



Conception Objet et Programmation C#,  
IUT département Informatique,  
1ère année, 2015-2016

## Programmation Objet en C#

# Petit Tutoriel n° Créer une Solution avec *Visual Studio*

Une application C#, qui pour nous correspondra à une *solution*, peut comporter plusieurs *modules*, ou *packages*, qui dans le vocabulaire propre à *Visual Studio*, peuvent prendre la forme de *projets* ou d'*assemblies*. De plus, des *namespaces* permettent, au niveau du langage, d'éviter les noms de classes ou de variables d'un module à l'autre. Par exemple, deux modules différents peuvent avoir une classe de même nom (comme une classe `Test` pour des tests unitaires de méthodes). Il n'y aura pas de confusion car les classes `Test` seront dans des *namespace* différents.

Dans l'exemple suivant, nous créons une solution *UrBooksLTD*, avec un projet éponyme (c'est du grec et ça veut dire "avec le même nom"), et comprenant un autre projet intitulé *Metier*, qui comprendra, comme son nom l'indique, les classes métier.

Voici, sur la figure 1, la boîte de dialogue obtenue par l'action *Fichier->Nouvelle Solution*, qui permet de créer notre solution *UrBookLTD*, de type *Application Console* en C#, et en choisissant son emplacement sur le disque dur.

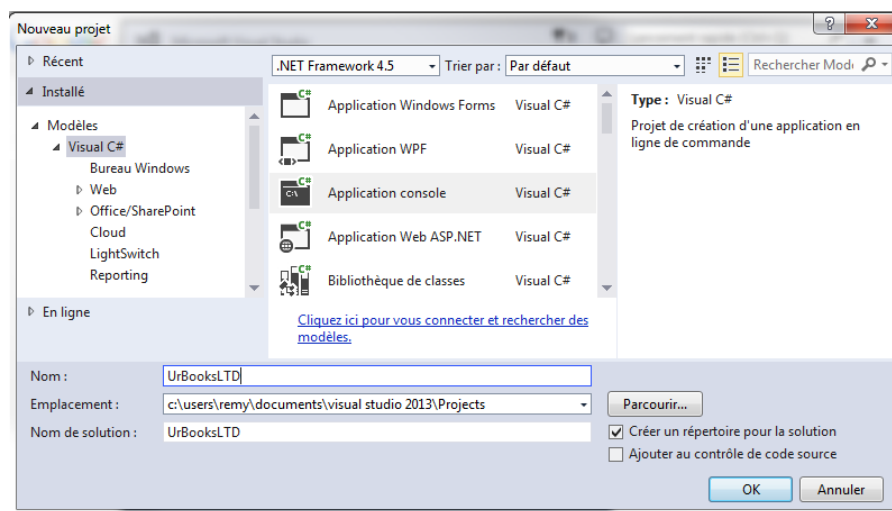


FIGURE 1 : Création d'une solution pour une application console *Visual C#*

Sur la figure 2, on peut observer l'état par défaut de la solution. Sur la figure 3, on visualise les propriétés de la solution, qui définit notamment la classe de démarrage, qui contient le `Main` à exécuter (ici `UrBookLTD.Program`), et la *namespace* par défaut.

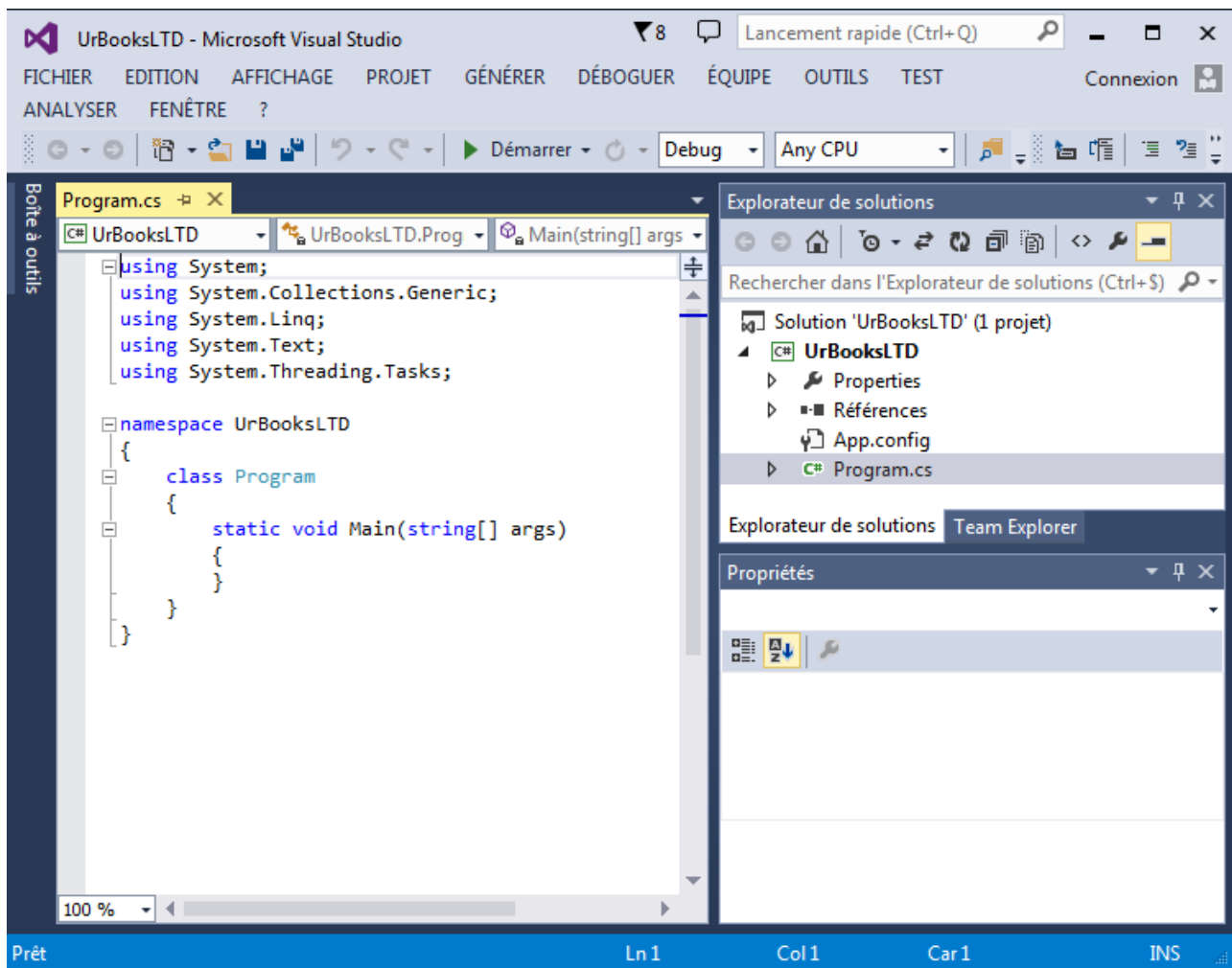


FIGURE 2 : La solution avec ses fichiers par défaut et la classe de test Program avec sa méthode Main.

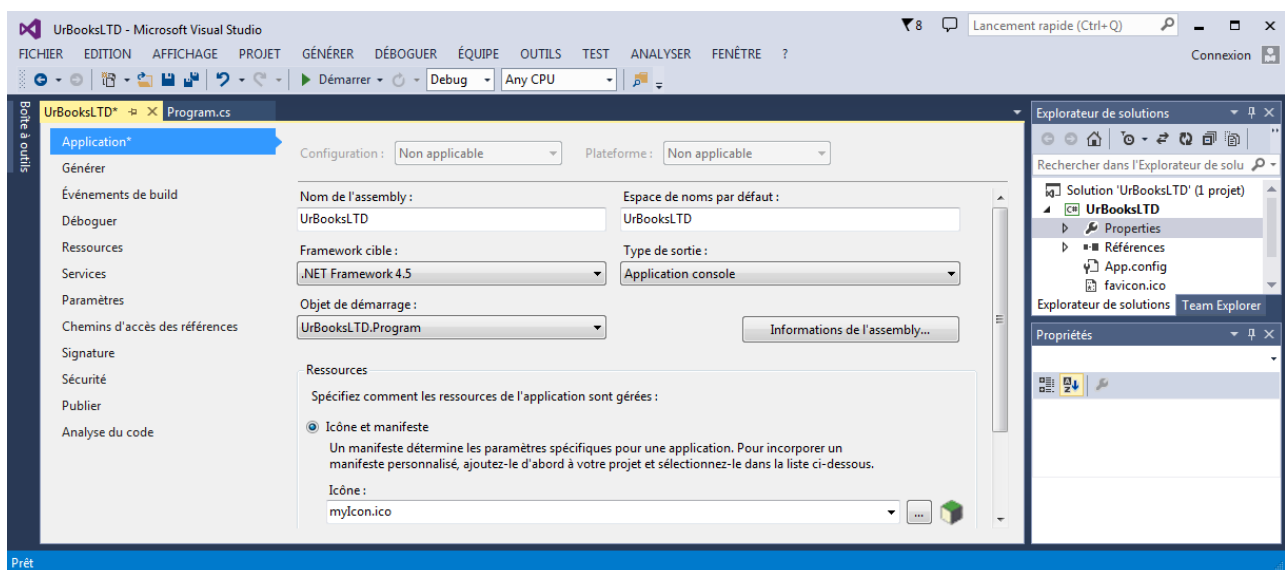


FIGURE 3

La figure 4 montre comment créer un nouveau projet, que l'on va ajouter à notre solution, et destiné à contenir nos classes métier. Ce projet ne contiendra pas de méthode `Main`, et il sera donc de type *Bibliothèque de Classes C# (DLL)*.

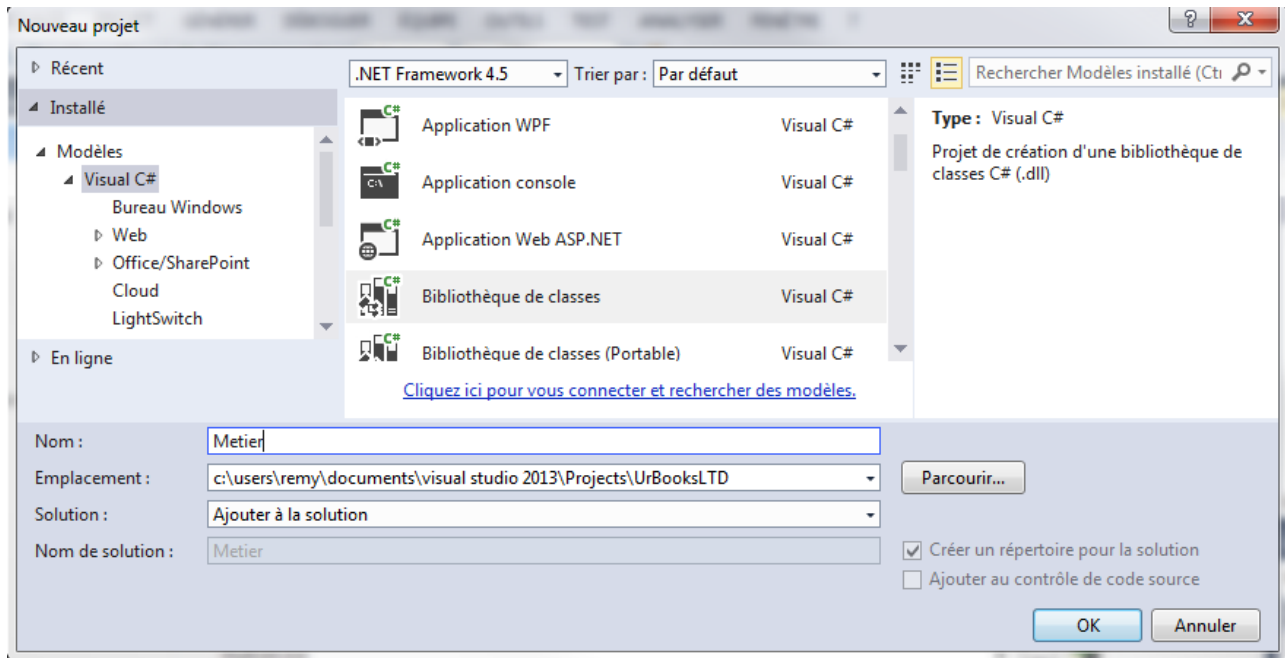
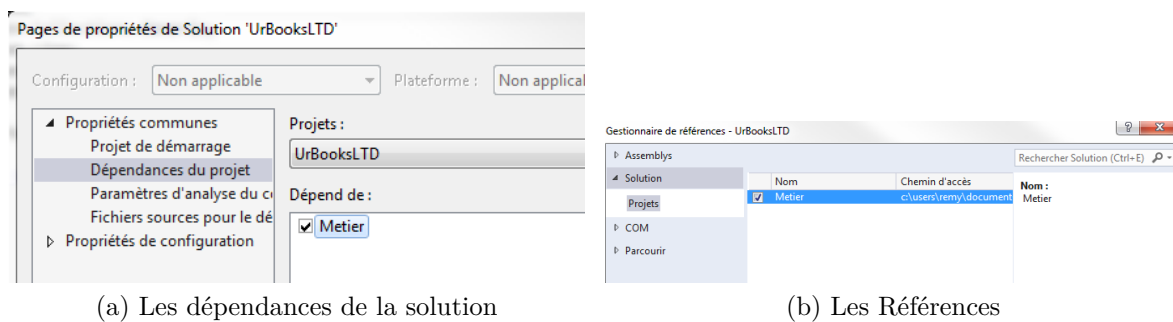


FIGURE 4 : Création d'un projet *Metier* pour les classes métier

La figure 5 montre comment, à la fois dans les dépendances et dans les références de l'application *UrBooksLTD*, nous définissons le lien avec le projet *Metier*, pour pouvoir utiliser les classes métier à partir du projet *UrBooksLTD*, et que le *namespace Metier* soit reconnu. Notons qu'il ne peut pas y avoir de dépendances cycliques entre les projets, et que, si besoin, nous utiliserons des techniques de réduction ou d'inversion de dépendances dans la conception objet de nos modules pour éliminer les éventuelles dépendances cycliques.



(a) Les dépendances de la solution

(b) Les Références

FIGURE 5 : Permettre l'utilisation des classes métier depuis le projet *UrBooksLTD* qui contient la méthode `Main`

L'aide de *Visual Studio*, très complète, permet très simplement, par simple *click* sur une méthode, ou encore par une recherche ou un index, de voir la documentation des *namespaces*, classes, interfaces ou méthodes du langage *C#*. Pour une utilisation *offline*, il faut avoir préalablement téléchargé les paquets correspondant de l'aide. Dans les options de l'aide, on choisit alors "Utiliser la Visionneuse d'Aide" (voir la figure 6).

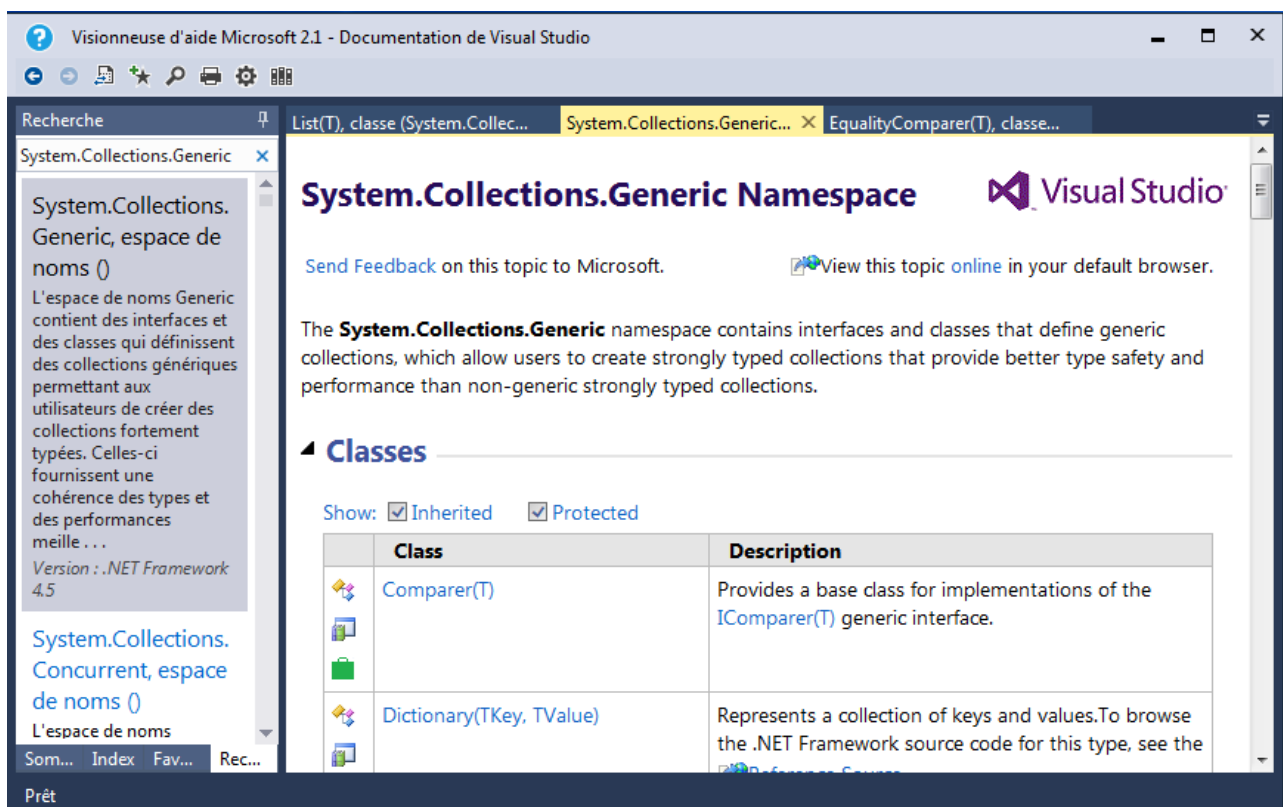


FIGURE 6 : La visionneuse d'aide de *Visual Studio* (ici, la documentation du *namespaces* des collections génériques)