

## TP 4 - Insérer un module de calculs sur les complexes

### Objectifs :

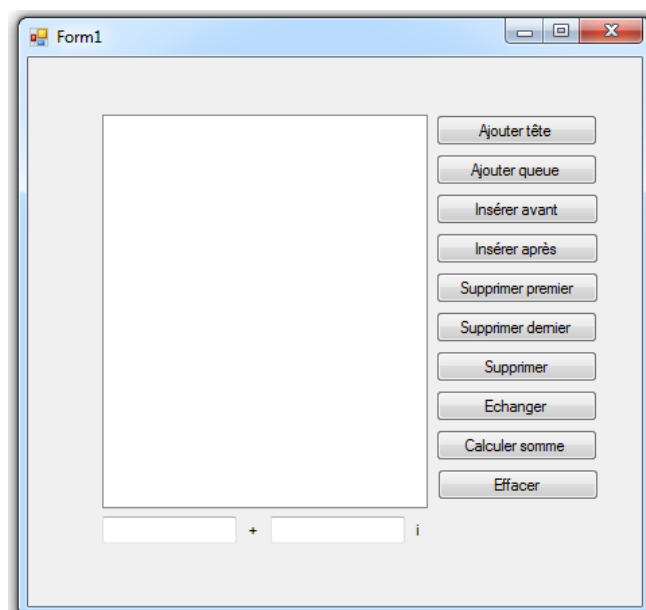
- Interface : Insérer une nouvelle feuille dans le projet.  
Utilisation d'une listBox pour l'affichage des données saisies, gestion des clics de l'utilisateur.
- Conception : Création d'un tableau indicé de complexes pour le stockage des nombres entrés par l'utilisateur.  
Réalisation des sous-programmes qui gèrent ce tableau (TD 2 et 3).
- Difficulté : Synchronisation du contenu du tableau et de la liste.  
Mise à jour d'un programme.

### Enoncé

- 1 – Créer une nouvelle feuille dans le projet (Nouvel élément, Windows Form)
- 2 – Insérer une ListBox, les zones de saisie et les boutons suivant l'image ci-dessous.
- 3 – Créer une structure pour les nombres complexes et une autre pour un tableau de complexes. Définir une variable globale qui contiendra le tableau de complexes synchronisé avec la ListBox.
- 4 – Après avoir renommé les boutons, créer les fonctions appropriées (ici on n'autorisera la sélection que d'un seul élément à la fois).
- 5 – Rendre actifs ou inactifs les boutons suivant la situation (liste vide, un élément sélectionné ou non, etc.).
- 6 – Ajouter un bouton Trier qui organisera la liste des complexes du plus petit au plus grand module.

### Remarques :

- 1 – Les principales fonctions des listBox sont accessibles par `lbx_complexes.items` (voir plus loin)
- 2 – Les indices d'une listBox commencent à zéro, supprimer un élément décale tous les suivants (il n'y a pas d'indice non utilisé entre le premier et le dernier élément).
- 3 – Une seule zone sert de saisie et de résultat.
- 4 – L'affichage des nombres complexes doit être la plus convenable possible "- 2i est préférable à 0 + - 2i par exemple".



Dans un deuxième temps, autoriser la sélection d'éléments multiples et modifier les fonctions nécessaires en ce sens (Seul le tri conservera l'ensemble des nombres complexes, sélectionnés ou non).

## Aide

Une listBox ListComplexes est un objet qui contient de nombreux attributs (ou propriétés).

Dans ce TP, voici ceux qui peuvent vous être utiles :

Dans les propriétés : SelectionMode : One (une seule sélection possible) ou MultiExtended (Sélection multiple autorisée, avec Maj pour sélectionner une plage et Ctrl pour sélectionner un à un les éléments).

ListComplexes.Items (type : Collection)	contient la collection (cf. plus bas) des éléments de la liste
ListComplexes.Items.Count	renvoie le nombre d'éléments dans la liste
ListComplexes.Items.Insert(id, item)	insère l'objet item à la position id
ListComplexes.Items.RemoveAt(id)	supprime l'élément à la position id (il n'y a alors plus d'élément sélectionné)
ListComplexes.Items.Clear()	supprime l'ensemble des éléments de la liste
ListComplexes.SelectedIndex (type : Integer)	contient l'indice de l'élément sélectionné dans la liste
ListComplexes.SelectIndices (type : Collection)	contient la collection des indices des éléments sélectionnés

Une collection est elle aussi un objet qui contient donc des attributs (ou propriétés) et sur laquelle peuvent s'appliquer des méthodes (sorte de fonctions):

ListComplexes.SelectIndices.Item(index) renvoie l'élément à la position index de la collection ListComplexes.SelectIndices

ListComplexes.SelectIndices.count renvoie le nombre d'éléments dans la collection ListComplexes.SelectIndices