Université Internationale de Casablanca

UNIVERSITÉ RECONNUE PAR L'ÉTAT

POO: Langage C#

Pr. EL OUKKAL Sanae

Plan

- Eléments de base
- Structures de contrôle
- Classes & objets
- Héritage, polymorphisme & interfaces
- Autres éléments de C#

Introduction: POO

- De manière superficielle, le terme « orienté objet » signifie que l'on organise le programme comme une collection d'objets dissociés comprenant à la fois une structure de données (attributs) et des comportements (méthodes) dans une même entité.

Introduction Langage c#

- Langage de programmation crée par Anders Heljberg (Microsoft) en 2001.
- Langage orienté objet, inspiré de C++ et Java.
- Permet d’utiliser la framework .NET de manière optimale.

Méthode Main

- Chaque assemblage exécutable a au moins une méthode statique Main() dans une de ses classes.

- Cette méthode représente le point d'entrée du programme, c'est-à-dire que le thread principal créé automatiquement lorsque le processus se lance ;va commencer par exécuter le code de cette méthode.

Méthode Main

- Il peut y avoir éventuellement plusieurs méthodes Main() (chacune dans une classe différente) par programme.

- Le cas échéant, il faut préciser au compilateur quelle méthode Main() constitue le point d'entrée du programme.

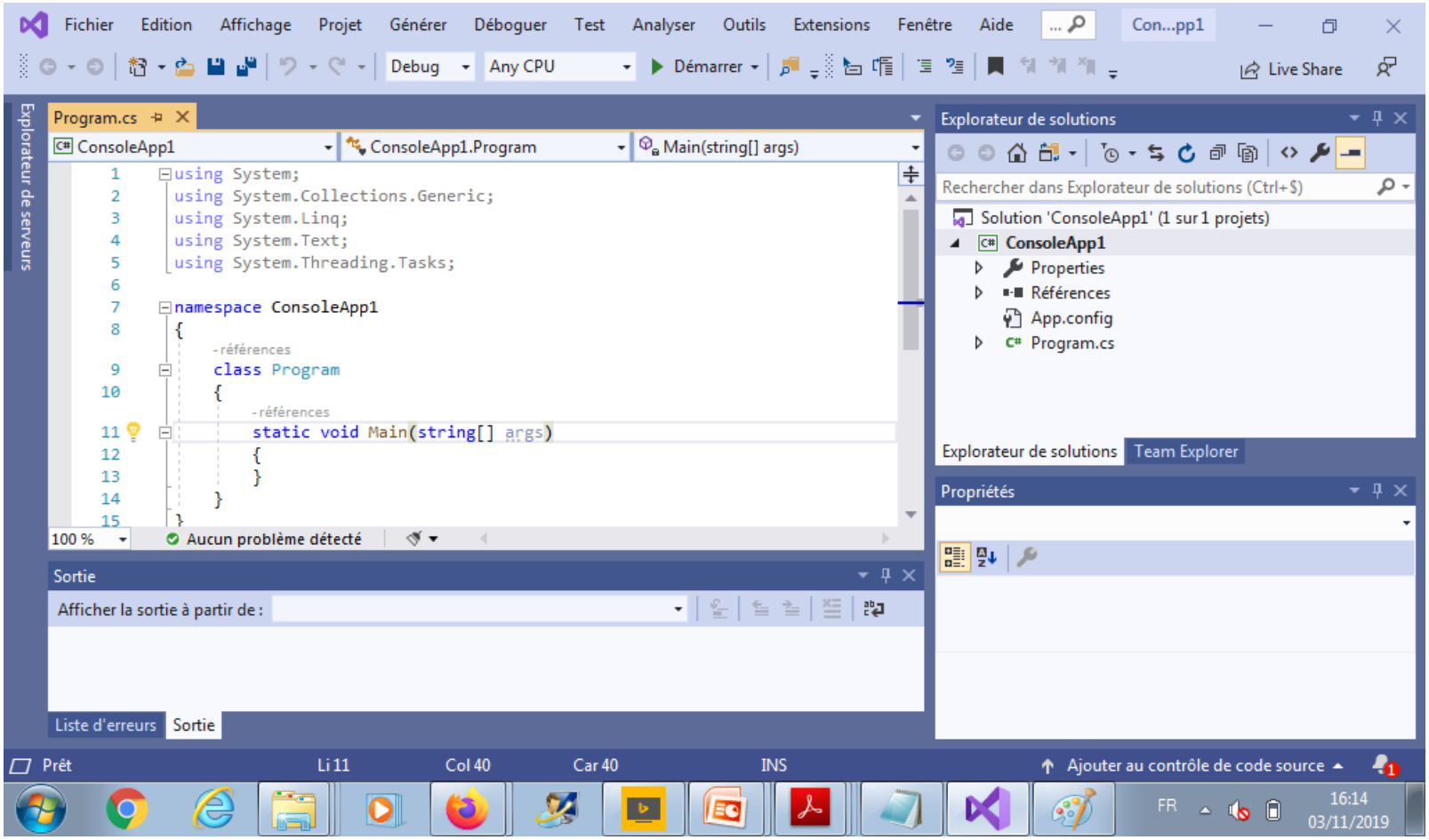
Exécution du Code C#

- Code C#: Code source
  - Traduction en un langage intermédiaire (MSIL) par Microsoft.
  - L'utilisation du langage précédent pour avoir un CLR, connu dans .NET framework.
  - Le rôle de CLR est de traduire le CLR en langage (Native Code).
  - À partir de ce (Native Code), pour qu'on puisse avoir le résultat sur l'écran.

Structure d'un programme type console


```
using System;
namespace nom
{
class Program {
      static void  Main(string[]  args) {
                  // vos lignes de codes ici }
      }
}
```

Interface Visual Studio



UTILISATION DE BASE DU LANGAGE C#

Écriture et Lecture en C#

- Comme tous les langages de programmation, C# a lui aussi des fonctions qui permettent d'afficher sur l'écran, ainsi que de lire au clavier.

- Dans la POO, on les appelle *les méthodes*.

- Chacune de ces méthodes se trouve dans la classe ***Console***.

Écriture en C#

- Méthode d'écriture: WriteLine(), Write()
  - Afin de pouvoir utiliser l'une de ces deux méthodes, il faut toujours faire référence au nom de la classe à la quelle elles appartiennent.
  - Syntaxe:

Console.WriteLine(" Message "); : affiche le message et revient à la ligne.

Console.Write(" Message "); : affiche le message et reste sur la même ligne.

Lecture en C#

- Méthode de lecture: ReadLine(), Read()
  - ReadLine() est une méthode qui permet de lire la valeur saisie au clavier. Cette valeur est considérée comme chaîne de caractère.
  - Read() permet la saisie des entiers, sauf que elle les converti en code ASCII.
  - Syntaxe:

Console.ReadLine(); : te donne la possibilité de saisir une valeur.

Console.Read(); saisir des entiers.

Lecture en C#

- D'où la nécessité d'utiliser des méthodes qui permettent la conversion.

- Il existe deux méthodes (fonctions):
  - Parse(nomVar);
  - Convert.To(Type)(nomVar);

- *Syntaxe:*
  - int a = int.Parse(Console.Readline());

Lecture en C#

- *Exemple d'application:*

  - Réaliser une saisie au clavier du prénom, le nom et l'âge de l'utilisateur. Après la saisie, afficher un message à l'écran:

==========

Nom: 'la valeur'

Prénom: 'la valeur'

Age : 'la valeur'

==========

Types des Variables

Type	Description	Rang
Byte	Entier non-sigtné (8bits)	0 à 255
Sbyte	Entier signé (8 bius)	-128 à 127
short	Entier signé (16-bit)	-32,768 à 32,767
ushort	Entier non signé (16-bit)	0 à 65,535
int	Entier signé (32-bit)	-2,147,483,648 à 2,147,483,647
uint	Entier non signé (32-bit)	0 à 4,294,967,295
long	entier signé (64-bit)	-9,223,372,036,854,775,808 à 9,223,372,036,854,775,807
ulong	Entier non signé (64-bit)	0 à 18,446,744,073,709,551,615

Types des Variables

Type	Description	Rang
float	Réel (32-bits)	-3.402823e38 à 3.402823e38
double	Réel (64-bits)	-1.79769313486232e308 à 1.79769313486232e308
decimal	Utilisé dans le domaine financier et monétaire (128-bits)	(+ or -)1.0 x 10e-28 to 7.9 x 10e28
char	Caractère (16-bit)	Tous les caractères valide
bool	Logique (8-bit)	True ou False

Types des Variables

Type	Description	Rang
object	Base de tous les autres types.	
string	Séquence de caractère	
DateTime	Représente la date et l'heure	0:00:00am 1/1/01 à 11:59:59pm 12/31/9999

Déclaration et Assignation

- Déclaration d'une variable

```
int a;
int b = 5;
double c = a + b;
```

- Assignation d'une valeur

```
a = 6;
c = a + 4;
c += 9;
```

Déclaration et Assignation

- Déclaration d'une constante

  const float PI = 3.14;

- Commentaire

  // Commentaire sur une ligne

  /* Commentaires sur plusieurs lignes */

Opérateurs

- Arithmétique:

a+b	Addition
a-b	Soustraction
a/b	Division
a%b	Modulo
a+=b	a = à la somme de a+b
a-=b	a = à la soustraction de a-b
a*=b	a= au produit de a*b
a/=b	a=à la division de a/b
a%=b	a= à la valeur de a%b

Opérateurs

- Conditionnel:

a==b	égal
a!=b	Différent
a>b	Supérieur
a<b	Inférieur
a>=b	Supérieur ou égal
a<=b	Inférieur ou égal

- Le résultat obtenu de ces opérateurs est une valeur booléenne.

Mots réservés au C#

abstract	as	base	bool	break
byte	case	catch	char	checked
class	const	continue	decimal	default
delegate	do	double	else	enum
event	explicit	extern	false	finally
float	for	foreach	Goto	if
is	lock	long	namespace	new
private	protected	public	readonly	ref
static	string	struct	switch	this
true	try	typeof	uint	ulong

Conditions en C#

- L'utilisation des conditions en c# est le même avec les autres langages de programmation (C par exemple)
- Syntaxe:

```
if (booleen){
bloc;
}
```

```
if (booleen){
bloc;
}
else{
bloc;
}
```

```
if(booleen){
bloc;
}
else if(booleen){
bloc;
}
else{
bloc;
}
```

Le Multichoix

- L'instruction *switch* peut être utilisée lorsqu'une variable prend beaucoup de valeurs (multichoix).
- *Syntaxe:*

```
Switch(variable){
case val1:
        instruction;
        break;
case val2:
        instruction;
        break;
default:
        instruction;}
```

Boucle en C#

- D'une manière générale, les boucles permettent d'exécuter un bout de code tant qu'une condition est vrai.

- Les boucles sont:
  - Boucle for,
  - Boucle foreach, // utilisée surtout pour l'affichage
  - While / do while

Boucle en C#: for

- Elle permet de répéter un bout de code tant qu'une condition est vraie.
- Souvent cette condition est un compteur.

- *Syntaxe:*

```
for(initialisation; condition d'arrêt; compteur){
Bloc d'instruction;
}
```

Boucle en C# foreach

- Cette boucle est particulièrement conçue pour parcourir les éléments d'un tableau ou d'une collection.

- On ne spécifie pas de compteur, on se contente de lui dire de prendre les éléments de l’ensemble, les uns après les autres.

Boucle en C# foreach

- *Syntaxe:*

foreach (*type nom_de_variable* in *nom_de_l'ensemble*)
{
    instructions;
}

Boucle en C#: différence entre for et foreach

- La variable que l'on utilise pour effectuer le balayage ne prend pas successivement les différentes valeurs d'un indice, mais directement, les différentes valeurs du tableau ou de la collection eux-mêmes

- La boucle foreach impose de déclarer cette variable dans l'instruction de la boucle : pas question donc de réutiliser une variable déjà déclarée auparavant

- Les éléments traités dans une boucle foreach sont en lecture seule. On ne peut donc ni les modifier, ni les ajouter, ni les supprimer.

Boucle en C#
While

- Il s'agit bien entendu de l'équivalent du TantQue..

- Syntaxe:

```
while (booléen)
{
   instructions;
}
```

Exemples:

1) Écrire un programme qui permet de calculer le factoriel d'un entier positif non nul saisi au clavier. Il faudra valider la nature du chiffre saisie.

2) Écrire un programme qui donne la possibilité de saisir un entier non nul au clavier, et affiche à l'utilisateur le jour correspondant.

Tableau:
Tableau à 1 dimension:

- Déclaration avec initialisation:
  - La déclaration d'un tableau à une dimension est codée en ajoutant des parenthèses [ ] à droite du type de données.
  - Il est possible d'initialiser le tableau avec des valeurs entre des accolades. (comme en C)

Tableau:
Tableau à 1 dimension

- Déclaration sans initialisation:
  - La déclaration d'un tableau et l'instanciation d'un tableau utilise l'opérateur *new*.
  - Le type des valeurs contenues dans le tableau ainsi que leur nombre, doit être précisé.
  - La propriété Length du tableau contient la taille du tableau.
  - Chacun des éléments contient une valeur par défaut :
    - 0 pour les entiers,
    - 0.0 pour les réels,
    - '\0' pour un caractère vide;
    - Et null pour une chaîne vide.

Tableau:
Tableau à 1 dimension

- Syntaxe de déclaration:

```
int[] nomtableau = new int [10];
```

- Parcours du tableau:

```
for (int index = 0; index < nomtableau.Length; index ++)
{
  instructions;
}
```

La classe Array

- Cette classe offre des méthodes pour la manipulation de tableaux à une dimension. En voici quelques-unes.

Méthode	**Valeur de retour**	**Description**

Sort(...)	void	Trie en ordre croissant les éléments du tableau fourni.
		Par exemple :
		// Trier tous les éléments du tableau.
		Array.Sort(nomtableau);
		
		// Trier les éléments de l'index 2 à 9.
		Array.Sort(nomtableau, 2, 9);

La classe Array

Méthode	Valeur de retour	Description

Reverse(...)	void	Inverse les éléments du tableau fourni. Le premier devient le dernier, le deuxième devient l'avant-dernier, etc.
		Par exemple :
		// Trier en ordre décroissant tous les éléments du tableau.
		Array.Sort(nomtableau);
		Array.Reverse(nomtableau);

La classe Array

Méthode	Valeur de retour	Description

IndexOf(...)	int	Recherche la valeur spécifiée et retourne l'index de la première occurrence de cette valeur dans le tableau. Retourne ***-1*** si la valeur n'est pas trouvée.
		Par exemple :
		
		int valeurCherchee = 88;
		int indexTrouve = Array.IndexOf(nomtableau, valeurCherchee);

La classe Array

Méthode	Valeur de retour	Description
Clear(...)		Affecte à une plage d'éléments la valeur 0 pour un tableau d'entiers, à partir d'un index spécifié, pour un nombre d'éléments donné. Par exemple : Array. Clear(nomtableau, 0, 5);

La classe Array

Méthode	Valeur de retour	Description
Copy(...)	void	Copie, un certain nombre d'éléments d'un tableau vers un autre tableau, en commençant par le premier élément. Par exemple : Array.Copy(tableauSource, tableauDestination, nombreElementsACopier);

Tableau de 2 dimensions

- Déclaration avec initialisation:
  - La déclaration d'un tableau à 2 dimensions est codée en ajoutant des parenthèses [,] à droite du type de données.
  - Pour l'initialiser, les valeurs sont codées entre des paires d'accolades.
- *Syntaxe:*

type de données [,] nomTableau = {{ valeur11, valeur12, valeur13, ...},{valeur21, valeur22, valeur23, ...};…..};

BON COURAGE