

Création d'applications multifenêtres sous DELPHI

Formulaire d'affichage
Boîte de dialogue

Ricco Rakotomalala
Université Lumière Lyon 2

Pourquoi plusieurs fenêtres ?

Meilleure organisation de l'interface, échange d'informations entre formulaires : (1) Certaines fenêtres servent à l'affichage de résultats : formulaires d'affichage, **fenêtre non modale** ; (2) d'autres à interagir avec l'utilisateur : boîtes de dialogues, **fenêtre modale**.



Dans tous les cas il s'agit de formulaires (2 fichiers créés pour chaque formulaire : *.dfm et *.pas)



On s'intéresse aux fenêtres créées par l'utilisateur dans ce support. Il existe par ailleurs des fenêtres standards Windows (ex. OpenFileDialog, SaveDialog, ...)

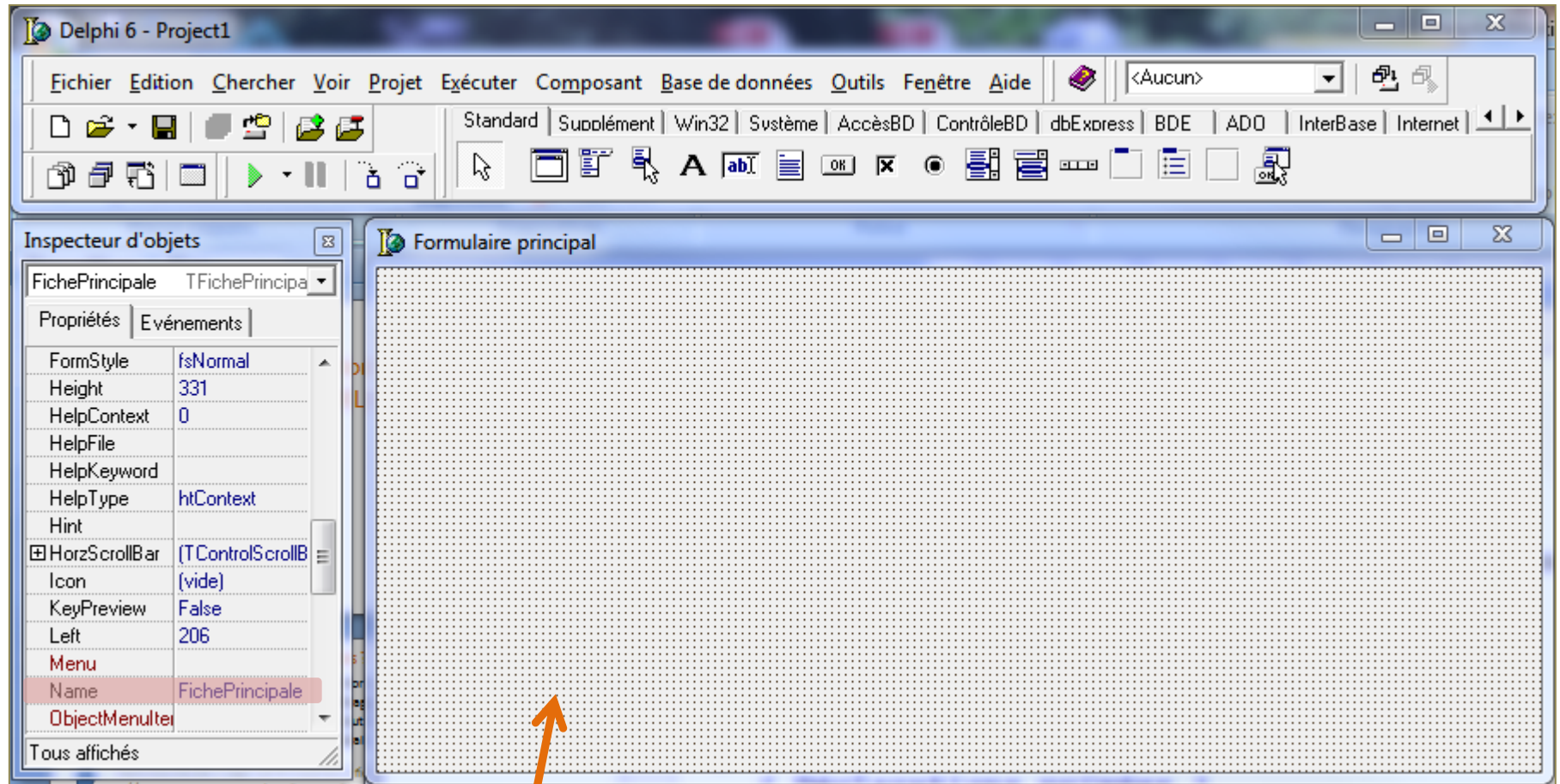


On s'intéresse à l'organisation SDI (Single Document Interface) dans ce support. Un formulaire joue le rôle de fenêtre principale, les autres gravitent autour.

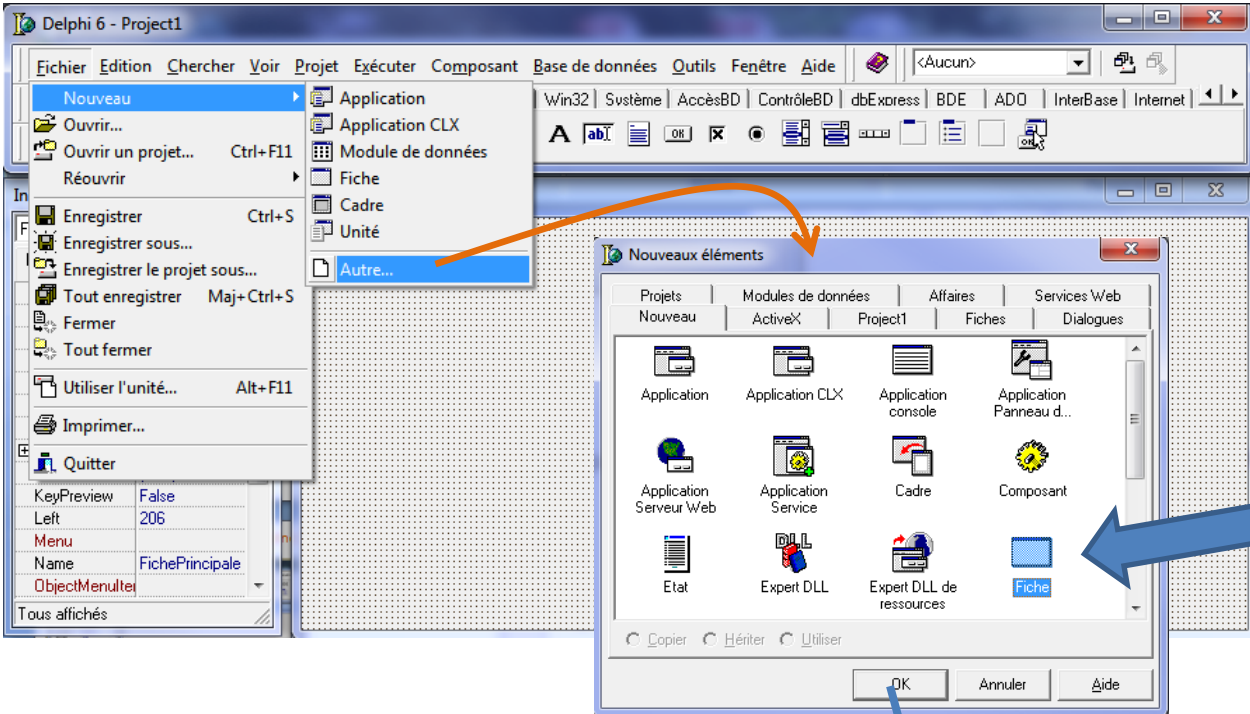
Affichage d'informations

FORMULAIRE D'AFFICHAGE

Point de départ, formulaire principal



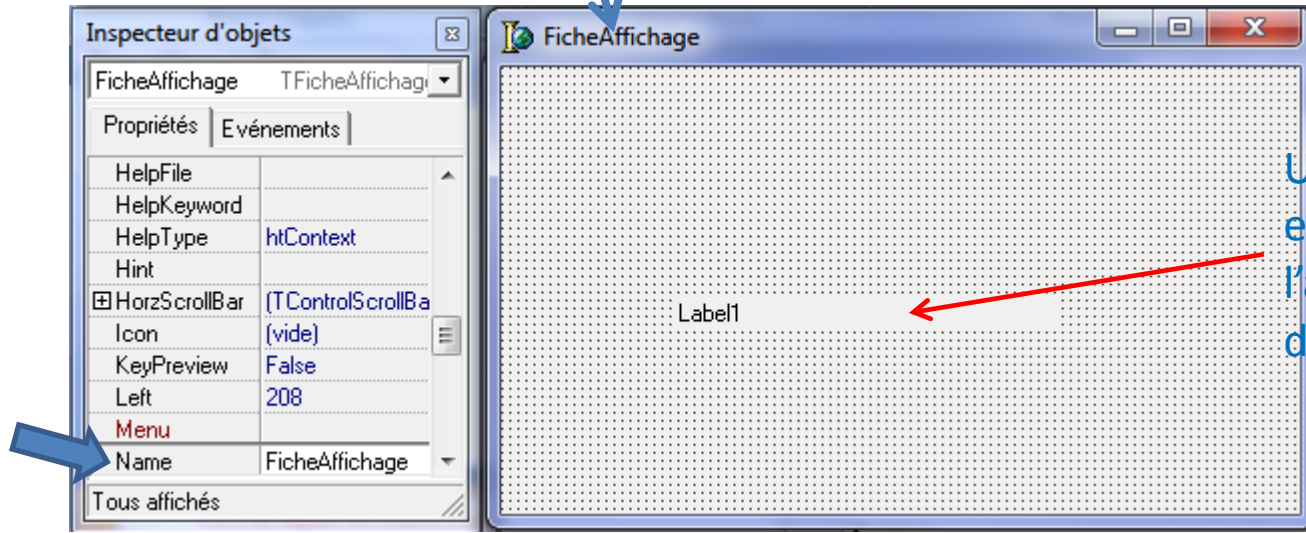
Le premier formulaire créé fait office de fenêtre principale (avec les menus)
On peut modifier la fenêtre principale d'une application (cf. PROJET / OPTIONS / onglet FICHES)



Ajouter un nouveau formulaire

Formulaire = Fiche dans la terminologie DELPHI

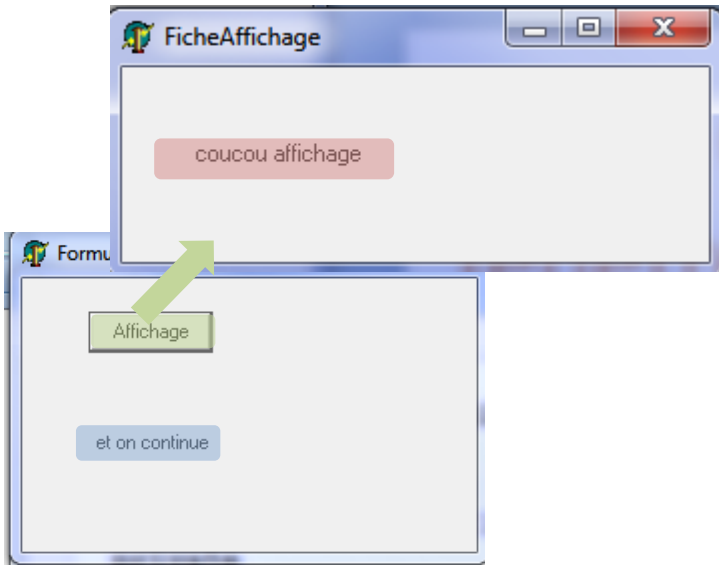
Bien noter la propriété « Name » qui définit le nom de variable du formulaire



Un « label » est mis pour l'affichage d'informations

Toujours tout enregistrer dès la création d'un formulaire, Delphi demande un **nom d'unité *.pas** (ex. unit2.pas)(le *.dfm est créé automatiquement).

Appel du formulaire d'affichage à partir de la fenêtre principale



Appel du formulaire d'affichage à partir du formulaire principal.

Programmation du « clic » sur le bouton « Affichage »

```
Unit1.pas
Unit1 | Unit2 |
type
  TFichePrincipale = class(TForm)
    Button1: TButton;
    labelFrmPrincipal: TLabel;
    procedure Button1Click(Sender: TObject);
  private
    { Déclarations privées }
  public
    { Déclarations publiques }
  end;

var
  FichePrincipale: TFichePrincipale;

implementation
  uses Unit2;
  {$R *.dfm}

  procedure TFichePrincipale.Button1Click(Sender: TObject);
  begin
    //màj du label du formulaire d'affichage
    FicheAffichage.Label1.Caption:= 'coucou affichage';
    //faire apparaître le formulaire
    FicheAffichage.Show();
    //suite normale
    labelFrmPrincipal.Caption:= 'et on continue';
  end;
```

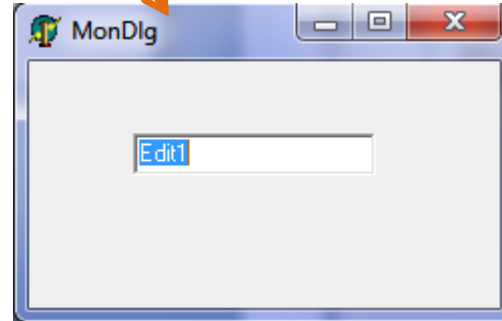
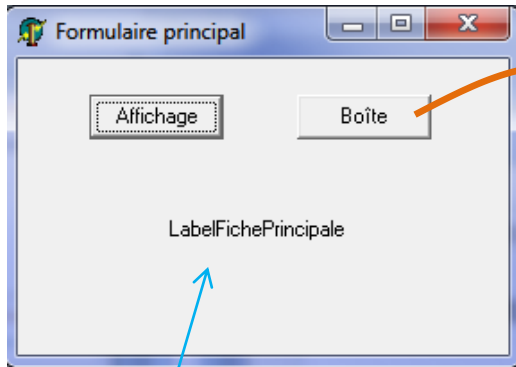
Très important, sinon le formulaire n'est pas accessible

Dès le formulaire affiché, les instructions suivantes sont exécutées ! Ici, le label du formulaire principal est modifié. Show() est non bloquant. On a une **fenêtre non modale**.

Récupération d'informations saisies par l'utilisateur

BOÎTE DE DIALOGUE

Une boîte de dialogue est aussi un formulaire (fiche). C'est l'appel qui modifie son comportement.



On ne peut pas accéder (cliquer sur) au formulaire principal tant que la boîte de dialogue n'est pas fermée.

```
Unit1.pas
Unit1 | Unit3 | Unit2

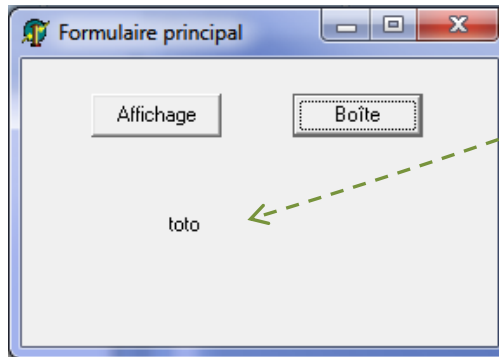
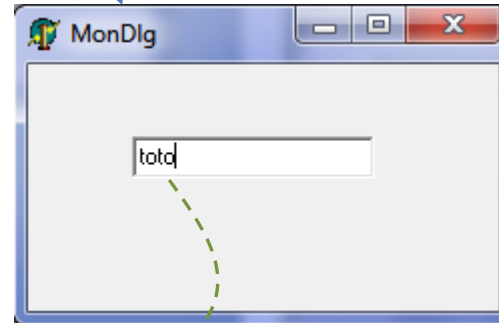
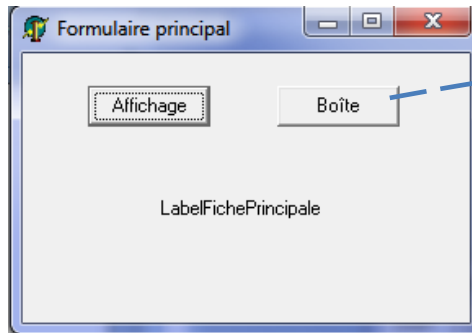
uses Unit2, Unit3;

{$R *.dfm}

procedure TFichePrincipale.Button2Click(Sender: TObject);
begin
  //affichage de la boîte avec ShowModal() !!!
  MonDlg.ShowModal();
  //suite normale
  labelFrmPrincipal.Caption:= 'et on continue';
end;
```

Ce code n'est pas exécuté tant que la boîte de dialogue n'est pas fermée

Récupération d'informations en provenance d'une boîte de dialogue.



Le contenu de la zone d'édition peut être récupéré, une fois seulement la boîte fermée.

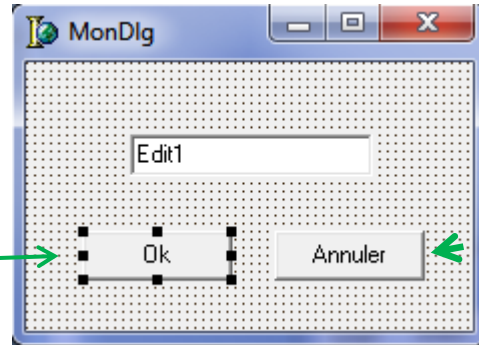
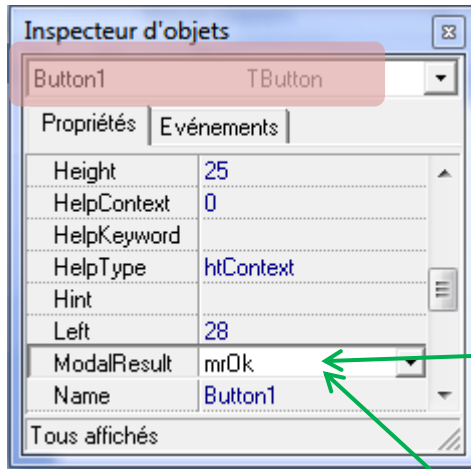
```
Unit1.pas
Unit1 | Unit3 | Unit2 |
uses Unit2, Unit3;

{$R *.dfm}

procedure TFichePrincipale.Button2Click(Sender: TObject);
begin
  //affichage de la boîte avec ShowModal() !!!
  MonDlg.ShowModal();
  //récupération du contenu de la zone d'édition
  labelFrmPrincipal.Caption:= MonDlg.Edit1.Text;
end;
```

La propriété **ModalResult** des boutons définit le code de retour renvoyé par **ShowDialog()**

Boîte de dialogue : gestion des boutons de réponse, la propriété « **ModalResult** »



ModalResult = mrCancel

```
procedure TFichePrincipale.Button2Click(Sender: TObject);  
begin  
  //affichage de la boîte avec ShowModal() !!!  
  if (MonDlg.ShowModal() = mrOk) Then  
  begin  
    //récupération du contenu de la zone d'édition  
    //uniquement si l'utilisateur a cliqué sur OK  
    labelFrmPrincipal.Caption:= MonDlg.Edit1.Text;  
  end;  
end;
```

Appel et gestion de la réponse de la boîte de dialogue dans le formulaire principal

Une vision globale des éléments constitutifs d'un projet DELPHI

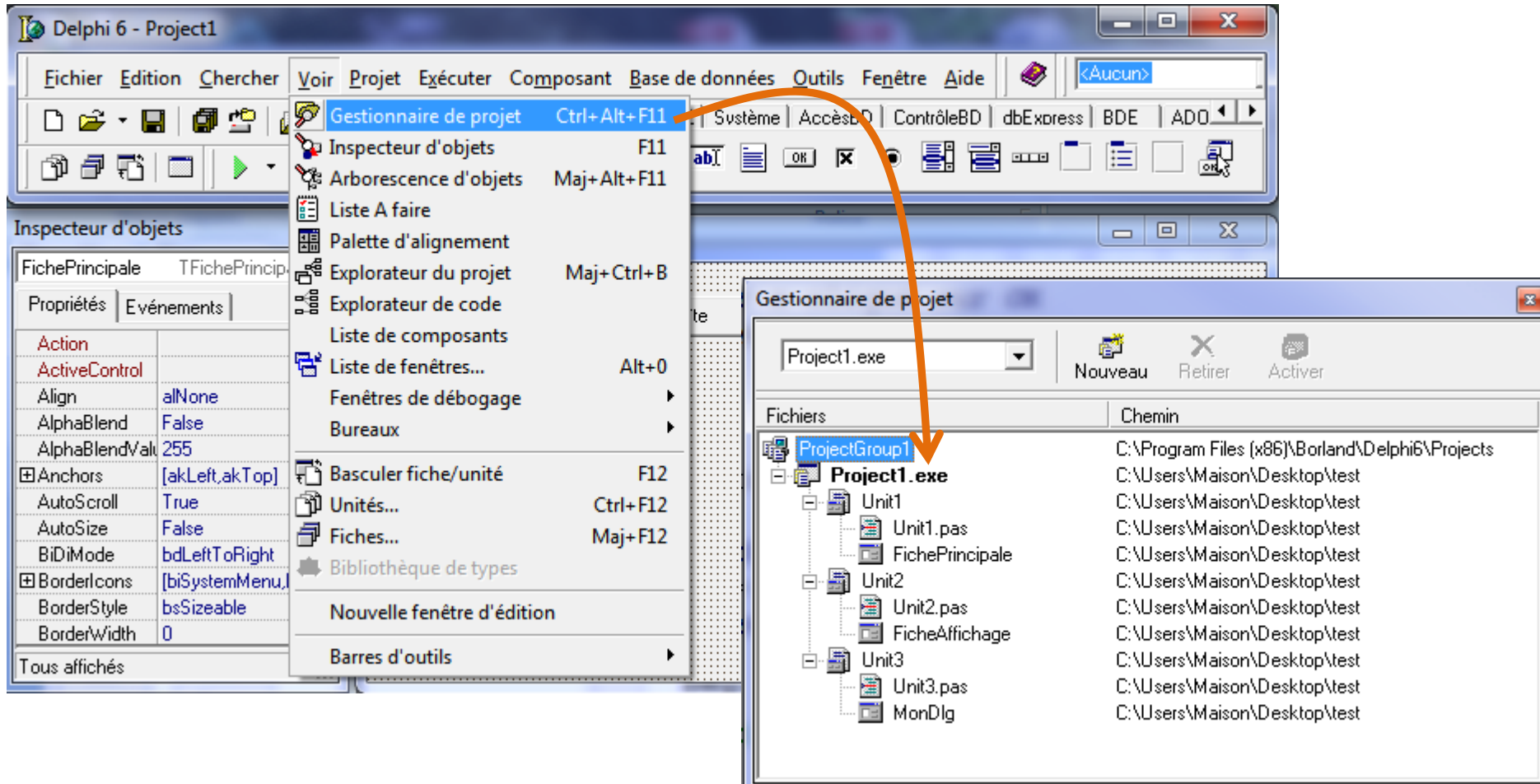
GESTIONNAIRE DE PROJET DELPHI

Vision globale d'un projet

Gestionnaire de projet

Avec le gestionnaire de projet, on obtient une vision globale des éléments constitutifs d'un projet :

- 1 Programme principal (*.dpr) → compilé en *.exe
- Formulaires et boîtes de dialogue (*.dfm et *.pas)
- Unités (calcul, structures, etc.) (*.pas)



FIN...

Les mêmes concepts sont – à peu de choses près – présents dans tous les langages de programmation...