

# ▼ C# Fundamentals Interfaces

# Interfaces:

## Interfaces (Guia de Programação em C#)

20/02/2020 • 4 minutos para o fim da leitura • 

Uma interface contém definições para um grupo de funcionalidades relacionadas que uma `classe` não abstrata ou uma `struct` deve implementar. Uma interface pode definir `static` métodos, que devem ter uma implementação. A partir do C# 8,0, uma interface pode definir uma implementação padrão para membros. Uma interface não pode declarar dados de instância, como campos, propriedades implementadas automaticamente ou eventos de propriedade.

Usando interfaces, você pode, por exemplo, incluir o comportamento de várias fontes em uma classe. Essa funcionalidade é importante em C# porque a linguagem não dá suporte a várias heranças de classes. Além disso, use uma interface se você deseja simular a herança para structs, pois eles não podem herdar de outro struct ou classe.

# Interfaces:

Você define uma interface usando a palavra-chave `interface` como mostra o exemplo a seguir.

C#

 Copiar

```
interface IEquatable<T>
{
    bool Equals(T obj);
}
```

# Interfaces:

O exemplo a seguir mostra uma implementação da interface `IEquatable<T>`. A classe de implementação, `Car`, deverá fornecer uma implementação do método `Equals`.

C#

 Copiar

```
public class Car : IEquatable<Car>
{
    public string Make {get; set;}
    public string Model { get; set; }
    public string Year { get; set; }

    // Implementation of IEquatable<T> interface
    public bool Equals(Car car)
    {
        return (this.Make, this.Model, this.Year) ==
            (car.Make, car.Model, car.Year);
    }
}
```

# Interfaces:

- Links de documentação oficial:
- **Documentação:**
- <https://docs.microsoft.com/pt-br/dotnet/csharp/programming-guide/interfaces/>





# Dúvidas?