

1) Après le chargement de Delphi, changer la légende de la feuille *Form1* (propriété *Caption*).

Ajouter une étiquette (*Label*) sur la feuille et changer sa légende. Changer également la police de caractères (propriété *Font*).

Encadrer l'étiquette à l'aide d'un composant biseau (*Bevel*) de la palette de composants supplémentaires.

Télécharger l'image *delphi.bmp* à l'adresse <http://sceco-nte.univ-lyon2.fr/Darmont/meconometrie/> ; insérer un composant *Image* sur la feuille ; modifier sa propriété *Picture* pour utiliser l'image *delphi.bmp*.

Ajouter un composant forme (*Shape*) sur la feuille. Modifier sa propriété *Shape*. Passer le en arrière plan (bouton de droite de la souris) et encadrer l'image avec.

Regarder le code de l'unité associée à la feuille. Tous les éléments que vous y avez inséré y apparaissent.

Sauvegarder le projet (menu Fichier/Enregistrer le projet sous...) : unité, feuille et projet.

Exécuter le projet (F9).

Résultat attendu :



2) Créer un nouveau projet (menu Fichier/Tout fermer, puis Nouvelle application).

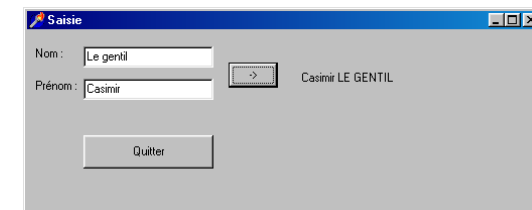
Disposer sur la feuille deux boîtes d'édition (*Edit*) qui vont servir à saisir le nom et le prénom d'une personne. Mettre leur propriété *Text* à vide et les nommer « Nom » et « Prenom ».

Ajouter une étiquette dans laquelle on placera la concaténation du prénom et du nom en majuscules. Nommer-la (propriété *Name*) « Resultat ».

Ajouter un bouton de commande (*Button*) nommé « Go » qui servira à placer la concaténation du nom et du prénom dans l'étiquette Résultat. Double-cliquer sur le bouton. La procédure *TForm1.GoClick* est automatiquement générée. Programmer l'affectation de la propriété *Caption* de « Resultat » à la concaténation des propriétés *Text* de « Nom » et « Prenom ». Utiliser la fonction *UpperCase* pour passer le nom en majuscules.

Ajouter un bouton nommé « ByeBye » et de légende « Quitter » servant à sortir de l'application (commande *halt*).

Résultat attendu :



Code :

```
procedure TForm1.GoClick(Sender: TObject);
begin
    Resultat.Caption:=Prenom.Text+' '+UPPERCASE(Nom.Text);
end;

procedure TForm1.ByeByeClick(Sender: TObject);
begin
    halt;
end;
```

3) Créer un nouveau projet (menu Fichier/Tout fermer, puis Nouvelle application).

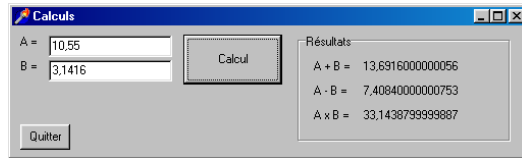
Placer sur la feuille deux boîtes d'édition nommées « A_chaine » et « B_chaine » qui serviront à saisir deux nombres réels A et B.

Ajouter une boîte de groupe (*GroupBox*) de légende « Résultats » et y placer six étiquettes : trois de légendes « A + B », « A – B » et « A x B », respectivement et trois nommées « Somme », « Difference » et « Produit » qui contiendront les résultats.

Ajouter un bouton « Calcul » permettant d'effectuer les calculs. Attention : les propriétés *Text* des boîtes d'édition sont des chaînes de caractères. Avant de les utiliser pour des calculs, il faut les convertir en réels à l'aide de la fonction *StrToFloat*. Inversement, avant de placer les résultats dans les propriétés *Caption* des étiquettes, il faudra convertir les réels obtenus en chaînes à l'aide de la fonction *FloatToStr*.

Ajouter un bouton « Quitter ».

Résultat attendu :



Code :

```
procedure TForm1.CalculClick(Sender: TObject);
var a, b, s, d, p: real;
begin
  { Conversion des chaînes dans a et b }
  a:=StrToFloat(A_chaine.Text);
  b:=StrToFloat(B_chaine.Text);
  { Calculs }
  s:=a+b;
  d:=a-b;
  p:=a*b;
  { Conversion des résultats dans les chaînes d'affichage }
  Somme.Caption:=FloatToStr(s);
  Difference.Caption:=FloatToStr(d);
  Produit.Caption:=FloatToStr(p);
end;
```

4) Créer un nouveau projet (menu Fichier/Tout fermer, puis Nouvelle application).

On souhaite effectuer le même traitement qu'à la question 3), mais en donnant cette fois le choix de l'opération à l'utilisateur.

Placer deux boîtes d'édition nommées « Ach » et « Bch » pour la saisie.

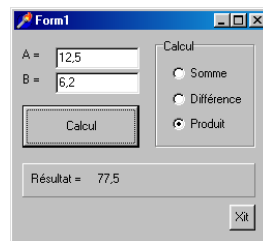
Ajouter un groupe de boutons radio (*RadioGroup*) et placer à l'intérieur trois boutons radio (*RadioButton*) nommés « Somme », « Différence » et « Produit ». Initialiser la propriété *Checked* de « Somme » à true.

Prévoir de mettre le résultat dans une étiquette placée sur un volet (*Panel*).

Ajouter un bouton « Calcul » et éditer le code correspondant.

Ajouter un bouton « Quitter ».

Résultat attendu :



Code :

```
procedure TForm1.CalculClick(Sender: TObject);
var a,b,r: real;
```

```
begin
  a:=StrToFloat(Ach.Text);
  b:=StrToFloat(Bch.Text);
  if Somme.Checked then r:=a+b;
  if Difference.Checked then r:=a-b;
  if Produit.Checked then r:=a*b;
  Res.Caption:=FloatToStr(r);
end;
```

5) Créer un nouveau projet (menu Fichier/Tout fermer, puis Nouvelle application).

On souhaite concevoir un programme qui, à partir d'un prix brut, calcul un prix final en tenant compte des contraintes suivantes :

- la TVA s'applique soit au taux normal (20,6 %), soit au taux réduit (5,5 %) ;
- le produit peut être vendu avec une marge de 30 % ou à prix coûtant (appliquer le surcoût sur le prix TTC).

NB : Quatre cas de figure peuvent donc se présenter.

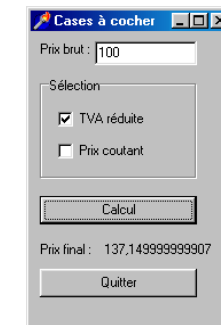
Placer une boîte d'édition nommée « Brut » pour la saisie du prix brut.

Utiliser deux cases à cocher (*CheckBox*) nommées « tva_reduite » et « prix_coutant » pour la sélection des options de taux de TVA et de marge.

Prévoir de mettre le résultat dans une étiquette nommée « Final ».

Ajouter un bouton « Calcul » et éditer le code correspondant.

Résultat attendu :



Code :

```
procedure TForm1.CalculClick(Sender: TObject);
var pf: real;
begin
  pf:=StrToFloat(Brut.Text);
  if tva_reduite.Checked then pf:=pf*1.055
  else pf:=pf*1.206;
  if not(prix_coutant.Checked) then pf:=pf*1.3;
  Final.Caption:=FloatToStr(pf);
end;
```

6) Créer un nouveau projet (menu Fichier/Tout fermer, puis Nouvelle application).

On souhaite gérer à l'écran deux listes d'éléments (chaînes de caractères) : les ajouts dans

chacune de ces listes, le passage d'un élément d'une liste à l'autre ou le transfert de tous les éléments d'une liste dans l'autre.

Placer une boîte d'édition nommée « Element » pour la saisie des éléments.

Ajouter deux boutons radio nommés « Bliste1 » et « Bliste2 » pour la sélection de la liste où ajouter l'élément.

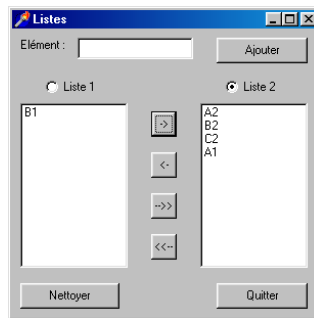
Ajouter un bouton « Ajouter » permettant cet ajout.

Ajouter deux boîtes de liste simples nommées « Liste1 » et « Liste2 ».

Ajouter quatre boutons nommés « Transfert12 », « Transfert21 », « Tout12 » et « Tout21 » permettant les transferts entre listes, dans les deux sens. Utiliser les propriétés *Items*, *ItemIndex*, et la méthode *Clear* des listes, ainsi que la propriété *Count* d'*Items* et les méthodes *Add* et *Delete*.

Ajouter un bouton « Nettoyer » permettant de réinitialiser les listes.

Résultat attendu :



Code :

```
procedure TForm1.AjouterClick(Sender: TObject);
begin
    if Bliste1.Checked then Liste1.Items.Add(Element.Text)
    else Liste2.Items.Add(Element.Text);
    Element.Text:='';
end;

procedure TForm1.NettoyerClick(Sender: TObject);
begin
    Liste1.Clear;
    Liste2.Clear;
end;

procedure TForm1.Transfert12Click(Sender: TObject);
begin
    Liste2.Items.Add(Liste1.Items[Liste1.ItemIndex]);
    Liste1.Items.Delete(Liste1.ItemIndex);
end;

procedure TForm1.Transfert21Click(Sender: TObject);
begin
    Liste1.Items.Add(Liste2.Items[Liste2.ItemIndex]);
    Liste2.Items.Delete(Liste2.ItemIndex);
end;
```

```
procedure TForm1.Tout12Click(Sender: TObject);
var i: integer;
begin
    for i:=0 to Liste1.Items.Count-1 do
        Liste2.Items.Add(Liste1.Items[i]);
    Liste1.Clear;
end;

procedure TForm1.Tout21Click(Sender: TObject);
var i: integer;
begin
    for i:=0 to Liste2.Items.Count-1 do
        Liste1.Items.Add(Liste2.Items[i]);
    Liste2.Clear;
end;
```