

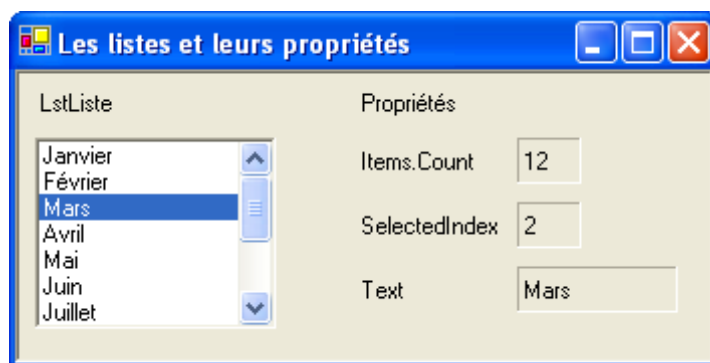
Listes diverses

➤ ListBox – Les listes et leurs propriétés

Objectif

Le but de cet atelier est de découvrir les fonctionnalités de bases communes aux différentes listes proposées dans le Framework.NET. L'exemple porte sur un contrôle de type ListBox.

Résultat attendu



Traitement effectué

En sélectionnant un élément dans la liste les 3 propriété (Items.Count, SelectedIndex et Text) sont automatiquement mises à jour.

Eléments utiles

Classe	Type	Membre	Description
ListBox	P	Items	Collection des éléments de la liste
Items	M	Add	Ajoute un élément à la collection Items
Items	P	Count	Nombre d'éléments dans la collection
ListBox	P	SelectedIndex	Index de l'élément sélectionné (à partir de 0)
ListBox	P	SelectedItem	Élément sélectionné
ListBox	P	Text	Texte correspondant à l'élément sélectionné
ListBox	E	SelectedIndexChanged	Lors de la sélection d'un élément

Si vous avez du temps

Essayer de découvrir d'autres membres intéressants de la classe ListBox susceptibles d'être utilisés selon vos besoins.

➤ Les différents types de Listes

Objectif

Mettre en œuvre les différents types de listes à notre disposition : ListBox, checkedListBox et comboBox.

Au chargement du formulaire, remplir les différentes listes :

ListBox : Janvier, Février, Mars, Avril.

checkedListBox : Mai, Juin, Juillet, Août.

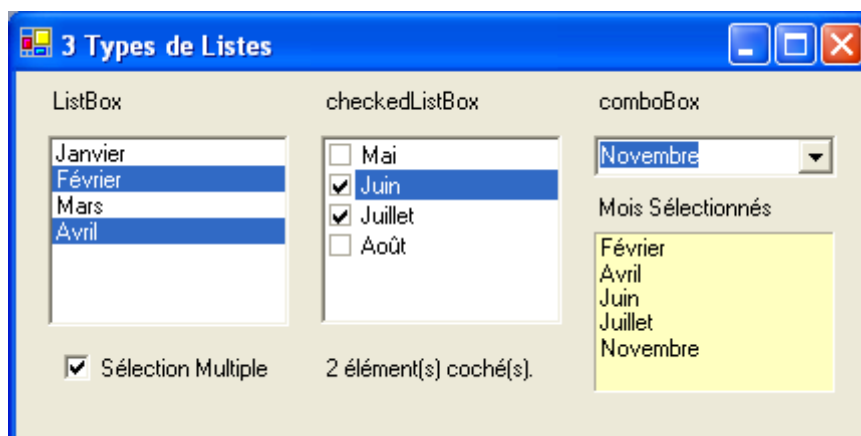
comboBox : Septembre, Octobre, Novembre, Décembre.

La case à cocher ‘Sélection Multiple’ permet selon son état de sélectionner un ou plusieurs éléments dans la ListBox.

Une étiquette indique en permanence le nombre d’éléments cochés dans la checkedListBox.

Une dernière étiquette récapitule les mois sélectionnés (cochés pour la checkedListBox) dans les trois listes. Cette information se met à jour en temps réel dès que l’utilisateur modifie la sélection d’une des trois listes.

Résultat attendu



Eléments utiles

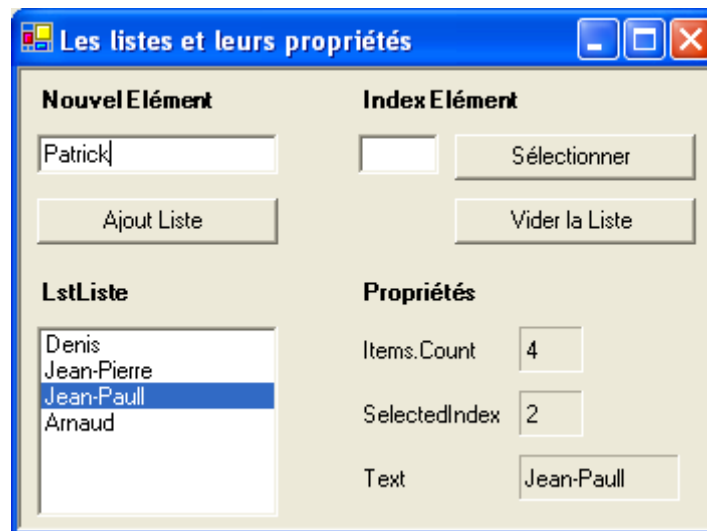
Classe	Type	Membre	Description
Tous contrôles	E	KeyUp	Touche relâchée
Tous contrôles	E	MouseUp	Souris relâchée
Toutes Listes	E	SelectedIndexChanged	Sélection modifiée
Toutes Listes	M	Add	Ajout d’éléments
Toutes Liste	P	SelectedItem	Éléments sélectionné
ListBox	P	SelectionMode	Sélection Unique, Multiple, Etendue
ListBox	P	SelectedItems	Collection des éléments sélectionnés
checkedListBox	P	CheckedItems	Collection des éléments cochés

➤ ListBox - Approfondissement

Objectif

Etre en mesure de réaliser les opérations principales concernant les contrôle de type ListBox.

Résultat attendu



Traitement effectué

Lors du lancement de l'application, la liste 'LstListe' est vide. Le peuplement de la liste s'effectue en saisissant des valeurs dans la zone 'Nouvel Elément' puis en pressant le bouton 'Ajout Liste'. Une fois un nouvel élément ajouté, le focus se repositionne sur 'Nouvel Elément' qui est remise à blanc.

Lorsque l'utilisateur sélectionne un élément, les propriétés affectées s'affichent à droite. Il est également possible de sélectionner un élément en saisissant son N° dans la zone 'Index Elément' et en pressant le bouton 'Sélectionner'.

Enfin, le bouton 'Vider la liste' comme son nom le laisse supposer remet la liste à blanc.

Eléments utiles

Classe	Type	Membre	Description
ListBox	P	SelectedIndex	Index de l'élément sélectionné (à partir de 0)
Items	P	Count	Nombre d'élément dans la collection
Control	M	Focus	Place le curseur (caret) sur le contrôle
int	M	Parse	Convertit la chaîne d'un nombre en nombre entier

Si vous avez du temps

Faites en sorte qu'un utilisateur maladroit ne puisse pas provoquer une erreur d'exécution de votre application. Exemple : s'il saisit un N° d'élément qui n'existe pas ou sous forme de texte.

Assurez-vous également qu'une valeur ne puisse pas être ajoutée deux fois dans la liste.

➤ Manipulation des Listes

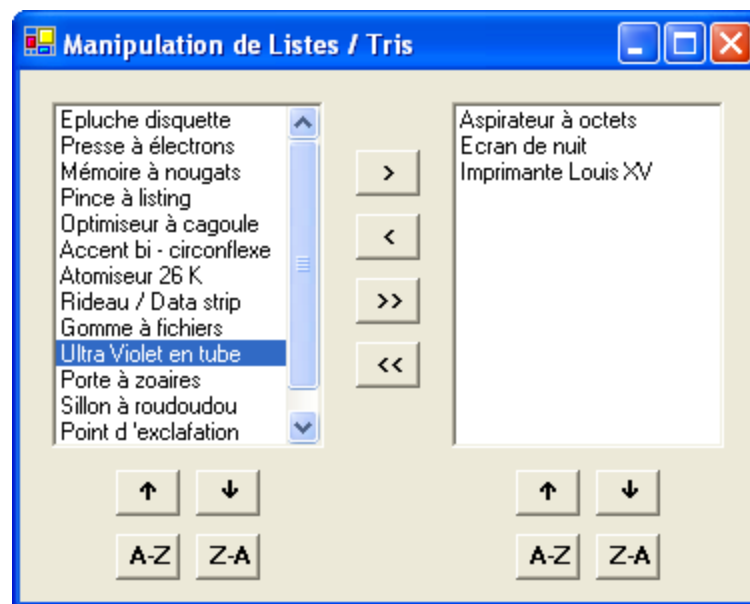
Objectif

Approfondissement des Listes

Créer une application permettant le déplacement d'éléments d'une liste dans une autre, ainsi que le tri de chacune des listes élément par élément.

En option, faire le tri de chacune des listes en ordre croissant et décroissant en une seule fois.

Résultat attendu



Traitement effectué

La liste de gauche s'alimente au chargement de la feuille. La liste de droite est vide au départ et est alimentée par les deux boutons '>' et '>>'.



Rôle des différents boutons :

- > Déplace l'élément sélectionné de la liste de gauche à celle de droite,
- < Déplace l'élément sélectionné de la liste de droite à celle de gauche,
- >> Déplace la totalité de la liste de gauche vers celle de droite,
- << Déplace la totalité de la liste de droite vers celle de gauche,
- ↑ Déplace l'élément sélectionné d'une ligne vers le haut (Tri Manuel),
- ↓ Déplace l'élément sélectionné d'une ligne vers le bas.

En option :

- A-Z Trie la liste correspondante par ordre alphabétique,
- Z-A Trie la liste correspondante par ordre inverse,

« Travaux Pratique »

Faites en sorte que les différents boutons ne soient accessibles qu'à bon escient. Exemple Si l'utilisateur sélectionne le premier élément d'une liste, le bouton  devra être désactivé, de même s'il sélectionne le dernier élément d'une liste, le bouton  sera désactivé.

Eléments utiles

Classe	Type	Membre	Description
ListBox	M	Items.Add	Ajoute un élément à la ListBox
ListBox	M	Items.Remove	Supprime l'élément spécifié
ListBox	M	Items.Clear	Efface tous les éléments de la liste
ListBox	M	Items.Insert	Insère un élément à une position donnée
ListBox	M	Items.AddRange	Ajoute une collection d'éléments à une liste

Conseils

Avant de vous lancer dans la programmation des différents tris, n'hésitez pas à réaliser un algorithme papier / crayon et à le 'faire tourner sur papier' à partir d'un jeu d'essai.

Bon Tri...