

Application de la gestion des erreurs sur le tp Commerciaux

Nous allons mettre en oeuvre le mécanisme de gestion des exceptions présenté sur le site : <http://grand.patrice.free.fr/net/courscharp/exception/exception.htm>

- 1) L'utilisateur ne sélectionne pas un fichier valide à désérialiser (menu Fichier).

Sans gestion particulière, le programme plante ; une `SerializationException` n'a pas été gérée. Il suffit de mettre le code dans un bloc try :

```
private void ouvrirToolStripMenuItem_Click(object sender, EventArgs e)
{
    try
    {
        this.openFileDialog1.ShowDialog();
        nomFichier = this.openFileDialog1.FileName;
        this.leService = PersisteServiceCommercial.charge(nomFichier);
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}
```

- 2) L'utilisateur oublie de sélectionner le fichier et accède directement aux menus concernant les commerciaux

Dans ce cas, le service commercial n'a pas été initialisé par la désérialisation. Plutôt que de demander au gestionnaire d'erreurs de traiter le problème on peut se contenter ici de n'ouvrir les formulaires que si le service commercial n'est pas null.

```
private void ajouterToolStripMenuItem_Click(object sender, EventArgs e)
{
    if (this.leService != null)
    {
        FrmAjoutCommercial f = new FrmAjoutCommercial(leService);
        f.MdiParent = this;
        f.Show();
    }
    else
        MessageBox.Show("sélectionner un fichier au préalable");
}
```

- 3) Dans le formulaire d'ajout d'un commercial, l'utilisateur ne fait pas de saisies valides.

Ceci peut arriver dans deux cas :

- L'utilisateur oublie de saisir le nom ou le prénom
- L'utilisateur oublie de sélectionner une valeur de la liste déroulante

Application de la gestion des erreurs sur le tp Commerciaux

Dans ces deux cas, le programme plante bien sûr. Nous sommes là dans le cas d'une validation de saisie que nous allons traiter ainsi :

```
private void btnValider_Click(object sender, EventArgs e)
{
    try
    {
        if (this.lstPuissance.SelectedItem == null)
            throw new Exception("il faut sélectionner une puissance");
        int puissance = Convert.ToInt32(this.lstPuissance.SelectedItem.ToString());
        char c = 'A';
        if (this.rdBtB.Checked)
            c = 'B';
        if (this.rdBtC.Checked)
            c = 'C';
        if (this.txtNom.Text == "" || this.txtPrenom.Text == "")
            throw new Exception("les champs nom et prénom doivent être remplis");
        Commercial cm = new Commercial(this.txtNom.Text, this.txtPrenom.Text, puissance, c);
        leService.ajouterCommercial(cm);
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}
```

Cette solution est préférable à un *try* simple qui ne fera pas apparaître de manière explicite les raisons des erreurs et surtout à des validations classiques sur les contenus des champs qui utiliseraient des si imbriqués bien peu clairs et mélangeant le code fonctionnel et la gestion des erreurs.