转载自CAD世界论坛普天同庆老师的作品。深表感谢！！  
  
Autocad VBA初级教程(第一课:入门)  
1.为什么要写这个教程  
市面上ACAD VBA的书不多，它的帮助是英文版的，很多人看不懂。其实我转行已经好几年了，而且手艺也慢慢生疏了，写个教程对自己来说也是一次复习。  
  
2.什么是Autocad VBA？  
VBA是Visual Basic for Applications的英文缩写，它是一个功能强大的开发工具，学好VBA可以成倍甚至成百、成万倍提高工作效率，在工作中，有很多任务仅用ACAD 命令不可能完成的，只要学好VBA就可以做到，相信到时候您一定会得到同事的佩服、老板的器重。  
  
3、VBA有多难？  
相信大家都知道Basic是的含义。应该承认，我的水平还不高，错误之处在所难免，如果大家发现错误一定要提出批评，以便及时更正。  
  
4、怎样学习VBA？  
介绍大家一个学习公式：信心+恒心=开心。仔细阅读本教程，完成例题，在学习的过程中一定要多思考，多想一些是什么、为什么。本教程将陆续发布在CAD世 界论坛上，您不需要付费就可以学习。本作者在此郑重承诺：关于本教程中有任何疑问，可以跟贴提问，只要有时间，本人一定会耐心解答。我不会发到任何人的邮 箱中，您自己在论坛上找就可以了，请不要再向我索要这份教程。  
  
5、现在我们开始编写第一个程序：画一百个同心圆  
第一步：复制下面的代码  
第二步：在模型空间按快捷键Alt+F8，出现宏窗口  
第三步：在宏名称中填写C100，点“创建”、“确定”  
第四步：在Sub c100()和End Sub之间粘贴代码  
第五步：回到模型空间，再次按Alt+F8，点击“运行”  
  
Sub c100()  
Dim cc(0 To 2) As Double ’声明坐标变量  
cc(0) = 1000 ’定义圆心座标  
cc(1) = 1000  
cc(2) = 0  
For i = 1 To 1000 Step 10 ’开始循环  
Call ThisDrawing.ModelSpace.AddCircle(cc, i \* 10) ’画圆  
Next i  
End Sub  
  
也许您还看不懂上面的代码，这没有关系，只要能把同心画出来就可以了，祝您成功。

Autocad VBA初级教程 (第二课 编程基础)  
  
下面是源码：  
Sub c100()  
Dim cc(0 To 2) As Double ’声明坐标变量  
cc(0) = 1000 ’定义圆心座标  
cc(1) = 1000  
cc(2) = 0  
For i = 1 To 1000 Step 10 ’开始循环  
Call ThisDrawing.ModelSpace.AddCircle(cc, i \* 10) ’画圆  
Next i  
End Sub  
  
先看第一行和最后一行：  
Sub C100()  
……  
End Sub  
C100是宏的名称，也叫过程名称，当用户执行C100时程序将运行sub　和end sub之间的所有指令。  
  
第二行：  
Dim cc(0 To 2) As Double ’声明坐标变量  
后半段“’声明坐标变量”自动变为绿色字体，它是代码语句的注释，它不会影响程序运行，它的作用是告诉阅读者程序员的想法。对于简单的程序，一般不需要写注释，如果要编写非常复杂的程序，最好要多加注释，越详细越好，对于程序员来说，这是一个好习惯。  
电脑真正编译执行的是这条语句：Dim cc(0 To 2) As Double  
它的作用就是声明变量。  
Dim是一条语句，可以理解为计算机指令。  
它的语法：Dim变量名 As 数据类型  
本例中变量名为CC，而括号中的0 to 2声明这个CC是一个数组，这个数组有三个元素：CC(0)、CC(1)、CC(2)，如果改为CC(1 to 3)，则三个元素是CC(1)、CC(2)、CC(3)，有了这个数组，就可以把坐标数值放到这个变量之中。  
Double是数据类型中的一种。ACAD中一般需要定义坐标时就用这个数据类型。在ACAD中数据类型的有很多，下面两个是比较常用的数据类型，初学者要有所理解。  
Long（长整型），其范围从 -2,147,483,648 到 2,147,483,647。  
Variant 它是那些没被显式声明为其他类型变量的数据类型，可以理解为一种通用的数据类型，这是最常用的。  
  
下面三条语句  
cc(0) = 1000 ’定义圆心座标  
cc(1) = 1000  
cc(2) = 0  
它们的作用是给CC变量的每一个元素赋,值其顺序是X、Y、Z坐标。  
  
  
For i = 1 To 1000 Step 10 ’开始循环  
……  
Next i ’结束循环  
这两条语句的作用是循环运行指令，每循环一次，i值要增加10，当i加到 1000时，结束循环。  
i也是一个变量，虽然没有声明i变量，程序还是认可的，VB不是C语言，每用一个变量都要声明，不声明就会报错。简单是简单了，这样做也有坏处，如果不小心打错了一个字母，程序不会报错，如果程序很长，那就会出现一些意想不到的错误。  
step后面的数值就是每次循环时增加的数值，step后也可以用负值。  
例如：For i =1000 To 1 Step -10   
很多情况下，后面可以不加step 10  
如：For i=1 to 100，它的作用是每循环一次i值就增加1  
Next i语句必须出现在需要结束循环的位置，不然程序没法运行。  
  
下面看画圆命令：  
Call ThisDrawing.ModelSpace.AddCircle(cc, i \* 10)  
Call语句的作用是调用其他过程或者方法。  
ThisDrawing.ModelSpace是指当前CAD文档的模型空间  
AddCircle是画圆方法  
Addcicle方法需要两个参数：圆心和半径  
CC就是圆心坐标，i\*10就是圆的半径，本例中，这些圆的半径分别是10、110、210、310……  
  
本课到此结束，下面请完成一道思考题：  
1.以(4,2)为圆心，画5个同心圆，其半径为1-5

Autocad VBA初级教程 (第三课 编程基础二)  
  
  
有人提出了下面的问题：  
绘制三维多段线时X、Y值在屏幕上用鼠标选取，Z值用键盘输入  
本课将讲解这个问题。  
  
为了简化程序，这里用多条直线来代替多段线。以下是源码：  
Sub myl()  
Dim p1 As Variant ’申明端点坐标  
Dim p2 As Variant  
p1 = ThisDrawing.Utility.GetPoint(, "输入点:") ’获取点坐标  
z = ThisDrawing.Utility.GetReal("Z坐标:") ’用户输入Z坐标值  
p1(2) = z ’将Z坐标值赋予点坐标中  
On Error GoTo Err\_Control ’出错陷井  
Do ’开始循环  
p2 = ThisDrawing.Utility.GetPoint(p1, vbCr & "输入下一点:") ’获取下一个点的坐标  
z = ThisDrawing.Utility.GetReal("Z坐标:") ’用户输入Z坐标值  
p2(2) = z ’将Z坐标值赋予点坐标中  
Call ThisDrawing.ModelSpace.AddLine(p1, p2) ’画直线  
p1 = p2 ’将第二点的端点保存为下一条直线的第一个端点坐标  
Loop  
Err\_Control:  
End Sub  
  
先谈一下本程序的设计思路：  
1、获取第一点坐标  
2、输入第一点Z坐标  
3、获取第二点坐标  
4、输入第二点Z坐标  
5、以第一、二点为端点，画直线  
6、下一条线的第一点=这条线的第二点  
7、回到第3步进行循环  
如果用户没有输入坐标或Z值，则程序结束。  
  
首先看以下两条语句：  
p1 = ThisDrawing.Utility.GetPoint(, "输入点:") ‘获取点坐标  
……  
p2 = ThisDrawing.Utility.GetPoint(p1, vbCr & "输入下一点:") ’获取下一个点的坐标  
这两条语句的作用是由用户输入点用鼠标选取点坐标，并把坐标值赋给p1、p2两个变量。ThisDrawing.Utility.GetPoint（）在 ACAD中这是最常用的方法之一，它需要两个参数，在逗号前面的参数应该是一个点坐标，它的作用是在屏幕上画一条线，前一个端点位于点坐标位置，后一个端 点跟随鼠标移动，逗号之前可以什么都不填，这时没有线条会跟随鼠标移动，但逗号必须保留。  
逗号后面使用一串字符，程序在命令行显示这串字符，这不难理解。  
VbCr通常代表一个回车符，而在这个语句中，它的作用是在命令行不显示“命令：”  
&的作用是连接字符。举例：  
“爱我中华 ”&”抵制日货 ”&”从我做起”  
  
z = ThisDrawing.Utility.GetReal("Z坐标:") ’用户输入Z坐标值  
由用户输入一个实数  
  
On Error GoTo Err\_Control ’出错陷井  
……  
Err\_Control:  
On Error是出错陷井语句，在程序出错时将执行On Error 后面的语句  
GoTo Err\_contorl 是程序跳转语句，它的作用是在程序中寻找Err\_control:，并执行这一行后面的语句，本例中Err\_Control:后就是结束宏，所以只要出现错误，程序中止。  
  
Do ’开始循环  
……  
Loop ‘结束循环  
这个循环就历害了，它会无休止地进行循环，好在本例中已经有了一个出错陷井，当用户输入回车时，由于程序没有得到点或坐标，程序出错，跳出循环，中止程 序。如果要人为控制跳出循环，可以在代码中用Exit Do语句跳出循环。在For 变量 和Next 变量之间如果要跳出循环，那么只要在循环体内加一个Exit for 就可以跳出循环，关于这方面的例程以后会讲到。  
  
Call ThisDrawing.ModelSpace.AddLine(p1, p2) ’画直线  
画直线方法也是很常用的，它的两个参数是点坐标变量  
  
本课到此结束，请做思考题：  
连续画圆，每次要求用户输入圆心、半径，当用户不再输入圆心或半径时程序才退出

Autocad VBA初级教程 (第四课 程序的调试和保存)  
  
  
人非圣贤，孰能无过，初学者在编写复杂程序时往往会出现一些意想不到的错误，所以程序的调试显得尤为重要，随着学习的深入，以后我们需要经常进行程序调试。事实上，对于那些资深程序员来说，调试程序也是一项不可或缺的重要工作。  
  
首先，在程序输入阶段，应该充分利用VBA编辑器的智能功能。当你在写代码时，输入一些字母后，编辑器可以自动列出合适的语句、对象、函数供你选择，可以 用上下键选择，然后按TAB键（它位于“Q”键左边）确认。当输入一个回车符后程序会自动对这条语句进行分析，如果出现错误就会提示。  
我们经常碰到的麻烦是程序的运行结果和预计的不一样，一般我会这样做：首先要想一想可能是哪一个变量有问题，然后去监视这个变量(或表达式)，在程序合适的位置设置断点，这样可以使程序停下来看一看这个变量有没有按照我的设想在变化。下面我举一个简单的例子，先看源代码：  
sub test()  
for i=2 to 4 step 0.6  
next i  
end sub  
这是一个非常简单的循环，每一次循环i便会增加0.6，当循环3次后i值就变为4.4，但问题是每一次循环时i值变为多少？  
第一步：在菜单中选“调试”—“添加监视”，在表达试中填“i”，点击确定，这时你会看到临视窗口中会多一行。  
第二步：把光标移到代码窗口中的“next i”行，按一下“F9”，于是每当程序运行到这里时就会暂停了。  
好，一切就绪，请按F5执行程序，在监视窗口中C值立刻变为2，再按F5继续，C值为2.6，再按几次F5，直到程序结束，这样我们就成功监视了C值的变化。  
第三步：在next i行再按一次F9，清除断点。监视的表达式的右键菜单选择“删除监视”。  
另外，还可以用“逐语句”、“逐过程”、“运行到光标处”等方法进行调试，这些都在调试菜单中，操作比较简单，请读者自行领悟。  
  
到目前为止，我们所做的工程都是“嵌入式工程”，它只是嵌入在当前的Autocad图形文件中， 以后打开这个文件时代码才会加载，如果别的dwg文件也要使用，那就需要把代码导出为.bas文件，供其他dwg文件导入。在VBA编辑器的“文件”菜单 中有这两个功能，一试便知。  
ACAD VBA还有一种工程叫“通用式工程”，只要进入ACAD就可以运行，程序可以在不同用户、不同的图形文件中共享，但是由于VBA功能太强，有时候会出现一些意想不到的事情，所以在学习阶段请暂时不要这样做。  
  
本课结束，请做思考题；监视下列代码中的i和j的值，注意，此题虽然要监视2个变量，但是在代窗口中只要设置1个断点就足够了。  
sub test()  
for i=2 to 4 step 0.6  
for j=-5 to 2 step 5.5   
next j  
next i  
end sub

Autocad VBA初级教程 (第五课 画函数曲线)  
  
  
先画一组下图抛物线。  
下面是源码：  
Sub myl()  
Dim p(0 To 49) As Double ’定义点坐标  
Dim myl As Object ’定义引用曲线对象变量  
co = 15 ’定义颜色  
For a = 0.01 To 1 Step 0.02 ’开始循环画抛物线  
For i = -24 To 24 Step 2 ’开始画多段线  
j = i + 24 ’确定数组元素  
p(j) = i ’横坐标  
p(j + 1) = a \* p(j) \* p(j) / 10 ’纵坐标  
Next i ’至此p(0)-p(40)所有元素已定义，结束循环  
Set myl = ThisDrawing.ModelSpace.AddLightWeightPolyline(p) ’画多段线  
myl.Color = co ’设置颜色属性  
co = co + 1 ’改变颜色，供下次定义曲线颜色  
Next a  
End sub  
  
为了鼓励大家积极思考，从本课开始，我不再解释每一条语句的作用，只对以前没有提过的语句进行一些解释，也许你一时很难明白，建议用上一课提到的跟踪变量、添加断点的办法领悟每一条语句的作用，如果有问题不懂请跟贴提问。  
在跟踪变量p时请在跟踪窗口中单击变量p前的+号，这样可以看清数组p中每一个元素的变化。  
  
ACAD没有现成的画抛物线命令，我们只能用程序编写多段线画近似抛物线。理论上，抛物线的X值可以是无限小、无限大，这里取值范围在正负24之间。  
  
程序第二行：Dim myl As Object ’定义引用曲线对象变量  
Object也是一种变量类型，它可以把变量定义为对象，本例中myl变量将引用多段线，所以要定义为Objet类型。  
  
看画多段线命令：  
Set myl = ThisDrawing.ModelSpace.AddLightWeightPolyline(p) ’画多段线  
其中括号中的p是一个数组，这个数组的元素数必须是偶数，每两个元数作为一个点坐标。  
等号前面部分“Set myl”的作用就将myl变量去引用画好的多段线。  
myl.Color = co ’设置颜色属性。在ACAD中，颜色可以用数字表示，本例中co会增值，这样就会有五彩缤纷的效果。  
  
本课第二张图：正弦曲线，下面是源码：  
Sub sinl()  
Dim p(0 To 719) As Double ’定义点坐标  
For i = 0 To 718 Step 2 ’开始画多段线  
p(i) = i \* 2 \* 3.1415926535897 / 360 ’横坐标  
p(i + 1) = 2 \* Sin(p(i)) ’纵坐标  
Next i  
ThisDrawing.ModelSpace.AddLightWeightPolyline (p) ’画多段线  
ZoomExtents ’显示整个图形  
End Sub  
  
  
p(i) = i \* 2 \* 3.1415926535897 / 360 ’横坐标  
横坐标表示角度，后面表达式的作用是把角度转化弧度  
ZoomExtents语句是缩放命令，它的作用是显示整个图形，消除图形以外的区域  
  
本课思考题：画一条抛物线：y=0.5\*x\*x+3，其中X取值范围在正负50之间

Autocad VBA初级教程 (第六课 数据类型的转换)  
  
  
上一节课我们用一个简单的公式把角度转化为弧度，这样做便于大家理解。不过VBA中有现成的方法可以转换数据类型。  
  
我们举例说明：  
jd = ThisDrawing.Utility.AngleToReal(30, 0)  
这个表达式把角度30度转化为弧度，结果是.523598775598299。  
AngleToReal需要两个参数，前面是表示要转换角度的数字，而后面一个参数可以取值为0-4之间的整数，有如下意义：  
0：十进制角度；1：度分秒格式；2：梯度；3：弧度；4：测地单位  
例：id= ThisDrawing.Utility.AngleToReal("62d30’ 10""", 1)  
这个表达式计算62度30分10秒的弧度  
  
再看将字符串转换为实数的方法：DistanceToReal  
需要两个参数，前一个参数是表示数值的字符串，后面可以取值1-5，表示数据格式，有如下意义：  
1：科学计数；2：十进制；3：工程计数——英尺加英寸；4：建筑计数——英尺加分数英寸；5：分数格式。  
例：以下表达式得到一个12.5的实数  
temp1 = ThisDrawing.Utility.DistanceToReal("1.25E+01", 1)  
temp2 = ThisDrawing.Utility.DistanceToReal("12.5", 2)  
temp3 = ThisDrawing.Utility.DistanceToReal("12 1/2", 5)  
而realtostring方法正好相反，它把一个实数转换为字符串。它需要3个参数  
第一个参数是一个实数，第二个参数表示数据格式，含义同上，最后一个参数表示精确到几位小数。  
temp1= ThisDrawing.Utility.RealToString(12.5, 1, 3)  
得到这个字符串：“1.250E+01”，  
  
下面介绍一些数型转换函数：  
Cint，获得一个整数，例：Cint(3.14159) ，得到3  
Cvar，获得一个Variant类型的数值，例：Cvar("123" & "00")，得到”12300”  
Cdate，转换为date数据类型，例：MyShortTime = CDate("11:13:14 AM")  
  
下面的代码可以写出一串数字，从000-099。  
  
Sub test()  
  
Dim add0 As String  
Dim text As String  
Dim p(0 To 2) As Double  
p(1) = 0 ’Y坐标为0  
p(2) = 0 ’Z坐标为0  
For i = 0 To 99 ’开始循环  
If i &lt; 10 Then ’如果小于10  
add0 = "00" ’需要加00  
Else ’否则  
add0 = "0" ’需要加0  
End If  
text = add0 & CStr(i) ’加零，并转换数据  
p(0) = i \* 100 ’X坐标  
Call ThisDrawing.ModelSpace.AddText(text, p, 4) ’写字  
Next i  
  
End Sub  
  
  
重点解释条件判断语句：  
If 条件表达式 Then   
……  
Else  
……  
End if  
  
如果满足条件那么程序往下执行，到else时不再往下执行，直接跳到End if后面  
如果不满足条件，程序跳到else后往下运行。  
  
Call ThisDrawing.ModelSpace.AddText(text, p, 4) ’写字  
这是写单行文本，需要三个参数，分别是:写的内容、位置、字高

Autocad VBA初级教程 (第七课 写文字)  
  
  
客观地说，ACAD写字功能不够历害，而用VBA可以使写字效率更高。比较正规的做法是把定义文字样式，用样式来控制文字的特性。我们还是用实例来学习，先看下面一段代码，它的作用是先创建一个文字样式，然后用这个文字样式写一段多行文本。  
  
Sub txt()  
  
Dim mytxt As AcadTextStyle ’定义mytxt变量为文本样式  
Dim p(0 To 2) As Double ’定义坐标变量  
p(0) = 100: p(1) = 100: p(2) = 0 ’坐标赋值  
Set mytxt = ThisDrawing.TextStyles.Add("mytxt") ’添加mytxt样式  
  
mytxt.fontFile = "c:\windows\fonts\simfang.ttf" ’设置字体文件为仿宋体  
mytxt.Height = 100 ’字高  
mytxt.Width = 0.8 ’宽高比  
mytxt.ObliqueAngle = ThisDrawing.Utility.AngleToReal(3, 0) ’倾斜角度（需转为弧度）  
  
  
ThisDrawing.ActiveTextStyle = mytxt ’将当前文字样式设置为mytxt  
Set txtobj = ThisDrawing.ModelSpace.AddMText(p, 1400, "{做到老，学到老}\P" & "此心自光明正大，过人远矣")  
  
txtobj.LineSpacingFactor = 2 ’指定行间距  
txtobj.AttachmentPoint = 3 ’右对齐（1为左对齐，2为居中）  
  
End Sub  
  
我们看这条语句  
Set mytxt = ThisDrawing.TextStyles.Add("mytxt")   
添加文本样式并赋值给mytxt变量，只需要一个参数：文本样式名  
  
fontfile、height、width、ObliqueAngle是文本样式最常用的属性  
  
Call ThisDrawing.ModelSpace.AddMText(p, 1400, "{做到老，学到老}\P" & "此心自光明正大，过人远矣")  
这条语句是写文本，需要三个参数。第一个参数p是坐标，1400是宽度，最后一个参数是文本内容，其中\p是一个回车符  
  
扩大字符间距用\T数字，例：\T3abc，使文字abc的间距扩大3 部，n取值范围是0.75-3  
  
在论坛中有一个经常被同好提及的问题：如何使用文字叠加。举例说明：123\S+0.12^-0.34  
\S是格式字符，^是分隔符，前面的数字在上，后面的数字在下。  
  
\C是颜色格式字符，C后面跟一个数字表示颜色  
  
\A是对齐方式，\A0，\A1,\A2分别表示底部对齐、中间对齐和顶部对齐

Autocad VBA初级教程 (第八课:图层操作)  
  
  
先简单介绍两条命令：  
  
1、这条语句可以建立图层：  
ThisDrawing.Layers.Add("新建图层")  
在括号中填写图层的名称。  
  
2、设置为当前的图层  
ThisDrawing.ActiveLayer=图层对象  
注意，等号右边的变量不能用图层名称，必须使用一个有效的图层变量  
  
以下一些属性在图层比较常用：  
LayerOn 打开关闭  
Freeze 冻结  
Lock锁定  
Color 颜色  
Linetype 线型  
  
  
看一个例题：  
1、先在已有的图层中寻找一个名为“新建图层”的图层  
2、如果找到这个图层，显示该图层的信息，并提示用户是否需要设置为当前图层，如果用户确认，则设置为当前图层。  
3、如果图层没有找到，新建一个名为“新建图层”的图层，设置为黄色，HIDDEN线型，并把这个图层设置为当前图层  
  
Sub mylay()  
  
Dim lay0 As AcadLayer ’定义作为图层的变量  
Dim lay1 As AcadLayer  
  
findlay = 0 ’寻找图层的结果的变量，0没有找到，1找到  
  
For Each lay0 In ThisDrawing.Layers ’在所有的图层中进行循环  
  
If lay0.Name = "新建图层" Then ’如果找到图层名  
findlay = 1 ’把变量改为1标志着图层已经找到  
msgstr = lay0.Name + "已经存在" + vbCrLf  
msgstr = msgstr + "图层状态：" + IIf(lay0.LayerOn = True, "打开", "关闭") + vbCrLf  
msgstr = msgstr + "图层" + IIf(lay0.Freeze = True, "已经", "没有") + "冻结" + vbCrLf  
msgstr = msgstr + "图层" + IIf(lay0.Lock = True, "已经", "没有") + "锁定" + vbCrLf  
msgstr = msgstr + "图层颜色号：" + CStr(lay0.Color) + vbCrLf  
msgstr = msgstr + "图层线型：" + lay0.Linetype + vbCrLf  
msgstr = msgstr + "图层线宽：" + CStr(lay0.Lineweight) + vbCrLf  
msgstr = msgstr + "打印开关" + IIf(lay0.Plottable = False, "关闭", "打开") + vbCrLf + vbCrLf  
msgstr = msgstr + "是否设置为当前图层？"  
If MsgBox(msgstr, 1) = 1 Then ’如果用户点击确定  
If Not lay0.LayerOn Then lay0.LayerOn = True ’打开  
ThisDrawing.ActiveLayer = lay0 ’把当前图层设为已经存在的图层  
End If  
Exit For ’结束寻找  
End If  
Next lay0  
  
If findlay = 0 Then ’没有找到图层  
Set lay1 = ThisDrawing.Layers.Add("新建图层") ’增加一个名为“临时图层”的图层  
lay1.Color = 2 ’图层设置为黄色  
  
ltfind = 0 ’找到线型的标志，0没有找到，1找到  
For Each entry In ThisDrawing.Linetypes ’在现有的线型中进行循环  
If StrComp(entry.Name, "HIDDEN") = 0 Then ’如果线型名为"HIDDEN"  
ltfind = 1 ’标志为已找到线型  
Exit For ’退出循环  
End If  
Next entry ’结束循环  
  
If ltfind = 0 Then ’没有找到线型  
ThisDrawing.Linetypes.Load "HIDDEN", "acadiso.lin" ’加载线型  
End If  
lay1.Linetype = "HIDDEN" ’设置线型  
  
ThisDrawing.ActiveLayer = lay1 ’将当前图层设置为新建图层  
End If  
  
End Sub  
  
在寻找图时时我们用到for each……next 语句  
它的语法是这样的：  
For Each 变量 In 数组或集合对象  
……  
exit for   
……  
next 变量  
它的作用是在数组或集合对象中进行循环，每循环一次，变量就成为数组或集合对象中的一个元素。本例在所有的图层对象中进行循环，每循环一次layo变量就代表一个图层  
在循环体中遇到exit for 语句则退出循环，如果没有 exit for，循环将在所有的元素都操作一遍后结束。  
  
If lay0.Name = "新建图层" Then  
lay0.name代表这处图层的图层名  
  
IIf(lay0.LayerOn = True, "打开", "关闭")  
这是一个简单判断语句，语法如下：  
iif（判断表达式，返回值1，返回值2）  
当判断表达式成立，函数值=返回值1，如果表达式不成立，函数值=2  
  
MsgBox(msgstr, 1)   
Mgbox显示一个对话框，第一个参数是对话框显示的内容  
第二个参数可以控制对话框上的按钮。  
0 只有确认按钮  
1 确认、取消  
2 终止、重试、忽略  
3 是、否、取消  
4 是、否  
MsgBox获得值如下：  
确认：1  
取消：2  
终止：3  
重试：4  
忽略：5  
是：6  
否7  
初学者不需要死记硬背，能有所了解就行了  
  
ACAD图层中最麻烦的就是线型问题了，本例先寻找一个HIDDEN线型，如果找不到就加载这个线型，用这条语句：  
ThisDrawing.Linetypes.Load "HIDDEN", "acadiso.lin"   
ThisDrawing.Linetypes.Load后需要两个参数，一个是线型的名称，另外一个是线型文件的名称。

Autocad VBA初级教程 (第九课:创建选择集)   
  
1.在创建对象的同时可以直接引用.以前的例程中已经做过多次了,现在复习一下,看例程:先随机画300个圆,在画圆时直接引用,然后再把这些圆根本大小修改颜色.  
  
Sub c300()  
  
Dim myselect(0 To 300) As AcadEntity ’定义选择集数组  
Dim pp(0 To 2) As Double ’圆心坐标  
  
For i = 0 To 300 ’循环300次  
pp(0) = 3000 \* Rnd: pp(1) = 3000 \* Rnd: pp(2) = 0 ’设置圆心坐标  
Set myselect(i) = ThisDrawing.ModelSpace.AddCircle(pp, Rnd \* 30 + 1) ’画不同大小的圆  
Next i  
  
For i = 1 To 300  
If myselect(i).Radius &gt; 10 Then ’判断圆的直径是否大于10  
myselect(i).color = Int(255 \* Rnd + 1) ’大圆颜色改为随机数  
Else  
myselect(i).color = 0 ’小圆改为白色  
End If  
Next i  
  
ZoomExtents ’缩放到显示全部对象  
  
End Sub  
  
  
pp(0) = 3000 \* Rnd: pp(1) = 3000 \* Rnd: pp(2) = 0  
这一行实际上应该是三条语句,用三行合并为一行,用冒号分开  
rnd是随机数函数,它的数值为0-1之间的小数,3000\*rnd得到的数值就是在0-3000之间的随机数  
  
Set myselect(i) = ThisDrawing.ModelSpace.AddCircle(pp, Rnd \* 30 + 1)  
这条语句的作用是以pp点坐标为圆心,画一个圆,半径是1-30之间的随机数,赋值给myselect选择集.  
  
2.提标用户在屏幕中选取  
选择语句这样写:Set sset = ThisDrawing.SelectionSets.Add("ss1"),其中”ss1”是一个选择集名称,这个参数可以随意写一个,注意不要重复就可以了.  
下面的例程是让用户选择对象,然后把选中的对象改为绿色,最后把选择集删除  
  
Sub mysel()  
  
Dim sset As AcadSelectionSet ’定义选择集对象  
Dim element As AcadEntity ’定义选择集中的元素对象  
  
Set sset = ThisDrawing.SelectionSets.Add("ss1") ’新建一个选择集  
sset.SelectOnScreen ’提示用户选择  
  
For Each element In sset ’在选择集中进行循环  
element.color = acGreen ’改为绿色  
Next  
sset.Delete ’删除选择集  
  
End Sub  
  
3.选择全部对象  
用select方法,参数为acSelectionSetAll ,看例程,这个程序选择全部对象,显示选中的对象,并计算对象数.  
Sub allsel()  
Dim sel1 As AcadSelectionSet ’定义选择集对象  
Set sel1 = ThisDrawing.SelectionSets.Add("s") ’新建一个选择集  
Call sel1.Select(acSelectionSetAll) ’全部选中  
sel1.Highlight (True) ’显示选择的对象  
sco= sel1.Count ’计算选择集中的对象数  
MsgBox "选中对象数:" & CStr(sco) ’显示对话框  
End Sub  
  
3.运用select方法  
上面的例题已经运用了select方法,下面讲一下select的5种选择方式:  
1:择全部对象(acselectionsetall)  
2.选择上次创建的对象(acselectionsetlast)  
3.选择上次选择的对象(acselectionsetprevious)  
4.选择矩形窗口内对象(acselectionsetwindow)  
5.选择矩形窗口内以及与边界相交的对象(acselectionsetcrossing)  
还是看代码来学习.其中选择语句是:  
Call sel1.Select(Mode, p1, p2)  
Mode已经定义为5,也就是选择矩形窗口内以及与边界相交的对象,p1和p2是两个点坐标,  
Sub selnew()  
Dim sel1 As AcadSelectionSet ’定义选择集对象  
Dim p1(0 To 2) As Double ’坐标1  
Dim p2(0 To 2) As Double ’坐标2  
p1(0) = 0: p1(1) = 0: p1(2) = 0 ’设置坐标1  
p2(0) = 300: p2(1) = 300: p2(2) = 0 ’设置坐标1  
Mode = 5 ’把选择模式存入mode变量中  
Set sel1 = ThisDrawing.SelectionSets.Add("sel3") ’新建一个选择集  
Call sel1.Select(Mode, p1, p2) ’选择对象  
sel1.Highlight (ture) ’显示已选中的对象  
End Sub

Autocad VBA初级教程 (第十课:画多段线和样条线)   
  
  
画二维多段线语句这样写：  
set lobj= ThisDrawing.ModelSpace.AddLightweightPolyline(fitpoint)  
AddLightweightPolyline后面需一个参数，存放顶点坐标的数组  
  
画三维多段线语句这样写：  
Set lobj = ThisDrawing.ModelSpace.Add3DPoly(fitpoint)  
Add3dpoly后面需一个参数，就是顶点坐标数组  
  
画二维样条线语句这样写：  
Set lobj = ThisDrawing.ModelSpace.AddSpline(fitPoints,startT, endT)  
Addspline后需要三个参数，第一个是拟合点数组，后面两个分别是起点切向和终点切向。  
  
下面看例题。这个程序是第三课例程的改进版。原题是这样的：  
绘制三维多段线时X、Y值在屏幕上用鼠标选取，Z值用键盘输入。  
细心的读者是否还记得这个小程序已经在第三课的回贴中发布了。先分析一下编程细路：  
用动态数组存放多段线的所有顶点坐标，获得新坐标后就把前一次画的多段线删除，画出多段线，并将线段第二点的端点保存为下一条线段的第一个端点坐标，以下是源码：  
  
Sub myl()  
Dim p1 As Variant ’申明端点坐标  
Dim p2 As Variant  
Dim l() As Double ’声明一个动态数组  
Dim templ As Object  
p1 = ThisDrawing.Utility.GetPoint(, "输入点:") ’获取点坐标  
z = ThisDrawing.Utility.GetReal("Z坐标:") ’用户输入Z坐标值  
p1(2) = z ’将Z坐标值赋予点坐标中  
ReDim l(0 To 2) ’定义动态数组  
l(0) = p1(0)  
l(1) = p1(1)  
l(2) = z  
On Error GoTo Err\_Control ’出错陷井  
Do ’开始循环  
p2 = ThisDrawing.Utility.GetPoint(p1, vbCr & "输入下一点:") ’获取下一个点的坐标  
z = ThisDrawing.Utility.GetReal("Z坐标:") ’用户输入Z坐标值  
p2(2) = z ’将Z坐标值赋予点坐标中   
lub = UBound(l) ’获取当前l数组中元的元素个数  
ReDim Preserve l(lub + 3)  
For i = 1 To 3  
l(lub + i) = p2(i - 1)  
Next i  
If lub &gt; 3 Then  
templ.Delete ’删除前一次画的多段线  
End If  
Set templ = ThisDrawing.ModelSpace.Add3DPoly(l) ’画多段线  
p1 = p2 ’将第二点的端点保存为下一条直线的第一个端点坐标  
Loop  
Err\_Control:  
End Sub  
  
我们学一学动态数组，这是非常有用的。所谓动态数组，简单讲就是数组元素可以改变的特殊数组。  
这样定义数组：Dim l( ) As Double   
赋值语句：  
ReDim l(0 To 2)   
l(0) = p1(0)  
l(1) = p1(1)  
l(2) = z  
重新定义数组元素语句：  
lub = UBound(l) ’先要获取当前l数组中元的元素个数,用ubount函数计算。  
ReDim Preserve l(lub + 3)  
重定义时数组中要加三个元素，用以存放坐标值，所以定义元素为lub+3。给数值赋值的语句是一样的。  
  
再看画多段线语句：  
Set templ = ThisDrawing.ModelSpace.Add3DPoly(l) ’画多段线  
在最后面括号中的l就是存放点坐标的动态数组，这条语句在画多段线的同时把多段线作为对象赋值给templ 变量，这样下面的删除语句就可以直接引用这个变量了。  
  
删除语句：  
templ.Delete  
因为已经知道对象是templ，所以只要在对象后面用delete方法就可以了。  
  
下面的例程更加实用些，它的功能是把样条线转化为多段线。编程思路：用户选择一根样条线，计算样条线的拟合点，然后把所有的拟合点存放到数组中，最后用这个数组画多段线。  
  
Sub sp2pl()  
Dim getsp As Object ‘获取样条线的变量  
Dim newl() As Double ‘多段线数组  
Dim p1 As Variant ‘获得拟合点点坐标  
ThisDrawing.Utility.GetEntity getsp, po, "本程序将样条曲线转为多段线。请选择样条曲线"  
sumctrl = getsp.NumberOfControlPoints ‘计算样条线中一共有多少拟合点  
ReDim newl(0 To sumctrl \* 3 - 1) ‘重定义数组   
For i = 0 To sumctrl - 1 ‘开始循环，  
p1 = getsp.GetControlPoint(i) ‘把拟合点坐标存到p1变量中  
For j = 0 To 2  
newl(i \* 3 + j) = p1(j)  
Next j  
Next i  
Set templ = ThisDrawing.ModelSpace.Add3DPoly(newl) ‘画样条线  
End Sub  
  
下面的语句是让用户选择样条线：  
ThisDrawing.Utility.GetEntity getsp, po, "本程序将样条曲线转为多段线。请选择样条曲线"  
ThisDrawing.Utility.GetEntity 后面需要三个参数：  
第一个getsp用于存放获得用户选择的对象变量，第二个po是用户鼠标点下时的位置，第三个是一段字符串，显示在提示栏。

Autocad VBA初级教程 (第十一课:动画基础)   
  
说实话，用ACAD玩动画绝对比不上专业的三维动画软件，不过这年头自称精通CAD的工程师太多了，而学会用ACAD玩动画一定能提高您的竞争实力， 想象一下您用ACAD动画向老板演示零件的装配方法……  
  
下面的例题是一个最简单的动画，由用户选择一个要移动的对象，然后定位起点和终点，程序会慢慢地把对象移动。  
  
移动方法：object.move 起点坐标,端点坐标  
  
Sub testmove()  
  
Dim p0 As Variant ’起点坐标  
Dim p1 As Variant ’终点坐标  
Dim pc As Variant ’移动时起点坐标  
Dim pe As Variant ’移动时终点坐标  
Dim movx As Variant ’x轴增量  
Dim movy As Variant ’y轴增量  
Dim getobj As Object ’移动对象  
Dim movtimes As Integer ’移动次数  
  
ThisDrawing.Utility.GetEntity getobj, po, "请选择移动对象"  
p0 = ThisDrawing.Utility.GetPoint(, "起点:")  
p1 = ThisDrawing.Utility.GetPoint(p0, "终点:")  
pe = p0  
pc = p0  
motimes = 3000  
movx = (p1(0) - p0(0)) / motimes  
movy = (p1(1) - p0(1)) / motimes  
For i = 1 To motimes  
pe(0) = pc(0) + movx  
pe(1) = pc(1) + movy  
getobj.Move pc, pe ’移动一段  
getobj.Update ’更新对象  
Next  
End Sub  
  
先让用户选择移动的对象、起点、终点，motimes动画是设置动画帧数，这个例程是让对象移动3000小段，movx和movy是每一段移动的距离，然后进行一个3000次的循环，每一次循环移动一小段距离。  
  
看第二个例题：做一个小轮子在一条山坡上运动的动画，山坡为正弦曲线。请注意，应该用偏移法获得轮圆心的轨迹才是正确的。编程思路：先画好圆和横轴，然后画山坡，偏移获得移动轨迹线，放到隐藏图层中，然后根据轨迹线移动圆和轴，在移动之前先旋转轴线。  
  
旋转方法：object. rotate 基点,角度  
偏移方法： object.offset(偏移量)  
  
Sub moveball()  
Dim ccball As Variant ’圆  
Dim ccline As Variant ’圆轴  
Dim cclinep1(0 To 2) As Double ’圆轴端点1  
Dim cclinep2(0 To 2) As Double ’圆轴端点2  
Dim cc(0 To 2) As Double ’圆心  
Dim hill As Variant ’山坡线  
Dim moveline As Variant ’移动轨迹线  
Dim lay1 As AcadLayer ’放轨迹线的隐藏图层  
Dim vpoints As Variant ’轨迹点  
Dim movep(0 To 2) As Double ’移动目标点坐标  
cclinep1(0) = -0.1: cclinep2(0) = 0.1 ’定义圆轴坐标  
Set ccline = ThisDrawing.ModelSpace.AddLine(cclinep1, cclinep2) ’画直线  
Set ccball = ThisDrawing.ModelSpace.AddCircle(cc, 0.1) ’画半径为0.1的圆  
Dim p(0 To 719) As Double ’申明正弦线顶点坐标  
For i = 0 To 718 Step 2 ’开始画多段线  
p(i) = i \* 3.1415926535897 / 360 ’横坐标  
p(i + 1) = Sin(p(i)) ’纵坐标  
Next i   
Set hill = ThisDrawing.ModelSpace.AddLightWeightPolyline(p) ’画正弦线即山坡曲线  
hill.Update ’显示山坡线  
moveline = hill.Offset(-0.1) ’球心运动轨迹线  
vpoints = moveline(0).Coordinates ’获得规迹点  
Set lay1 = ThisDrawing.Layers.Add("hidelay") ’创建名为"hidelay"的图层  
lay1.LayerOn = False ’关闭图层  
moveline(0).Layer = "hidelay" ’将轨迹线放到关闭的图层中  
ZoomExtents ’显示整个图形  
For i = 0 To UBound(vpoints) - 1 Step 2  
movep(0) = vpoints(i) ’计算移动的轨迹  
movep(1) = vpoints(i + 1)  
ccline.Rotate cc, 0.05 ’旋转直线  
ccline.Move cc, movep ’移动直线  
ccball.Move cc, movep ’移动圆  
cc(0) = movep(0) ’把当前位置作为下次移动的起点  
cc(1) = movep(1)  
For j = 1 To 50000 ’这次循环的目的是让小球移动得慢一点,循环量应根据自已的电脑速度设置  
j = j \* 1  
Next j  
ccline.Update ’更新  
Next i  
End Sub  
  
本课结束，请做思考题：在一条直线上画两个同样大小的圆，然后让这两个圆沿直线相向而行，到互换位置后停下来，尺寸自定