



## TP\_WPF : Création d'une application de voyages

### Objectifs :

- Mettre en œuvre la programmation objet en C#.
- Mettre en œuvre la création, la stylisation et la liaison de données d'une application WPF.

---

Le but de ce TP est de réaliser une application permettant de visualiser les voyages proposés par une agence. Elle permettra également de les modifier, supprimer ou d'en ajouter.

Pour développer cette application, le minimum de code sera écrit pour montrer et utiliser la puissance de WPF.

### Travail à réaliser :

## I/Partie 1 : ObservableCollection

1. Ajoutez à votre projet la classe représentée en annexe 1.

**\*\*\*Faites valider le code de votre classe\*\*\***

2. Compléter le fichier *MainWindow.xaml.cs* en ajoutant un attribut privé de type *ObservableCollection*, nommé voyages, initialisé à null.
3. Complétez le constructeur de cette classe avec le code fourni dans le fichier *constructeur.txt*.

## 2/Partie 2 : Binding simple

La vue de l'application sera constituée :

- d'un contrôle **ListBox** qui affiche le nom des voyages,
- d'un contrôle **GroupBox** qui affiche les différentes informations du voyage sélectionné (nom du séjour, pays, url du drapeau, description, date du prochain départ, nombre de jours, prix et indice de satisfaction). Ceci est réalisé en utilisant des contrôles **Label** pour indiquer le type d'information et des contrôles **TextBox** pour indiquer la valeur. Pour le contrôle GroupBox, réaliser un Binding sur le **DataContext** faisant référence à l'élément sélectionné du contrôle ListBox.

Il va de soi que :

- le texte de la description du voyage (comme les autres) doit toujours être lisible et ceci quelle que soit la taille de l'application,
- les prix doivent être indiqués en € avec le séparateur de millier français,



- les dates ne doivent pas préciser d'heure et doivent être indiquées en français.

**Il est conseillé de placer les contrôles dans une grille ce qui permettra un redimensionnement des contrôles lorsque la fenêtre sera elle-même redimensionnée.**

*Cette partie doit se faire sans ajouter de code C# (donc uniquement de l'écriture dans le fichier XAML).*

*Vérifier que les modifications réalisées sur les informations sont bien prises en compte dans l'attribut voyages.*

*Vérifier que lorsque le nom du séjour est modifié, la modification est immédiatement visible dans la liste.*

**\*\*\*Faites valider cette partie\*\*\***

## 2/Partie 2 : DataTemplate

Afin d'éviter de sélectionner un séjour pour connaître le pays, l'item du contrôle ListBox devra être modifié pour contenir le drapeau du pays, le nom du séjour et une représentation de l'indice de satisfaction. La modification se fera à l'aide des contrôles suivants :

- Un contrôle **ToolTip** affichera le nom du pays lorsque la souris passera sur cet item. Cela permettra d'aider ceux qui ne connaissent pas les drapeaux
- Un contrôle **ProgressBar** permettra d'afficher l'indice de satisfaction ; il est possible d'utiliser une autre solution : un contrôle **Rectangle** inclus dans un contrôle **ViewBox** lui-même inclus dans un contrôle **Border**. Il existe évidemment d'autres solutions que vous pouvez mettre en œuvre...

*Les différents indices de satisfaction doivent pouvoir être comparés et ceci est plus facile s'ils ont tous la même taille*

*Cette partie doit également se faire sans ajouter de code C# (donc uniquement de l'écriture dans le fichier XAML). ...*

**\*\*\*Faites valider cette partie\*\*\***

## 3/Partie 3 : Utilisation de Triggers

Afin de mettre en évidence les séjours particulièrement bien notés, il faudra créer des triggers qui vont :

- mettre en rouge et en gras le nom dans l'item du contrôle ListBox lorsque l'indice de satisfaction est de 5,
- mettre en gras le nom dans l'item du contrôle ListBox lorsque l'indice de satisfaction est de 4. Cette action doit avoir lieu dès que la note change.



*Cette partie doit se faire également sans ajouter de code C# (donc uniquement de l'écriture dans le fichier XAML)*

**\*\*\*Faites valider cette partie\*\*\***

## 4/Partie 4 : Utilisation du Convertisseur

Lors du lancement de l'application, aucun séjour n'est sélectionné mais il est possible de saisir des informations dans les contrôles TextBox. Il serait plus judicieux de les rendre inopérant tant qu'aucun séjour n'est sélectionné. Pour cela, il faut ajouter au projet une classe qui hérite de l'interface **IValueConverter** et qui permet à la méthode **Convert** de retourner *false* s'il n'y a aucun élément sélectionné et *true* sinon.

Complétez votre fichier XAML afin de ne valider les contrôles TextBox que lorsqu'il y a un élément du contrôle ListBox de sélectionné.

*Cette partie doit se faire sans ajouter de code C# dans le fichier MainWindow.xaml.cs*

**\*\*\*Faites valider cette partie\*\*\***

## 5/Partie 5 : Les règles de validation

L'indice de satisfaction doit être compris entre 0 et 5. Il doit être impossible de saisir une valeur qui n'est pas un nombre ou qui est un nombre hors de ces limites.

- Ajoutez une classe qui hérite de la classe **ValidationRule** et qui permet de vérifier que le texte saisi est bien un nombre et que ce nombre est bien compris entre 0 et 5. L'utilisateur devra être averti visuellement et textuellement de son erreur (style).
- Complétez le fichier XAML afin de valider l'indice de satisfaction.

*Cette partie doit se faire sans ajouter de code C# dans le fichier MainWindow.xaml.cs.*

Le prix est obligatoirement strictement positif. Sur le même principe,

- Compléter votre application pour obliger une saisie correcte et signaler une erreur de saisie.

## 6/Partie 6 : Le style

Maintenant que vous avez géré les fonctionnalités de votre application, il est temps de la modifier afin qu'elle devienne une belle application !

## 7/Partie 7 : Fonctionnalités supplémentaires (partie optionnelle)

Comme dans l'application fournie, vous pouvez ajouter les fonctionnalités suivantes (dans l'ordre que vous souhaitez) :

- Ajoutez un bouton **Désélectionner** qui désélectionne le séjour sélectionné et efface les informations des contrôles.



- Ajoutez un bouton **Ajouter** qui ajoute un nouveau séjour à la liste, ensuite remplit par les contrôles TextBox.
- Ajoutez un bouton **Supprimer** qui supprime le séjour sélectionné.
- Vérifiez que l'url du drapeau est valide,
- Ajoutez un zoom qui garde la même taille de fenêtre mais qui permet aux déficients visuels de mieux lire les différentes informations (balise **ScaleTransform**),

