C# - FICHE PRATIQUE N° 5 – Solutions et compléments

Une petite gestion de personnel
Accès aux données centralisé, utilisation d'une relation.

A/ Un dataset commun

Cette « centralisation » des données peut sembler un peu complexe au premier abord. En fait, il suffit de bien intégrer la solution et de la systématiser. Il faut noter que les formulaires de l'application sont totalement indépendants de la source réelle des données et du SGBD utilisé.

> Dans le formulaire principal de l'application (x tables à gérer dans y formulaires) :

- Créer les objets d'accès aux données : une connexion, x DataAdapters, un DataSet.
- Remplir les DataAdapters dans le constructeur, et mettre en place la gestion des événements « RowChanged » et « RowDeleted » des x tables.
- Déclarer un objet de type formulaire pour chaque formulaire de l'application.
- Instancier ces objets formulaires dans le constructeur en utilisant le constructeur à un argument défini dans chacun d'entre eux.

Remarque : dans le cas où un formulaire nécessiterait un accès à d'autres objets concernant les données (un SqlCommand par exemple), il est préférable de lui passer en paramètre le formulaire principal lui-même, et de créer une propriété public fournissant un accès au DataSet à ces autres objets.

> Dans chaque formulaire utilisant un accès aux données :

- Ajouter un contrôle de navigation.
- Déclarer un objet de type « DataSet typé » (dbDs par exemple).
- Ajouter un constructeur paramétré par un objet « DataSet typé ». Ce constructeur doit appeler le constructeur sans argument, affecter l'objet « dbDs », initialiser les navigateurs à l'aide de cet objet et mettre en place les liaisons de données.

Remarque : on trouve vite plus rapide d'établir les liaisons de données par programme, d'autant plus qu'il devient possible de régénérer le groupe de données sans les perdre…

Un détail : certains étudiants auront oublié de refixer une valeur par défaut pour les champs « cadre » et « sexe »…

B/ Un troisième formulaire

Attention, l'ordre de chargement des tables du DataSet dans le constructeur de Fm_principal est maintenant impératif : les services avant les employés. Une autre solution consiste à demander la désactivation des contraintes avant le chargement (propriété « EnforceConstraints » du DataSet, voir l'application fournie).

Il est utile de réfléchir avant d'écrire le constructeur du formulaire. On peut établir le tableau suivant :

Contrôles	Propriétés	Comment obtenir les données ?
tb_code	Text	tp1_service.code
tb_designation	Text	tp1_service.designation
tb_numero	Text	tp1_service.serviceemploye.numero à initialiser dans une fonction surAjout lecture seule
tb_nom	Text	tp1_service.serviceemploye.nom
tb_prenom	Text	tp1_service.serviceemploye.prenom
cb_cadre	Checked	tp1_service.serviceemploye.cadre
gb_sexe rb_masculin rb_feminin		A traiter par programme fonction surDeplact pour afficher les données traitement du clic sur un bouton radio pour les mises à jour
tb_salaire	Text	tp1_service.serviceemploye.salaire
cb_service	SelectedValue	tp1_service.serviceemploye.sce la propriété dataSource de la liste est liée à la table service