

逆变：

对于泛型类型参数，`in` 关键字可指定类型参数是逆变的。可以在泛型接口和委托中使用 `in` 关键字。

逆变使你使用的类型可以比泛型参数指定的类型派生程度更小。这样可以隐式转换实现变体接口的类以及隐式转换委托类型。引用类型支持泛型类型参数中的协变和逆变，但值类型不支持它们。

如果类型仅用作方法参数的类型，而不用作方法返回类型，则类型可以在泛型接口或委托中声明为逆变。 `Ref` 和 `out` 参数不能为变体。

具有逆变类型参数的接口使其方法接受的参数的类型可以比接口类型参数指定的类型派生程度更小。例如，因为在 .NET Framework 4 的 [IComparer<T>](#) 接口中，类型 `T` 是逆变的，所以可以将 `IComparer(Of Person)` 类型的对象分配给 `IComparer(Of Employee)` 类型的对象，而无需使用任何特殊转换方法（如果 `Employee` 继承 `Person`）。

协变：

具有协变类型参数的接口使其方法返回的类型可以比类型参数指定的类型派生程度更大。例如，因为在 .NET Framework 4 的 [IEnumerable<T>](#) 中，类型 `T` 是协变的，所以可以将 `IEnumerabe(Of String)` 类型的对象分配给 `IEnumerable(Of Object)` 类型的对象，而无需使用任何特殊转换方法。

可以向协变委托分配相同类型的其他委托，不过要使用派生程度更大的泛型类型参数。