

## Scénario 5 Jours :

### Jour 1 :

#### Matin :

- Présentation des participants :  
10h30
- Définir une proposition de TP "fil rouge" ! Si une copie de bdd "maison" est accessible.
- .net et .net framework  
.net : <https://fr.wikipedia.org/wiki/.NET> open source, avec un sous-ensemble d'API mais pas de designer. CLR et CIL  
<https://learn.microsoft.com/fr-fr/dotnet/core/whats-new/dotnet-6>  
.net framework : [https://fr.wikipedia.org/wiki/.NET\\_Framework](https://fr.wikipedia.org/wiki/.NET_Framework) application pour Windows CLI pour être utilisable avec tous les langages supportés. CLR : la machine virtuelle exécutant un bytecode CIL.  
Trois couches :
  - Base Class Library ( correspondrait au Win32 )
  - ADO : accès aux données.
  - API pour les GUIs : Application Web ( ASP.net) ou Application de bureau ( Windows Forms ).
- Découverte de l'environnement de développement API + IDE Visual Studio 2022 :  
[C:\Users\fm\Documents\CoursFormation\CoursCSCoreDotNet08112022\dotnet\\_cs.odp](C:\Users\fm\Documents\CoursFormation\CoursCSCoreDotNet08112022\dotnet_cs.odp)
- Présentation de l'IDE en créant une première application .net avec framework net4.8 :  
installation des différents modules.  
-> où se trouve l'exécutable.  
-> examiner : les fichiers générés modifiables, les fichiers régénérés à chaque build, les fichiers binaires, ... L'arborescence du projet.  
-> comment se passe le débogage.
- Même chose avec .net 6.0.
- Les divers types d'application avec VS22 et plus particulièrement en cs.
- C# 10

#### Après midi :

- Installer le concepteur de classes ( classDesigner ).
- découverte de la POO avec C#. Ajouter <partial> ( todo divers)

#### Classes et classes partielles

- Propriétés
- Interfaces
- Enums
- Structs
- Namespaces

#### Template

IntroPOO\_CS, IntroPOO1HasA, IntroPOO2AsA, IntroPOO2IsA, IntroPOODivers\_cs.  
Eventuellement PetitCoursAlgo\_cs

Jour 2 :

Matin :

- Suite POO avec C#.
- TP : Application -> début de mise en oeuvre du TP "fil rouge".

Après midi :

- Les assemblies. Les assemblies partagés sont dans <C:/Windows/Assembly>.
- Réflexion ( introspection ) sur des assemblies et des métadonnées.
- Boxing et unboxing : valeur -> objet ( implicite ) et objet -> valeur par conversion de type.
- Le CLI.net : <https://learn.microsoft.com/fr-fr/dotnet/core/tools/>  
Développement sans visual studio.

Evènements et delegates

- Lambda expressions
- Méthodes anonymes
- Méthodes d'extension
- **Generics**
- Exceptions
- Boxing
- Nullable types
- Finalisation

delegate\_cs, exception\_cs, petitCoursAlgo\_cs

Lambda expressions, Méthode anonymes ( todo ).

Jour 3 :

Matin :

passage d'argument en ligne de commande : argumentOS\_cs

GUI : UWP et .net Composants Windows\_cs ou NETApiGUI\_cs

Les types incontournables : Object, String...

- Gestion des nombres et dates
- Les collections d'objets
- Collections et généricité

Après midi :

Application au TP "fil rouge".

Jour 4 :

Matin :

Données persistantes :

- Les Fichiers : LesFichier\_cs et les IO-> voir CoursWindowsIOT08112022. (IO avec uwp,

DotNetRaspiAPI -> Blink avec C# et I2C)

- BDD. connexionsBDD\_cs
- ADO.
- DataView, source de données.
- LINQ.

– LINQ to Objects

Après midi :

Application au TP "fil rouge".

Jour 5 :

Matin :

- Client/Serveur.
- Déploiement.

Après midi :

Application au TP "fil rouge".