
TP 13

Les objectifs de ce treizième TP sont :

- la conception d'interfaces graphiques en XAML,
- l'utilisation de Data-Binding,
- l'utilisation de Data-Binding avec des UserControl.

OBJECTIFS

Refaire la même application WPF que le TP précédent, en exploitant au maximum les possibilités du Data-Binding.

Puisqu'il s'agit de votre premier travail utilisant du Data-Binding en XAML, voici quelques instructions pour vous guider...

STYLE

Ajoutez un `Style` qui s'appliquera à toutes les `TextBox` et `TextBlock` de vos fenêtres. Vous pourrez pour cela utiliser un Dictionnaire de Ressources. Amusez-vous à modifier les propriétés préparées par ce `Style`.

LISTBOX DES PERSONNES



Ajoutez une ressource à votre fenêtre principale correspondant à la propriété statique `Personnes` de la classe `Personne` de l'assemblage `giContacts`. Pour cela, vous pouvez utiliser l'élément suivant :

```
<x:Static x:Key="listContacts" Member="contacts:Personne.Personnes" />
```

où `listContacts` est la clé, et `contacts` le préfixe associé à l'espace de noms `clr giContacts`. Reliez la propriété `ItemsSource` de la `ListBox` des personnes à cette ressource statique.

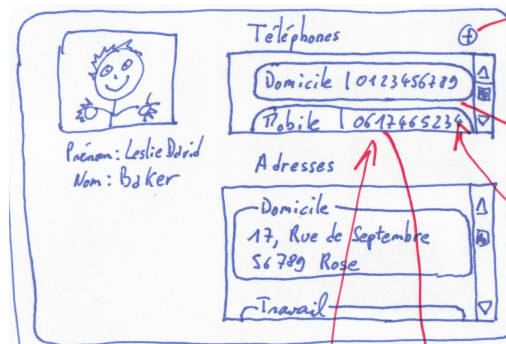
L'affichage des instances de `Personne` ne devrait pas être très beau... sans modifier la méthode `ToString` de `Personne`, modifiez la propriété `ItemTemplate` de la `ListBox` pour qu'elle affiche *Prénom NOM* de chaque personne, en testant les trois méthodes suivantes :

- `MultiBinding + StringFormat`,
- `MultiBinding + Converter`
- `Binding + Converter`.

Triez les personnes par ordre alphabétique de noms à l'aide d'une `CollectionView`.

DÉTAILS D'UN CONTACT

Utilisez un `ContentControl` pour stocker les détails du contact sélectionné.
Pour la suite, vous ne devez plus gérer l'événement `SelectionChanged` de la `ListBox`.



PHOTO, PRÉNOM ET NOM D'UN CONTACT

Reliez le prénom et le nom du contact aux `TextBlock` concernées en Binding.
Reliez l'image à la propriété `ImageSource` de la `Personne` sélectionnée à l'aide d'un `Converter`.

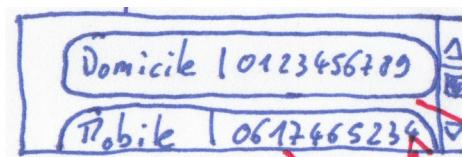
Rendez les nom et prénom modifiables (utilisez des `TextBox`). Que faut-il modifier dans le métier pour que l'affichage dans la `ListBox` soit mis à jour ?
(faites les modifications nécessaires...).



LISTE DE TÉLÉPHONES

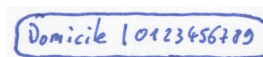
Modifiez la `ListView` de téléphones, en reliant en binding sa propriété `ItemsSource` à la collection de téléphones du contact sélectionné.

Modifiez la propriété `ItemTemplate` de la `ListView` pour utiliser le `UserControlTéléphone` en Binding pour l'affichage du numéro. Contastez que cela marche très moyennement...



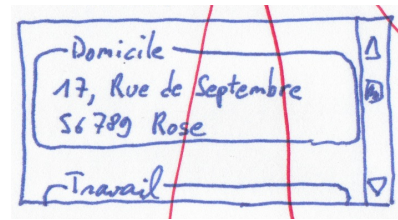
TÉLÉPHONE USERCONTROL

Modifiez le `UserControl`, et en particulier, transformez sa propriété Téléphone en `DependencyProperty`. Reliez le `UserControl` à la fenêtre en Binding et constatez que tout marche mieux...



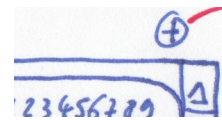
LISTE D'ADRESSES

Recommencez les deux dernières étapes mais cette fois-ci pour les adresses.



AJOUT ET SUPPRESSION D'UN NOUVEAU NUMÉRO DE TÉLÉPHONE

Lors de l'ajout ou de la suppression d'un téléphone lors du TP12, vous (...) avez dû non seulement mettre à jour la collection de téléphones du contact sélectionné, mais également la collection de



UserControlTéléphone de la ListView. À l'aide du *Binding*, ne faites

plus cette dernière mise à jour en *code-behind*. Que faut-il modifier dans le métier pour que cela marche ?