

Les Bases de données avec DELPHI

KHADRAOUI Fairouz

Faculté des sciences économiques, commerciales et
sciences de la gestion

Khadraoui.F@gmail.com

Public cible

LMD 3: Finance Comptabilité Audit
semestre 5

Table des matières

Objectifs

I - Présentation Générale

II - Test de pré-requis

III - Pré Tests

1. Pré Test1

2. Pré Test2

IV - Le cours

1. Chapitre 1 : Éléments de la programmation

1.1. Présentation du langage Delphi

1.2. Environnement Borland Delphi 7

1.3. Notions de Projet, composant et unité

1.4. Programmer avec les composants

1.5. Exercice : Activité Globale

2. Chapitre 2 : Conception d'une BDD

2.1. définition d'une Base De Données

2.2. Exemples de BDD

2.3. Propriétés d'une BDD

2.4. Système de Gestion de Bases de Données (SGBD)

2.5. Modèle Relationnel

2.6. Structure d'une BDD

2.7. Démarche de conception

2.8. Études de cas

2.9. Étapes de réalisation d'une BDD

2.10. Exercice : Activité Globale

V - Post Test

Solution des exercices

Glossaire

Abréviations

Bibliographie

Webographie

Plan du cours



- 1
Eléments de programmation avec Delphi
- 2
Conception et création de Bases de données
- 3
Les requêtes et les états

2013-2014

réalisé par Khadraoui.F

Eléments de programmation avec Delphi

Introduction

- **Delphi** est un environnement de programmation largement employé pour créer des applications Windows.
- Delphi permet d'utiliser le langage **Pascal**
- Comme beaucoup de logiciels, Delphi existe en plusieurs **versions** (Delphi I, ..., Delphi 7, Delphi 2006, ..., Delphi 2010, Delphi XE)
- Le présent cours utilise le **Borland Delphi 7**
- Le but du cours est d'apprendre la programmation **des Bases de données** avec Delphi

Références du cours

- Livre : Introduction à la programmation Orientée Objet : [PASCAL-DELPHI](#) (Frédérique BEALIEU, Edition BERTI, 536 page)
- Site : <http://delphi.developpez.com>
- Document électronique en arabe : (محمد خالد ناصر أغا) [لغة البرمجة دلفي](#)

réalisé par Khadraoui.F

2

Eléments de programmation avec Delphi

Environnement Borland Delphi 7 (Edition entreprise)

The screenshot shows the Borland Delphi 7 IDE interface. Numbered callouts identify the following elements:

- 1: Barre d'outils (Toolbar)
- 2: Barre de menus (Menu bar)
- 3: Palette des composants (Component palette)
- 4: fiche (Form)
- 5: Arborescence des objets de la fiche (Object Inspector)
- 6: Inspecteur d'objets (Object Inspector)
- 7: Editeur de code (Code editor)

réalisé par Khadraoui.F 3

Eléments de programmation avec Delphi

Environnement Borland Delphi 7 (Edition entreprise)

Exemple de fiche

The screenshot shows a Delphi form titled "somme de deux entiers". It contains two text labels, two text edit boxes, and a button labeled "Somme". Red callouts identify these as:

- composants LABEL (pointing to the text labels)
- composant BUTTON (pointing to the "Somme" button)
- composants EDIT (pointing to the text input fields)

Exécution de la fiche

The screenshot shows the form during execution. The first text box contains the value "12", the second contains "3", and the "Somme" button is highlighted. The result "15" is displayed in red text next to the button.

réalisé par Khadraoui.F 4

Eléments de programmation avec Delphi

Environnement Borland Delphi 7 (Edition entreprise)

Vue arborescente des objets

réalisé par Khadraoui.F 5

Eléments de programmation avec Delphi

Environnement Borland Delphi 7 (Edition entreprise)

Inspecteur d'objets

réalisé par Khadraoui.F 6

Eléments de programmation avec Delphi

Environnement Borland Delphi 7 (Edition entreprise)

```

unit Unit2:
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, StdCtrls;
type
  TForm2 = class(TForm)
    Edit1: TEdit;
    Edit2: TEdit;
    Label1: TLabel;
    Label2: TLabel;
    Button1: TButton;
    Label3: TLabel;
  procedure Button1Click(Sender: TObject);
  private
    { Déclarations privées }
  public
    { Déclarations publiques }
  end;
var
  Form2: TForm2;
implementation
($R *.dfm)
procedure TForm2.Button1Click(Sender: TObject);
var a,b,c:integer;
begin
  a:=strtoint(edit1.text);
  b:=strtoint(edit2.Text);
  c:=a+b;
  label3.Caption:=inttostr(c);
end;
end.

```

Editeur de code

déclaration des composants de la fiche

code pascal qui correspond au fonctionnement du bouton

réalise par Khadraoui.F 7

Eléments de programmation avec Delphi

Notions de base de la programmation sous Delphi

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

réalisé par Khadraoui.F 8

Éléments de programmation avec Delphi

Notions de base de la programmation sous Delphi

Fichiers d'un projet

- ➔ Un projet Delphi est constitué d'un **fichier projet**, d'**unités** et de **fiches**
 - ➔ Le fichier projet porte l'extension **.DPR**
 - ➔ Chaque fiche est stockée dans un fichier avec l'extension **.DFM**
 - ➔ Chaque unité est stockée également dans un fichier portant l'extension **.PAS**
 - ➔ La fiche et son unité associée portent le **même nom** de fichier (Delphi ne demande ce nom qu'une seule fois et l'utilise pour l'unité et pour la fiche)

EXEMPLE : si la fiche est stockée dans un fichier **Somme.DFM**, son unité associée sera stockée dans le fichier **Somme.PAS**

réalisé par Khadraoui.F

9

Éléments de programmation avec Delphi

Notions de base de la programmation sous Delphi

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

réalisé par Khadraoui.F

10

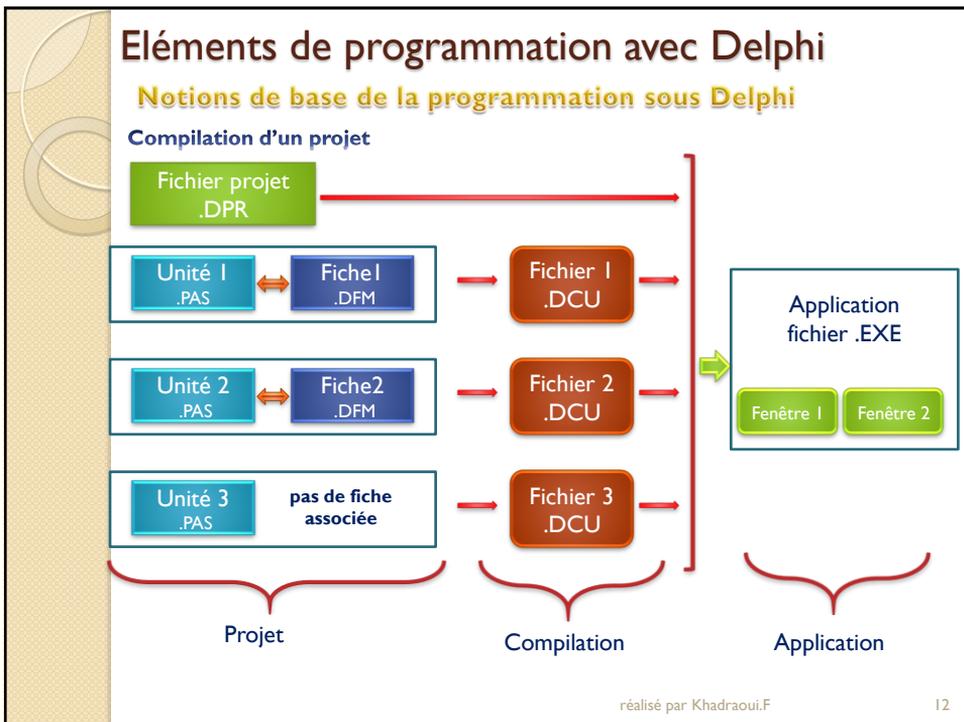
Eléments de programmation avec Delphi

Notions de base de la programmation sous Delphi

Compilation d'un projet

Extension	Description
DPR	Delphi Project : contient l'unité principale du projet
DFM	Delphi ForM (fiche Delphi) : contient la structure de la fiche
PAS	PAScal : contient une unité écrite en Pascal
DCU	Delphi Compiled Unit (unité compilée Delphi) : forme compilée et combinée d'un .PAS et d'un .DFM optionnel
~???	Fichier de sauvegarde temporaire
EXE	EXEcutable : le résultat final de la compilation. Il est seul suffisant pour le fonctionnement de l'application
RES	RESource : contient les ressources de l'application tel que son icône.
DOF DSK CFG	Ces fichiers contiennent les options du projet, les options d'affichage

réalisé par Khadraoui.F 11



Éléments de programmation avec Delphi

Notions de base de la programmation sous Delphi

Structure du fichier Projet

- ➔ Le texte du fichier projet (.DPR) est généré entièrement par Delphi (et non pas par le programmeur). Il représente le programme principal
- ➔ Pour afficher le texte de ce fichier, deux méthode:
 - Menu **Voir | Unités** (CTRL+F12) puis sélectionner le nom du fichier projet
 - Menu **Projet| Voir le source**

réalisé par Khadraoui.F

13

Éléments de programmation avec Delphi

Notions de base de la programmation sous Delphi

Structure du fichier Projet

1. En tête du programme
2. Le bloc **USES** précise la liste des unités utilisées par le projet
3. Bloc des instructions (entre Begin et End) précise les premières instructions à exécuter au démarrage de l'application

```

Unit1 Project1
program Project1; 1
uses 2
  Forms,
  Unit1 in 'Unit1.pas' (Form1);
{$R *.res}
begin 3
  Application.Initialize;
  Application.CreateForm(TForm1, Form1);
  Application.Run;
end.
  
```

REMARQUES

- Le texte entre { et } en bleu représente un commentaire (ignoré par Delphi)
- Le texte entre { et } en vers et précédé de \$ représente une directive de compilation

réalisé par Khadraoui.F

14

Éléments de programmation avec Delphi

Notions de base de la programmation sous Delphi

Notion de composant

- ➔ Delphi permet de créer des **applications Windows** (possibilité de l'interaction avec l'utilisateur à l'aide des: fenêtres, boutons, case à cocher, menus, listes,)
- ➔ Chaque élément d'une fenêtre est dit : **Composant** et il est accessible à partir de la **palette des composants**



réalisé par Khadraoui.F

15

Éléments de programmation avec Delphi

Notions de base de la programmation sous Delphi

Propriétés et événements

- ➔ Chaque **fiche**, chaque **composant** possède une liste de **propriétés** et une liste **d'événements**
- ➔ Les **propriétés** sont des paramètres réglables pour un composant. Par exemple : les dimensions, les couleurs, les polices, le titre d'une fenêtre, le texte d'un bouton,....
- ➔ une partie **d'événements** est provoquée lors de l'utilisation des composants et des fenêtres tel que les clics et les mouvements de la souris, les frappes de touches. D'autre partie d'événements est provoquée lorsqu'une fenêtre soit visible, invisible, lorsqu'une case à cocher est cochée ou un élément d'une liste est sélectionné.

réalisé par Khadraoui.F

16

Éléments de programmation avec Delphi

Notions de base de la programmation sous Delphi

Notion d'unité

- ➔ Une unité est un fichier séparé pouvant contenir des **constantes**, des **types**, des **variables** et des **procédures et fonctions** disponibles pour la construction d'autres applications.
- ➔ L'utilisation des unités permet de partager des données et des procédures et fonctions entre plusieurs applications.

réalisé par Khadraoui.F

17

Éléments de programmation avec Delphi

Notions de base de la programmation sous Delphi

Syntaxe

```

UNIT Nom_unite; 1
INTERFACE 2
  [ Déclarations publiques ]
IMPLEMENTATION 3
  [ Déclarations privées ]
  [ Corps des procédures et des fonctions ]
INITIALIZATION 4
  [ Code d'initialisation ]
FINALIZATION 5
  [ Code de finalisation ]
END.

```

5. Ce code s'exécute avant la fin du pgm utilisant l'unité

1. le nom de l'unité figure dans la partie **USES** d'un programme ou d'une unité qui utilise cette unité.
2. Tout ce qui est placé ici est visible pour toute entité utilisant l'unité
3. Cette partie définit les **procédures** et les **fonctions** déclarées dans la partie interface. On peut également y définir des **types**, **constantes**, **variables** et **sous programmes locaux**.
4. Ce code s'exécute avant la première action du pgm utilisant l'unité

réalisé par Khadraoui.F

18

Éléments de programmation avec Delphi

Notions de base de la programmation sous Delphi

Exemple d'unité

Unit formes ;

Interface

Const pi=3.14 ;

Function surfCercle(R: Real): real;

Function PerimRect (Long, Larg:Real):Real;

Implementation

Function surfCercle; {Pas besoin de répéter les paramètres}

Var s: real;

Begin

s:=R*R*Pi;

Result :=s;

End ;

Function PerimRect;

Begin

result :=(long+larg)*2;

End ;

End.

réalisé par Khadraoui.F

19

Éléments de programmation avec Delphi

Notions de base de la programmation sous Delphi

Utilisation

.....

Uses formes;

Var A,B,C,P: real;

.....

P:=PerimRect(A,B);

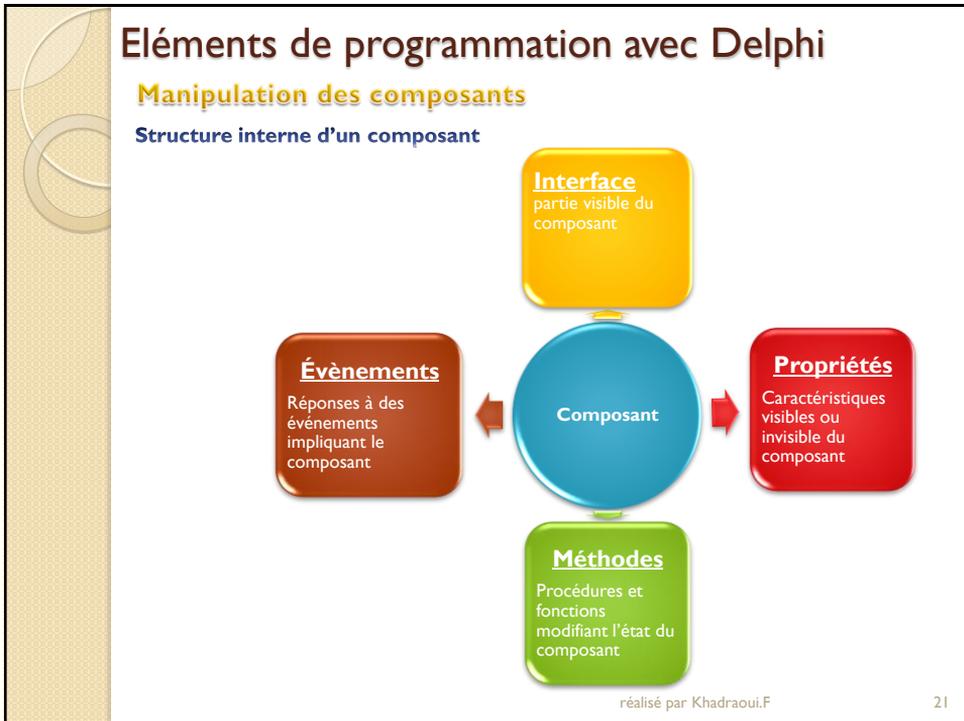
.....

Label.caption := FloatToStr (surfCercle(c));

....

réalisé par Khadraoui.F

20



Eléments de programmation avec Delphi

Manipulation des composants

Manipulation des propriétés

Deux moyens pour modifier les propriétés d'un composant :

1. Utilisation Delphi

- Manipulation visuelle : par exemple pour dimensionner un composant, il faut le sélectionner puis déplacer les poignées avec la souris
- Par inspecteur d'objets :

Poigné

Le composant

La propriété

Le composant

La valeur de la propriété

Inspecteur d'objets	
Button2 TButton	
Propriétés Événements	
DragMode	dmManual
Enabled	True
Font	(TFont)
Height	45
HelpContext	0
HelpKeyword	
HelpType	htContext
Hint	
Left	350

réalisé par Khadraoui.F 22

Éléments de programmation avec Delphi

Manipulation des composants

Manipulation des propriétés

2. Utilisation du code

Pour utiliser une propriété d'un composant dans du code pascal (précisément dans une procédure interne à la fiche qui contient le composant) il faut écrire :

- Le nom du composant (sa propriété Name)
- Un point (.)
- Le nom de la propriété

EXEMPLE

Button2 . Height := 45;

Le composant

La propriété

réalisé par Khadraoui.F

23

Éléments de programmation avec Delphi

Manipulation des composants

Manipulation des méthodes

- Les méthodes sont des procédures et des fonctions internes aux composants.
- Les méthodes font tout le travail d'un composant.
- De même qu'une propriété, une méthode doit être précédée par le nom du composant qu'il manipule.

EXEMPLES

Button2 . Hide ; Pour masquer le bouton

Button2 . Show ; Pour réafficher le bouton

réalisé par Khadraoui.F

24

Éléments de programmation avec Delphi

Manipulation des composants

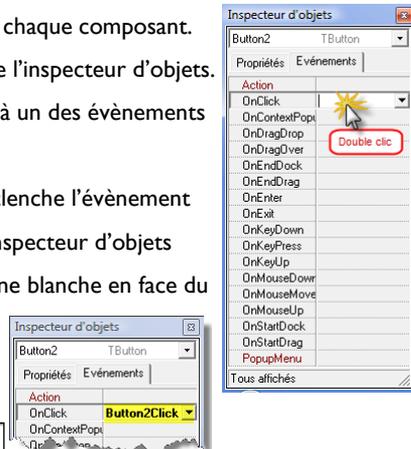
Manipulation des événements

- ➔ La liste des événements est spécifique à chaque composant. Elle est listée dans l'onglet **événement** de l'inspecteur d'objets.
- ➔ il est possible d'assigner une procédure à un des événements d'un composant, comme suit :
 - ☞ Sélectionner le composant qui déclenche l'évènement
 - ☞ Aller à l'onglet **événement** de l'inspecteur d'objets
 - ☞ Effectuer un double clic dans la zone blanche en face du nom de l'évènement à traiter
 - ☞ la procédure sera générée

```

procedure TForm1.Button2Click(Sender: TObject);
begin
|
end;

```



réalisé par Khadraoui.F

25

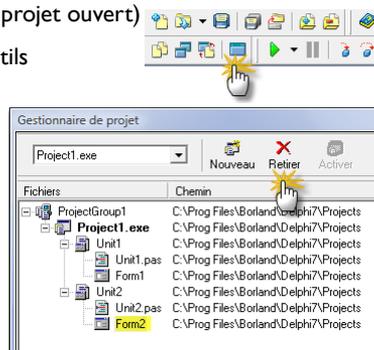
Éléments de programmation avec Delphi

Manipulation des composants

Les composants les plus utilisés

La fiche : composant « Form »

- ➔ La fiche est un composant **conteneur** (peut contenir d'autres composants)
- ➔ La fiche ne se crée pas depuis la palette des composants
- ➔ Pour créer une nouvelle fiche (Ajoutée au projet ouvert)
 - l'outil **Nouvelle fiche** de la barre d'outils
 - menu **Fichier | Nouveau | Fiche**
- ➔ Pour retirer la fiche du projet :
 - Menu **voir | Gestionnaire de projet**
 - La commande **Retirer**
- ➔ Pour afficher la liste des fiches du projet :
 - Menu **Voir | Fiches**



réalisé par Khadraoui.F

26

Éléments de programmation avec Delphi

Manipulation des composants

Propriétés d'une fiche

Propriété	Type	description
Name	String	Indique le nom du composant utilisé dans le code de l'application
Caption	String	Spécifie une chaîne de texte permettant à l'utilisateur d'identifier le composant. Pour la fiche le texte s'affiche sur sa barre de titre
Height	Integer	Indique la taille verticale du composant, exprimée en pixels.
Width	Integer	Détermine la taille horizontale, exprimée en pixels, du composant ou de la fiche.
Left	Integer	Détermine la coordonnée horizontale, exprimée en pixels relativement à la fiche, du bord gauche d'un composant.
Color	Tcolor	Indique la couleur d'arrière-plan du composant.
Font	Tfont	Détermine les attributs du texte écrit au-dessus ou dans le composant.
Hint	String	Contient la chaîne de texte (conseil ou info bulle) apparaissant lorsque l'utilisateur déplace la souris au-dessus du composant. Il faut aussi affecter True à la valeur de la propriété ShowHint
Visible	Boolean	Détermine si le composant apparaît à l'écran.

réalisé par Khadraoui.F

27

Éléments de programmation avec Delphi

Manipulation des composants

La fiche : composant « Form »

REMARQUE

Les propriétés précédentes sont valables pour plusieurs composants qui vont venir par la suite de ce chapitre

Évènements d'une fiche

Evènement	Description
OnClick	Se produit quand l'utilisateur clique sur le composant. Pour une fiche, il se produit si l'utilisateur clique sur une zone vide de la fiche ou sur un composant désactivé.
OnDbClick	Se produit quand l'utilisateur double-clique avec le bouton gauche de la souris alors que le pointeur de la souris est au-dessus du composant.
OnMouseMove	Se produit quand l'utilisateur déplace le pointeur de la souris au-dessus d'un contrôle.
OnClose	Se produit quand la <u>fiche</u> se ferme.
OnHide	Se produit quand la <u>fiche</u> est cachée (c'est-à-dire quand la propriété Visible de la fiche prend la valeur False).
OnShow	Se produit quand la <u>fiche</u> est affichée (c'est-à-dire quand la propriété Visible de la fiche prend la valeur True).

réalisé par Khadraoui.F

28

Éléments de programmation avec Delphi

Manipulation des composants

La fiche : composant « Form »

Méthodes d'une fiche

Méthode	Description
Close	Est une procédure sans paramètre qui ferme la <u>fiche</u> Quand la fiche principale de l'application se ferme, l'application se termine.
Show	Est une procédure sans paramètre qui affiche la <u>fiche</u>
Hide	Est une procédure sans paramètre qui cache la <u>fiche</u>

Le composant « Label »



- Un composant Label permet d'inclure facilement du texte sur une fiche.
- Il peut contenir jusqu'à 255 caractères

Propriétés d'un Label

En plus les propriétés Name, Caption, Width, Height, Left, Hint, ShowHint, Color, Font, visible, le composant label possède les propriétés suivantes :

réalisé par Khadraoui.F

29

Éléments de programmation avec Delphi

Manipulation des composants

Propriétés d'un Label

Propriété	Type	description
AutoSize	Boolean	Spécifie si le label se redimensionne automatiquement pour s'adapter à son contenu. La valeur par défaut de cette propriété est True
WordWrap	Boolean	Autorise les retours à la ligne pour permettre d'afficher plusieurs lignes à l'intérieur du label. AutoSize doit être False pour permettre l'utilisation de plusieurs lignes.

Le composant « Edit »



- Un composant Edit permet de proposer une **zone d'édition**.
- Ce composant permet d'entrer une information quelconque tapée au clavier.

Propriétés d'un Edit

En plus les propriétés Name, Width, Height, Left, Hint, ShowHint, Color, Font, visible, AutoSize (pour ajuster la largeur) le composant Edit possède les propriétés spécifiques suivantes :

réalisé par Khadraoui.F

30

Éléments de programmation avec Delphi

Manipulation des composants

Propriétés d'un Edit

Propriété	Type	description
Text	String	Contient le texte entré dans la zone d'édition. C'est aussi en modifiant cette propriété que l'on fixe le contenu de la zone
MaxLength	Integer	Spécifie le nombre maximum de caractères que l'utilisateur peut entrer dans la zone Edit. Mettre 0 pour ne pas donner de limite (par défaut)
PasswordChar	Char	À utiliser lorsqu'on veut masquer les caractères tapés, comme pour les mots de Passe. Utiliser par exemple le caractère « * » pour masquer et le caractère « #0 » pour ne pas masquer

Évènements d'un Edit

En plus les évènements OnClick, OnDbClick, OnMouseMove, le composant Edit possède l'évènement spécifique suivant :

Evènement	Description
OnChange	Se produit quand le texte de la zone de saisie a peut-être changé.

réalisé par Khadraoui.F

31

Éléments de programmation avec Delphi

Manipulation des composants

Méthodes d'un Edit

Méthode	Description
Clear	Est une procédure sans paramètre qui efface tout le texte de la zone de saisie.
ClearSelection	Est une procédure sans paramètre qui efface le texte sélectionné de la zone de saisie.

Le composant « Button »



→ Le composant Button sert en général à proposer à l'utilisateur une action.

Propriétés d'un Button

Comme la plupart des composants, le composant Button possède les propriétés Name, Caption, Width, Height, Left, Hint, ShowHint, Font, Visible, WordWrap

Évènements d'un Button

Les évènements OnClick, OnMouseMove

réalisé par Khadraoui.F

32

Éléments de programmation avec Delphi

Manipulation des composants

Le composant « ListBox »



- ↳ ListBox permet d'afficher une collection d'éléments dans une liste déroulante..
- ↳ Il est possible de sélectionner un ou plusieurs éléments (mais pas d'édition).

Propriétés d'un ListBox

En plus les propriétés Name, Width, Height, Left, Hint, ShowHint, Color, Font, visible, le composant ListBox possède les propriétés spécifiques suivantes :

Propriété	Type	description
Sorted	Boolean	Spécifie si les éléments de la liste sont triés par ordre alphabétique.
MultiSelect	Boolean	Détermine s'il est possible de sélectionner plusieurs éléments à la fois.
Items	TString	Est une propriété Objet qui contient les éléments (les chaînes de caractères) qui apparaissent dans la liste. Elle dispose ses propres propriétés et méthodes: Count : est une propriété (Integer) qui indique le nombre des éléments de la liste. Add(str) : est une méthode (fonction) qui ajoute la chaîne « str » à la liste des éléments Clear : est une méthode (procédure) qui vide la liste.

réalisé par Khadraoui.F

33

Éléments de programmation avec Delphi

Manipulation des composants

Méthodes d'un ListBox

Méthode	Description
Clear	Est une procédure sans paramètre qui efface tous les éléments de la liste. C'est l'équivalent de Items.Clear
ClearSelection	Est une procédure sans paramètre qui désélectionne tous les éléments de la liste (attention ! C'est différent de la méthode ClearSelection de Edit)

Le composant « Memo »



- ↳ Le composant Memo permet l'édition du texte sur plusieurs lignes.
- ↳ Il stocke le texte sous forme de lignes: chaque ligne est une chaîne de caractère.

Propriétés d'un Memo

En plus les propriétés Name, Width, Height, Left, Hint, ShowHint, Color, Font, visible, le composant Memo possède les propriétés spécifiques suivantes :

réalisé par Khadraoui.F

34

Éléments de programmation avec Delphi

Manipulation des composants

Propriétés d'un Memo

Propriété	Type	description
Lines	TString	Est une propriété Objet qui contient les lignes (les chaînes de caractères) du texte. Elle dispose des propriétés et méthodes analogues à celles de la propriété Items de ListBox

Evènements d'un Memo

Similaires à ceux de Edit (OnChange, OnClick, OnDbClick, OnMouseMove)

Méthodes d'un Memo

Similaires à celles d'un Edit (Clear, ClearSelection)

Le composant « BitBtn »



- BitBtn (bouton Bitmap) est un composant de l'onglet **Supplément** de la palette des composants.
- Il représente un bouton avec une image sur sa face..
- Les boutons bitmap se comportent de la même manière que les composants bouton.

réalisé par Khadraoui.F 35

Éléments de programmation avec Delphi

Manipulation des composants

Propriétés d'un BitBtn

En plus les propriétés Name, Caption, Width, Height, Left, Hint, ShowHint, Color, Font, Visible, ce composant possède les propriétés spécifiques suivantes

Propriété	Type	description
Glyph	TBitmap	Spécifie le bitmap (image) qui apparaît sur le bouton BitBtn.
Kind	TBitBtnKind	Détermine le type du bouton bitmap. Le type TBitBtnKind définit plusieurs constantes pour des boutons Standards : <div style="display: flex; flex-wrap: wrap; padding: 5px;"> <div style="margin-right: 20px;"> bkOK :  </div> <div style="margin-right: 20px;"> bkNo :  </div> <div style="margin-right: 20px;"> bkCancel :  </div> <div style="margin-right: 20px;"> bkYes :  </div> <div style="margin-right: 20px;"> bkHelp :  </div> <div> bkClose :  </div> </div>

Evènements et méthodes d'un BitBtn

Similaires à ceux de Button

réalisé par Khadraoui.F 36

Bases De Données avec Delphi

The diagram features two stacks of circular discs. The left stack, with blue and white discs, is labeled 'Eléments de programmation avec Delphi'. A large white arrow points from this stack to the right stack, which has red and white discs and is labeled 'Conception et Réalisation d'une base de données'. The background is a dark blue space with glowing light trails.

2013-2014 1 Khadraoui.F

Plan

1. Conception d'une base de données
 - Définition d'une Base De Données
 - Exemples de BDD
 - Propriétés d'une BDD
 - Système de Gestion de Base de Données (SGBD)
 - Le modèle relationnel
 - Structure d'une BDD
 - Démarche de conception
 - Etudes de cas
 - Etapas de réalisation d'une BDD

2013-2014 2 Khadraoui.F



Conception d'une Base de données

Introduction

- Le **volume des données** généré par les activités humaines croît avec le développement de notre civilisation.
- Ces données sont souvent gérées par des **moyens informatique**
- Les données d'un établissement d'enseignement, d'une entreprise, d'une banque, d'un hôpital,etc sont souvent des ensembles de **même structure**. Par exemple dans une entreprise:
 - ❖ Liste des membres du **personnel** : pour chaque personne, on enregistre le nom, prénom, adresse, sexe, date de naissance, date de recrutement, fonction dans l'entreprise, etc.
 - ❖ Liste de produits dans le **stock** : pour chaque produit, on garde : la référence, désignation, quantité en stock, prix unitaire, taux TVA, etc.

Définition

Une **Base De Données (BDD)** est un ensemble structuré de données enregistrées sur des supports accessibles par l'ordinateur, pour satisfaire un ensemble d'utilisateurs de façon sélective.

2013-2014 Khadraoui.F



Conception d'une Base de données

Exemples de BDD

Entreprise

Données	Données
<ul style="list-style-type: none"> • Produits • Clients • Fournisseurs • Commandes • Factures • 	<ul style="list-style-type: none"> • Employés • Salaires • Congés • Retraites • Mutations • Avancements •
<div style="background-color: #c0c0c0; border-radius: 50%; width: 60px; height: 60px; margin: 0 auto; display: flex; align-items: center; justify-content: center;"> <p style="margin: 0;">BDD du Stock</p> </div>	<div style="background-color: #00a68a; border-radius: 50%; width: 60px; height: 60px; margin: 0 auto; display: flex; align-items: center; justify-content: center;"> <p style="margin: 0; color: white;">BDD du Personnel</p> </div>

2013-2014 Khadraoui.F

 **Conception d'une Base de données**

Exemples de BDD

Etablissement d'enseignement

données	données
<ul style="list-style-type: none"> • Etudiants • Modules • Formations • Enseignants • Notes • 	<ul style="list-style-type: none"> • Livres • Thèses • Emprunts • Adhérents •
BDD de la scolarité	BDD de la bibliothèque

2013-2014 5 Khadraoui.F

 **Conception d'une Base de données**

Propriétés d'une BDD

- **Exhaustivité des données**
la BDD doit contenir toutes les informations requises par les traitements attendus.
Exemple : si une personne est absente dans la BDD du personnel, elle risque de ne pas être payée
- **Données non redondantes (unicité)**
une information ne sera pas répétée plusieurs fois dans une BDD.
Exemple : si une personne est enregistrée deux fois, elle risque de toucher double paye !?

2013-2014 6 Khadraoui.F



Conception d'une Base de données

Propriétés d'une BDD

- **Données structurées**
 les données d'une BDD doivent être réparties d'une manière logique en des ensembles de même structure.
Exemple : on ne peut pas mettre ensemble les informations des produits et celles des clients, il faut les séparer.
- **Données en liaison**
 les données d'une même BDD doivent avoir des relations entre elles et ne peuvent être toutes indépendantes les unes des autres.
Exemple : les notes dans une BDD d'un établissement d'enseignement dépendent des étudiants et des formations.

2013-2014 Khadraoui.F

7



Conception d'une Base de données

Système de Gestion de Base de Données (SGBD)

Définition

Le logiciel qui gère une BDD s'appelle un **Système de Gestion de Base de Données (SGBD)** (**DBMS** en anglais, pour **Data Base Management System**).

Exemples

Delphi, ACCESS, DBASE, FoxPro, Oracle, Clipper, SQL server, , etc.

Fonctionnalités

- ✓ Définir la structure de la BDD.
- ✓ Conserver les données sur les Disques.
- ✓ Rechercher les données selon les critères des utilisateurs.
- ✓ Assurer la sécurité et confidentialité des données surtout dans un environnement multiutilisateurs.

2013-2014 Khadraoui.F

8

 **Conception d'une Base de données**

Le modèle relationnel

Un modèle de données (de BDD) est un ensemble de règles selon lesquelles les données sont structurées et organisées. Il existe plusieurs modèles de BDD :

- Modèle HIERARCHIQUE
- Modèle RESEAU
- Modèle Orienté Objets
- Modèle **RELATIONNEL** (le plus répandu)

2013-2014 9 Khadraoui.F

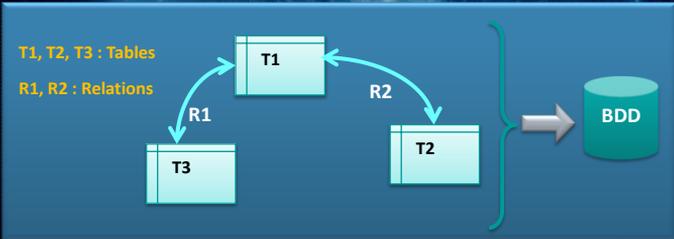
 **Conception d'une Base de données**

Structure d'une BDD

Une BDD est un ensemble de données organisé sous forme d'une ou plusieurs tables de données.

Chaque table représente un ensemble de données relatives à un domaine ou une entité bien déterminée.

Des liaisons existent entre tables qui dépendent les unes des autres, c'est la raison pour laquelle, elles sont rassemblées dans une même BDD.



2013-2014 10 Khadraoui.F



Conception d'une Base de données

Table

Une table est un ensemble de données organisées en lignes et en colonnes.

- Les lignes sont appelées : **Enregistrements**
- Les colonnes sont appelées : **Champs**



```

graph TD
    BD[Base de données] --> T1[Table 1]
    BD --> T2[Table 2]
    BD --> T3[Table 3]
    T1 --> E1[Enregistrement 1]
    T2 --> E2[Enregistrement 2]
    T3 --> E1
    T3 --> E2
    E1 --> C1[Champ1]
    E1 --> C2[Champ2]
    E1 --> C3[Champ3]
    E2 --> C1
    E2 --> C2
    E2 --> C3
    
```

2013-2014 Khadraoui.F



Conception d'une Base de données

Exemples de tables

Table de produits



Réf.ence	Désignation	Prix unitaire	Quantité en stock
01	Lait liquide	25 DA	3000
02	Café	400 DA	200
03	Sucre	60 DA	100
04	Lait en poudre	170 DA	50

Enregistrements

2013-2014 Khadraoui.F



Conception d'une Base de données

Exemples de tables

Table des Employés

Nom	Prénom	Adresse	Sexe	Date naissance	Date recrutement
Allaoui	Ahmed	25 rue Biskra, Batna	M	12/04/70	02/06/94
Zidani	Samia	15 cité 150 log, Batna	F	14/01/80	10/04/05
Ben Cherif	Mohamed	N 123 bloc A2, cité 1200 log, Batna	M	02/11/78	13/09/01

2013-2014 13 Khadraoui.F



Conception d'une Base de données

Démarche de conception

Une BDD peut stocker une quantité considérable d'informations, celles-ci peuvent évoluer dans le temps et peuvent subir des mises à jours.

Vu l'importance d'une BDD, sa création doit être réfléchie. La meilleure façon de la concevoir et d'anticiper tous les besoins en information et à long terme. Toutefois, une BDD peut être conçue par approche successive.

Avant de se lancer dans la création d'une BDD, on devrait prendre quelques instants pour structurer nos besoins en information. On propose les étapes d'analyse suivantes :

2013-2014 14 Khadraoui.F



Conception d'une Base de données

Demarche de conception

- 1.** Définir l'objet ou le domaine d'étude et décrire tous les paramètres le décrivant.
- 2.** Faire un découpage des informations (selon leurs natures et dépendances) jusqu'au plus petit élément significatif.
- 3.** Structurer la BDD en des tables
- 4.** Déterminer les relations qui peuvent exister entre les tables



Schéma de la BDD

2013-2014
15
Khadraoui.F



Conception d'une Base de données

Etudes de cas

1. Suivi des notes des étudiants dans un établissement d'enseignement

Etudiants
NumIns*
Nom
Prénom
DateN

Notes
NumIns*
CodeM*
Note

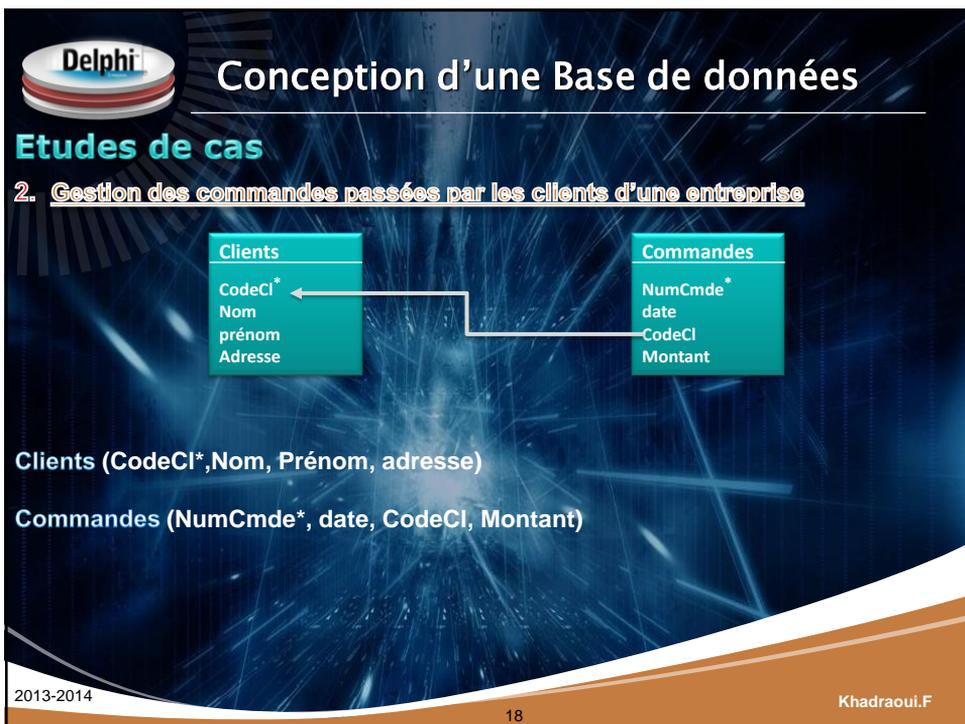
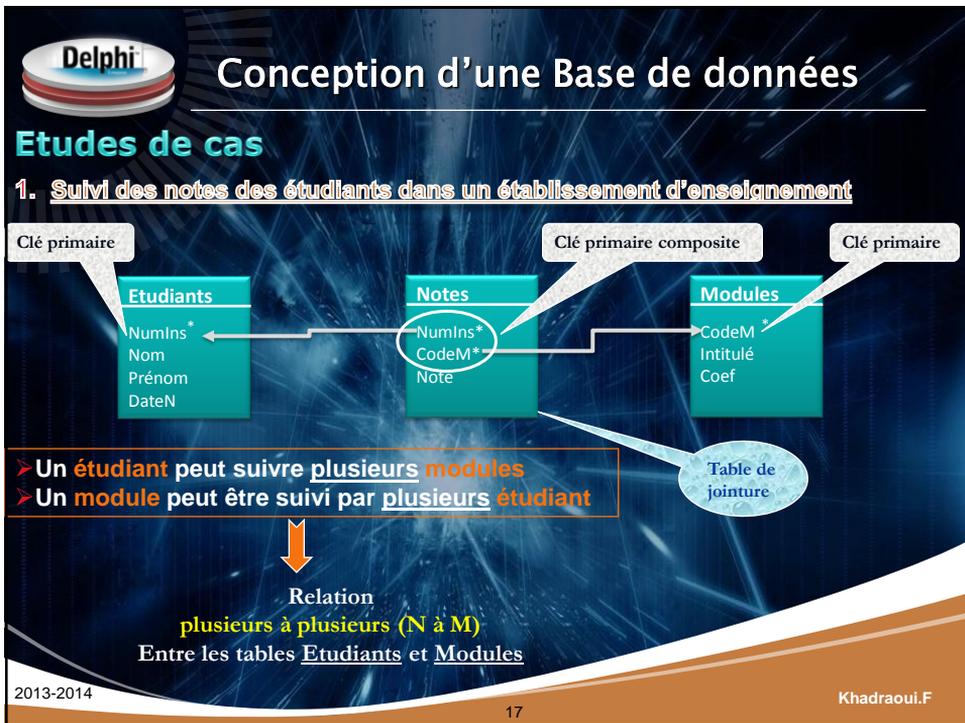
Modules
CodeM*
Intitulé
Coef

Etudiants (NumIns*, nom, prénom, dateN)

Modules (CodeM*, Intitulé, Coef)

Notes (NumIns*, CodeM*, Note)

2013-2014
16
Khadraoui.F



Delphi Conception d'une Base de données

Etudes de cas

2. Gestion des commandes passées par les clients d'une entreprise

Clé primaire

Clients
CodeCl*
Nom
prénom
Adresse

Commandes
NumCmde*
date
CodeCl
Montant

Clé primaire

Clé étrangère

➤ Un **Client** peut passer **plusieurs Commandes**

➤ Une **Commande** peut être passée par **un seul client**

↓

Relation
un à plusieurs (1à N)
Entre les tables **Commandes** et **Clients**

2013-2014 Khadraoui.F

Delphi Conception d'une Base de données

Etudes de cas

3. Gestion des bureaux des enseignants

Enseignants
CodeEns*
Nom
prénom
Spécialité
NumBur

Bureaux
NumBur*
NumBloc
Etage

Enseignants (CodeEns*, Nom, Prénom, Spécialité, NumBur)

Bureaux(NumBur*, NumBloc, Etage)

2013-2014 Khadraoui.F

Delphi Conception d'une Base de données

Etudes de cas

3. Gestion des bureaux des enseignants

Enseignants
 CodeEns*
 Nom
 prénom
 Spécialité
 NumBur

→

Bureaux
 NumBur*
 NumBloc
 Etage

➤ Un **enseignant** peut avoir un **seul Bureau**
 ➤ Un **bureau** peut être attribué à un **seul enseignant**

↓

Relation
un à un (1 à 1)
Entre les tables Enseignants et Bureaux

2013-2014 Khadraoui.F

Delphi Conception d'une Base de données

Etapes de réalisation d'une BDD

Une fois l'analyse et la conception de la BDD terminées, on doit passer à sa réalisation à l'aide d'un SGBD. Cela nécessite les étapes :

- ✓ **Création des tables de la BDD** : la création d'une table passe par deux étapes :
 - ✓ Création de sa structure (définition des **champs**)
 - ✓ Saisie des informations (création des **enregistrements**)
- ✓ **Définition des relations entre tables**
- ✓ **Interrogation de la BDD** : Une BDD ne doit pas être sans intérêt, elle doit restituer des informations à la demande des utilisateurs. Une demande constitue une interrogation (requête) de la BDD.
- ✓ **Etablissement des états à imprimer** : l'utilisateur d'une BDD a toujours besoin d'afficher et d'imprimer des listings (états) contenant des données obtenues à partir des tables des requêtes.

2013-2014 Khadraoui.F