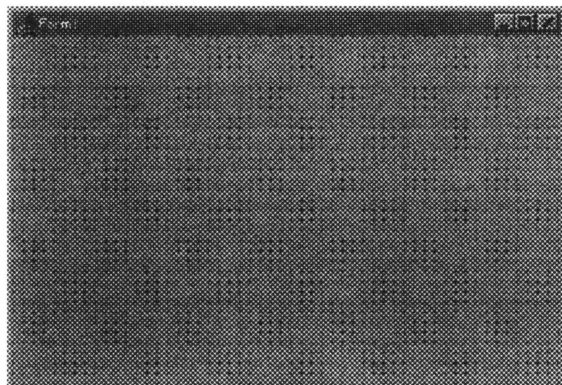


図 5: フォーム

オブジェクトインスペクタ オブジェクトのインスタンスのプロパティの一部とそのインスタンスに対する操作をそれぞれ、Properties, Events としてリストアップした表です (図 6)。

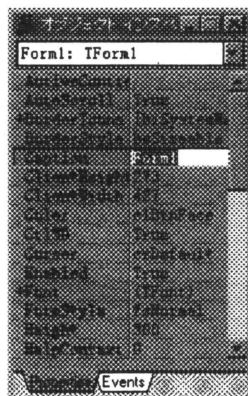


図 6: オブジェクトインスペクタ

コードエディタ オブジェクトのインスタンスに対する操作の受け答えを手続き (procedure) として記述するためのエディタです。作成された手続きを一般にイベントハンドラといいます。一般的によく使われるエディタと比べて遜色のない Delphi プログラミング専用のエディタです。起動画面では、フォームの後ろに隠れていて見えません。最近の高機能エディタではよく見受けられる決まった言葉を自動的に挿入してくれる機能⁵があります。この機能のおかげで、Delphi の統合環境ではプログラムを白紙の状態から書くことは皆無です (図 7)。

⁵ 高機能エディタ mule 上で動作する AUCTeX や YaTeX などでおなじみの機能です。

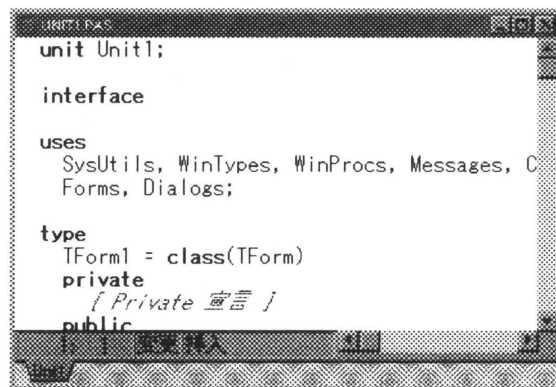


図 7: コードエディタ

3 簡単な Delphi プログラム

3.1 プログラムの例

Form のインスタンス Form1 上でマウスをクリックしたら、クリックした箇所に円を書くプログラムを書いてみましょう。

1. オブジェクトインスペクタのタブ Events をクリックします。さらに、リスト中の OnMouseDown の右欄をマウスの左ボタンでダブルクリックします。
2. コードエディタが Form1 の上に現れ、以下の手続き文が新たに自動的に挿入されます。マウスをクリックしたとき (OnMouseDown) のイベントハンドラです。

```

procedure TForm1.FormMouseDown(Sender: TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: Integer);
begin

end;

```

3. 上の手続き文の **begin** と **end** の間に次の文を挿入します。TForm1.FormMouseDown と Form1.Canvas.Ellipse の意味は、第 4 章で解説します。

```

Form1.Canvas.Ellipse(x, y, x + 20, y + 20)

```

4. メインメニューの [実行] から [実行] を選ぶと、ドットがないフォームが現れます。フォーム上を適当にマウスでクリック (右でも左でもかまいません) してください。クリックごとに白い円が描かれるはずです。
5. 実行したプログラムは必ず終了させます。通常の Windows95 のアプリケーションと同様に右上の 3 つのボタンのうち右端の [X] を押して下さい。

タイプミスなどのエラーが見つかったときは、自動的にエラー箇所へカーソルが移動し、エラーの理由を教えてください⁶。

従来の MS-DOS でのプログラミングでは、これだけのことを行うためにかなりのプログラミングが必要でした。Delphi では、これくらいであれば実質 1 行だけでプログラムが済むため、プログラマは頭脳活動の大部分をソフトウェアのデザインに費すことができることになります。

3.2 プログラムの構造

前章で書いたプログラムは 1 行ですが、Delphi はかなりの部分を前もって自動的に書いてくれます。また、フォームの形などの情報もデフォルトの状態でかなりの部分が用意されています。この章では、Delphi で作成するプログラムの基本構造を述べます。

Delphi が自動的に作成したプログラムはデフォルトで UNIT1.PAS (図 8) という名前がつけられています (PAS は Pascal の意味です)。このプログラムをユニットと呼びます。ユニットは interface 部と implementation 部⁷の 2 つから成ります。

```
unit1 Unit1;

interface
    SysUtils, WinTypes, WinProcs, Messages, Classes,
    Graphics, Controls, Forms, Dialogs;

type
    TForm1 = class(TForm)
    private
        {Private 宣言}
    public
        {Public 宣言}
    end;

var
    Form1: TForm1;

implementation

    {$R *.DFM}

end.
```

図 8: デフォルトのユニット

また、メインメニューの [表示] から [プロジェクトソース] を選ぶと図 9 のような PROJECT1.DPR と名付けられたプログラムが現れます (DPR は Delphi Project の意味です)。このプログラムをプロジェクトと呼びます。実は、これが Delphi のメインプログラムです。Pascal 言語を御存知の方にはなじみ深いキーワードがならんでいます。Delphi による Windows プログラミングでは、こちらのプログラムを手作業で修正することはまずありません。

⁶この辺りは Borland 社 Turbo Pascal の伝統をしっかり受け継いでいます。

⁷interface 部で var において宣言した変数、手続き、関数は、プロジェクト全体を通じて有効です。一方、implementation 部で宣言した変数は、宣言したユニット内だけで有効になります。

```

program Project1;

uses
  Forms,
  Unit1 in 'UNIT1.PAS' {Form1},

  {$R *.RES}

begin
  Application.CreateForm(TForm1, Form1);
  Application.Run;
end

```

図 9: デフォルトのプロジェクト

Delphi のプログラムは必ず予約語 unit, program, library のいずれかで始まり、それに識別子が続き、セミコロンで終わります。ファイル名は識別子と一致してはなりません。上記のプログラムでは、4 行目で UNIT1.PAS というファイル中のモジュール Unit1 を使用すると宣言しています。

3.3 プログラムの保存

ユニットとプロジェクトの両方を保存します。最初はユニットです。ユニットを保存するには、メインメニューの [ファイル] から [ユニットに名前を付けて保存] を選びます。ファイル名をたとえば Myunit.pas とします。確認したら [OK] ボタンあるいは **Enter** キーを押して下さい。

ユニットが保存できたら、プロジェクトを保存する前に、一度プロジェクトを覗いてみましょう。4 行目が自動的に変更されていることに注意して下さい (図 10)。

```

program Project1;

uses
  Forms,
  Myunit in 'MYUNIT.PAS' {Form1},

  {$R *.RES}

begin
  Application.CreateForm(TForm1, Form1);
  Application.Run;
end

```

図 10: ユニット保存後のプロジェクト

プロジェクトを保存するには、メインメニューの [ファイル] から [プロジェクトに名前を付けて保存] を選びます。このときファイル名には、ユニットと同じ名前 (ここでは、MYUNIT) を使ってはいけません。たとえば、プロジェクトのファイル名を Test.dpr とします。逆に、プロジェクトをユニットより先に保存しようとした場合、プロジェクトがユニットを参照するという原則か

ら、ユニットの保存を促されます。

この2つのプログラム以外に、TEST.opt(Option File の意味です)、Test.res(Resource の意味です)などが自動的に作成されます。また、作成したフォームの形などの情報は Myunit.dfm(Delphi Form の意味です)に保存されます。これらは全て必要なファイルですので削除しないでください。

4 オブジェクトインスペクタ – Properties と Events

オブジェクト Form のインスタンス Form1⁸のどこかを左クリックすると Form1 のためのオブジェクトインスペクタが表示されます。表示されたリストには Properties と Events があります。オブジェクトインスペクタの Properties リストに表示されているプロパティは、ユーザが直接変更可能なオブジェクトの大きさ、形、色などです。ユーザが直接変更できないプロパティの変更はメソッドと呼ばれる手続きを介して行います。この手続きをリストアップした表がオブジェクトインスペクタの Events リストです。

この章では Form1 のプロパティの変更方法を2つ述べます。

デフォルトのフォームは 273 × 427 の広さ、clBtnFace という色を持っています。これらを変更するには2通りの方法があります。

1. オブジェクトインスペクタによる変更

オブジェクトインスペクタの Properties リストにある ClientHeight の右横の数字をクリックし、400 にして Enter キーを押して下さい。Enter キーを押すと同時にフォームの縦の長さが 400 になります。同じように ClientWidth の数字を 400 に変更してみましょう。色の変更は Color で行います。clWhite にしてみましょう。

2. プログラムによる変更

第3章でふれたオブジェクトインスペクタの Events リストの OnMouseDown の右の空白をクリック後、現れたイベントハンドラの雛型を修正して、次のようにして下さい。

```
procedure TForm1.FormMouseDown(Sender: TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: Integer);
begin
  Form1.ClientHeight := 400;
  Form1.ClientWidth := 400;
  Form1.Color := clWhite
end;
```

実行後、フォーム上でマウスをクリックすると、いままでのフォームが 400 × 400 の白いフォームに早変わりします。

インスタンスの性質を変更するときは、一般に、

インスタンスの名前. プロパティ名. プロパティ名... プロパティ名 := 値;

とします。この例でも、Form1 の縦の長さを変更するために Form1.ClientHeight := 400; のようにしています。

また、イベントハンドラは、一般に次のような形をしています。

⁸いつも目の前にあります。