



alura

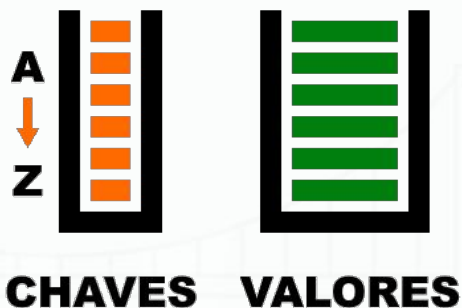
Cursos Online de Tecnologia

C# Collections

Parte 2

Coleções Ordenadas

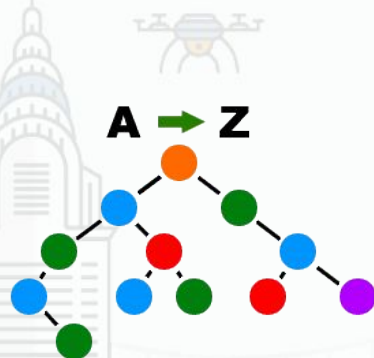
Sorted List



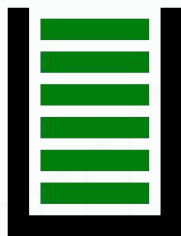
```
IDictionary<string, Aluno> sorted  
    = new SortedList<string, Aluno>();  
  
sorted.Add("VT", new Aluno("Vanessa", 34672));  
sorted.Add("AL", new Aluno("Ana", 5617));  
sorted.Add("RN", new Aluno("Rafael", 17645));  
sorted.Add("WM", new Aluno("Wanderson", 11287));
```

Coleções Ordenadas

Sorted Dictionary



CHAVES



VALORES

```
IDictionary<string, Aluno> sorted
    = new SortedDictionary<string, Aluno>();

sorted.Add("VT", new Aluno("Vanessa", 34672));
sorted.Add("AL", new Aluno("Ana", 5617));
sorted.Add("RN", new Aluno("Rafael", 17645));
sorted.Add("WM", new Aluno("Wanderson", 11287));
```

Coleções Ordenadas

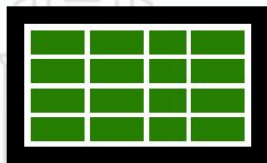
Sorted Set



```
alunos.IsSubsetOf(outroConjunto);  
alunos.IsSupersetOf(outroConjunto);  
alunos.SetEquals(outroConjunto);  
alunos.ExceptWith(outroConjunto);  
alunos.IntersectWith(outroConjunto);  
alunos.SymmetricExceptWith(outroConjunto);  
alunos.UnionWith(outroConjunto);
```

Arrays Multidimensionais

Matriz Retangular, Matriz Multidimensional



```
string[,] resultados = new string[4, 3];
```

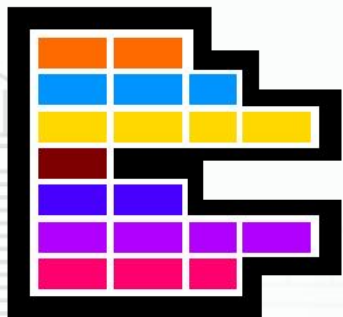
```
resultados[0, 0] = "Alemanha";  
resultados[1, 0] = "Argentina";  
resultados[2, 0] = "Holanda";  
resultados[3, 0] = "Brasil";
```

```
resultados[0, 1] = "Espanha";  
resultados[1, 1] = "Holanda";  
resultados[2, 1] = "Alemanha";  
resultados[3, 1] = "Uruguai";
```

```
resultados[0, 2] = "Itália";  
resultados[1, 2] = "França";  
resultados[2, 2] = "Alemanha";  
resultados[3, 2] = "Portugal";
```

Arrays Multidimensionais

Matriz Denteada



```
string[][] familias = new string[3][];
```

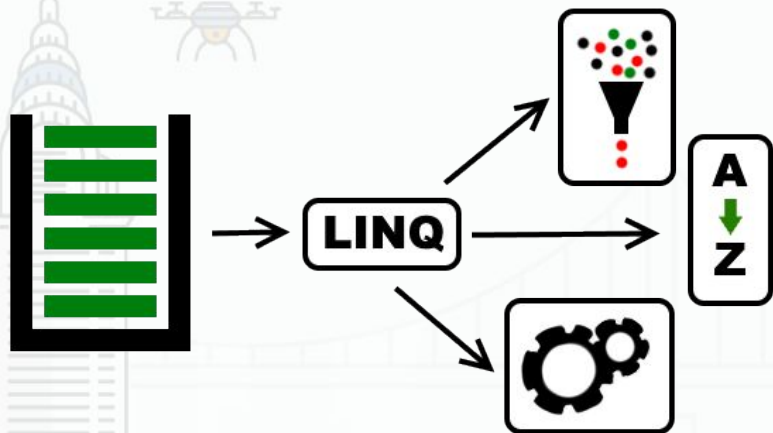
```
familias[0] = new string[] { "Fred", "Wilma", "Pedrita" };
```

```
familias[1] = new string[] { "Homer", "Marge", "Lisa", "Bart", "Maggie" };
```

```
familias[2] = new string[] { "Florinda", "Kiko" };
```

LINQ

Filtrando, Ordenando e Transformando



```
IEnumerable<string>  
consulta = meses  
    .Where(m => m.Dias == 31)  
    .OrderBy(m => m.Nome)  
    .Select(m => m.Nome.ToUpper());
```

LINQ

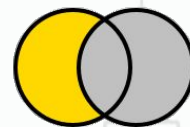
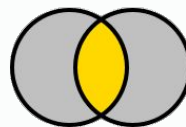
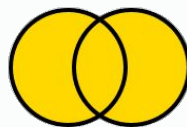
Pulando e Obtendo faixas de elementos



```
//Pegar o primeiro trimestre  
var consulta = meses.Take(3);  
//Pegar os meses depois do primeiro trimestre  
var consulta2 = meses.Skip(3);  
//Pegar os meses do terceiro trimestre  
var consulta3 = meses.Skip(6).Take(3);  
//Pegar os meses até que o mês comece com a letra 'S'  
var consulta4 = meses.TakeWhile(m => !m.Nome.StartsWith("S"));  
//Pular os meses até que o mês comece com a letra 'S'  
var consulta5 = meses.SkipWhile(m => !m.Nome.StartsWith("S"));
```


LINQ

Operações de Conjuntos



```
var consulta = seq1.Concat(seq2);

Console.WriteLine("União de duas sequências");
var consulta2 = seq1.Union(seq2);

Console.WriteLine("União de duas sequências com comparador");
var consulta3 = seq1.Union(seq2,
    StringComparer.InvariantCultureIgnoreCase);

Console.WriteLine("Interseção de duas sequências");
var consulta4 = seq1.Intersect(seq2);

Console.WriteLine("Exceto: elementos de seq1 que não estão em seq2");
var consulta5 = seq1.Except(seq2);
```

Convertendo Coleções

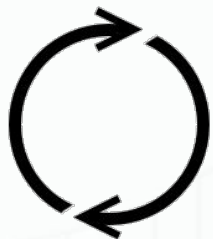
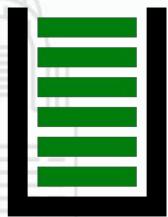
Covariância



```
IList<string> listaMeses = new List<string>
{
    "Janeiro", "Fevereiro", "Março",
    "Abril", "Maio", "Junho",
    "Julho", "Agosto", "Setembro",
    "Outubro", "Novembro", "Dezembro"
};
IEnumerable<object> enumObj = listaMeses; //COVARIÂNCIA
```

Iteração de Coleções

O Laço Foreach



```
var meses = new string[]  
{  
    "Janeiro", "Fevereiro", "Março",  
    "Abril", "Maio", "Junho",  
    "Julho", "Agosto", "Setembro",  
    "Outubro", "Novembro", "Dezembro"  
};  
  
foreach (var mes in meses)  
{  
    meses[0] = meses[0].ToUpper();  
    Console.WriteLine(mes);  
}
```