**泛型代码中的默认关键字（C# 编程指南）**

在泛型类和泛型方法中产生的一个问题是，在预先未知以下情况时，如何将默认值分配给参数化类型 T：

* T 是引用类型还是值类型。
* 如果 T 为值类型，则它是数值还是结构。

给定参数化类型 T 的一个变量 t，只有当 T 为引用类型时，语句 t = null 才有效；只有当 T 为数值类型而不是结构时，语句 t = 0 才能正常使用。解决方案是使用 **default** 关键字，此关键字对于引用类型会返回 null，对于数值类型会返回零。对于结构，此关键字将返回初始化为零或 null 的每个结构成员，具体取决于这些结构是值类型还是引用类型。对于可以为 null 的值类型，默认返回 [System.Nullable<T>](https://msdn.microsoft.com/zh-cn/library/b3h38hb0.aspx)，它像任何结构一样初始化。

以下来自 GenericList<T> 类的示例显示了如何使用 **default** 关键字。有关更多信息，请参见[泛型概述](https://msdn.microsoft.com/zh-cn/library/0x6a29h6.aspx)。

namespace ConsoleApplication1

{

class Program

{

static void Main(string[] args)

{

// Test with a non-empty list of integers.

GenericList<int> gll = new GenericList<int>();

gll.AddNode(5);

gll.AddNode(4);

gll.AddNode(3);

int intVal = gll.GetLast();

// The following line displays 5.

System.Console.WriteLine(intVal);

// Test with an empty list of integers.

GenericList<int> gll2 = new GenericList<int>();

intVal = gll2.GetLast();

// The following line displays 0.

System.Console.WriteLine(intVal);

// Test with a non-empty list of strings.

GenericList<string> gll3 = new GenericList<string>();

gll3.AddNode("five");

gll3.AddNode("four");

string sVal = gll3.GetLast();

// The following line displays five.

System.Console.WriteLine(sVal);

// Test with an empty list of strings.

GenericList<string> gll4 = new GenericList<string>();

sVal = gll4.GetLast();

// The following line displays a blank line.

System.Console.WriteLine(sVal);

}

}

// T is the type of data stored in a particular instance of GenericList.

public class GenericList<T>

{

private class Node

{

// Each node has a reference to the next node in the list.

public Node Next;

// Each node holds a value of type T.

public T Data;

}

// The list is initially empty.

private Node head = null;

// Add a node at the beginning of the list with t as its data value.

public void AddNode(T t)

{

Node newNode = new Node();

newNode.Next = head;

newNode.Data = t;

head = newNode;

}

// The following method returns the data value stored in the last node in

// the list. If the list is empty, the default value for type T is

// returned.

public T GetLast()

{

// The value of temp is returned as the value of the method.

// The following declaration initializes temp to the appropriate

// default value for type T. The default value is returned if the

// list is empty.

T temp = default(T);

Node current = head;

while (current != null)

{

temp = current.Data;

current = current.Next;

}

return temp;

}

}

}

**请参阅**

[System.Collections.Generic](https://msdn.microsoft.com/zh-cn/library/system.collections.generic.aspx)

[C# 编程指南](https://msdn.microsoft.com/zh-cn/library/67ef8sbd.aspx)

[泛型（C# 编程指南）](https://msdn.microsoft.com/zh-cn/library/512aeb7t.aspx)

[泛型方法（C# 编程指南）](https://msdn.microsoft.com/zh-cn/library/twcad0zb.aspx)

[.NET Framework 中的泛型](https://msdn.microsoft.com/zh-cn/library/ms172192.aspx)