**相等比较（C# 编程指南）**

有时必须比较两个值是否相等。在某些情况下，您测试的是“值相等性”（也称为“等效性”），意即两个变量包含的值相等。而在其他情况下，则必须确定两个变量是否引用内存中的同一基础对象。这种类型的相等性称为“引用相等性”（或“标识”）。本主题描述这两种相等性，并提供指向其他主题的链接以了解更多信息。

**引用相等性: 需要使用Object.** **ReferenceEquals方法来比较**

引用相等性是指两个对象引用均引用同一基础对象。这可以通过简单的赋值来实现，如下面的示例所示。

using System;

class Test

{

public int Num { get; set; }

public string Str { get; set; }

static void Main()

{

Test a = new Test() { Num = 1, Str = "Hi" };

Test b = new Test() { Num = 1, Str = "Hi" };

bool areEqual = System.Object.ReferenceEquals(a, b);

// False:

System.Console.WriteLine("ReferenceEquals(a, b) = {0}", areEqual);

// Assign b to a.

b = a;

// Repeat calls with different results.

areEqual = System.Object.ReferenceEquals(a, b);

// True:

System.Console.WriteLine("ReferenceEquals(a, b) = {0}", areEqual);

// Keep the console open in debug mode.

Console.WriteLine("Press any key to exit.");

Console.ReadKey();

}

}

在此代码中，创建了两个对象，但在赋值语句后，这两个引用所引用的是同一对象。因此，它们具有引用相等性。使用 [ReferenceEquals](https://msdn.microsoft.com/zh-CN/library/system.object.referenceequals.aspx) 方法来确定两个引用是否引用同一对象。

引用相等性的概念仅适用于引用类型。由于在将值类型的实例赋给变量时将建立值的副本，因此值类型对象无法具有引用相等性。因此，永远不能有引用内存中的同一位置的两个未装箱结构。此外，如果您使用 [ReferenceEquals](https://msdn.microsoft.com/zh-CN/library/system.object.referenceequals.aspx) 来比较两个值类型，结果将始终为 **false**，即使对象中包含的值都相同也是如此。这是因为每个变量都会被装箱到单独的对象实例中。有关更多信息，请参见[如何：测试引用相等性（标识）（C# 编程指南）](https://msdn.microsoft.com/zh-CN/library/dd183759.aspx)。

**值相等性,使用== 或者equals函数**

值相等性是指两个对象包含相同的一个或多个值。对于基元值类型（例如 [int](https://msdn.microsoft.com/zh-CN/library/5kzh1b5w.aspx) 或 [bool](https://msdn.microsoft.com/zh-CN/library/c8f5xwh7.aspx)），针对值相等性的测试简单明了。您可以使用 [==](https://msdn.microsoft.com/zh-CN/library/53k8ybth.aspx) 运算符，如下面的示例所示。

int a = GetOriginalValue();

int b = GetCurrentValue();

// Test for value equality.

if( b == a)

{

// The two integers are equal.

}

对于大多数其他类型，针对值相等性的测试较为复杂，因为它需要您了解类型对值相等性的定义方式。对于具有多个字段或属性的类和结构，值相等性的定义通常是指所有字段或属性都具有相同的值。例如，如果 pointA.X 等于 pointB.X，并且 pointA.Y 等于 pointB.Y，则可以将两个 **Point** 对象定义为相等。

不过，并不要求类型中的所有字段均相等。只需子集相等即可。在比较不属于您的类型时，应确保明确了解相等性对于该类型是如何定义的。有关如何在您自己的类和结构中定义值相等性的更多信息，请参见[如何：为类型定义值相等性（C# 编程指南）](https://msdn.microsoft.com/zh-CN/library/dd183755.aspx)。

**浮点值的值相等性**

由于二进制计算机上浮点算法的不精确性，因此浮点值（[double](https://msdn.microsoft.com/zh-CN/library/678hzkk9.aspx) 和 [float](https://msdn.microsoft.com/zh-CN/library/b1e65aza.aspx)）的相等比较会出现问题。有关更多信息，请参见主题 [System.Double](https://msdn.microsoft.com/zh-CN/library/system.double.aspx) 中的备注。