1. Gérer les différents contextes d'affichage

Il n'est pas toujours aisé de gérer les différentes situations d'affichage associées au déroulement du processus de traitement des informations au sein d'une interface graphique.

Je vous propose ici une démarche qui vise à simplifier la gestion et la mise au point de ces différents contextes d'affichage.

Cette démarche autorisera une meilleure prévention des erreurs.

Illustration dans le cadre de la gestion des prêts d'une bibliothèque.

1.1. 1ère étape

Définir un type énumération qui regroupera les différents contextes.

Identifier les différentes situations à traiter.

Les définir comme items d'une énumération.

La liste des éléments de l'énumération pourra facilement évoluer par la suite au fur de la découverte de nouveaux contextes.

```
enum ContextesAffichage
{
    Initial,
    PretInitial,
    ExemplaireDetails,
    PretsListe,
    ExemplaireNonEmpruntable,
    ExemplaireEmpruntable
}
```

1.2. 2ème étape

Une méthode qui centralise la gestion des propriétés des contrôles fonction du contexte. Extrait ci-dessous.

Les méthodes SuspendLayout et ResumeLayout permettent d'éviter de trop fréquentes modifications de la surface d'affichage.
Il ne faut pas chercher à mutualiser le code commun à différents contextes mais privilégier la simplicité et la facilité de maintenance.

```
private void GererContextesAffichage(ContextesAffichage contexte)
{
    this.SuspendLayout();
    switch (contexte)
    {
        case ContextesAffichage.Initial:
            gbAdherent.Visible = true;
            gbPrets.Visible = false;
            break;
        case ContextesAffichage.PretInitial:
            gbPretDetails.Visible = false;
            pretDataGridView.Visible = false;
            gbPrets.Visible = true;
            btnAjouterPret.Visible = false;
            btnAnnuler.Visible = false;
            break;
        default:
            break;
    }
    this.ResumeLayout();
}
```

Il vous suffit ensuite d'invoquer la méthode de gestion des contextes avec la bonne valeur de l'énumération en fonction de l'étape et du contexte.

```
public FrmPret()
{
    InitializeComponent();
    GererContextesAffichage(ContextesAffichage.Initial);
}
```