

STRUCTURES DE DONNEES

✓ Entités et évènements

→ Collections

LinQ

Synthèse

LIST<>

Usage

```
var l = new List<int>();
```

Méthodes utiles

Add, Remove, Clear
[], Count

DICTIONARY<K, V>

Principe

Association clé/valeur

Accès quasi instantané par clé

Syntaxes

```
var d = new Dictionary<string, double>();  
d["France"] = 3.4 ;  
if(d.ContainsKey("France")) ...
```

IENUMERABLE<T> / YIELD

Usages

Interface utilisée pour toute énumération

Utilisable avec foreach, linq, thread safe, ...

Yield

Idéal pour les méthodes/propriétés retournant une collection

Ni allocation, ni création de collection intermédiaire

```
IEnumerable<...> GetXxx(...) {  
    for(...) yield return ... ;  
}
```

STRUCTURES DE DONNEES

✓ Entités et évènements

✓ Collections

→ LinQ

Synthèse

LINQ

Language Integrated Query

Requête intégrée au langage

Elimine/optimise les sélections/tris/filtres/traitements

Evaluation paresseuse = au moment du parcours

Structure minimale

```
var maires = from v in villes  
            select v.maire;
```

```
Ou var maires = villes.Select(v => v.maire);
```

LINQ

Méthodes (usuel)

```
var liste = villes.Where(...).OrderBy(...).Select(...);
```

Notation LinQ (désuet)

```
var groupes =   from v in villes  
                let tranche = v.Population / 10000  
                join r in regions on v.Region equals r  
                group v by tranche into categorie  
                where categorie.Key >= 20  
                orderby categorie.Key(descending)  
                select categorie;
```

STRUCTURES DE DONNEES

- ✓ Entités et évènements
- ✓ Collections
- ✓ LinQ
- ➔ Synthèse

RESUME

Types valeurs, types références et record

Evènements pour une autre forme d'IoC

Collections usuelles du C# selon les besoins

Langage intégré d'interrogation et de traitement