

L'environnement de programmation Delphi

Cours Delphi LMD3, Gestion (GRH, GE)

Lakehal k.

Module: informatique 4

Lakehal

1. L'interface de Delphi

The image shows the Delphi 7 IDE interface with several callouts pointing to specific components:

- Barre d'outils**: Points to the toolbar at the top of the IDE.
- Barre de menus**: Points to the menu bar at the top of the IDE.
- Palette de composants**: Points to the component palette at the top right of the IDE.
- Arborescence des objets de la fiche**: Points to the Object Inspector on the left side of the IDE.
- Inspecteur d'objets**: Points to the Object Inspector on the left side of the IDE.
- fiche**: Points to the central design canvas (Form1) where the form is being designed.
- Editeur de texte**: Points to the code editor window (Unit1.pas) on the right side of the IDE.

The code editor window (Unit1.pas) contains the following code:

```
unit Unit1;  
  
interface  
  
uses  
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, I  
  Dialogs;  
  
type  
  TForm1 = class(TForm)  
  private  
    { Private declarations }  
  public  
    { Public declarations }  
  end;  
  
var  
  Form1: TForm1;  
  
implementation  
  
{$R *.dfm}  
  
end.
```

1. L'interface de Delphi

- a) Conception de fiches : la palette des composants
- Une fiche constitue l'interface (ou une partie de l'interface) d'une application.
- Pour concevoir une fiche, il suffit d'y insérer des composants prêts à l'emploi : **boutons, listes, menus...** listés dans la palette des composants.
- Un clic sur le contrôle, puis un autre sur la fiche cible suffisent (un double clic insère le composant au milieu de la fiche active).

1. L'interface de Delphi

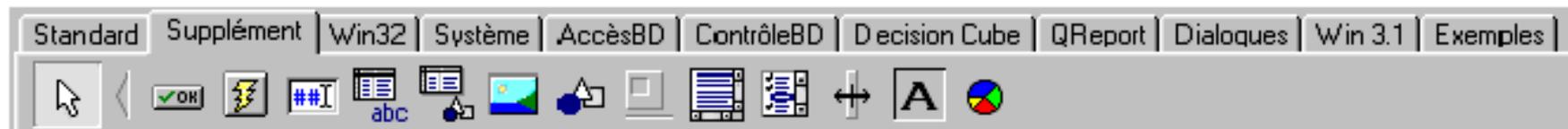
- Composants standards



Contrôle	Nom Pascal	Description
Menu principal	MainMenu	Barre de menus
Menu surgissant	PopupMenu	Menu contextuel
Étiquette	Label	Zone d'affichage de texte non modifiable
Édition	Edit	Boîte d'édition permettant la saisie ou la modification d'une ligne de texte
Mémo	Memo	Boîte d'édition permettant la saisie ou la modification d'un texte de plusieurs lignes
Bouton	Button	Bouton de commande simple
Case à cocher	CheckBox	Sélection de choix entre plusieurs
Bouton radio	RadioButton	Sélection d'un choix entre plusieurs
Boîte de liste simple	ListBox	Liste [déroulante] d'éléments
Boîte de liste combinée	ComboBox	Liste déroulante d'éléments avec possibilité d'édition sans une boîte d'édition
Barre de défilement	ScrollBar	Ascenseur vertical ou horizontal
Boîte de groupe	GroupBox	Cadre conteneur de contrôles
Groupe de boutons radio	RadioGroup	Cadre conteneur de boutons radio
Volet	Panel	Volet conteneur de contrôles

1. L'interface de Delphi

- Composants supplémentaires



Contrôle	Nom Pascal	Description
Bouton bitmap	BitBtn	Bouton de commande avec image bitmap
Turbo bouton	SpeedButton	Icône dans une barre d'outils
Masque de saisie	MaskEdit	Boîte d'édition permettant des saisies formatées
Grille de chaînes	StringGrid	Tableau d'affichage de chaînes
Grille d'affichage	DrawGrid	Tableau d'affichage de données
Image	Image	Zone d'affichage d'une image bitmap, d'une icône ou d'un métafichier Windows
Forme	Shape	Forme géométrique (ellipse ou rectangle)
Biseau	Bevel	Ligne ou rectangle 3D
Boîte de défilement	ScrollBar	Conteneur de composant possédant des barres de défilement
Boîte de liste à cocher	CheckListBox	Liste [déroulante] d'éléments à cocher
Séparateur	Splitter	Séparateur mobile
Texte statique	StaticText	Étiquette contenant un descripteur de fenêtre
Graphique	Chart	Graphique type Excel

1. L'interface de Delphi

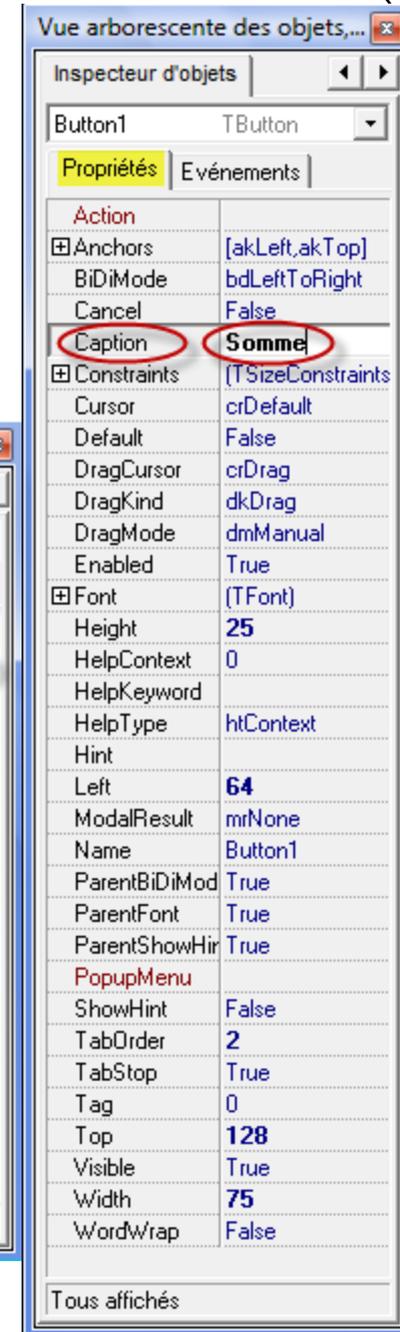
- Composants boîtes de dialogue



Boîte de dialogue	Nom Pascal	Description
Ouvrir	OpenDialog	Ouverture d'un fichier
Enregistrer sous	SaveDialog	Sauvegarde d'un fichier
Ouvrir image	OpenPicture	Ouverture d'un fichier image
Enregistrer image	SavePicture	Sauvegarde d'un fichier image
Fontes	FontDialog	Sélection d'une police de caractère
Couleurs	ColorDialog	Sélection d'une couleur dans la palette
Imprimer	PrintDialog	Impression d'un document
Configuration impression	PrinterSetupDialog	Paramétrage de l'imprimante
Recherche	FindDialog	Recherche d'une donnée
Remplacer	ReplaceDialog	Recherche et remplacement d'une donnée

1. L'interface de Delphi

- b) L'inspecteur d'objets
- Cet outil est dédié à la gestion des composants. La fenêtre de l'inspecteur contient deux volets:
 - la liste des propriétés (attributs) du composant courant,
 - la liste des événements associés au composant courant.



1. L'interface de Delphi

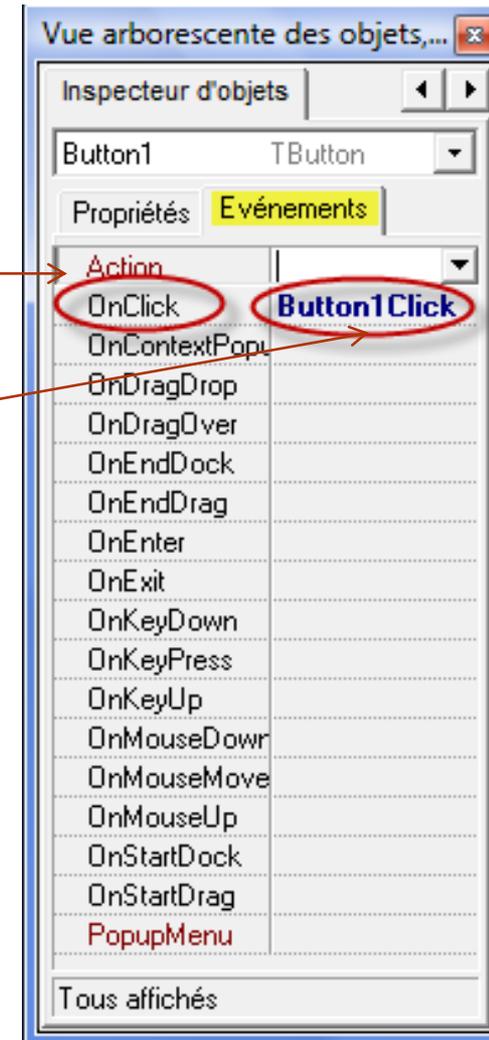
- **Propriétés**
- Les **noms des propriétés** sont placés dans la colonne de gauche (dans l'ordre alphabétique) et les valeurs sur la ligne correspondante à droite.
- Pour donner une **valeur à une propriété**, le plus simple est de remplacer sa valeur par défaut par une nouvelle valeur dans la boîte d'édition prévue à cet effet.

1. L'interface de Delphi

- La propriété **Name** est particulièrement importante car elle permet d'accéder au composant depuis les programmes. Par défaut, Delphi lui confère une valeur peu explicite (ex. Form1, Button1...).
- Il est préférable d'utiliser des noms plus « parlants ».
- Les propriétés visibles dans l'inspecteur sont modifiables lors de la phase de conception

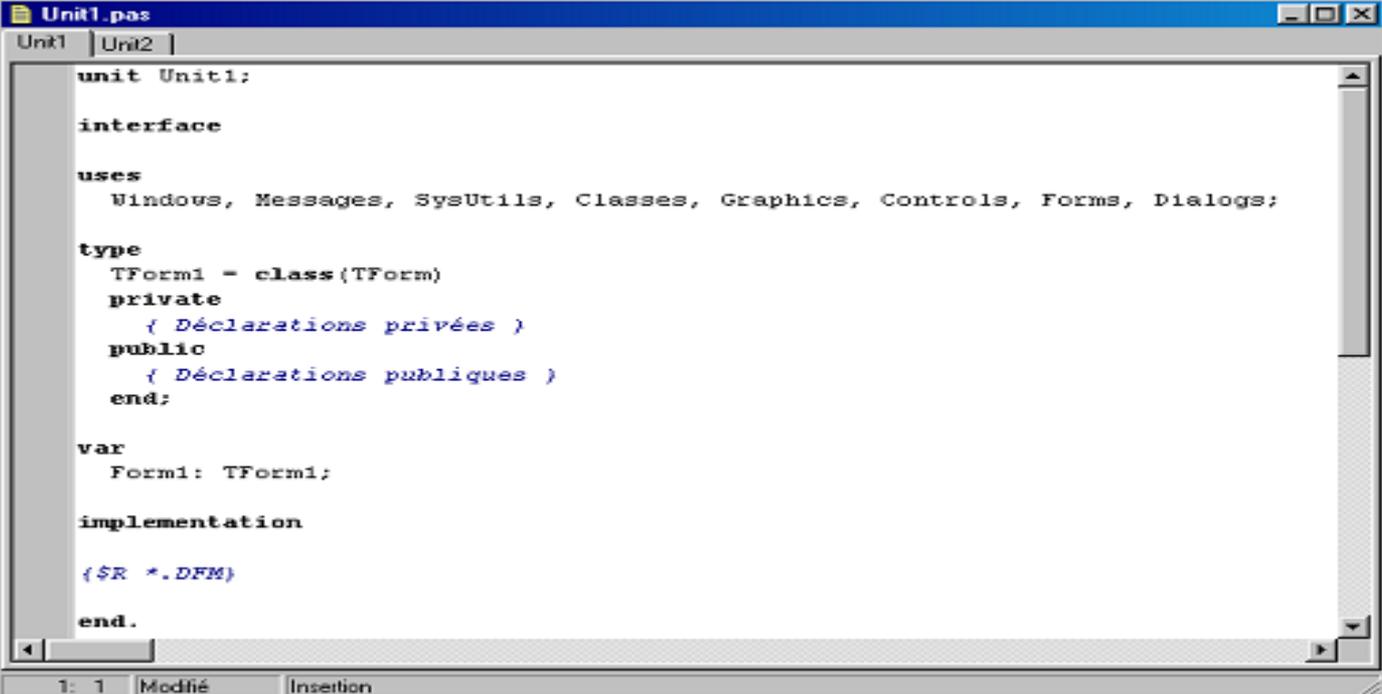
1. L'interface de Delphi

- Événements
- La colonne de gauche contient l'ensemble des événements associés à un composant donné.
- La colonne de droite permet d'associer un gestionnaire d'événements à l'événement correspondant (i.e., un ensemble d'instructions Pascal exécutées lorsque l'événement survient).



1. L'interface de Delphi

- c) L'éditeur de code
- Les fichiers de code source composant un projet sont rassemblés dans l'éditeur de code.



```
Unit1.pas
Unit1 | Unit2 |
unit Unit1;

interface

uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs;

type
  TForm1 = class(TForm)
  private
    { Déclarations privées }
  public
    { Déclarations publiques }
  end;

var
  Form1: TForm1;

implementation

{$R *.DFM}

end.
```

- À chaque fiche est associée une unité identifiée par un onglet situé en haut de la fenêtre de l'éditeur.

1. L'interface de Delphi

- d) Les menus
- **Menu Fichier** : création, ouverture, enregistrement...
- **Menu Édition** : annulation, copier, coller...
- **Menu Chercher** : chercher, remplacer...
- **Menu Voir** : gestionnaire de projets, inspecteur d'objets, débogage...
- **Menu Projet** : compilation de code source, options de compilation...
- **Menu Exécuter** : exécution de programmes, débogage...
- **Menu Composant** : création de nouveau composant, configuration de la palette de composants...
- **Menu Bases de donnée**
- **Menu Outils** : options, utilitaires...
- **Menu Aide** : très utile surtout pour un débutant!

2. Projets et programmes

Notion de projet

- ➔ Delphi permet de créer une seule **application** à la fois, ouverte en tant que **projet**
- ➔ Un projet est l'état **non compilé** d'une application
- ➔ Une application Windows est constituée de **fenêtres**. Le projet Delphi qui génère l'application contient donc ces fenêtres : les **fiches**
- ➔ A chaque fiche est adjointe une seule **unité** (code en langage **Pascal** qui décrit le *contenu* de la fiche (boutons, menus, cases à cocher,...) et son *fonctionnement* (ce qui se passe lorsqu'on clique un bouton) et bien d'autres choses)

REMARQUE : il est possible d'utiliser des unités qui ne sont associées à aucune fiche

2. Projets et programmes

- Concrètement, un **projet** est un fichier binaire d'extension **.dpr** contenant des liens vers tous les éléments du projet (fiches, unités, etc.), eux mêmes stockés dans des fichiers séparés.

Fichier	Extension	Contenu
Projet	.dpr	Code d'initialisation du programme
Code source	.pas	Code source des procédures, fonctions, fiches, ou composants (explicitement implémenté par l'utilisateur ou automatiquement généré par Delphi)
Fiches	.dfm	Caractéristiques et propriétés des composants sous forme binaire
Ressources	.res	Fichier binaire contenant les ressources Windows utilisées par l'application (ex. l'icône du programme)
Code objet compilé	.dcu	
Programme exécutable	.exe	

2. Projets et programmes

Compilation d'un projet

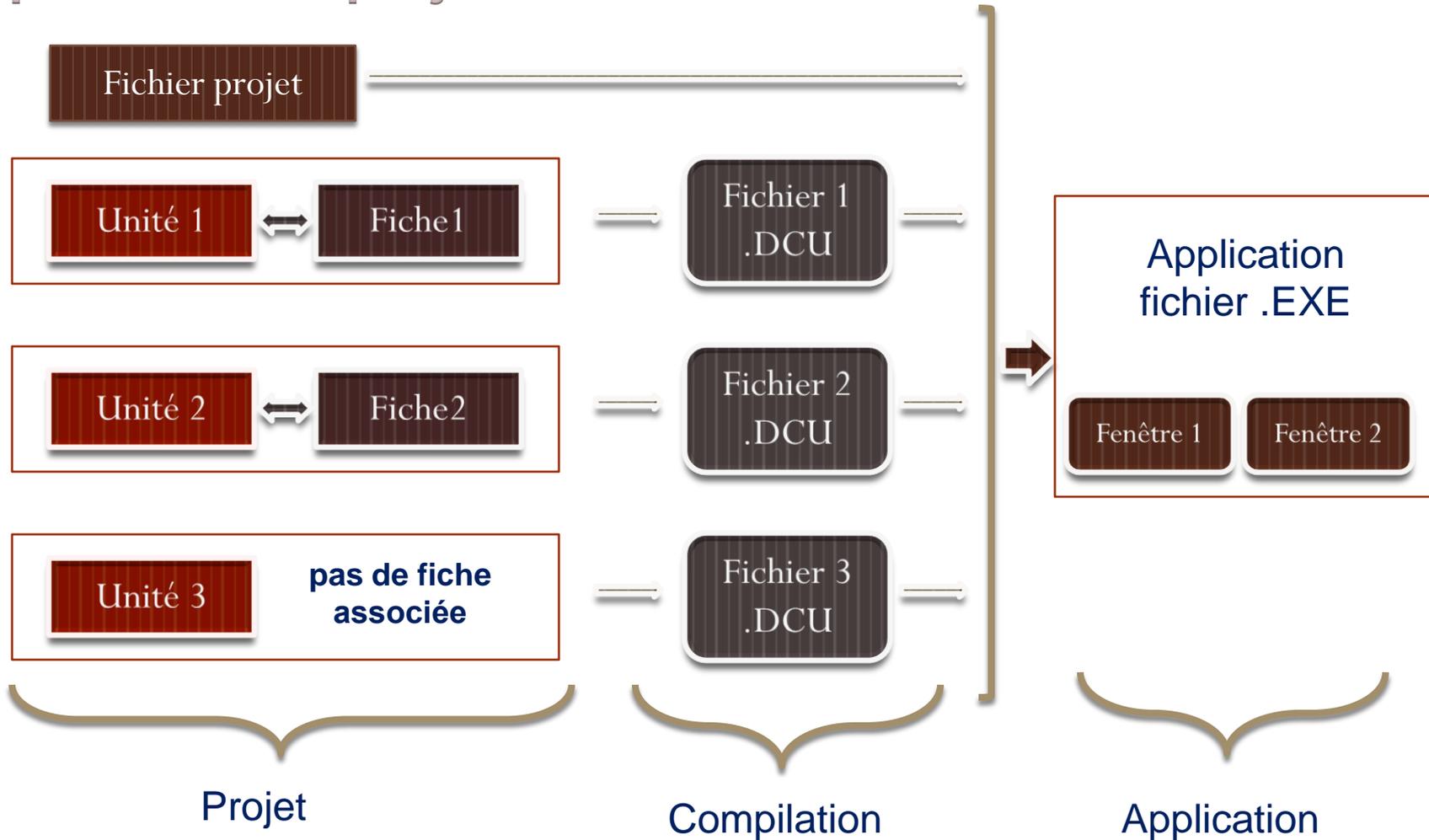
➡ Lors de la compilation, chaque fiche (fichier .DFM) avec son unité (fichier .PAS) seront transformées en un seul fichier (**.DCU**)

➡ Tous les fichiers .DCU seront transformés en un seul fichier exécutable **.EXE** qui représente l'application Delphi générée.

REMARQUE: Le fichier .EXE porte le même nom que le fichier projet .DPR

2. Projets et programmes

Compilation d'un projet



2. Projets et programmes

- Il n'existe qu'un seul fichier .dpr par application. Par défaut, il contient le code suivant.

```
Program Project1;  
Uses Forms,  
    Unit1 in 'UNIT1.PAS'; {Form1}  
  
{ $R *.RES }  
Begin  
    Application.CreateForm(TForm1, Form1);  
    Application.Run(Form1);  
End.
```

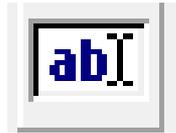
3. Quelques composants standard

- TButton : 
- C'est un composant qui nous permet d'utiliser des boutons d'actions dans nos fiche. 
- Les propriétés les plus importantes de ce composant sont:
 - Name qui donne un nom au bouton.
 - Caption qui donne un titre au bouton.
- On peut changer ce titre (Button1) directement dans l'inspecteur d'objet par la valeur voulue par exemple Ouvrir.

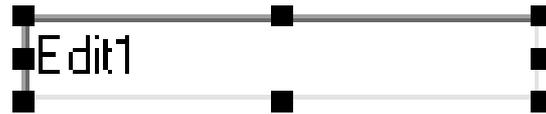


3. Quelques composants standard

- TEdit :

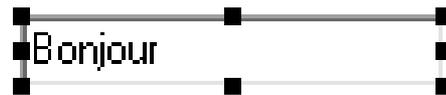


- C'est un composant qui nous permet d'utiliser des champs de saisie.



- La propriété la plus importante de ce composant est la propriété Text qui définit le texte contenu dans le champ de saisie de TEdit.

- On peut modifier la valeur directement dans l'inspecteur d'objet.



- Ou par programmation par l'instruction: `Edit1.Text := 'Bonjour';`

3. Quelques composants standard

- TLabel : 
- C'est un composant qui nous permet d'utiliser des étiquettes dans nos fiche.



- Les propriétés les plus importantes de ce composant sont:
 - Name qui donne un nom au label.
 - Caption qui donne un titre au label.
- On eut changer ce titre (Label1) directement dans l'inspecteur d'objet par la valeur voulue par exemple Cliquer sur le bouton Ouvrir.



3. Quelques composants standard

- TlistBox : 

- C'est un composant qui nous permet d'utiliser des listes dans nos fiche.

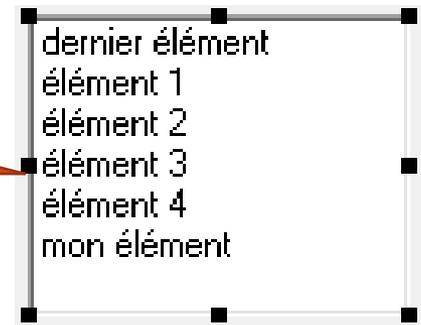


- Les propriétés les plus importantes de ce composant sont:
 - Items qui permet de saisir les éléments de la liste.
 - Sorted qui permet d'ordonner la liste dans l'ordre alphabétique croissant.

Sorted à
False



Sorted à
True



3. Quelques composants standard

- Pour les composants déjà vu il existe plusieurs propriétés en commun:
- La taille du composant est modifiable soit directement par la souris soit à travers les valeurs des propriétés **Height** et **width**.
- La propriété **visible** qui permet de cacher (`visible := false;`) ou visualiser (`visible := true;`) le composant sur la fiche.
- La propriété **left** permet de déterminer la position du composant par rapport au bord gauche de la fiche.

3. Quelques composants standard

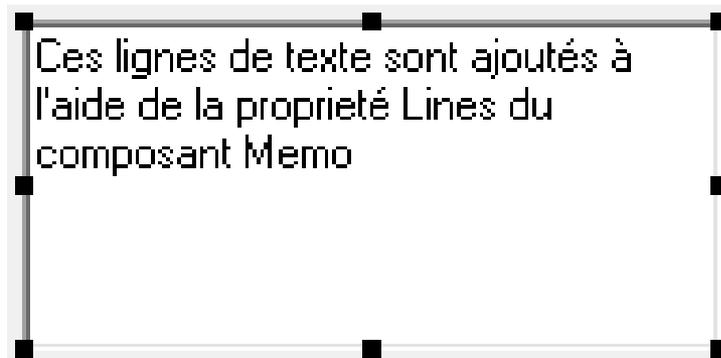
- L'événements communs et important entre ces composants est l'événement:
 - `OnClick` qui permet de déterminer le gestionnaire d'événement associé à l'événement click de souris.
- Pour le composant `TEdit` il y a l'événement `OnChange` qui permet de déclencher un gestionnaire d'événement lors du changement de texte dans le champ de saisie.
- Il existe plusieurs autre événement, `OnExit`, `OnEnter`, `OnKeyPress`, `OnKeyDown`, `OnDblClick`, `OnMouseMove`, ...etc. on va les détailler au besoin incha Allah.

3. Quelques composants standards

- TMemo :



- C'est un composant qui nous permet d'ajouter des lignes de texte (des petits paragraphes) à nos fiches.
- Les propriétés les plus importantes de ce composant sont:
 - **Name** qui donne un nom au composant TMemo.
 - **Lines** qui permet d'ajouter des lignes de texte sur la fiche
 - **Font** qui permet de personnaliser la police du texte dans le Memo.



4. Quelques composants supplément

- TBitBtn : 
- C'est un composant qui nous permet d'utiliser des boutons d'actions dans nos fiches avec une icône à côté du titre du bouton.
- Les propriétés les plus importantes de ce composant sont:
 - **Name** qui donne un nom au bouton.
 - **Glyph** qui définit l'image de l'icône à mettre sur le bouton
 - **Caption** qui donne un titre au bouton.



Quelques composants supplément

- TImage :



- C'est un composant qui nous permet ajouter des images à nos fiches.
- Les propriétés les plus importantes de ce composant sont:
 - **Name** qui donne un nom au composant TImage.
 - **Picture** qui définit l'image qu'on veut mettre sur la fiche
 - **Stretch** permet d'ajuster la taille de l'image à la taille du composant TImage. Cette propriété est booléenne; elle prend soit vrai soit faux.



5. Quelques composants Dialogue

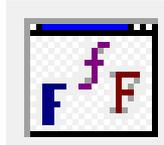
- TOpenPicture :



- C'est un composant qui nous permet d'ajouter une boîte de Dialogue ouverture d'images à nos fiches.
- La propriété la plus importante de ce composant est:
 - **FileName** qui contient le nom de l'image à ouvrir sous forme de chemin d'accès.
- Une de ses fonction est la fonction Execute qui permet de lancer l'exécution de ce composant (l'ouverture de la boîte de dialogue ouvrir image) .

5. Quelques composants Dialogue

- TFontDialog :



- C'est un composant qui nous permet d'ajouter une boite de Dialogue police à nos fiches.
- La propriété la plus importante de ce composant est:
 - **Font** qui contient le style initial de la police.
- La fonction Execute permet de lancer l'exécution de ce composant (l'ouverture de la boite de dialogue police) .