

Formation C++
Tp : programme de saisie d'informations

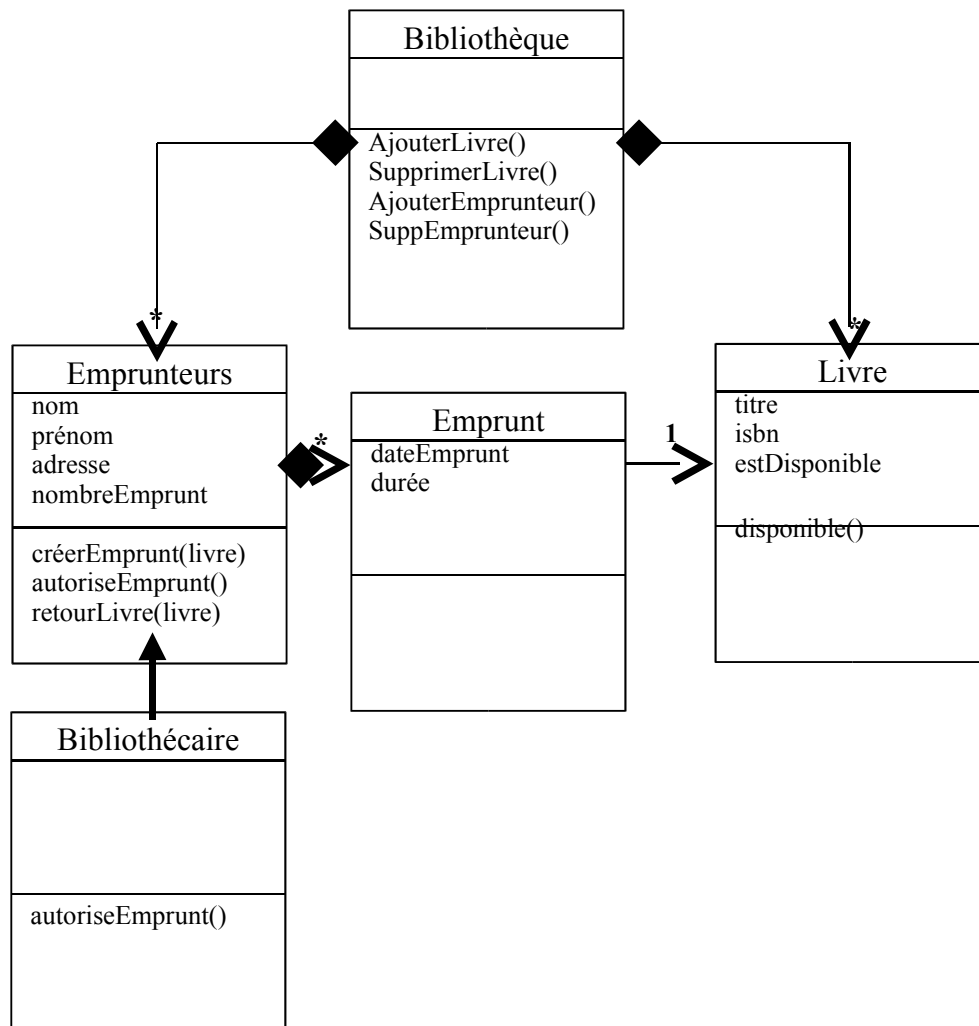
I. Objectif du Tp

Le but de ce Tp est de réaliser une application permettant la saisie et le contrôle d'informations. Nous utiliserons comme support une application de prêt de livre.

II. Modélisation

La bibliothèque possède des emprunteurs et des livres. Un abonné peut emprunter 5 livres en même temps, alors qu'un bibliothécaire peut en emprunter 10. Il n'existe qu'un seul exemplaire pour chaque livre.

D'où le modèle suivant pour représenter notre système :



La bibliothèque contient une liste de livres et une liste d'emprunteurs qui peuvent être des abonnés ou des bibliothécaires. Les listes seront gérées sous forme de tableaux.

1 – Ecrivez les classes Emprunteurs, Bibliothécaire, Emprunt, Livre et Bibliothèque.

Valider le fonctionnement à l'aide d'entrées/sorties clavier, terminal.

2 - Dessinez l'application permettant la saisie des informations suivant le modèle ci-dessous.

The screenshot shows a Windows application window titled "bibli". It has a standard Windows title bar with a close button. The main area is divided into two sections. The top section, labeled "Emprunteur", contains four input fields: "Nom" (with a dropdown arrow), "Prénom", "N°" (with a dropdown arrow), and "Cp". To the right of these fields are three buttons: "Ajouter", "Supprimer", and "Modifier". The bottom section, labeled "Livres", contains two input fields: "Titre" (with a dropdown arrow) and "ISBN". To the right of these fields are two buttons: "Ajouter" and "Supprimer". At the bottom center of the window are two buttons: "Emprunter" and "Quitter".

- Nommez les différents objets graphiques (*ALT+DoubleClick / Id*)
- Vous ajouterez à l'application des contrôles pour les objets devant être manipulés. (*CRTL+DoubleClick sur les objets*).
- Fixez l'ordre de déplacement par tabulation dans l'application à l'aide du menu « layout »/ « tab order ».
- Ajoutez les actions sur les boutons de l'application. (*CRTL+DoubleClick sur fenêtre principale*).
- Agrandissez la zone de liste des combo box en cliquant bouton gauche sur la flèche puis en agrandissant la hauteur du combo.

3 – Créez l'ajout et la suppression de livres. Le champ ISBN ne doit être composé que de chiffres et caractères '-'. Le livre ne doit pas déjà exister. Les champs sont vérifiés lors d'un click sur les boutons ajouter et supprimer. Lorsque l'utilisateur choisit un titre dans la liste, le champ ISBN doit être mis à jour.

Rq : Pour lire les données saisies dans une zone de texte, il faut utiliser :

```
Controleur_zone.GetWindowText( CString chaine);  
Controleur_zone est l'objet a lire.  
Chaine un objet où sera mis le résultat.
```

De même la méthode *SetWindowText(Cstring)* permet de modifier le contenu de la zone.

L'événement *OnSelChange* sur un objet de type combo permet d'être appelé lorsque la sélection faite par l'utilisateur change. Il est possible de connaître la nouvelle chaîne affichée par la méthode suivante :

```
Combo_controleur.GetLBText(Combo_controleur.GetCurSel(), CString);
```

Il est possible de créer une boîte de dialogue indiquant une erreur de la façon suivante :

```
MessageBox( « Message », « titre », MB_ICONERROR | MB_OK );
```

Ajout de livre

```
void CBibliDlg::OnLivAdd()
{
    // Verifie le format de la chaine
    CString isbn;
    m_emp_isbn.GetWindowText(isbn);
    for ( int i = 0 ; i < isbn.GetLength() ; i++ )
        if ( !(isbn[i] > '0' && isbn[i] < '9') && isbn[i] != '-' )
            break;
    if ( i < isbn.GetLength() ) {
        MessageBox(_T("Format Isbn incorrect"),
            _T("Isbn incorrect"),
            MB_ICONERROR | MB_OK );
    } else {
        // Format valide, on lance l'ajout
        CString titre;
        m_liv_titre.GetWindowText(titre);

        if ( myBib->ajouterLivre(titre,isbn) == false ) {
            // Creation d'une fenetre pour prevenir de
            // l'erreur
            MessageBox(_T("Impossible d'ajouter ce
livre"),
                _T("Titre incorrect"),
                MB_ICONERROR | MB_OK );
        } else {
            myBib->creerListLivre(&m_liv_titre);
            m_emp_isbn.SetWindowText("");
        }
    }
}
```

Mise a jour isbn sur modification titre

```
void CBibliDlg::OnSelchangeTitre()
{
    CString titre;
    m_liv_titre.GetLBText(m_liv_titre.GetCurSel(),titre);

    CString isbn = myBib->donnerIsbnFromTitre(titre);
    m_emp_isbn.SetWindowText(isbn);
}
```

Suppression de livre

```
void CBibliDlg::OnLivSup()
{
    // Recuperation du titre s'il en est un de sélectionné
    CString titre;
    if ( m_liv_titre.GetCurSel() != CB_ERR ) {
        m_liv_titre.GetLBText(m_liv_titre.GetCurSel(),titre);

        // Récupération de l'isbn associé
        CString isbn = myBib->donnerIsbnFromTitre(titre);

        // Suppression du livre de la base
        if ( myBib->supprimerLivre(isbn) == false ) {
            MessageBox(_T("Ce livre n'existe pas"),
                _T("Suppression impossible"),
                MB_ICONERROR | MB_OK );
        } else {
            // Mise a jour des éléments graphiques
            myBib->creerListLivre(&m_liv_titre);
            m_emp_isbn.SetWindowText("");
        }
    }
}
```

Création de la liste des livres pour le combo

```
void Bibliotheque::creerListLivre(CComboBox * cbox)
{
    // On commence par vider le combo
    cbox->ResetContent();

    // Maintenant on peut mettre a jour simplement
    for (int j=0 ; j < nbLivre ; j++ )
        cbox->AddString(listLivre[j]->getTitre());
}
```

4 – Réalisez la même chose pour les emprunteurs. Toutefois, plutôt que de faire les vérifications de syntaxe lors de l'appui sur un bouton, créez vos propres objets dérivant de la classe CEdit. Vous pourrez surcharger la méthode *OnKillFocus* appelée à chaque fois que l'élément est modifié.

5 – Ajoutez l'emprunt et la restitution de livres.

Exemple de vérification de données numériques

```
void NumEdit::OnKillfocus()
{
    CString num;
    this->GetWindowText(num);

    for ( int i = 0 ; i < num.GetLength() ; i++ )
        if ( !(num[i] > '0' && num[i] < '9') )
            break;

    if ( i < num.GetLength() ) {
        MessageBox(_T("Ce champ ne doit contenir que des chiffres"),
            _T("Syntaxe champ numérique !"),
            MB_ICONERROR | MB_OK );
        this->SetWindowText("");
    }
}
```