cencal centro de formação profissional para a indústria de cerâmica

Programação em C#

António Ramos | Novembro 2022

Co-Financiado por:

IEFP APICER associação portuguesa das indústrias de cerâmica e cristalaria ARO Associação Empresarial da Região Oeste REPÚBLICA PORTUGUESA POCH PORTUGAL 2020 UNIÃO EUROPEIA Fundo Social Europeu

Enum

Permitem ao programador utilizar nomes friendly para constantes numerárias.

Usa-se a keyword "enum" antes do nome da variável, sendo os valores passados dentro dos parênteses separados por virgula.

```
enum WeekDayNumber
{
 Sun = 1, Mon = 2, Tues = 3, Wed = 4, Thurs = 5, Fri = 6, Sat = 7
}
```

Struct

Representam estruturas de dados, podem conter campos, métodos, propriedades como as classes.

Estes são utilizados para definir estruturas de dados complexas, pares de chave-valor, entre outros.

```
public struct Money
{
  public string Currency { get; set; }
  public double Amount { get; set; }
}
```

```
Money money = new Money();
Console.WriteLine(money.Currency);
Console.WriteLine(money.Amount);
```

Dictionary<TKey,TValue>

Permitem criar listas/coleções genéricas de pares valor-chave. As suas chaves devem ser únicas e os seus valores são acessíveis pela chave myDictionary[key]

Podem ser utilizados os métodos Add(), Remove(), Update() para manipular as coleções ao longo do programa.

Dictionary<TKey,TValue>

```
// Create a new dictionary of strings, with string keys.
Dictionary<string, string> openWith =
  new Dictionary<string, string>();

// Add some elements to the dictionary. There are no
// duplicate keys, but some of the values are duplicates.
openWith.Add("txt", "notepad.exe");
openWith.Add("bmp", "paint.exe");
openWith.Add("dib", "paint.exe");
openWith.Add("rtf", "wordpad.exe");
```

Dictionary<TKey,TValue>

Para verificar se a chave já existe no dicionário utiliza-se o método ContainsKey().

```
if (openWith.ContainsKey("txt"))
{
  Console.WriteLine(openWith["txt"]);
}
```

LINQ

Funcionalidade que permite efetuar queries aos dados no nosso programa, podendo efetuar filtragens em listas de dados.

```
List<int> numbers = new() { 5, 4, 1, 3, 9, 8, 6, 7, 2, 0 };

IEnumerable<int> filteringQuery =
    from num in numbers
    where num < 3 || num > 7
    select num;
```

LINQ

O LINQ também pode ser utilizado através de métodos:

```
var average = numbers.Average();

var concatenationQuery = numbers.Concat(numbers);

var largeNumbersQuery = numbers.Where(c => c > 8);
```