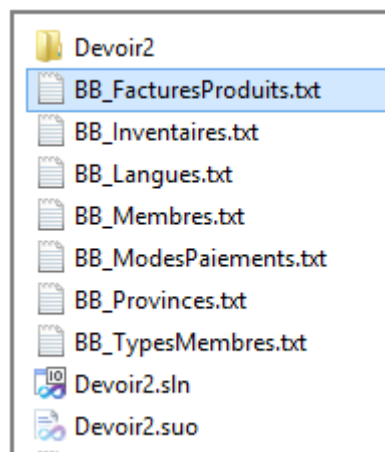


Devoir #02

SPÉCIFICATIONS

UTILISATION ET MANIPULATION DE COLLECTIONS PROGRAMMATION ÉVÉNEMENTIELLE UTILISATION DE TABLEAUX VALIDATION DE DONNÉES UTILISATION DE FICHIERS DE DONNÉES

- Sous **c:\Temp**, créer un nouveau répertoire de travail nommé **Devoir2**
- Télécharger le fichier de démarrage compressé **511_Devoir02_A2017_Demarrage.zip** d'**eCité** et déposez-le dans le répertoire **C:\Temp\Devoir2**.
- Décompressez le fichier de démarrage **511_Devoir02_A2017_Demarrage.zip** fourni et assurez-vous que son contenu est décompressé dans le répertoire **C:\Temp\Devoir2**.
- Cette décompression devrait avoir créé un sous-répertoire **Devoir2**, ainsi que les



fichiers suivants :

- Ouvrir le fichier de solution **Devoir2.sln**, Microsoft Visual Studio 2010 s'ouvre.

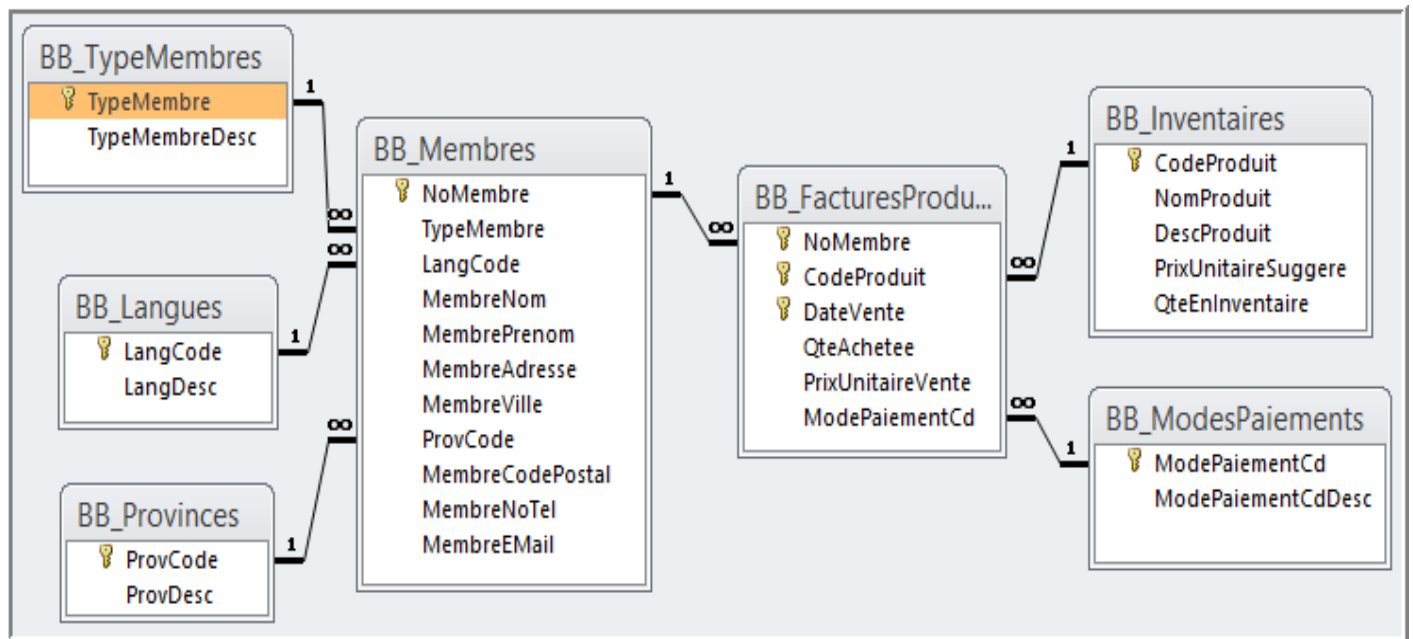
Devoir #02

INTRODUCTION

Dans cet exercice, vous allez expérimenter principalement l'utilisation de collections (chargement à partir de fichiers de données, lecture, mise-à-jour, suppression, et écriture des collections dans des fichiers délimités).

Vous expérimenterez **OBLIGATOIREMENT** avec la programmation en routines (Sub et fonctions) afin d'effectuer les validations et les divers traitements requis.

Les fichiers de données fournis sont liés comme suit :



- **Un membre peut avoir 0, une ou plusieurs factures de produits.**
- **Le membre habite une province, parle une langue et est catégorisé par type de membre**
- **Un membre reçoit des factures pour les produits achetés et la facture est payée selon un certain mode de paiement.**

Note #1 : Toutes les données contenues dans les fichiers fournis sont valides

Note #2 : Si des données sont modifiées sur la partie '**Membre**' du formulaire, lors de la mise-à-jour de la collection des **Membres** (Mise à jour les données - membre), vous devez vous assurer au préalable que les données sont validées et des messages appropriés (via MsgBox) sont affichés s'il y a lieu.

Devoir #02

Note #3 : Sur la partie '**Facture**' du formulaire, Seule la navigation est permise, les modifications ne seront pas sauvegardées.

Note #4 : Les données des collections **Types de membre**, **Provinces**, **Langues**, **Inventaires** et **Modes de paiement** serviront uniquement pour la validation des changements, alimenter les collections d'items des listes déroulantes (**ComboBox**) et pour affichage des descriptions dans les zones de texte (**TextBox**) désactivées.

Note #4 : Les structures des données des collections sont fournies :

```
Public Structure Membres
    Dim NoMembre As String
    Dim TypeMembre As String
    Dim LangCode As String
    Dim MembreNom As String
    Dim MembrePrenom As String
    Dim MembreAdresse As String
    Dim MembreVille As String
    Dim ProvCode As String
    Dim MembreCodePostal As String
    Dim MembreNoTel As String
    Dim MembreEMail As String
End Structure

Public Structure Provinces
    Dim ProvCode As String
    Dim ProvDesc As String
End Structure

Public Structure Langues
    Dim LangCode As String
    Dim LangDesc As String
End Structure

Public Structure TypesMembres
    Dim TypeMembre As String
    Dim TypeMembreDesc As String
End Structure
```

CONCEPTION DE L'INTERFACE

L'interface suivante est fournie et ne devrait pas nécessiter de changements. Vous noterez les champs désactivés. Ils doivent le demeurer.

Vous noterez aussi l'utilisation des contrôles '**MaskedTextBox**' pour faciliter et valider l'acquisition de certaines données. Noter aussi l'affichage de certains défauts.

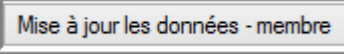
Devoir #02

- Soit le formulaire suivant qui vous est fourni (Affichage initial) : onglet **Membre** :

The screenshot shows a Windows application window titled "Devoir2". Inside, there is a tab labeled "Membre". The form contains the following fields and controls:

- # membre: Text input with "M" entered.
- Nom: Text input.
- Prenom: Text input.
- Adresse: Text input.
- Ville: Text input.
- Province: Dropdown menu showing "QC", followed by a text input and the label "← Nom de la province".
- Code postal: Text input.
- No de téléphone: Text input with a "(" prefix.
- Courriel: Text input.
- Langue: Dropdown menu showing "FR", followed by a text input and the label "← Description de la langue".
- Type de membre: Dropdown menu showing "MADU", followed by a text input and the label "← Description du type de membre".

Below the form, there are four navigation buttons: "<<", "<", ">", and ">>". To the right of these is a button labeled "Mise à jour les données - membre". At the bottom of the window, there are three buttons: "Charger les collections de données à partir des fichiers", "Sauvegarder les collections de données dans des fichiers", and "Fermer l'application".

- Dans ce formulaire, aucun des boutons n'est fonctionnel. Ils affichent seulement un message.
- Vous noterez qu'il n'y a aucun bouton pour ajouter ou supprimer des membres.
- Nous limiterons cet exercice à la navigation dans la collection et à la mise-à-jour des données sur les membres ().

PROGRAMMATION REQUISE :

Exécuter les étapes de programmation suivantes en testant entre chaque étape.

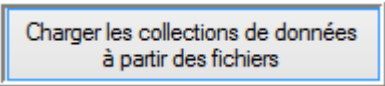
EXIGENCES QUI SERONT ÉVALUÉES : pour chacune de ces étapes, vous **DEVEZ** vous assurer de créer des **Sub** et/ou **function** réutilisables.

Devoir #02

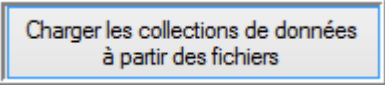
ÉTAPE #1 :

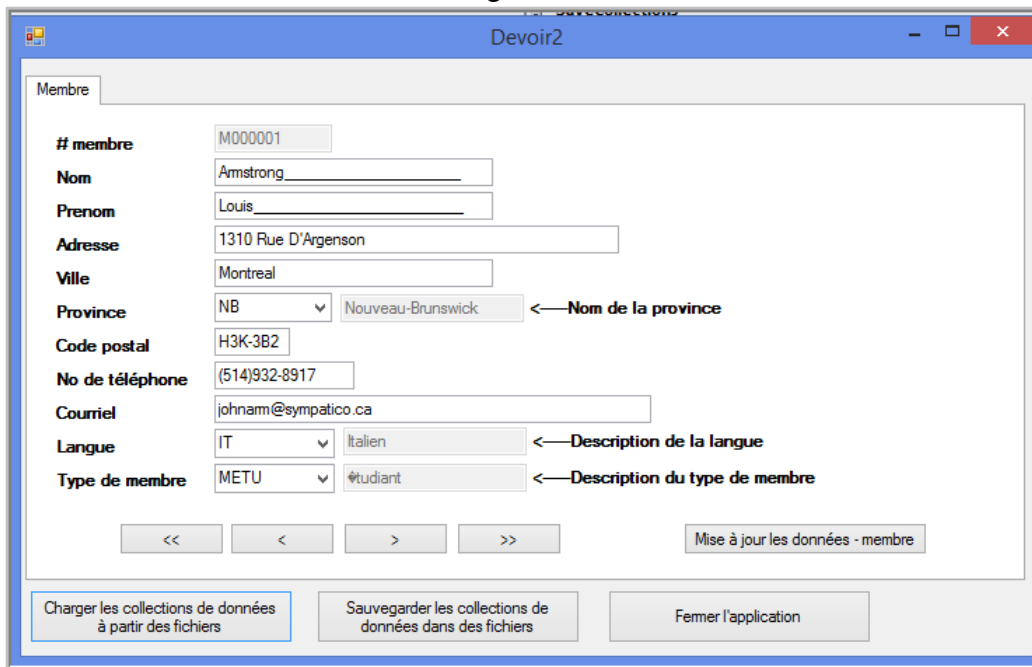
Ajouter le code pour charger les 4 collections de données ().

ÉTAPE #2 :

Modifier le code existant derrière le bouton  pour qu'à la fin du chargement des collections, on charge les 3 collections des **Items** des **ComboBox** du formulaire. (Province, Langue, Type de membre)

ÉTAPE #3 :

Modifier le code existant derrière le bouton  pour qu'à la fin du chargement des collections et des **Items** des **ComboBox**, on affiche le premier membre de la collection des membres sur l'onglet **Membre**,



ÉTAPE #4 :

Modifier maintenant le code existant pour que lors de l'affichage d'un membre, on affiche également le nom de la province, la description de la langue et la description du type de membre dans les zones de texte désactivées sur l'onglet **Membre**,

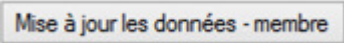
Devoir #02

ÉTAPE #5 :

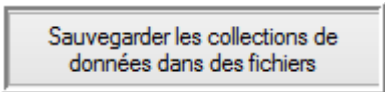
Si on clique sur un des boutons suivants () dans l'onglet **Membre** du formulaire, on doit afficher respectivement le 1^{er}, le précédent, le suivant ou le dernier membre de la collection des membres.

- N'oubliez pas qu'il n'y a peut-être pas de **Membre** précédent ou suivant.
- N'oubliez pas non plus toutes les initialisations des zones désactivées sur les 2 onglets.

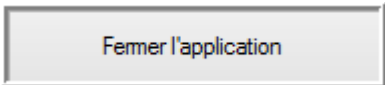
ÉTAPE #6 :

Si on clique sur le bouton suivant () dans la partie **Membre** du formulaire, on s'assurera de mettre à jour le membre dans la collection de données. Ne pas oublier de marquer la collection **Membres** comme ayant été modifiée.

ÉTAPE #7 :

Si on clique sur le bouton suivant () au bas du formulaire, si (et seulement si) la collection des **Membres** a été modifiée, on devra la sauvegarder en l'écrivant par-dessus le fichier existant. La collection des **Membres** est la seule qui doit être sauvegardées dans le cadre de ce devoir.

ÉTAPE #8 :

Si on clique sur le bouton suivant () au bas du formulaire, on ferme l'application. L'application ne fermera pas si la collection des **Membres** a été modifiée, et qu'elle n'a pas été sauvegardée.

À TRANSMETTRE :

- **Sauvegarder votre projet DEVOIR2**
- **Fermer VS2010**
- **Compressez le contenu (sous-dossier et tous les fichiers [en incluant les fichiers de données]) du dossier C:\Temp\Devoir2**
- **Déposez votre fichier compressé dans le pigeonier sur eCité avant l'échéance prévue.**

Bon Travail.