
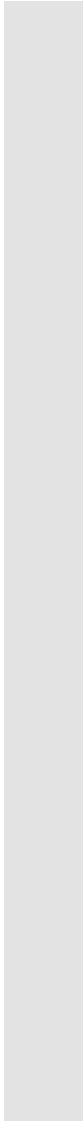


TP 1 – C# Basics

- 
- 
- Installation Environnement de travail
 - TP
 - Exercice 1
 - Exercice 2
 - Exercice 3
 - Tips / Helpers

- Visual Studio 2022 Community
 - Web
 - Database
- NetCore 7.X
 - <https://dotnet.microsoft.com/download>

Charges de travail

Composants individuels

Modules linguistiques

Emplacements d'installation

Web et cloud (4)



Développement web et ASP.NET

Générez des applications web en utilisant ASP.NET Core, ASP.NET, HTML/JavaScript ainsi que des conteneurs



Développement Python

Modification, débogage, développement interactif et contrôle de code source pour Python.



Développement Azure

Kits Azure SDK, outils et projets pour le développement d'applications cloud et la création de ressources à l'aide du



Développement Node.js

Générez des applications réseau scalables via Node.js, un runtime JavaScript piloté par des événements asynchrones.



Bureau et mobile (5)



Développement .NET Desktop

Générez des applications WPF, Windows Forms et console en C#, Visual Basic et F# à l'aide du .NET Core et du .NET



Développement pour la plateforme Windows universelle

Créez des applications pour la plateforme Windows universelle en C#, VB ou éventuellement C++.



Développement mobile en C++

Générez des applications multiplateformes pour iOS, Android ou Windows en C++.



Développement Desktop en C++

Générez des applications C++ modernes pour Windows à l'aide des outils de votre choix, notamment MSVC, Clang,



Développement mobile en .NET

Générez des applications multiplateformes pour iOS, Android ou Windows avec Xamarin.



Jeux (2)



Développement de jeux avec Unity



Développement de jeux en C++



Modification de Visual Studio Enterprise 2019 - 16.8.2

Charges de travail

universelle en C#, VB ou éventuellement C++.



Développement mobile en C++

Générez des applications multiplateformes pour iOS, Android ou Windows en C++.



Jeux (2)



Développement de jeux avec Unity

Créez des jeux 2D et 3D avec Unity, un puissant environnement de développement multiplateforme.



Développement de jeux en C++

Utilisez toute la puissance du C++ pour générer des jeux professionnels basés sur DirectX, Unreal ou Cocos2d.



Autres ensembles d'outils (6)



Stockage et traitement des données

Connectez, développez et testez des solutions de données avec SQL Server, Azure Data Lake ou Hadoop.



Applications de science et analyse des données

Langages et outils permettant de créer des applications de science des données, notamment en Python et F#.



Développement d'extension Visual Studio

Créez des composants additionnels et des extensions pour Visual Studio, notamment de nouvelles commandes, des



Développement Office/SharePoint

Créez des compléments Office et SharePoint, des solutions SharePoint, ainsi que des compléments VSTO en C#, VB et



Développement Linux en C++

Créez et déboguez des applications s'exécutant dans un environnement Linux.



Développement multiplateforme .NET Core

Générez des applications multiplateformes en utilisant .NET Core, ASP.NET Core, HTML/JavaScript, ainsi que des



Exercice 1

- Créer un projet "Classroom" en mode console
- Créer une classe Student représentant un étudiant
 - Nom / Prénom
 - Promo
 - Date de naissance
- Création d'un étudiant depuis la console via Readline()
- Affichage d'un étudiant en particulier via console
- Affichage de tous les étudiants créés ordonnées par ordre alphabétique ascendant OU descendant selon la selection de l'utilisateur.

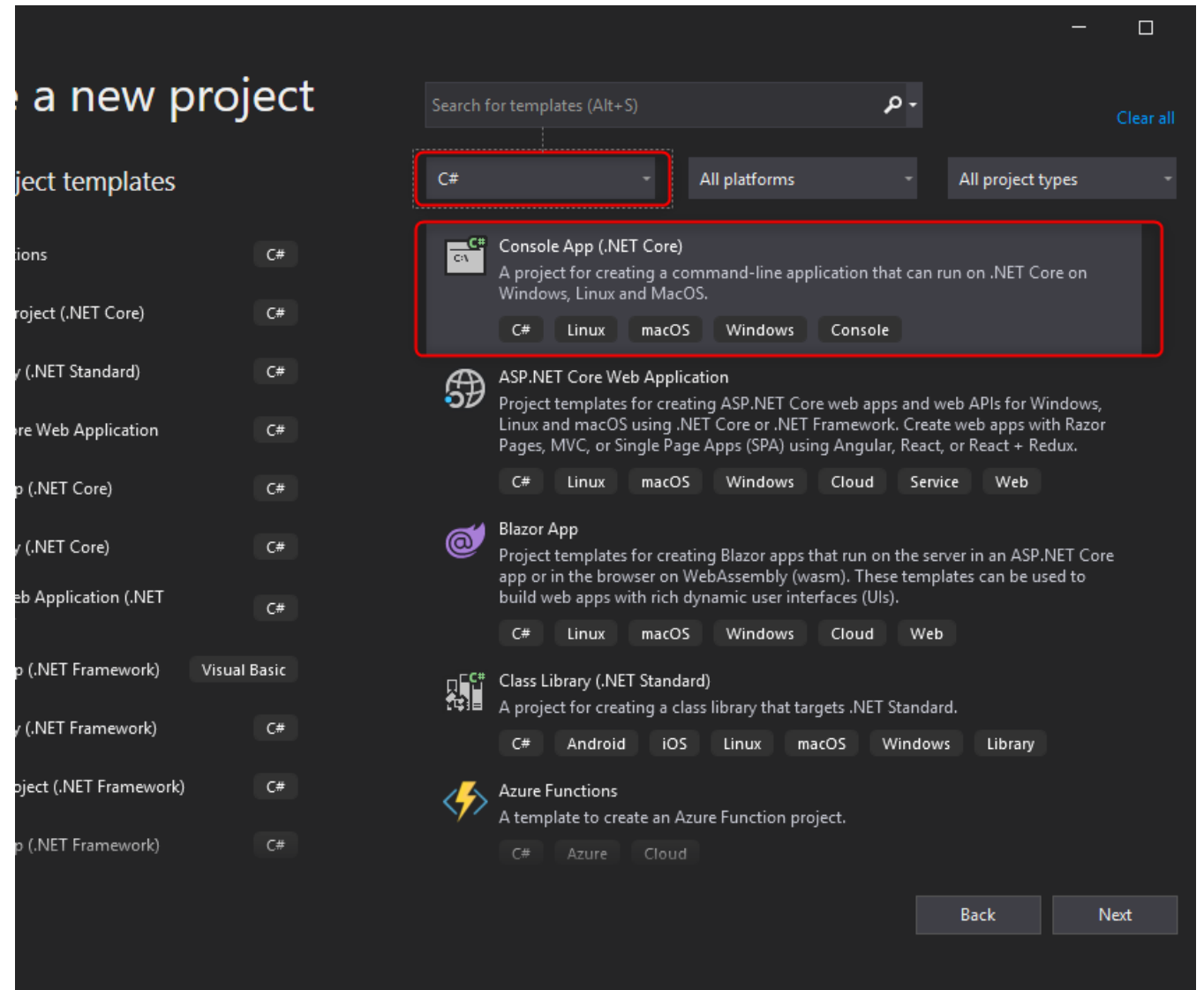
Exercice 2

- Créer une classe Lecture représentant un cours
 - Nom
 - Professeurs
 - Salles de cours associés
 - Liste des étudiants participants au cours
- Création du cours via console
- Affichage via la console
- Associer Students a un cours via la console

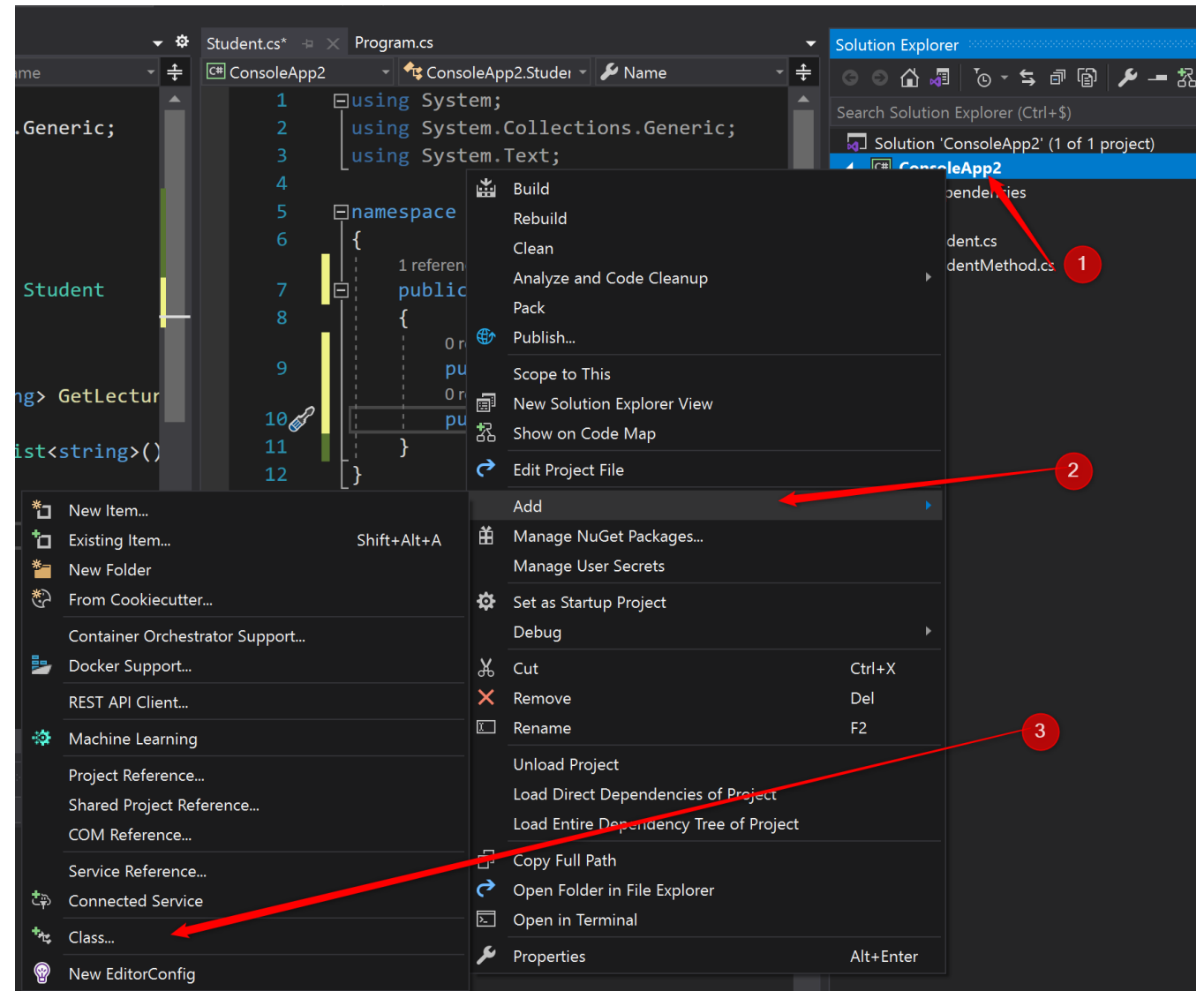
Exercice 3

- Ajouter une salle de cours
- Un cours doit avoir une salle, avec des étudiants.
- Affichage via la console de toutes les informations ordonnées

Création d'un projet Console .NetCore

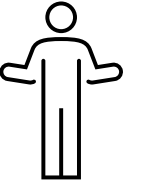


Création d'une classe



Utilisation Console Readline() / WriteLine()

N'oubliez pas un Readline() en fin du main afin que l'application ne se ferme pas sans votre accord



```
static void Main(string[] args)
{
    Console.WriteLine("Hello World!");

    Console.WriteLine("Choose your option");
    foreach (string menuItem in _menu)
    {
        Console.WriteLine(menuItem);
    }
    string option = Console.ReadLine();
    Console.WriteLine($"you choose {option}");

    Console.ReadLine();
}
```

```
C:\Users\pierl\source\repos\
Hello World!
Choose your option
1 - Student
2 - Lecture
1
you choose 1
_
```

Liens

MSDN : <https://docs.microsoft.com/>

- <https://docs.microsoft.com/fr-fr/dotnet/api/system.linq.enumerable.orderby?view=net-5.0>

Stackoverflow n'est pas toujours la bonne solution.

- 1er choix : MSDN
- 2eme : Blog / article de personnes MVP Microsoft
- 3eme : StackOverflow