

Première Séance

Etape1 : Structure d'une application MDI (multiple document interface)

MDI : « Les applications d'interface multidocument (MDI) vous permettent d'afficher simultanément plusieurs documents, chacun dans une fenêtre (vue) différente. Les applications MDI comportent souvent un élément de menu Fenêtre contenant des sous-menus qui permettent de passer d'une fenêtre ou d'un document à un autre »

1.1 Création de la fenêtre principale et des fenêtres enfants

- Création de l'application Windows Graphe,
- Modification du nom de la classe principal Formulaire en **EditGraphes** ; cette classe sera utilisée comme fenêtre principale en positionnant **IsMDIContainer**

« Propriété Form.IsMdiContainer [C#] : Obtient ou définit une valeur indiquant si le formulaire est un conteneur de formulaires enfants d'interface multidocument (MDI, Multiple Document Interface). »

- Création de la classe fenêtre enfant par l'ajout d'un formulaire Windows, la classe aura pour nom **VueGraphe**

1.2 Création du menu principal et du menu fenêtre enfant

- Ajout d'un menu au formulaire principal avec l'item Fichier (nouveau, quitter) et l'item Fenêtre (cascade, mosaïque verticale et mosaïque horizontale).
- Ajout d'un menu principal au formulaire fenêtre enfant avec comme un unique item Fichier (Fermer).

1.3 Traitants d'évènements associés aux items des menus

- Ajout des traitants d'évènements associés aux items : Nouveau, Cascade, Mosaïque Horizontale et Verticale, Fermer et Quitter.

Le choix par l'utilisateur de l'item nouveau provoquera la création d'une instance **VueGraphe** ; le formulaire s'affichera par l'appel à la méthode **show**. On positionnera correctement la propriété **MdiParent** afin de rattacher cette instance à la fenêtre principale, la propriété **Text** pour différencier les différentes vues en utilisant par exemple un compteur incrémenté à chaque création.

Note : « La propriété **MdiChildren** vous permet d'obtenir des références à tous les formulaires enfants MDI actuellement ouverts dans un formulaire parent MDI. Pour créer un formulaire enfant MDI, assignez à sa propriété **MdiParent** le **Form** qui doit constituer son formulaire parent MDI. Vous pouvez avoir recours à cette propriété pour effectuer une boucle sur tous les formulaires enfants MDI de façon à réaliser des opérations, telles que

l'enregistrement de données dans une base de données, lorsque le formulaire parent MDI est fermé ou pour mettre à jour des champs de ces formulaires enfants d'après les actions accomplies dans l'application. »

La fermeture d'un formulaire enfant sera réalisée par l'appel à la méthode **close** ; la fin de l'application par l'appel à la méthode **exit** de la classe **Application**.

La lecture de la documentation concernant la méthode **Form.LayoutMdi** vous permettra de réaliser les différentes réorganisations des formulaires enfants (cascade, mosaïque ...).

1.4 Ajout de la barre d'état à la fenêtre principale

On ajoutera une barre d'état (**StatusStrip**) à la fenêtre principale composée de deux zones : la première permettant d'afficher un message d'aide à l'utilisateur, la seconde affichant l'heure (heure qui pourra être mise à jour à l'aide d'un **timer**).