

泛型

- C#泛型是C#语言中的一个重要特性，它允许程序员在编译时指定变量的类型，从而提高代码的类型安全性和可移植性
- C#泛型使用<>符号来指定变量的类型。例如，以下代码定义了一个泛型类MyClass，该类有一个泛型方法MyMethod()

```
public class MyClass<T>
{
    public void MyMethod(T value)
    {
        // Do something with value
    }
}
```

- 要使用泛型类，需要在创建对象时指定类型参数。例如，以下代码创建了一个MyClass对象，并将其类型指定为int

```
MyClass<int> myClass = new MyClass<int>();

//现在，可以将int类型的值传递给MyMethod()方法。例如，以下代码将10传递给MyMethod()方法
myClass.MyMethod(10);
```

- C#泛型还可以用于定义泛型接口和泛型方法

泛型可以提高代码的类型安全性和可移植性

- 泛型可以提高代码的类型安全性和可移植性，如果使用泛型，可以确保在将对象传递给方法时，该对象的类型与方法的参数类型匹配。这可以帮助避免类型转换错误
- 提高代码的可移植性。例如，如果使用泛型，可以将代码从一个平台移植到另一个平台，而无需进行任何更改

泛型的方法或属性

方法或属性	描述
GetType()	获取泛型类型的类型对象
IsGenericType()	判断是否为泛型类型
IsGenericTypeDefinition()	判断是否为泛型类型的定义
GetGenericTypeDefinition()	获取泛型类型的定义

方法或属性	描述
GetGenericArguments()	获取泛型类型的类型参数
MakeGenericType()	创建泛型类型的具体类型
IsInstanceOfType()	判断对象是否是指定类型的泛型类型
As()	将对象转换为指定类型的泛型类型

- | T | 获取泛型类型的类型参数