TP Listes

I) Création du projet

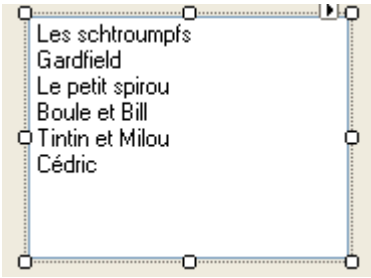
- Créer un nouveau projet C# nommé **Listes**.
- Renommer **Form1.cs** en **FormListes.cs**
- Donner la valeur **Exemple de listes** à la propriété **Text** du formulaire.
- Ajouter au formulaire un contrôle **ListBox** avec les propriétés suivantes:

Name	listBoxAnimaux
Location	64;21
Size	160;21

Les éléments d'une zone de listes sont membres de leur collection Items.
Chaque élément de la collection est repéré par un index, (l'index commence à 0)

II) Ajout d'éléments à la liste.

- Séléctionner la propriété **Items** de la **ListBox**, et cliquer sur le bouton Générer …
- Saisir des chaines de caractères, exemple:
- Valider par OK



Nous allons ajouter un élément à la liste grâce à du code:

- Ajouter au formulaire, un nouveau bouton avec les propriétés suivantes:

Name	buttonAjouter
Location	77;148
Size	130;23
Text	Ajouter un élément

- Créer le gestionnaire Click de ce bouton (par double-clic sur ce bouton) et compléter le avec le code:

```
private void buttonAjouter_Click(object sender, EventArgs e)
{
    listBoxAnimaux.Items.Add("Titeuf");

}
```

ATTENTION Cette valeur ne sera pas conservée lorsque le programme sera fermé.

- Compiler et tester le projet.
- Essayer la modification suivante:

```
private void buttonAjouter_Click(object sender, EventArgs e)
{
    //listBoxAnimaux.Items.Add("Titeuf");
    listBoxAnimaux.Items.Insert(0,"Titeuf");

}
```

Titeuf devient le n° 0 dans la liste, il s'affiche en début de liste.

Nous allons supprimer un élément de la liste grâce à du code:

- Ajouter au formulaire un nouveau bouton avec les propriétés suivantes:

Name	buttonSupprimer
Location	77;177
Size	130;23
Text	Supprimer un élément

- Créer le gestionnaire Click de ce bouton, et compléter le code de la façon suivante:

```
private void buttonSupprimer_Click(object sender, EventArgs e)
{
    listBoxAnimaux.Items.Remove("Titeuf");
}
```

La méthode **Remove()** parcourt la liste à partir de l'index 0, et supprime le 1° élément correspondant au texte, elle ne supprime que la 1° occurrence.

- Tester le programme en ajoutant plusieurs éléments "Titeuf" et en les supprimant.
- Modifier le gestionnaire de la façon suivante, et tester le projet

```
private void buttonSupprimer_Click(object sender, EventArgs e)
{
    //listBoxAnimaux.Items.Remove("Titeuf");
    listBoxAnimaux.Items.RemoveAt(3);
}
```

Nous allons effacer complètement le contenu de la liste grâce à du code.

- Ajouter au formulaire un nouveau bouton avec les propriétés suivantes:

Name	buttonEffacer
Location	77;206
Size	130;23
Text	Effacer la liste

- Créer son gestionnaire click et compléter le avec le code suivant:

```
private void buttonEffacer_Click(object sender, EventArgs e)
{
    listBoxAnimaux.Items.Clear();
}
```

- Compiler et tester le projet

Remarque: Les méthodes **Add(), Insert(), Remove(), RemoveAt(), Clear,** etc… sont des méthodes de la collection Items, et non de la zone de liste elle-même.

Les propriétés **SelectedItem** et **SelectedIndex:**

SelectedItem retourne le texte de l'élément sélectionné, un chaîne vide si aucun élément n'est sélectionné.

SelectedIndex retourne l'index de l'élément sélectionné, -1 si aucun élément n'est sélectionné.

- Ajouter au formulaire un nouveau bouton avec les propriétés suivantes:

Name	buttonAfficher
Location	77;231
Size	130;23
Text	Afficher la sélection

- Créer son gestionnaire click et compléter le avec le code suivant:

```
private void buttonAfficher_Click(object sender, EventArgs e)
{
    MessageBox.Show("Vous avez séléctionné " + listBoxAnimaux.SelectedItem
        + " dont l'index est de " + listBoxAnimaux.SelectedIndex);

}
```

Remarques:

Pour autoriser la sélection multiple d'éléments, il faut modifier la propriété SelectionMode par:

- soit la valeur **MultiSimple** (clic sur un élément modifie son état sélectionné)
- soit la valeur **MultiExtended** (sélection d'éléments à l'aide des touches Ctrl ou Maj)

Pour connaître les éléments sélectionnés, il faut utiliser **SelectedItems.**

La propriété **Sorted** permet de trier une liste.

- Tester le projet en donnant la valeur **true** à la propriété **Sorted** de la liste.

III) Ajout d'une ComboBox

A l'opposé de la ListeBox la ComboBox permet à l'utilisateur de saisir des valeurs.

- Redéfinir la taille du formulaire à 300;330.
- Placer un contrôle ComboBox avec les propriétés suivantes:

Name	ComboBoxCouleurs
Location	64;264
Size	160;21

- Séléctionner la propriété **Items** de la **ComboBox**, et cliquer sur le bouton Générer …
- Saisir des valeurs, exemple: Noir, Bleu, Or, Vert, Rouge, Jaune, valider par OK
- Compiler, tester le projet et la ComboBox, il est possible de saisir du texte…
- Pour empêcher la saisie du texte, arrêter le projet et modifier la propriété suivante:

DropDownStyle	DropDownList

- Compiler et tester le projet.

Remarques:

Le codage pour l'ajout, suppression, effacement d'éléments de liste est comparable à la ListBox.