


```
计算N 3 ' ). result:=N 3. end. function
```

```
Double1(N:Integer):integerstdcall. begin messagebox(0, ' 计算N 2  
' , ' 计算N 2 ' ,mb_ok). result:=N 2. end. exports Triple name
```

```
' Tr ' , Double name ' Do ' , Triple1 name ' TrM ' , Double1  
name ' DoM ' . Triple,Double,Triple1,Double1. {$R *.RES} begin
```

```
end. 其中函数：Triple：把传入值加三 Double：把传入值加二
```

```
Triple1：把传入值加三并显示提示 Double1：把传入值加二并
```

```
显示提示 从这个例子中可以看出DLL 程序的几个规则：1)
```

```
在DLL 程序中，输出函数必须被声明为stdcall，以使用标准的
```

```
Win32 参数传递技术来代替优化的Register。（说明：
```

```
在Delphi中Register方式是缺省的调用约定，这个约定尽量采  
用寄存器来传递参数，传递次序从左到右，最多可用到3
```

```
个CPU 的寄存器，如果参数多于3 个，剩下的就通过栈来传  
送，使用寄存器传送可保证参数传递的速度最快。而stdcall
```

```
方式是通过Windows 的标准调用来传递参数，传递秩序从左  
到右，这种方式适合调用Windows 的API，在DLL 中，当然
```

```
要使用这种方式)。2) 所有的输出函数都必须列在exports子  
句下面，这使的子例程在DLL外部就可以看到。 exports Triple
```

```
name ' Tr ' , Double name ' Do ' , Triple1 name ' TrM ' ,  
Double1 name ' DoM ' . 列出了用户使用这个函数的接口名字
```

```
。虽然别名不是必须的，但最好给个别名，以使用户程序更  
容易找到这个函数，同时还要指出Delphi 6.0取消了Delphi 5.0
```

```
中允许使用的index，如果还用Index指明接口名字，Delphi  
6.0中将提示错误。实例中给出了两种提示方法，主要想说明
```

```
一个问题：showmessage( ' ' )，是VCL 提供的函数，由于多  
次编译VCL，做出的程序会比较大小。而messagebox(0, ' ' , ' ' ,
```

' ,mb_ok) 是Windows提供的API 函数，做出的程序会比较小。这就是说，编写DLL 程序的时候，要尽量避免多次编译VCL。作为一个实例，这里把两种方法都列出来了。保存编译：Projct---Build Mydll 这就完成了一个简单的动态连接库的编写。

二、动态连接库的调用 首先在implementation下做调用声明

```
const gdi32= ' mydll.dll ' .function  
triple(n:integer):integer.stdcall.external gdi32 name ' Tr ' .  
function Double(N:Integer):integer.stdcall.external gdi32 name  
' Do ' .function triple1(n:integer):integer.stdcall.external gdi32  
name ' TrM ' .function  
Double1(N:Integer):integer.stdcall.external gdi32 name ' DoM ' .
```

以后程序中就可以作为普通的函数使用了，例如：procedure TForm1.Button1Click(Sender: TObject). var N:integer. begin N:=updown1.position. edit1.text:=inttostr(triple(N)). end. 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