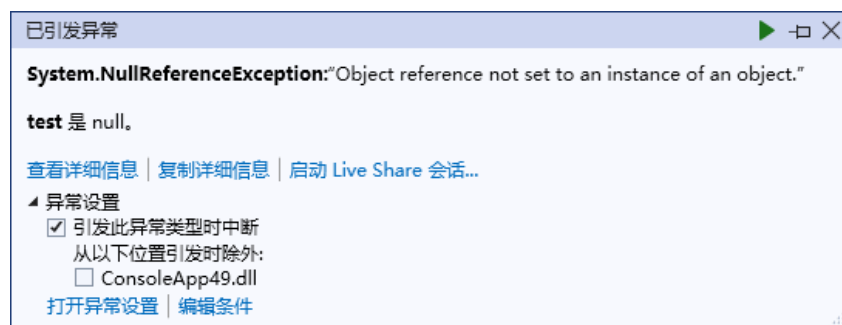


第14期-可空的引用类型

2022年5月19日 13:37

类型介绍

在 C# 8.0 之前，所有引用类型都可以赋 null 值，导致运行时引发 System.NullReferenceException 异常的可能性。可为 null 引用类型可帮助在编译期间避免这些异常。



```
// Declare variable and assign it as null.
FooBar fooBar = null;

// Dereference variable by calling ToString.
// This will throw a NullReferenceException.
_ = fooBar.ToString();
```

开启可空的引用类型支持

#nullable disable: 将可为空注释和警告上下文设置为“已禁用”。

#nullable enable: 将可为空注释和警告上下文设置为“已启用”。

```
<nullable>Enable</nullable>
```

运算符

- null 包容 (!) 运算符
- null 合并 (??) 运算符
- null 条件 (?.) 运算符

解析可为空警告的方法

- 添加必要的 null 检查。
- 添加 ? 或 ! 可为 null 的注释。
- 添加描述 null 语义的特性。
- 正确初始化变量。

使用可空性特性注解表达意图

Attribute	Category	含义
AllowNull	Precondition	不可为 null 的参数、字段或属性可以为 null。
DisallowNull	Precondition	可为 null 的参数、字段或属性应永不为 null。
MaybeNull	后置条件	不可为 null 的参数、字段、属性或返回值可能为 null。
NotNull	后置条件	可为 null 的参数、字段、属性或返回值将永不为 null。
MaybeNullWhen	有条件后置条件	当方法返回指定的 <code>bool</code> 值时，不可为 null 的参数可以为 null。
NotNullWhen	有条件后置条件	当方法返回指定的 <code>bool</code> 值时，可以为 null 的参数不会为 null。
NotNullIfNotNull	有条件后置条件	如果指定参数的自变量不为 null，则返回值、属性或自变量不为 null。
MemberNotNull	方法和属性帮助程序方法	当方法返回时，列出的成员不会为 null。
MemberNotNullWhen	方法和属性帮助程序方法	当方法返回指定的 <code>bool</code> 值时，列出的成员不会为 null。
DoesNotReturn	无法访问的代码	方法或属性永远不会返回。换句话说，它总是引发异常。
DoesNotReturnIf	无法访问的代码	如果关联的 <code>bool</code> 参数具有指定值，则此方法或属性永远不会返回。

<https://docs.microsoft.com/zh-cn/dotnet/csharp/language-reference/attributes/nullable-analysis>

可空的引用类型在 Entity Framework Core 中的应用

<https://docs.microsoft.com/zh-cn/ef/core/miscellaneous/nullable-reference-types>

参考资料

可为空引用类型

<https://docs.microsoft.com/zh-cn/dotnet/csharp/nullable-references>

解决可以为 null 的警告

<https://docs.microsoft.com/zh-cn/dotnet/csharp/nullable-warnings>

C# 中的 null 安全性

<https://docs.microsoft.com/zh-cn/learn/modules/csharp-null-safety>