

C：最强大的.NET语言之对象构造 PDF转换可能丢失图片或格式，建议阅读原文

[https://www.100test.com/kao\\_ti2020/253/2021\\_2022\\_C\\_\\_\\_EF\\_BC\\_9A\\_E6\\_9C\\_80\\_E5\\_c97\\_253833.htm](https://www.100test.com/kao_ti2020/253/2021_2022_C___EF_BC_9A_E6_9C_80_E5_c97_253833.htm) 简介 Visual C 开发小组花了大量的时间用于听取用户的意见，在对 .NET和C 经过仔细考量之后，决定在Visual C 2005中重新设计对通用语言运行时库（CLR）的支持，此项重新的设计被称为"C /CLI"，它将为使用及编写CLR类型提供更自然的语法。在本文中，主要探讨了新的语法，并将之与C#和托管C 这两个CLR平台上极其相近的语言进行比较，在文中也会适当地以图表给出其与本地C 的相似之处。通用语言运行时库（CLR）包括了一组规范，其是Microsoft .NET的基础，也是CLI的Microsoft版本实现。C /CLI语言设计的目标是为了对CLI提供更自然的C 支持，而Visual C 2005的编译器则在CLR上实现了C /CLI。当在仔细研究了Visual C 2005编译器和C /CLI语言设计之后，就会发现它们传达了两条重要的讯息；首先，Visual C 把自己定位于在CLR平台上的最低级编程语言，（看起来似乎没有必要使用其他语言了--包括MSIL）；其次，.NET编程应与本地C 编程一样自然。本文针对C 程序员，但并不想说服你放弃C#或者Visual Basic .NET。如果你非常喜欢C，并想继续使用传统C 提供的全部功能，而又想要C#般的编程效率，那本文正适合你。另外，本文并不是CLR或 .NET Framework的简介，而把重点放在Visual C 2005是怎样使你可以编写 .NET Framework上更优雅和高效的代码。对象构造 CLR定义了两种类型：值类型和引用类型。值类型被设计用于可进行高效地分配和访问，它们与C 的内置类型大体相似，你也能创建属于你自己的

类型，这就是Bjarne Stroustrup所称的具体类型；另一方面，引用类型被设计用于提供面向对象编程所需的特性，可用于创建有着层次结构的类：例如派生类和虚拟函数。另外在CLR中，引用类型自始至终都提供了额外的运行时特性，如自动内存管理--通常称为垃圾回收。同时，对引用类型和值类型，CLR也提供了精确的运行时类信息，这种特性通常被称为反射。值类型分配在堆栈上；而引用类型通常分配在托管堆中--这是由CLR垃圾回收机制所管理的堆。如果你在C中编写汇编代码，如平时那样，可在CRT堆中分配本地C类型，在将来，Visual C开发小组甚至允许你在托管堆中分配本地C类型，毕竟，垃圾回收对本地类型来说，也是一个极具吸引力的主题。本地C允许选择在何处创建一个特定的对象，任何类型都可分配在堆栈或CRT堆中。

```
// 分配在堆栈上
std::wstring stackObject. // 分配在CRT堆中
std::wstring* heapObject = new std::wstring.
```

如上所示，在何处分配对象是独立于类型的，主动权完全掌握在程序员的手中。另外，堆栈与堆的分配语法也是易于区别的。100Test 下载频道开通，各类考试题目直接下载。详细请访问 [www.100test.com](http://www.100test.com)