

Série de TP N° 1 : Syntaxe de base du langage C# / POO avec C#

Partie abordées dans cette série de TP :

- Syntaxe de base du langage C#
- Les tableaux
- Manipulation des chaînes de caractères
- La POO avec C#

Travail à faire

EX 1 :

Écrire un programme en C# qui permet de trouver la longueur d'une chaîne sans utiliser les fonctions de la bibliothèque C#

EX 2 :

Écrire un programme en C# qui permet d'afficher les caractères individuels d'une chaîne dans l'ordre inverse.

EX 3 :

Créer un programme en C# permettant d'ajouter dans un tableau d'entiers de taille N, un nombre X à la position i, et ceci sans écraser les valeurs existantes dans le tableau.

N, X et i sont saisies par l'utilisateur.

Vous pouvez remplir le tableau par des valeurs aléatoires avec un code du genre :

```
int valeurATrouver = new Random().Next(0, 100);
```

EX 4 :

Créer un projet C# de type Console dans lequel il faut créer les deux classes **Personne** et **Étudiant** définies ci-dessous :

Classe Personne		
Nom attribut ou méthode	Type	Description
numero	int	
nom	string	
prenom	string	
Propriétés (getters et setters) des attributs		

ToString		Surcharge de la méthode ToString
----------	--	----------------------------------

La classe **Étudiant** hérite de la classe **Personne**.

Classe Étudiant		
Nom attribut ou méthode	Type	Description
filiere	string	
notes	Tableau float	
Propriétés (getters et setters) des attributs		
CalculMoy		Calcule la moyenne des notes de l'étudiant, on suppose que tous les modules ont le même coefficient

Dans la classe Program, faire ce qui suit :

- ✓ Créer un tableau de 5 étudiants
- ✓ Renseigner les informations de tous les étudiants
- ✓ Affecter 5 notes aléatoires à chaque étudiant
- ✓ Afficher la moyenne de chaque étudiant

Toujours dans le même projet, ajouter les deux constructeurs suivants à la classe **Étudiant** :

- ✓ Un constructeur par défaut initialisant à zéro / vide tous les attributs d'un étudiant.
- ✓ Un constructeur acceptant en paramètres : le numéro, nom et prénom d'un étudiant

EX 5 :

Toujours dans le même projet, créer la classe Filière définie ci-dessous :

Classe Filière		
Nom attribut ou méthode	Type	Description
nomFil	string	Le nom de la filière
listeEtudiants	List	
Propriétés (getter et setter) de l'attribut nomFil		
Constructeur par défaut		
GetNbrEtudiants		Retourne le nombre d'étudiants dans la filière
AddEtudiant(Etudiant e)		Ajout l'étudiant à la filière
Chercher(Etudiant e)		Cherche l'étudiant e dans la filière par nom et prénom
DeleteEtudiant(Etudiant e)		Supprime l'étudiant e de la filière
AfficherListeEtudiants		Affiche la liste de tous les étudiants de la filière

Tester votre projet en créant des étudiants liés à différentes filières.