

I.U.T. G.E.I.I.
1ère Année

TP

DELPHI

Langage de Programmation Visuel Orienté Objet

I.U.T. G.E.I.I.
7 rue Jules Vallès
19 100 BRIVE

1. INTRODUCTION

DELPHI a deux principaux atouts:

- C'est un langage de **programmation visuel**, sous WINDOWS. Grâce à une bibliothèque d'objets visuels appelés composants: les VCL (Visual Component Library), la programmation est simplifiée. On n'écrit aucune ligne de code pour créer l'environnement d'une application, mais on clique et déplace simplement avec la souris des composants visuels (fenêtre, bouton, ascenseur ...).

- De plus c'est un **langage orienté objet** car il intègre le Pascal Objet. Mais nous nous ferons pas de POO (Option de 2^{ème} année)

2. OBJETS ET COMPOSANTS

OBJET = En informatique, un objet est une représentation d'un objet physique, défini par ses caractéristiques propres et par son comportement (ou manipulations possibles).

ex:

	<u>Caractéristiques</u>	<u>Comportements</u>
<i>Télévision</i>	<i>Couleur</i>	<i>Allumer</i>
	<i>Hauteur</i>	<i>Eteindre</i>
	<i>Profondeur</i>	<i>Changer de canal</i>
	<i>Largeur</i>	<i>Augmenter le son</i>
	<i>Taille écran</i>	<i>Modifier le contraste</i>
	<i>Tension alimentation</i>	<i>Diminuer la lumière</i>

Exemples d'objets informatiques souvent rencontrés dans les logiciels:

Bouton, liste, fenêtre, ...

COMPOSANT = Objet réutilisable dans différents contextes.

Les objets de DELPHI sont des composants VCL

1°) Lancer DELPHI

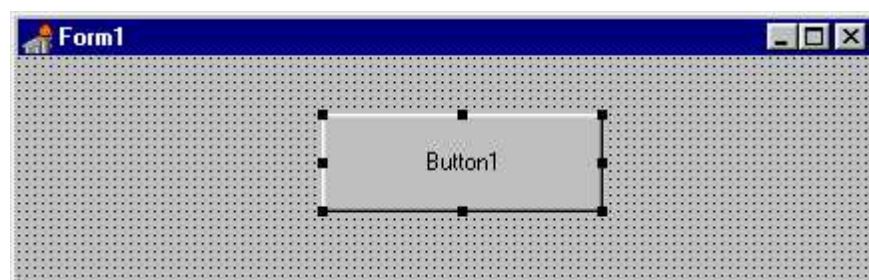
2°) Exécuter votre nouvelle application

3°) Enregistrer votre projet dans un nouveau répertoire

4°) Regarder les fichiers créés

Les VCL de DELPHI sont regroupés par pages: Standard, Supplément ..., et représentés par des icônes : *Regarder les différents composants qui sont à votre disposition.*

Mettre le composant "bouton" sur la feuille quadrillée appelée FORM1



3. PROPRIETES (de DELPHI)

PROPRIETE = Caractéristique descriptive d'un objet

Opérations sur les propriétés:

- Consultation (ou lecture) d'une propriété
- Modification (ou écriture) d'une propriété

Rq: certaines propriétés sont accessibles uniquement en lecture

Ex: Thermomètre

Propriétés

Température réelle

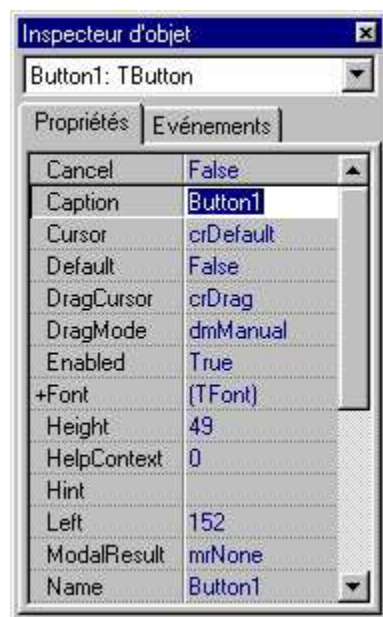
Température souhaitée

Opérations

Lecture

Lecture/Ecriture

Consulter les différentes propriétés du "bouton"



Rq: Ne pas confondre la propriété caption (=titre), et la propriété name (=Nom d'identification)

4. EVENEMENTS ET METHODES

EVENEMENT = Un fait qui se produit à un moment donné sous certaines conditions, pouvant déclencher des **actions**.

Ex: Lorsque la température ambiante dépasse la température souhaitée (= évènement), le chauffage est coupé (= action).

Un objet peut réagir à un ensemble d'évènements éventuels.

Consulter les différents évènements possibles d'un "bouton"

Ex : Evènement OnDragDrop: lâche le bouton après l'avoir fait glissé.

METHODE = Procédure (suite d'instructions) qui réagit à un évènement.

Cliquer 2 fois sur la case à droite de l'évènement OnClick

Il apparait la méthode TForm1.Button1Click. A l'exécution du programme cette procédure sera appelée dès que l'on cliquera sur le bouton Button1.

5. ENVIRONNEMENT DE DELPHI

5.1. Fenêtre Principale (DELPHI)

Comporte:

- +Menu classique
- +Barre de raccourcis à gauche
- +Palette de composants:
 - Standard : contrôles standards de Windows
 - Supplements : Contrôles supplémentaires classiques
 - Accès bases de données
 - Contrôle BD

...

5.2. Fiche (Form1)

La première fiche (Form1) sert à construire la fenêtre principale de notre application, en déposant dessus les différents composants de la palette.

5.3. Inspecteur d'objets

Permet de visualiser les différents objets composants la fiche, et la fiche elle-même.

Cliquer sur la fiche en haut à droite de l'inspecteur d'objets.

Pour chacun des objets on a leurs propriétés (modifiables) et les évènements pris en compte par l'objet.

5.4. Editeur de code (Unit1)

Il permet d'écrire les méthodes sous forme de procédures Pascal dans une Unité. Une fiche est associée à une unité.

Rq: On peut basculer de la fiche à l'unité par simple clic.

Un double clic sur un composant place le curseur sur la méthode réagissant à l'évènement OnClick du composant.

6. DEVELOPPEMENT D'UN PROJET

Ce fait en 5 étapes :

- 1°) Créer un projet
- 2°) Ajouter des composants
- 3°) Modifier leurs propriétés
- 4°) Ajouter des réactions aux évènements
- 5°) Exécuter
- 6°) Recommencer au 2°)

1°) Créer un nouveau projet (Créer un répertoire par application est obligatoire)

- Créer un répertoire**
- Cliquer sur "Fichier/Nouveau Projet"**
- Enregistrer**
- Exécuter** : On obtient une fenêtre vierge
- Stopper le programme**
- Réinitialiser le programme**

2°) Ajouter des composants

Ajouter un bouton

3°) Modifier les propriétés

- Modifier la valeur de la propriété** Caption: '**Attend**'
- Modifier le titre de la fiche** : '**Exercice 1**'
- Exécuter**
- Stopper**

4°) Ajouter des réactions à des évènements

- Cliquer sur l'onglet évènement de l'inspecteur d'objets**
- Double clique sur la case à droite de l'évènement** OnClick de button1
- Ajouter l'instruction:** button1.Caption:='Clic';

5°) Exécuter

Remarques:

- Tous les **contrôles** (=composants visibles à l'exécution) ont des propriétés indiquant leur position (Left, Top) et leur taille (Width, Height). Il est plus facile de modifier ces propriétés avec la souris par déplacement et déformation du contrôle, qu'en les changeant par des instructions Pascal.

- Certaines propriétés ne peuvent pas être modifiées directement par une affectation dans une instruction, mais il faut obligatoirement appeler une méthode qui le fera. L'objet est ainsi protégé des mauvaises manipulations.

EXERCICES

Pour tous les exercices, consulter l'aide en tapant F1, sur tous les composants, propriétés, et méthodes que vous utilisez.

Exercice 1



Afficher dans le ***LabelEtat*** le texte suivant :

- « Le bouton gauche vient d'être enfoncé » si on clique sur ButtonGauche
- « Le bouton droit vient d'être enfoncé » si on clique sur ButtonDroit

Exercice 2 :

- 1°)Réaliser un programme comportant 2 boutons:
- l'un contenant une valeur qui s'incrémente à chaque fois que l'on clique dessus
 - l'autre contenant 'RAZ', qui remet le 1er bouton à 0 par simple clic.



Rq: Utiliser les instructions

IntToStr : Transforme un Integer en un String

StrToInt : Fait l'inverse

2°) Modifier la propriété Enabled pour dévalider puis valider le bouton RAZ lorsque c'est nécessaire

Exercice 3

Ecrire une application qui fait le jeu 421 avec 3 dés.

Exercice 4

Ecrire un programme donnant pour un cercle, son rayon et son périmètre. Si l'utilisateur modifie la valeur du rayon,, le périmètre doit être mis à jour, et inversement.

Exercice 5

Ecrire une application de résolution d'équation du second degré

Exercice 6

1°) Ecrire une fonction qui retourne le nombre de voyelles dans un mot.

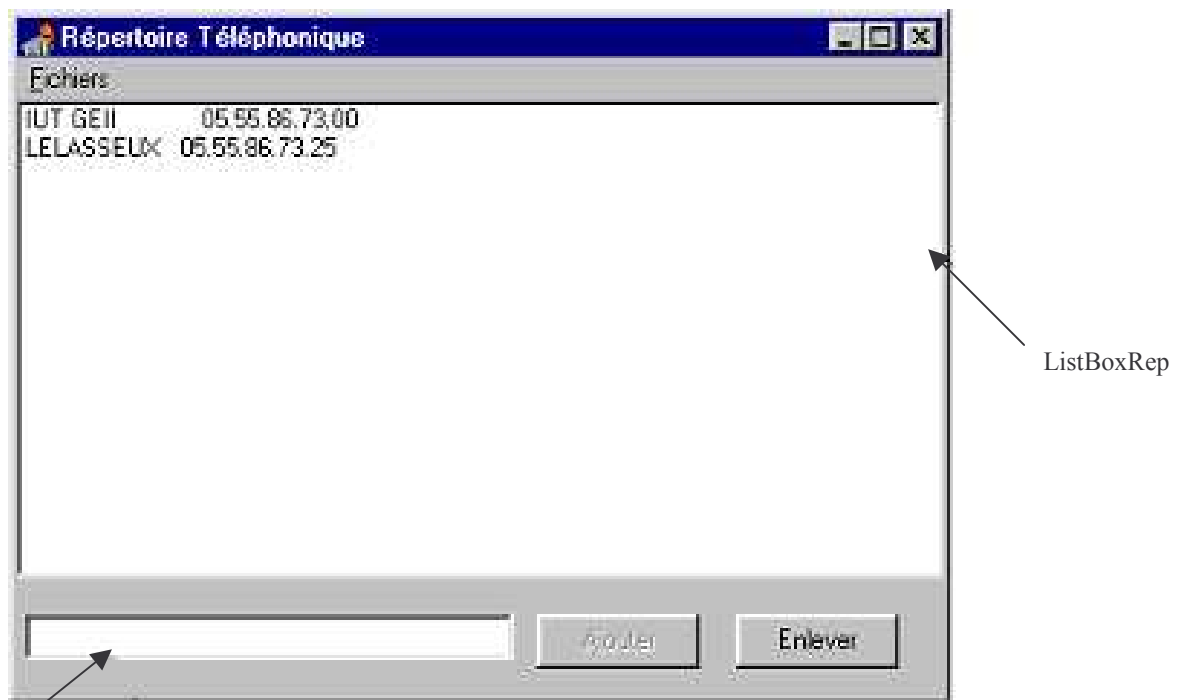
2°) Ecrire la fonction **Mot** qui retourne le premier mot d'une phrase donnée en paramètre, et supprime ce mot de la phrase.

Ex : si la phrase passée en paramètre est : « *Il fait beau et chaud* »,

La fonction mot retournera « *Il* »,

Et la phrase deviendra « *fait beau et chaud* »

Exercice 7 : Création d'un répertoire



EditLigne

Il s'agit de concevoir un répertoire téléphonique :

1°) Nommer les différents composants (Name) :

ListBoxRep (TListbox): Contient la liste des noms

On modifiera la propriété Align = Altop

Pourquoi ?

EditLigne (TEdit): Ligne où l'on tapera le nom et le numéro de
téléphone

Penser à effacer la propriété text

Pourquoi ?

ButtonAdd(TButton): Ajoute à la liste la personne saisie dans la ligne
d'édition.

ButtonSupp(TButton): Supprime de la liste la personne sélectionnée.

2°) On ne pourra pas cliquer sur le bouton ajouter si la ligne d'édition est vide.

3°) Ecrire l'action permettant d'ajouter le contenu de l'édition dans la liste

Propriété Items (=élément)

Consulter l'aide sur "Items"

Add (Méthode): Ajoute un élément dans une liste

Delete (Méthode): supprime l'élément sélectionné de la liste

Selected(Propriété): Booléen indiquant si un élément est sélectionné

Count(Propriété): indique le nombre d'éléments dans la liste

ItemIndex: Numéro de l'élément sélectionné

ItemIndex=0 pour le 1er élément
ItemIndex=-1 s'il n'y a aucun élément sélectionné

4°) Supprimer un élément

5°) Classer par ordre alphabétique la liste

6°) Remettre le texte de la ligne d'édition à vide après chaque ajout.

7. CREATION DE MENU

Deux phases:

- Conception du menu sous la forme de menu déroulant
- Ecriture du code réalisant les actions qui répondent aux commandes du menu

Nous allons créer un menu à partir de l'exercice 3 : Répertoire téléphonique.

7.1. Conception du menu

La conception du menu se fait à partir du composant MainMenu et de sa propriété Items qui permet de définir les sous-menus.

Ajouter sur la fiche du Répertoire Téléphonique le composant "MainMenu"



Rq: L'icône symbolisant MainMenu ne sera pas affichée à l'exécution, comme s'est le cas pour un bouton, mais une fenêtre classique de menu apparaîtra.

Cliquer 2 fois sur la propriété "Items"

Taper '&Fichier'

& permet de souligner la lettre qui le suit

Créer le menu suivant:



7.2. Répondre aux commandes du menu

1°) Quitter:

Pour quitter le programme l'instruction Pascal est: Close;

Réaliser la commande Quitter et vérifier son fonctionnement

2°) Enregistrer sous:

Cette commande devra enregistrer le répertoire créé dans un fichier . Classiquement sous Windows, une boîte de dialogue permet de choisir le chemin et le nom du fichier. Puisque cela existe sous Windows, nous pouvons l'utiliser dans DELPHI. C'est le composant SaveDialogBox qui se trouve dans la palette de composants **Dialogues**:



Double clic sur Enregistrer sous

Taper les instructions:

```
If Savedialog1.Execute Then  
    ListBoxRep.Items.SaveToFile(SaveDialog1.FileName);
```

Si la boîte de dialogue d'enregistrement de fichier s'exécute alors

Appeler la méthode de sauvegarde du fichier, avec comme paramètre le nom du fichier. Ce nom est celui saisi dans la boîte de dialogue.

3°) Ouvrir:

idem avec la boîte de dialogue d'ouverture de fichier:



Utiliser la propriété Filter permettant d'utiliser un filtre de fichier :

Fichier	Filtre
Texte	*.txt
Tous	*.*

Rq: Une application contient souvent les menus de manipulations de fichier (Ouvrir, Enregistrer, Enregistrer sous, Fermer, Chercher ...) d'édition (Couper, Copier, Coller, ...). A partir de DELPHI 2 la création de menu est simplifiée grâce à l'Expert Application qui propose ces principaux menus.