vinci
HAUTE ÉCOLE LÉONARD DE VINCI

institut paul lambin

.NET

Semaine 1

Introduction

Java vs C#

Points importants

- Annotations [ ]
- Conventions C#
- Package / Namespace
- Calendar / Datetime
- Interfaces /Collections
- Héritage
- Const / readonly / sealed
- Itérateurs
- Console.ReadLine / Console.WriteLine

Annotations

- C# utilise un mécanisme d'annotations
- Les annotations sont placées entre []
- Notamment utilisé pour la sérialisation
  - Ex : [Serializable]

Conventions C#

- Nom des champs -> _champ
- Méthode -> Commence par une lettre majuscule
  - Attention Main
- C# utilise des propriétés plutôt que getter/setter
  - Propriété en Majuscule !

```
private readonly DateTime _dateDeNaissance;
public DateTime DateDeNaissance
{
   get { return _dateDeNaissance;}
}
```

- Propriétés simples auto-implémentées

```
public DateTime DateDeNaissance { get ; set}
```

Package / Namespace

- Package -> organisation physique et logique
- Namespace -> organisation logique

Calendar /Datetime

- Datetime -> Classe de base pour représenter date et heure en C#

Interfaces /Collections

- Les interfaces commencent par un « I »
  - Ex : IList, IDictionary
- Les collections disposent d'une classe de base qui correspond au nom de l'interface sans le « I »
  - Ex : List, Dictionary, …
- Map en java -> Dictionary en C#

Héritage

- Java -> toutes les méthodes sont « virtual » c'est-à-dire qu'elles peuvent être redéfinies dans les classes enfants et que le type de l'objet est recherché à l'exécution
- C# -> par défaut les méthodes ne sont pas « virtual » c'est-à-dire que le type de l'objet à l'exécution sera la classe la plus haute dans la hiérarchie.
- Si on veut le même comportement qu'en Java
  - virtual(parent) et override (enfant)
- Implements / extends -> :

Const /readonly / sealed

- Java « Final » -> empêcher la redéfinition dans les classes enfants
- C#
  - « sealed » si on a utilisé des « virtual »
  - « const » -> assigné à la compilation
  - « readonly » -> assigné à l'exécution (uniquement dans constructeur ou déclaration)

vinci
HAUTE ÉCOLE LÉONARD DE VINCI

Iterateurs

- Iterator -> IEnumerator
  - GetEnumerator sur les listes, dictionnaires, …
- Boucle
  - it.MoveNext() (hasNext + next en Java)
  - it.Current

vinci
HAUTE ÉCOLE LÉONARD DE VINCI

Console

- Console.WriteLine == System.out.writeln
- Console.ReadLine == Scanner.nextLine