

VBA 入门指南

—献给在这里得到帮助但看不懂代码的朋友（已建电梯）

VBA 入门指南	1
1、【2009.1.16】初识 VBA	3
2、初识 VBA(续).....	4
3、便于对“类” 进行理解的只言片语	5
4、【2009.1.19】代码起步——关键字	7
4、【2009.1.20】代码起步——关键字（续）	8
5、【2009.1.21】代码起步——关键字（续）	11
6、【2009.1.22】代码起步——关键字（续）	15
7、【2009.1.27】代码起步——关键字(续).....	17
8、【2009.1.28】代码起步——关键字（续）	19
9、【2009.1.29】代码起步——关键字（续）	20
10、【2009.1.30】代码起步——我的程序.....	23
11、【2009.1.31】代码起步——我的程序（续）	25
12、【2009.2.1】代码起步——我的程序（续）	27
13、【2009.2.2】代码起步——我的程序（续）	29
14、【2009.2.3】代码起步——我的程序（续）	32
15、【2009.2.4】代码起步——我的程序（续）	35
16、【2009.2.6】代码起步——我的程序（续）	40
17、【2009.2.7】代码起步——我的程序（续）	43

此为 EXCELHOME 的 yagi2008 先生在 2009 年年关不辞辛劳进行的教学成果，
在此全面摘录下来，做成电子书，以方便大家。



yagi2008

为了方便大家查看本教程，特制作了索引：

【2009. 1. 16】初识 VBA [http://club.excelhome.net/viewth ...
d=701638#pid2488755](http://club.excelhome.net/viewth...d=701638#pid2488755)

【2009. 1. 17】初识 VBA(续) [http://club.excelhome.net/viewth ...
d=701638#pid2490947](http://club.excelhome.net/viewth...d=701638#pid2490947)

【2009. 1. 18】初识 VBA(续) [http://club.excelhome.net/viewth ...
d=701638#pid2492976](http://club.excelhome.net/viewth...d=701638#pid2492976)

【2009. 1. 19】代码起步——关键字 [http://club.excelhome.net/viewth ... d=701638#pid2493610](http://club.excelhome.net/viewth...d=701638#pid2493610)

【2009. 1. 20】代码起步——关键字

(续) <http://club.excelhome.net/viewth ... d=701638#pid2496077>

【2009. 1. 21】代码起步——关键字

(续) <http://club.excelhome.net/viewth ... d=701638#pid2498626>

【2009. 1. 22】代码起步——关键字

(续) <http://club.excelhome.net/viewth ... d=701638#pid2500468>

【2009. 1. 27】代码起步——关键字

(续) <http://club.excelhome.net/viewth ... d=701638#pid2504890>

【2009. 1. 28】代码起步——关键字

(续) <http://club.excelhome.net/viewth ... d=701638#pid2505205>

【2009. 1. 29】代码起步——关键字

(续) <http://club.excelhome.net/viewth ... d=701638#pid2506201>

【2009. 1. 30】代码起步——我的程序 [http://club.excelhome.net/viewth... d=701638#pid2507007](http://club.excelhome.net/viewth...d=701638#pid2507007)

【2009. 1. 31】代码起步——我的程序

(续) <http://club.excelhome.net/viewth ... d=701638#pid2507932>

【2009. 2. 1】代码起步——我的程序

(续) <http://club.excelhome.net/viewth ... d=701638#pid2509234>

【2009. 2. 2】代码起步——我的程序(续)

<http://club.excelhome.net/viewthread.php?tid=390711&page=11&fromuid=701638#pid2511074>

【2009. 2. 3】代码起步——我的程序(续)

<http://club.excelhome.net/viewthread.php?tid=390711&page=11&fromuid=701638#pid2513181>

【2009. 2. 4】代码起步——我的程序(续)

<http://club.excelhome.net/viewthread.php?tid=390711&page=12&fromuid=701638#pid2515382>

【2009. 2. 6】代码起步——我的程序(续)

<http://club.excelhome.net/viewthread.php?tid=390711&page=14&fromuid=701638#pid2519770>

来这里几天了，回复了一些帖子，但仔细想想，发现很多朋友虽然得到了版主、高手们的帮助，也拿到了可以实现其所要求的功能的范例 XLS 文件，但往往并不能理解其中的原理和代码，更不能举一反三化为自己的东西加以应用。为了能体现“授人以鱼不如授人以渔”的技术共享精神，特开此帖，希望本帖的

一些介绍能带着您进入 VBA 的大门。当然由于水平有限，难免出现错误，希望各位版主、高手们加以批评指正——写在开贴之初！

1、【2009.1.16】初识 VBA

今天，阳光明媚，清晨刚到办公室，泡好香茗一杯，便开始构思，该如何组织这篇帖子，看来我是作好了打持久战的准备，不管最终出来的是又臭又长的裹脚布还是无厘头的流水账，只要能对您带来一点帮助或者乐趣，我就很满意了。

既然题目是初识 VBA，那么我想先给大家一个直观的感受——什么是 VBA？，往往在一些 EXCEL 的教材中，都会讲：VBA 是一种自动化语言，它可以使常用的程序自动化，可以创建自定义的解决方案。就我看书学习的经验来说，我感到这样的定义式的描述带有很典型的中国式教材的风格，然而却不能让人立即理解，那么这样的定义有什么用呢？我给 VBA 下的定义是：VBA 是一些代码的组合。可能这个定义没有上面教材的定义那么严谨和准确，但我想你一看就应该能明白，原来 VBA 就是写代码，或者说就是写程序。（如果掌握了 VBA，居然我也能写程序了~~惊喜吧？）

那么你用了这么久的 EXCEL，可能从来没在 EXCEL 的 XLS 文件中看到过代码，难道就写在单元格里？那我就带你去找找，VBA 的代码藏在哪里吧，这也正式我今天要谈的第二个问题。当你打开一个 EXCEL 文件，你只能看到由许多单元格组成的工作表，这时，你可以通过【视图】菜单的【工具栏】项目，打开一个工具栏，名字叫“控件工具箱”，从左往右属第三个按钮，名字叫“查看代码”，点一下，就可以打开 VBA 代码编辑窗口。更简单的方法是，通过【ALT】+【F11】的快捷键，打开 VBA 代码编辑窗口，看到了吧，原来代码就是藏在这里的哦~~~

怎么写代码呢？你也太心急了点吧，你还没掌握命令、语法、关键字、对象、事件、属性等等内容，就想写代码？一口是吃不成个胖子的，还是让我来带你先认识下这里的每个角落吧，就像进了一间陌生的屋子，总归每个角落都得转到吧？最上面的几行，和 EXCEL 以及每一个微软平台下的应用程序都一样，总归是标题栏、菜单栏、工具栏等，接下来，看左边，有一个纵栏，写着“工程——VBAProject”，这里是工程资源管理窗口，里面通过树形列表展示了这个 XLS 文件（或者标准的应该叫“工作簿”）所包含的各个工作表对象以及工作簿对象本身。有什么用呢？你双击任何一个 SHEET，比如 SHEET1，试试？是不是在右侧打开了一个文档编辑窗口？那么你就能明白，如果是 SHEET1 的代码，就应该写在这里了，那么 SHEET2 的呢？当然就写在双击 SHEET2 后打开的窗口里咯~那么 SHEET3 呢？你烦不烦啊~~~人家都明白了!!! 再看看右侧的文档窗口吧，顶上有两个下拉列表框，一个写着“通用”，一个写着“声明”，什么东东啊？左边的那个是对象列表框，可以让你选择在这个 SHEET 里面包含的所有对象，右边的那个是过程列表框，可以让你选择和左侧当前选中的对象关联的系统或自定义过程。那么什么是对象？什么又是过程呢？今天没时间了，什么？领导要我马上去开会？不会加工资了吧？哈哈，明天继续！

2、初识 VBA(续)

看到这么多朋友的支持和期待，倍感压力啊！怎么感觉写不好就对不起观众呢？哎，书到用时方恨少，大家将就着看吧……

昨天我们在 VBA 代码编辑器里转了一圈，是不是有别有洞天的感觉啊？可是，这里对于大家都是陌生的环境，到底该怎么用这里的每一个工具呢？别心急，且听我慢慢道来。在讲代码之前，我首先必须介绍一下 OOP 的概念和基本知识。去去去，怎么你也学着那些教材里挑些专用名词来忽悠我们啊？呵呵，因为这样才能体现作者，也就是本人的水平啊！（别扔鸡蛋，说你呢！下次不说了，还不行吗？）所谓 OOP 就是指 Object Oriented Programming, 即面向对象的程序设计方法，这种程序设计思想主要是相对于以前的面向过程的程序设计方法而言的。下面我就给大家讲个故事，很久很久以前，写程序还是很专业很神秘的一种技术活，这个时候的程序员写出来的东西，是必须要按照他给你安排好的步骤来执行的，比方说，你运行一个程序，必须先输入一条打开数据表的命令，然后才能对这个数据表进行操作，否则就会报错，甚至对于有些设计不完善的程序就会崩溃。这样写出来的程序一个是缺乏灵活性，难道作为用户就不能自己决定先浏览还是先打印？另一方面，对于那时候写出来的程序，如果将来要修改或扩充功能，那将牵一发而动全身，也为在设计这些功能的时候是一步一步安排好的，如果要修改后面的步骤，必然会影响到前面的代码，这样导致了程序员非常累。（所以高薪啊~~其实也不过是重复劳动，呵呵）。后来，出来了一个聪明人，写出了一个叫“Smalltalk”的语言，这个语言本身并没有什么特别的地方，但其中其中表现出来的接近现实形态描述的对象思想却让整个软件设计理论提升了一个新的层次。说这些和我有什么关系啊？呵呵，下面的东西就和你有关了，也为现在的基于可视化架构的程序设计基本都引入了 OOP 的思想，所以如果你将 OOP 中最主要的概念搞清楚了，那么将对以后的学习奠定扎实的基础。

面向对象的程序设计思想中主要包含四个概念。对象、事件、方法和属性。别看这些概念这么复杂，我给大家举个例子就明白了。对象是什么？就像现实中看得见摸得着的东西一样，是实实在在的东西。比如，一个苹果，就可以作为一个对象，围绕对象概念，延伸了一个“类”的概念，什么是类呢？苹果就是一个类。晕，刚才还说苹果是对象呢，怎么一会又成了类了？其实作为类而言的苹果的概念，就是指区别于梨和橘子的苹果类的概念，而不是具体指某一个苹果。红苹果、绿苹果都属于苹果这个类，哪怕烂苹果也是。（别扔啊！）我们在设计程序的时候，可以定义一个自己的类，这样以后要用到这类对象的时候，就可以通过这个类来派生出来，而不用每次都去写定义。还有个好处是，以后修改对象属性

的时候，就可以通过修改类的属性来达到同时修改通过这个类派生的所有对象的属性的目的。对象的基本概念还有三个，属性、方法和事件。苹果的颜色可以看作苹果这个对象的一个属性，苹果的大小也可以看作他的一个属性，你自己还能说出什么属性吗？恩，重量、好坏、产地等等。慢点，价格是不是啊？这个问题留给大家思考吧。从上面的例子可以看出，属性是描述对象在某一方面的特征的，通过不同属性的值就可以区别开不同的对象。方法呢？对于苹果来说“吃”就是方法，啥？吃苹果也是方法？恩，别馋啊，对苹果而言，吃就是方法，因为这个是可以作用于这个对象的，还能举个例子吗？比方相对于表格对象的方法就可以是打开、关闭、修改等。方法是用来作用于对象，使其产生某种变化的。事件是什么呢？事件就是被设计用来触发某些代码来自动执行的，这个就叫做事件驱动机制。还记得刚才我们谈到的面向过程的程序设计方法的缺点吗？对，代码的执行顺序是由设计者在设计时指定的。这种吃力不讨好的事情，劝大家以后还是少做做。我们有了事件驱动机制以后，这样的情况就可以避免了。现在，可以通过将代码写在可能发生的事件中的办法，由用户决定啥时候运行这些代码，只要程序运行时，用户的操作行为触发了相应的事件，那么该事件中定义的代码，就会被自动执行了。

好了，该吃午饭了，有朋友回复说最好图文并茂，现在讲的是理论，在以后讲到代码时候，会加上插图和动画的。明天见！

3、便于对“类”进行理解的只言片语

类是抽象的，不仅仅在程序中有，生活中到处都是。

你的衣服是按照春夏秋冬 分门别类整理存放的，到超市购物，商品也是按各种大类，小类摆放的

类 就是分类、种类的意思。

类 是人为的、按照一定的特性进行区分的一组对象（大白话就是“东西”）

你我也都是 一 “类”

生物类--》动物类--》人类 --》会使用电脑的那一类人--》会用 EXCEL 的一类人--》上 EH 的一类人

类还可以设计，创建，重复使用，这回不拿人说事了拿电脑吧

电脑粗分一下 台式机、笔记本。

但是电脑都没有 点烟（就是汽车上的点烟器）这个功能，但是我需要这样一个电脑，于是我就设想有这么一台，并且做了简单的设计图纸。

于是一个新类——点烟电脑 就设计好了（这就等于在类模块中写代码），

现在 “点烟电脑” 还只处于设计图纸中，我要想使用就必须创建，我把图纸交给厂家，

厂家按设计图生产。这我才能使用（这就等于用 NEW 关键字 实体化一个类）。

我的朋友也想要一个 点烟电脑，我就不用再设计了，直接叫厂家再生产一个就行了（再 NEW 一个，类的重复使用）。

整个过程当中设计图纸这一步是关键（在类模块中写代码）。

整个 Windows 系统中有很多种 “类”，用于实现各种功能。VBA 可以直接调用。

但有些功能是我们自己想出来的，我们就只有设计一个新类，然后再调用。

谢谢 LDY 老师的精彩注释!!!欢迎更多的版主、朋友发表自己的想法和见解。

快过年了，早晨去了趟超市，那个人叫挤啊！本来想早点回来的，结果午饭到了 2 点才吃到。打开电脑，看到好多朋友都在关心这个帖子，而且好几位朋友提出了自己的见解和疑惑，想到昨天讲了“类”，今天真的感到了好“累”。

因为大家都刚接触 VBA，可能光听我介绍还没有直观的感受，为了帮助大家理解，下面先介绍两个方法，可以让大家看到更多的东西。首先要介绍的是对象浏览器。打开这个对话框有三种方法。（中国的教材和教育模式下的考试，最喜欢列举完成一件事情的方法，我就纳闷，只要能解决问题，知道了一条路，干嘛还要打破沙锅问到底？）一个是通过【视图】菜单的【对象浏览器】项目，另一个是在工具栏上点击【对象浏览器】按钮。最方便的是按【F2】快捷键。打开了以后，大家看到了一个新的窗口，这里列举了三种信息，最上面的部分是用来搜索和查找信息的，中间的部分分了三栏来分别列举“库”、“类”和“成员”信息。最下面是提示和说明部分，用来说明当前选中对象的信息。为什么要介绍对象浏览器呢？大家通过这个工具可以搜索到和 EXCEL 以及 VBA 有关的各种各样的“类”和“对象”。顺便说一下，这里说的“成员”基本相当于对象、属性、方法和事件；类是成员的上级；很多类又组成了一个类库，在这里叫做“库”。以后大家如果有不明白的类，就可以在这里找到这个类有哪些对象、属性、方法和事件，以及他们之间是什么层级关系。其次，简单介绍下如何使用帮助。其实按照微软的体系，VBA 的帮助包括了两个部分，一个是本地脱机帮助，也就是安装完 OFFICE 软件就自带的帮助，这里只是简单的介绍了 VBA 的基本信息。另一个部分则被微软放在了 MSDN 里面，这是微软单独提供给微软平台的开发人员的独立帮助工具，其中包含了非常详细的各种编程信息、技巧和案例。通过仔细研读 MSDN 里的内容，可以快速掌握之前不了解的编程知识，并且 MSDN 总是被微软的工程师们不断更新并保持着最新的状态。

下面，我们就继续昨天的内容，接着讲面向对象的编程。什么？昨天的都已经忘记了？那就先回头去看看再继续吧。今天主要给大家介绍 VBA 中最常用到的一些关键字。什么是关键字，关键字就是那些在 OFFICE 软件里已经被定义好了特定含义和用法的字符串。比方说，“IF”这个字符串，表示“如果”的意思，在 VBA 里用来表示条件语句的起始，那么关键字和普通字符串有什么区别呢？关键字被规定为不可以用来表示变量名、过程名、函数名等用户定义的信息，只能根据系统规定的含义使用。也就是说，你如果有个变量起个名字叫“IF”，你写一条赋值语句“IF = 1”，这样系统在编译的时候就会提示一个错误，而不能通过编译和运行。想看看 VBA 里一共有多少个关键字吗？从明天开始，我们将开始一一给大伙讲解，欢迎继续关注。

对于 VBA，我觉得比较烦的一点儿是：对里面的对象（类）的用法不大好自

学与理解。原有的帮助常常是分断的、原则性的，缺乏针对性。没有一个系统的、全面的、有较强针对性的帮助内容！或许找一本书来看，您会发现：书的知识还不如帮助呢！所以至今还未深入学习 VBA。

4、【2009.1.19】代码起步——关键字

“同学们好！”，“老——师——好——”，“请坐下，把书翻到 25 页，今天我们讲……”，才三岁的儿子已经开始模仿和憧憬着学校的生活了~~哈哈，当我们回忆起儿时上课的情景，亲切吧？那就让我们重温一下这份感受吧。

今天开始，我们将正式进入代码的学习阶段。我多年以来的授课经验告诉我，学代码应该从关键字开始。下面列举了一些常见的关键字给大家先了解下。

【框架类关键字】

工程 (Project): 是指用于创建一个应用程序的文件的集合。

对象 (Object): 可控制的某个东西，例如窗体和控件。

窗体 (Form): 应用程序的用户界面。

控件 (Control): 指的是各种按钮、标签、文本框等。

属性 (Property): 是指对象的特征，如大小、标题或颜色。

工作表 (Worksheet): 指 EXCEL 文件里的工作表，例如 sheet1、sheet2 等。

模块 (Module): 指在 VBA 工程中存放独立于用户定义对象代码的容器。

过程 (Sub): 容纳和组织代码的限定符号，一般和 End Sub 联用，不返回结果。

函数 (Function): 容纳和组织代码的限定符号，一般和 End Function 联用，并返回结果。

【控件类关键字】

标签 (Label): 用来显示文本。

文本框 (Textbox): 用来提供给用户输入文本。

命令按钮 (CommandButton): 用来组织和提供程序功能。

列表框 (ListBox): 用来提供给用户选择列表中的数据。

组合框 (ComboBox): 用来提供给用户下拉选择列表中的数据。

选项按钮 (OptionButton): 用来提供给用户指定单项数据，一般成组使用。

复选框 (CheckBox): 用来提供给用户指定多项数据，一般成组使用。

【声明类关键字】

Public: 声明公共类型的数据；

Private: 声明私有类型的数据；

Static: 声明静态类型的数据；

Dim: 声明数据类型；如：Dim myCell As Range

reDim: 定义未显式声明的数组的维数和元素;

Const: 声明常量数据; 如: Const limit As Integer = 33

As: 一般用于声明数据类型中的“As Type”子句;

Type: 声明用户自定义数据类型;

【数据类型关键字】

Byte: 字节类型;

Integer: 整型数值类型;

Long: 长整型数值类型;

String: 字符串类型;

Boolean: 逻辑类型;

Single: 单精度类型;

Double: 双精度类型;

Currency: 货币数值类型;

Decimal: 可以容纳小数的数值类型;

Variant: 任何数字值或字符串值;

Object: 对象类型;

用户自定义: 利用 TYPE 语句由用户自己定义的数据类型。

【运算符关键字】

这类关键字太多了, 而且有很多种分类方法, 比如按参数个数分为单目运算符、双目运算符、三目运算符和多目运算符等, 按参与运算的参数类型可以分为数值运算符、字符运算符、逻辑运算符、日期运算符、字节运算符等, 按照运算符的作用范围可以分为过程级运算符、函数级运算符和模块级运算符。

【程序结构类】

IF...ELSE...END IF: 分支判断选择语句;

DO WHILE...LOOP: 先判断后执行循环语句;

DO...LOOP WHILE: 先执行后判断循环语句;

FOR...NEXT: 限定次数循环语句;

GOTO...: 无条件转移语句。

一下子列举了这么多的关键字, 是不是眼睛都花了? 是不是对这些关键字还是没搞明白该怎么用? 没关系, 下面就给大家一一介绍。哎呀, 今天的时间又快到了, 明天继续吧, 别换频道啊~~

4、【2009.1.20】代码起步——关键字 (续)

大伙的热情支持，真的让我很感动，提出的很多建议对我也非常有启发，今后的帖子里将会不断吸收各位的好的意见和建议，尽我所能把这篇入门指南写好。

什么今天讲？不，今天讲什么？哎太过激动以至于有点语无伦次了。对了，今天先说说框架类关键字。列举如下：

【框架类关键字】

工程 (Project): 是指用于创建一个应用程序的文件的集合。

对象 (Object): 可控制的某个东西，例如窗体和控件。

窗体 (Form): 应用程序的用户界面。

控件 (Control): 指的是各种按钮、标签、文本框等。

属性 (Property): 是指对象的特征，如大小、标题或颜色。

工作表 (Worksheet): 指 EXCEL 文件里的工作表，例如 sheet1、sheet2 等。

模块 (Module): 指在 VBA 工程中存放独立于用户定义对象代码的容器。

过程 (Sub): 容纳和组织代码的限定符号，一般和 End Sub 联用，不返回结果。

函数 (Function): 容纳和组织代码的限定符号，一般和 End Function 联用，并返回结果。

记得我在学写程序的时候（那是很久很久以前的事情了），我最怕的就是看到这些 E 字母。我经常去看 HELP 里面的例子，然而那里面居然连变量和过程的名字都是很长的 E 文。没办法，谁叫人家微软是美国公司呢！（记得之前有过几个用中文写程序的编辑器，甚至还热过一阵某语言，但随着时间的流逝，好像现在也不知去向了……）那么怎么样才能记住这些又长又难记的单词呢？不用记，你如果用 VBA 编程，你会发现只要你输入了正确的第一个关键字之后，可以通过支持 VBA 语言的编辑器自动提示输入接下来的单词，是不是很简单啊~其实多使用几次，你也就记住了。还有一个可以判断是否输入正确的方法，就是你总是用小写字母书写关键字，如果正确，编辑器会自动修正你录入的关键字的第一个字母为大写。

Project，中文翻译为工程、项目等。这个关键字在 VBA 中表示一个工作簿所关联的代码的总容器。所有和这个工作簿（XLS 文件）关联的代码，都装在这个容器里。但这个容器不仅仅装代码，还能装好多东西，比如工作表、模块、过程、函数等。那么这个容器是不是一个对象呢？请大家结合前面的讲解思考下这个问题？答案是肯定的。（如果你答错了，请接着往下看；否则请跳过下一段。）

Object，对象，实体。在 VBA 中，工程、窗体、控件、工作表、模块等无一不是对象，因为这些都是实体，具备了独立存在的所有条件。比如，我问你书是对象嘛？是，因为书可以独立存在。那么颜色是对象嘛？不是，因为你无法单单拿出来一个叫“颜色”的东西，它只能作为其他东西的一种属性而存在。同样，工程，在 VBA 中是独立存在的，可以作为对象来看待。窗体、控件、工作表、模块也是一样。还有个小提示给大家，大家以后看到代码中某个对象名称后面跟了个 S，在 E 文里应该表示复数，即多于一个的意思。那么在 VBA 中表示什么意思呢？表示多个对象放在一起，我们给它们一个名字叫“集合”。集合是对象嘛？不是，是多个对象放在一起。这个千万要注意，因为中文是没有复数的表示方法的，所以不能理解为“一个人是人嘛？是；一群人是人嘛？当然是！”，那就错了，在 VBA 里“一群人”不是人，是人的集合！那么你可能又会将集合同我们之前讲的“类”搞在一起了，类是对象的抽象，还拿书做例子，一本实实在在的书

是一个对象，它可能是文学书，也可能是计算机书，当然可能是红色的，也可能是绿色的。而如果你脱离了具体的书而说“书”的概念，那么这时就指书这个类了。一个典型的例子是来自于生物界的分类。比如某只麻雀和某只鸽子，都是具体对象，而麻雀和鸽子都可以作为独立的一类鸟这个概念来描述，这时应该说成是麻雀类和鸽子类，同时它们又都属于鸟类这个概念，所以在类这个东西里，是可以分很多层次的，但除非具体到某一只鸟，不然就都是类的概念，而不是对象。集合呢？一群麻雀或者一群鸽子，就是集合，因为它们是由具体的很多只麻雀或鸽子组成的。干嘛费那么大劲说这些概念呢？可能你听得都快睡着了。但我还是要说，因为如果你不弄清楚类、对象和集合等概念，那么在以后写代码的时候，你就会犯糊涂，“怎么我写了一个‘对象’，就是不能用它的某个属性或方法呢？”，其实，很有可能你把对象写成了类或集合，因为他们就差一点点。（比如 Row 可以作为一个类，也可以作为一个对象，就看你如何声明它，同时，Rows 就是一个集合）。

我还要讲一下如何使用类、对象和集合。类一般在声明部分使用，比如 DIM A As Range, 这里 Range 就表示“行”这个“范围”类，而 A 被声明为属于“行”这个类的类型的变量，换句话说，A 在经过这样的声明以后，就可以用来引用某个具体的行对象了。比如 Set A = Sheet1.rows(1)。那么这时，A 就可以代表第一行这个对象了。这里的 ROWS 代表 Sheet1 的所有行对象的集合。一般集合有一个属性，叫做 COUNT，表示在这个集合中的对象的个数。同时，要用集合表示具体对象时，要在集合名称后加上一对括号，里面填写对象的索引值或名称。

Form，窗体。你一般在 Windows 里运行的程序，都具备可视化的功能。那么这就是 Form 的功劳。通过引用 FORM 对象，你可以对这个工程里的一些窗体进行属性的设置、事件的定义以及运行某些方法。

Control，控件。指的是各种按钮、标签、文本框等。它们可以装饰你的窗体，并在窗体中提供用户各种功能。从某种意义上讲，窗体也可以看做是控件的容器。

Worksheet，工作表。指 EXCEL 文件里的工作表，例如 sheet1、sheet2 等。通过引用这个对象，可以对工作表以及下级对象的属性进行设置、事件进行定义、方法进行执行。

Module，模块。指在 VBA 工程中存放独立于用户定义对象代码的容器。如果你想定义一些变量、过程或函数，并且想在各个对象（比如 Sheet1、Sheet2 等）中都能使用，并且可以互相传递值。那么就应该将这些声明或代码写在模块中。详细的以后用到再详述。

以上讲的都是对象，请大家参考对象的用法，具体操作一下如何声明、如何引用它们。至于如何通过对象设置属性、定义事件和执行方法，我们在具体讲解各类对象的时候再详细阐述。

Sub，过程。容纳和组织代码的限定符号，一般和 End Sub 联用，不返回结果。Function，函数。容纳和组织代码的限定符号，一般和 End Function 联用，并返回结果。这两个概念，本来应该在代码里讲的，这里只是提示一下。大家只要明白一个概念，除声明语句外其他代码都应该放在这两种容器内。

Property，属性。是指对象的特征，如大小、标题或颜色。最后讲讲这个。还记得上面讲到的对象的声明和引用的例子吗？刚讲过，应该不会忘吧，否则应该回头重新看下这篇文章了。现在，A 已经被声明为一个 Range 类对象，并且已经赋予 A 引用为 Sheet1 的第一行了。那么如果想对 Sheet1 表的第一行设置为值

都等于 0，应该如何做呢？其实只要对 Range 类对象的一个 Value 属性设置一下就可以了，即：A.value=0。把上面所有的代码都放在 Sheet1 的一个叫

“Worksheet_SelectionChange”的事件中，然后随使用鼠标点一个单元格试一下，是不是第一行的所有单元格都是 0？恭喜你，写出了第一个完整的程序！不仅如此，你还可以在设置属性的语句后面，再写一句“A.Select”，这个是什么意思？先运行下试试（别忘记先保存）。是不是在将第一行都赋值为 0 以后，又把第一行都选中了？这里执行了 Range 类对象的一个叫“Select”的方法，作用是选中对象。回过头来想想，事件是什么？看下这个事件的名称吧：

Worksheet_SelectionChange，翻译过来就是“工作表的选中项目发生改变”，也就是说只要该工作表（Sheet1）中的选中位置发生了变化，那么这个事件就会被触发，其中的代码就会被自动执行。现在明白什么叫“事件触发机制”了吧？换了以前基于过程的编程方法，那么必须规定啥时候这个单元格会被选中，而用户必须在这个规定的时候选中这个单元格，而现在有了事件触发机制，随便什么时候，只要满足事件触发的条件，这个动作就会被执行。

今天讲的所有的代码总结如下，你可以直接把它们复制粘贴到 Sheet1 的代码集中：

```
Private Sub Worksheet_SelectionChange( ByVal Target As Range) ' 声明一个 Worksheet_SelectionChange 事件的私有过程
```

```
Dim A As Range ' 声明一个变量 A，并制定为 Range 类型
```

```
Set A = Sheet1.Rows(1) ' 将 A 赋值为（引用）Sheet1 工作表的第一行
```

```
A.Value = 0 ' 设置 A 变量（表示 Sheet1 工作表的第一行）的 Value 属性值为 0
```

```
A.Select ' 执行 A 变量的 Select 方法
```

```
End Sub ' 事件过程结束标志
```

具体示例程序见下面的附件。（打开 EXCEL 表格后，可以通过按“【ALT】+【F11】”的方法打开代码编辑窗口，并通过双击左侧 Sheet1 工作表的方法打开该工作表下的代码集）。

 例 1.rar (5.65 KB)

明天继续。

5、【2009.1.21】代码起步——关键字（续）

昨天刚洗了车，今天就下雨了，哎，天意弄人~~“今年春节怎么过？出门嘛？”最近听得最多的就是这句话，不管是经济危机的影响还是交通拥堵一票难求的作用，今年明显出门的人少了许多，昨天偶然听到了广播里张信哲的《逃生》，其中一首“牡丹忧”让我想起了父母，立马跑去买了龙井和翡翠的平安扣，晚上看着嘴上埋怨却掩饰不了脸上高兴的父母，总算心安了一些。不管什么借口，今年在家陪陪爸妈吧！所谓“娘生儿，连心肉；儿行千里母担忧；儿思娘亲难叩首；娘想儿来泪双流……”

言归正传，继续昨天的话题，让我们来看看第二类关键字——控件，先列表如下：

【控件类关键字】

标签 (Label): 用来显示文本。

文本框 (Textbox): 用来提供给用户输入文本。

命令按钮 (CommandButton): 用来组织和提供程序功能。

列表框 (ListBox): 用来提供给用户选择列表中的数据。

组合框 (ComboBox): 用来提供给用户下拉选择列表中的数据。

选项按钮 (OptionButton): 用来提供给用户指定单项数据, 一般成组使用。

复选框 (CheckBox): 用来提供给用户指定多项数据, 一般成组使用。

先来说说, 什么是控件, 顾名思义, 控件就是提供给用户控制程序运行的组件。回忆一下, 以往我们用过的程序很多, 都用过哪些控件呢? 可能之前都没有注意, 但现在既然要做程序给别人用, 就得注意了。我们看到, 一般在我们打开的窗口上, 都会有些文字, 这些文字只是提供了一些信息, 比如告诉你这里是输入姓名的, 那里是选择性别等等, 这些文字用户并不能修改或删除, 那么这些文字是不是就是写在窗体上的呢? 不是, 窗体只是一个容器, 里面是空的, 就像不能在空气里写文字一样。VBA 给我们提供了一个可以写这些文字的控件, 我们给它一个名字叫“标签”, E 文叫“Label”。从我们上面描述的文字可以总结出标签控件的特点和作用: 无法修改, 主要用于显示提示信息。这是第一个控件。

第二个控件叫做“文本框”, E 文叫做“Textbox”。这个控件和标签有点类似, 也是放文字的, 但这里的文字是让用户输入的, 而不是显示给用户看的, 所以我们把文本框的特点和作用做个总结就是: 可以提供用户文字编辑功能, 主要用来供用户录入文字信息供程序处理或存入数据库。

第三个控件叫命令按钮, E 文叫“CommandButton”。这个控件的作用不用我多说了吧? 当你之前每次用鼠标点击按钮, 你可曾想过, 为什么这么多按钮会有各种各样不同的功能? 按钮除了能单击还能双击嘛? 按钮的作用是在按下的时候发生还是放开的时候发生呢? 所有这些问题的答案都在于事件是如何写的。按钮提供了单击、双击、按下、放开等事件, 如果代码写在了按钮的某个事件里, 那么在用户做出相应动作以后就会触发执行这些代码。

第四个控件叫列表框, E 文叫“ListBox”。这个控件的外观, 像一张一维表格 (只有列标题)。这个控件的作用除了可以提供给用户一些整齐的列表信息外, 还可以通过选中、单击、双击等事件, 实现用户选择标准化录入的效果。

第五个控件叫组合框, E 文叫“ComboBox”。这个控件比较灵活, 有三种形态。第一种是下拉列表的形式, 主要提供给用户选择列表里的值, 用户的选择只能限于下拉列表提供的内容, 不能自己输入, 并且选择只能是单选。第二种是用户除了可以选择下拉列表提供的内容外, 还可以输入自定义的内容, 即兼具了文本框和下拉列表框的作用。第三种是用户可以对下拉列表框提供的内容进行复选。组合框是比较常用且功能较为强大的控件之一。

还有两个控件，一个叫选项按钮，就是我们平时说的单选按钮，另一个叫复选框，这两个控件经常被成组提供，主要让用户选择某些项目，而避免了文本框的不规则文本录入的问题，达到了录入结果标准化的效果。


说了这么多控件，还没看见呢！？别急，下面就告诉你怎么添加控件。如图

() 单击“控件工具箱”

工具箱”工具栏中的某一个控件按钮，然后在 SHEET 表上就可以按住左键拖动鼠标勾画出某种控件，就这么简单。画好了，看到了，可没什么作用啊？呵呵，接下来就是要给这些控件“赋予生命”了。

再次看到“控件工具箱”工具栏，从左数第一个按钮（如图：





), 这个按钮的作用是进入设计模式和退出设计模式，如果在非设计模式下，那么对于你添加的控件就会按照你所定义的事件代码触发执行，如果要编辑这些按钮，就必须确保这个切换设计模式的按钮处于被按下的状态，这时候就进入了设计模式，这时对上述你添加的控件进行操作就不会触发你定义的事件代码，就可以提供给你编辑修改这些控件属性和代码的机会。请确认切换设计模式按钮处于按下状态，双击你添加在当前工作表中的任何一个控件，你就能看到自动打开了 VBA 编辑窗口，并生成了与被双击控件关联的一个事件，当前光标正在这个事件过程中一闪一闪的等你呢！请把目光移到代码编辑窗口的最顶端的两个下拉列表框那里（还记得前面讲的叫什么名字嘛？），左侧的下拉框显示的是当前控件的名称，右侧下拉框显示的是该控件的默认事件，（如图：) 比

如文本框，默认名称为“TextBox1”，默认事件为“Change”，这个事件表示当文本框内容发生变化的时候被触发执行。你可以下拉右侧的列表框，你会发现，和当前控件关联的事件有很多，至于如何用我们下面会逐个讲解。现在请重新回到 SHEET1 工作表界面，再重新选择另外一类控件，添加这个控件并双击进入编辑窗口，看看和这种（比如命令按钮）控件关联的事件有哪些？至此，知道如何给控件添加代码了嘛？

还有个小问题，你考虑过为什么这里出现的控件名称都是一样的，只是在最后添加了数字以示区别嘛？因为你没有修改控件的名称，所以就只能用默认的名称咯。其实每个控件都有名称，因为每个控件都是对象嘛！所以，设置控件的名称就是设置控件对象的一个属性。除了名称属性外，常见的还有哪些属性呢？我在这里讲一个查看属性的方法，大家不妨自己去看看各类对象的属性。所谓“师傅领进门，修行在个人”嘛！上面提到的“控件工具箱”工具栏还在吗？刚才讲了从左边数第一个按钮的作用，除了各类控件外，还有 2 个按钮，也就是从左边数第 2 个和第 3 个按钮没用过，你一定猜出来了，不错，第二个按钮是打开属性

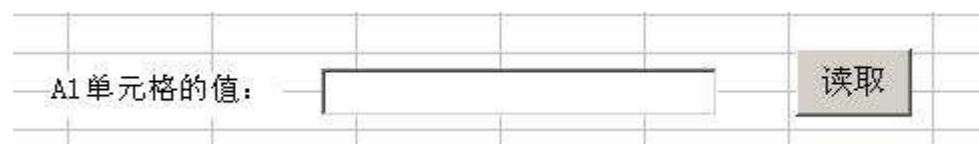


对话框的（如图：），点了这个按钮就可以看到当前选中对象的属性，而第三个按钮则和双击某个控件一样，是打开代码编辑窗口的。

最后，我想结合一个实例讲讲常见控件的常见属性和事件。（实例：例 2.rar (9.07 KB) ）

一般控件常见属性主要有如下几个：Enable，取值为逻辑值，通过设置该属性可以决定控件当前是否可用；Visible，取值为逻辑值，通过设置该属性可以决定控件当前是否可以被显示出来。对于标签控件，常见属性就是 Caption 属性，取值为文本，该属性决定了标签当前显示出来的文本。对于文本框、列表框、组合框控件，常见属性有 Value 属性，取值为文本，该属性决定了文本框、列表框和组合框的内容，可读写。对于命令按钮控件，常见属性有 Caption 属性，取值为文本，该属性决定了按钮当前显示出来的文本。

接下来，大家新建一个空白的 EXCEL 文档，在 Sheet1 工作表中，依次添加标签、文本框、命令按钮各一个，如图：



。设置标签控件的名称属性为：Lbl_A1；Caption 属性为“A1 单元格的值：”；文本框控件的名称属性为：Txt_A1；命令按钮控件的名称属性为：Cmd_A1；Caption 属性为“读取”；接着，双击命令按钮控件，打开代码编辑器窗口，并在 Cmd_A1_Click() 事件中，添加如下代码：Txt_A1.value = Cstr(Sheet1.Range("A1").value)。保存后，回到 Sheet1 工作表编辑界面，单击控件工具箱的左侧第一个按钮，即退

出设计模式。这时候，在 A1 单元格输入一个值后，只要按一下命令按钮，在文本框中就会出现 A1 单元格的值。

由于时间关系，今天只是简要的讲述了一些常见控件的常见属性、方法和事件，至于其他一些控件和大家以后遇到的新的控件的属性、事件和方法，可以参照我上面讲述的过程和方法自己去学习、练习和使用。

明天继续。

6、【2009.1.22】代码起步——关键字（续）

新年的脚步离我们越来越近了，越是临近过年，事情就越忙，办年货、筹备年夜饭……好不容易跑出来写点东西，不知道一会还会有啥事等着我呢！

不知道昨天讲的关于控件类的关键字大家有没有掌握，如果没搞清楚也没关系，我们接下来会专题详细讲解每一个控件的用法，只要大家耐心听下去。今天继续聊关键字的话题。今天要讲的是声明类关键字。先列示如下：

【声明类关键字】

Public：声明公共类型的数据；

Private：声明私有类型的数据；

Static：声明静态类型的数据；

Dim：声明数据类型；如：Dim myCell As Range

reDim：定义未显式声明的数组的维数和元素；

Const：声明常量数据；如：Const limit As Integer = 33

As：一般用于声明数据类型中的“`As Type`”子句；

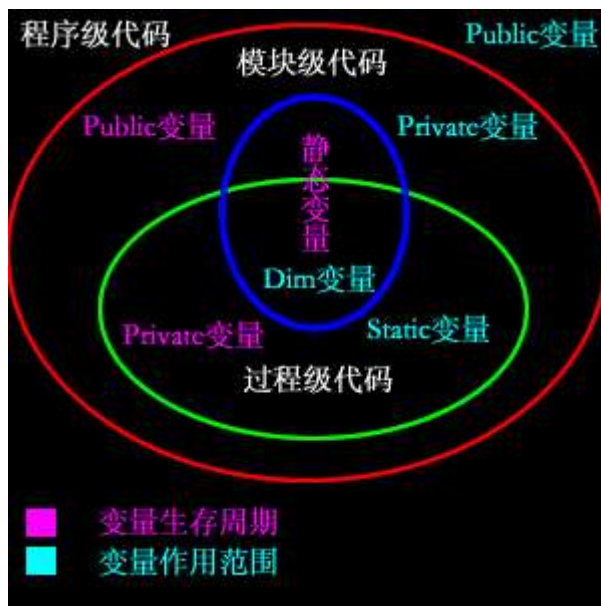
Type：声明用户自定义数据类型。

考虑到该类关键字属于一个相互联系的整体，并且一般在写代码的时候都会遇到，因此我想分三个部分来讲解。首先我们先了解下 VBA 代码的语法结构。每一种编程语言都有各自的语法组织结构，什么内容就应该放在什么地方，就像人身上的“部件”，如果鼻子长到了眼睛的上面，那就不属于正常的人了。VBA 代码一般都按照事件、过程、函数、模块、类模块等来组织，不管是系统内部定义好的还是开发人员自定义的过程、函数或模块，使用前都必须声明。声明也就是告诉 VBA 代码的编译器，我下面的程序中会用到这些变量、过程、函数和模块等。声明分两种，一种是显式声明，也就是将声明部分代码明确的写出来。另一种是非显式声明，意思正好和前一种相反。当您的模块中含有这条语句“`Option Explicit`”，那么系统就强制规定您必须显式声明该模块中的所有变量，否则将编译出错。如果没有出现上述语句，那么您可以在使用变量前不显式声明该变量，

系统编译时将默认该变量类型为 Variant。

声明语句可以命名和定义过程、变量、数组以及常数。当声明一个过程、变量或常数时，也同时定义了它的作用范围，而此范围是取决于声明位置以及用什么关键字来声明它。这就是我要讲的第二个问题。先说声明的位置，如果你在某个工作表的某个过程中声明了一个变量或函数，那么这个变量或函数的作用范围就局限于这个过程；如果你在一个工作表的（通用）部分声明了一个变量或函数，那么这个变量或函数的作用范围就局限于该工作表代码模块中。这就表明声明的位置可以决定变量或函数的作用范围。除了声明位置外，采用哪个关键字也可以决定被声明对象的作用范围。比如，如果你在一个自定义模块中，采用 PUBLIC 这个关键字来声明一个变量，那么这个变量在所有应用程序的所有没有使用 Option Private Module 的模块的任何过程中都是可用的（若该模块使用了 Option Private Module，则该变量只是在其所属工程中是公用的）。而采用 PRIVATE 声明的变量，其作用范围就只有该模块内部，称为私有变量。上述两个关键字只能在模块级别代码中使用。（所谓模块级别就是只在过程外部的代码，即某个工作表的<通用>部分中写的代码，区别于过程级别的代码，即过程中书写的代码）除此以外，还有一个 DIM 关键字，用这个关键字可以在模块级别中声明变量，也可以在过程级别中声明的变量；前者声明的变量对该模块中的所有过程都是可用的，而后者则只在过程内是可用的。STATIC 关键字是用来声明静态类型变量的，这是什么意思呢？举个例子你就清楚了，一般声明的过程级别变量，在程序代码运行完该过程之后，该变量的值就自动清空了，该变量也由于所在过程的运行完毕而结束了其生存周期。这样的过程级别的变量称为动态变量。那么有另外一种变量，虽然也是在过程级别中被声明的，但在该过程运行结束以后，该变量的值仍然保持不变，直至该变量所在过程所在的模块运行中止为止。这样的变量就叫做静态变量。静态变量只能在过程级别中使用，用于声明变量并分配存储空间。在整个代码运行期间都能保留使用 Static 语句声明的变量的值，并且其生命周期与包含该过程定义的模块相同。从作用范围来讲，这类变量由于在过程级别的代码中被声明，因此其作用范围只是局限于其所在的过程内部。

如下图所示：



今天的内容可能比较枯燥，但绝对重要。正确理解声明类关键字的用法，对把握变量、过程、函数等对象的作用范围和生命周期有着非常总要的作用。希望今天的讲解能对大家正确理解这部分关键字带来帮助。

“什么？要定年夜饭的菜单？这就来~~~”各位，明天见！

7、【2009.1.27】代码起步——关键字(续)

小别数日，今日终于回来了。先给大伙道一声“新年好”！

早晨一早驱车三个多小时，赶到了孩子他外婆家。到达的时候已然到了中午，自然免不了应酬一番。喝了点小酒，一下午头都昏昏的。到此时日落西山，方突然想起年前许下的今日之约，立即开机上网、奋起疾书，至此，才有了下面的文字，望诸位海涵！

还记得年前我们说完了声明类关键字，今天我们要讲的是数据类型关键字，列举如下。

【数据类型关键字】

Byte：字节类型；

Integer：整型数值类型；

Long：长整型数值类型；

String：字符串类型；

Boolean：逻辑类型；

Single：单精度类型；

Double：双精度类型；

Currency: 货币数值类型;

Decimal: 可以容纳小数的数值类型;

Variant: 任何数字值或字符串值;

Object: 对象类型;

用户自定义: 利用 TYPE 语句由用户自己定义的数据类型。

首先,我们先谈谈 VBA 中大伙见到的文字有哪些。大家在 VBA 中能够见到的文字,除了之前列举的几大类关键字(保留字)外,就是些数据和注释了。注释是以西文半角方式的单引号开头的文字,在 VBA 编辑器里显示为绿色。这些字主要用来注释解说变量或代码的用途。数据则有很多种,这才是我们今天要讲的主要内容。在 VBA 中,数据按照形态大致分为三类,常量、变量和数组。常量就是指在程序中已经指定的值,并且常量的值在程序中保持不变。举个例子,我们声明一个常量名字叫“Name”,我们在可以在声明的时候直接赋予这个常量一个值比如“张三”。那么在这个程序里,我们就不用直接引用“张三”这个名字了,我们在需要引用的时候只要用 Name 这个常量就可以替代了。可能有的朋友会说,既然 Name 就是代表“张三”,那我们为什么不直接在程序里写“张三”呢?这里有这么几个原因。一是有可能直接使用“张三”不方便,如果起个有意义的名字更便于阅读和维护代码。二是如果以后需要修改程序,需要将程序中使用到“张三”的地方都改成“李四”,那么如果用引用一个常量的这种方式的话,就只需要修改 Name 这个常量的值就可以了。变量和常量有点类似,都是以一个名字来命名并赋予一个值给它,不同之处在于,常量的值在整个应用程序代码中是不变的,而变量的值会随着代码位置的不同而发生变化。在这里提供一个小建议,在给变量起名字的时候最好起一些有意义的名字,否则在今后维护代码的时候会遇到困难。比如可以给整型变量加上 INT 的前缀,如果这个变量用来保存“编号”的,那么可以起这样的名字“INT_name”,如果是保存“生日”的日期型变量呢?可以这样起“DATE_birthday”,等等。再来说说数组,数组也是变量,不同之处是数组是一组变量的集合而不是单个变量。因此,数组一般由变量名和下标组成,例如:NAME(2)、k(5)等。数组按照下标的个数分为一维数组(只有一个下标)、二维数组(有两个下标)和多维数组(有多于 2 个下标)。除了普通变量能采用数组形式外,对象、控件、单元格等都可以采用数组的形式来表示。对于有规律的变量或对象采用数组的形式可以方便的配合循环语句来简化代码、提高执行效率,不过会占用更多的内存资源,因此是一种空间换时间的方法。

其次,在学习具体变量之前,我们还要谈谈计算机是如何保存和处理各类数据的。按照数据的种类和载体我们大致可以分为这么几类,数字、西文字符、中文字符、图形、动画、视频、音频等。这些数据在计算机中都被翻译为 2 进制来保存。比如数字,分为整形、浮点型、单精度、双精度、长整形等,按照 2 的 N 次幂的方法计算所能保存的最大数和最小数来确定范围;再比如数字型字符和西

文字符，都是按照 ASCII 编码来保存的，一个字符占用一个字节（8 位二进制）；中文字符由于数量众多，就采用国际标准交换代码来保存，一个字符占用二个字节（16 位二进制）；图形、动画、视频和音频数据就更为复杂，一般通过引用（在变量中保存能找到这些数据的地址）的方法保存。

最后，我们来看看学习变量需要重点掌握哪些内容。一是要十分清楚每种变量类型所对应的取值范围。根据我们第二点讲解的计算机保存各类数据的方法，对应不同类别变量的取值范围是存在不同的。BYTE 字节型变量，只能存放 1 个字节的内容，因此你不能将一个汉字字符保存到一个字节型变量中，也为一个汉字字符需要占用 2 个字节的空间，如果用 BYTE 型变量保存就会发生溢出错误。具体上述关键字所对应的每种变量的取值范围详见联机帮助。二是变量的本质是一块特定内存区域的名称，我们通过引用这个名称（即变量名）来使用这块系统分配给应用程序的特定内存存放数据。明白了这一点可以帮助我们更好地在程序中使用变量。三是变量的生存周期和作用范围。这点就是我们上一讲所重点讲述的内容。

今天太晚了，先简单谈几点吧，有关内容将在今后用到的时候详细表述。

8、【2009.1.28】代码起步——关键字（续）

看到这么多热心的朋友支持这个帖子，真的很高兴。能让这个帖子帮到更多需要的朋友，即使我多写点也感到很开心。今天该讲运算符关键字了，照旧先列出该类关键字。

【运算符关键字】

这类关键字太多了，而且有很多种分类方法，比如按参数个数分为单目运算符、双目运算符、三目运算符和多目运算符等，按参与运算的参数类型可以分为数值运算符、字符运算符、逻辑运算符、日期运算符、字节运算符等，按照运算符的作用范围可以分为过程级运算符、函数级运算符和模块级运算符。

先谈谈运算符的含义。到底什么是运算符？运算符就是连接各种类型的数值、常量、变量的运算符号。这里需要特别声明的是运算符与函数的区别。举个比较有代表性的例子，大家在 EXCEL 的单元格里写公式的时候，如果写 2 个或 2 个以上并列条件的时候，可能会遇到 AND 和 OR 的关系，以往有很多同事和朋友都来问过我，为什么我写的条件会报错。那么他们是如何写这些条件的呢？“x>1 or y<2”，这样写在 VBA 程序中是可以被接受并符合语法的，但在 EXCEL 单元格的公式中就会报错，因为在公式中，EXCEL 会将 AND 或者 OR 都作为函数来看待而不是运算符。按照定义，运算符是连接各类数据的符号，因此其一般前后都有数据，

但函数不同，一般是由函数名开头，后面跟上一对括号，括号里面是参数，彼此用逗号分开，因此在 EXCEL 的公式中，如果用到多个条件之间的关系，就应该采用函数形式而不是运算符的形式，例如 and(条件一，条件二)。

按照运算符所能连接的数据的数量，我们将运算符分为单目、双目、三目和多目运算符。单目运算符只能连接一个数据，比如 not 条件一，返回一个结果。这类运算符一般集中在逻辑运算中。大家可能见过这样的运算符号，“++”、“--”，这些也是单目运算符，如果数据在运算符前面意思是先返回该数据然后本身加、减 1；如果数据在运算符后面意思是数据先进行加减运算然后返回运算后的该数据，这一般常用于循环语句。二目运算符能连接 2 个数据，很多运算符都属于二目运算符，比如常见的+、-、*、/等等。三目运算符常见的有这么一种类型，“表达式 1? 表达式 2: 表达式 3”，意思是先对表达式 1 求值，如果表达式 1 条件成立（不为 0），则对表达式 2 求值，然后将该值作为整个表达式的结果返回；如果表达式 1 不成立，则对表达式 3 求值，并将该值作为整个表达式的结果返回。类似于条件函数。多目运算符类似于函数的形式，由多个参数参与运算，并返回 1 个或多个值。

按参与运算的参数类型，运算符可以分为数值运算符、字符运算符、逻辑运算符、日期运算符、字节运算符等。这个比较容易理解，在此就举几个常见运算符的例子供大家参考。数值运算符主要有“+*/”等，此外还有“^”符号，表示幂，例如 2^3，表示 2 的 3 次幂。字符运算符有“&”，表示连接两个字符的连接符号，此外还有“+-”符号，“+”表示字符串的连接，“-”表示剔除。逻辑运算符主要有“not、and、or、xor”等。XOR 表示异或的关系，就是如果存在如下表达式：A xor B，则如果 A 和 B 均为真或均为假，即 A 和 B 相同，则返回真，否则返回假。其他运算符大家可以参考联机帮助，或者也可以在我们今后遇到的时候听我详细讲解。

明天还剩最后一部分即语句类关键字了，这部分将结合常用的语句及程序结构来进行解说，敬请期待~~~. 1

9、【2009.1.29】代码起步——关键字（续）

今天一天都在带儿子，因为昨天老婆的腰闪了。哎，好在明天可以回自己家了，真是“归心似箭”啊~

今天是关键字的最后一讲，按照惯例，还是先将该类关键字列举如下：

【程序结构类】

IF...ELSE...END IF：分支判断选择语句；

DO WHILE...LOOP：先判断后执行循环语句；

DO...LOOP WHILE：先执行后判断循环语句；

FOR...NEXT：限定次数循环语句；

GOTO...：无条件转移语句。

今天我想主要谈谈程序结构问题。所谓程序结构，就是指程序代码的执行顺序，按照执行顺序的不同我们将程序结构大致分为以下三类：顺序结构、分支判断选择结构和循环结构。基本上我们目前能接触到的语句结构都离不开这三种类别。

先讲第一类，顺序结构。这类语句是按照程序代码书写的先后顺序一句一句从前往后顺序执行的，这类语句结构是最简单且容易理解的一类。那么如何才算一个完整的语句结束呢？这在不同的编程语言环境中是不尽相同的，比如 C 语言中以分号作为每个语句的结束标志，而 VBA 中，则以换行符作为一个语句的结束标志，换句话说，只要在同一行书写，哪怕再多的代码都被看作一条语句来执行。

第二类是分支判断选择结构。这类结构并不像顺序结构那样按顺序逐条执行代码，而需要按照判断语句的结果对各条分支语句进行选择执行，所以需要通过分支判断选择语句，对本结构代码加以控制执行。例如，“IF...ELSE...END IF”，这条语句以“IF”开始，至“END IF”结束，其中在 IF 所在行书写判断的条件表达式，“IF”和“ELSE”之间书写条件表达式为“真”时执行的代码，“ELSE”和“END IF”之间书写条件表达式为“假”时执行的代码。实例如下：

```
IF 条件 THEN
```

```
    语句 1
```

```
ELSE
```

```
    语句 2
```

```
END IF
```

在这里实例程序中，就体现了一个典型的分支判断选择结构，程序是这样执行的：当程序执行到 IF 语句时，系统会计算条件表达式的值，如果该值为“真”（即条件成立），那么就执行语句 1，然后结束该 IF 结构去执行 END IF 后面的代码；如果该值为“假”（即条件不成立），那么就执行语句 2，然后结束该 IF 结构去执行 END IF 后面的代码。所以通过这里实例可以看出，在分支判断选择结构中包含的语句不是每一个都被执行到的，其执行与否、何时执行由条件表达式控制。

1 条 IF 条件语句可以区分 2 种情况下的执行代码，那么多于 2 种情况该怎么办呢？可以通过在 IF 语句中嵌套 IF 语句来实现多条件判断执行的效果。实例如下：

```
IF 条件 1 THEN
```

```
    语句 1
```

```
ELSE
```

```
    IF 条件 2 THEN
```

```
        语句 2
```

```
ELSE
    语句 3
END IF
END IF
```

这和 EXCEL 中的 IF 函数的嵌套类似。除了 IF 语句外，该类结构还有一条语句，也可以实现多条件的分支判断选择结构，即：

```
SELECT CASE 条件
    CASE 值 1:
        语句 1
    CASE 值 2:
        语句 2
    CASE ...
        ...
    ELSE CASE:
        语句 3
END SELECT
```

这里通过计算条件表达式的值与各分支语句中的值进行比较来选择具体到哪个分支中执行相应的代码。

第三类是循环结构。和第二类分支结构类似，循环结构也需要通过循环语句来控制程序代码的执行。按照循环条件的位置和计算方法不同，循环语句大致分为如下三种：

DO WHILE...LOOP：先判断后执行循环语句；
DO...LOOP WHILE：先执行后判断循环语句；
FOR...NEXT：限定次数循环语句；

前 2 类是不限定次数或者在代码执行前无法确切指定循环次数的循环控制语句。实例程序分别为：

```
DO WHILE 条件
    语句 1
    语句 2
LOOP
```

这类语句先判断条件表达式的值，然后根据该值决定是否进入或继续执行循环体内部的语句 1 和 2。

```
DO
    语句 1
    语句 2
LOOP WHILE 条件
```

这类语句先执行循环体内部的语句 1 和 2, 然后进一步根据条件表达式的值决定是否再次回到 DO 语句继续执行循环体内的语句。

第 3 类是限定循环执行次数的循环语句。实例程序为:

FOR 控制变量=1 TO N STEP 步长

语句

NEXT

这里通过每次循环后控制变量的自动增加步长值, 然后与最大值 N 比较, 来决定是否继续执行循环体内的语句。

循环语句也是可以嵌套使用的。无论分支结构还是循环结构的控制语句, 都会打乱原先执行的语句顺序, 从而给理解和维护代码带来困难, 因此要恰当利用这些结构控制语句, 并加上相关注释, 以防代码出现错误或今后理解困难。

还有一条语句, 称为无条件转移控制语句, 即 GOTO 转移地址名称, 这条语句可以方便编程人员在无法确知转移条件和难于控制时将程序执行顺序强制转移到指定的地址。由于是无条件转移, 这种情况中存在着编程人员的主管随意性, 因此必须加上必要的注释, 并尽量少用, 以避免今后程序理解和维护的困难。

至此, 我们已经将 VBA 代码中常见的关键字都一一讲解完毕了, 想必大家今后在看到程序中的那些 E 文的时候也大致能知道这个属于什么类型的关键字、在这个位置有什么作用了, 至于如何才能熟练掌握并运用这些关键字写出自己的代码, 则是我们这个教程接下来的章节需要完成的任务, 敬请期待《VBA 入门指南——代码起步之我的程序》篇。

10、【2009.1.30】代码起步——我的程序

今夜鞭炮依旧, 浓睡不消残酒。坐在自己的床上, 那感觉真是舒服啊~~

从今天开始, 我们将开始讨论一个新的话题, 前一个话题让我们认识了许多 VBA 代码, 而这个话题我们将用这些代码写出我们自己的程序来。在开始写代码前, 我想先用今天的时间给大家介绍下如何培养良好的代码书写习惯以及 VBA 编辑环境提供的代码书写辅助功能。

先讲第一个问题, 我们都听说过这么一句话, “外行看热闹, 内行看门道”, 说的是内行只要一看便知某一个事物或某一个人是不是专业的水平。请看下面两段代码, 请你先看看哪段是由专业程序员写出来的?

(1)

```
Public a(10) as Integer
```

```
.....
```

```
Sub abs()
```

```

For i = 1 to 10
If a(i)<0 then
a(i)= - a(i)
End If
Next
End Sub

```

```

(2)
Public a(10) as Integer
.....
Sub abs()' 求绝对值
Dim i as Integer
    For i = 1 to 10
        If a(i)<0 then
            a(i)= - a(i)
        End If
    Next
End Sub

```

看出来了吗？为什么是第二段呢？其实这里至少有 3 个地方可以体现一种专业程序员良好的编程习惯和素养。第一，代码采用缩进格式书写。为什么要这样写呢？因为这样不但可以清楚的表达各种程序结构之间的关系和层次，而且能清晰地体现出语句之间的配对关系，并为他人阅读和今后维护提供方便。第二，变量先声明后使用，这样可以保证内存正确分配和使用变量所占空间，保证系统的正常稳定运行。第三，给代码加上必要的附注或注释，以方便他人阅读和今后维护。以上三点希望大家在开始写代码之初就养成这样的习惯，将对今后的代码之路有所裨益。

第二个问题比较简单，请看下图就应该明白了。



（标准工具栏）



（调试工具栏）

这样的两个工具栏，均可以通过 VBA 代码编辑窗口的视图菜单的工具栏菜单项下列出的“标准”和“调试”两项来打开。标准工具栏，左起按钮依次为：属性方法列表、常数列表、快速信息、参数信息和自动完成关键字；缩进、突出；设置断点、设置注释块、解除注释块；切换书签，下一个书签、上一个书签和清除所有书签。这里要着重提出的是缩进和突出功能，可以通过这两个按钮对选中的代码（可以一行或多行）进行缩进和突出功能，对修改代码结构和层次很有帮

助。还有，设置注释块和解除注释块分别可以对选中的代码进行转化为注释和解除注释的作用，这为调试程序中需要暂时屏蔽但不想删除的代码提供了很好的方法。调试工具栏，左起按钮依次为：运行、中断、重新设置；切换断点、逐语句、逐过程、跳出、本地窗口、立即窗口、监视窗口、快速监视和调用堆栈。这里主要用到的是调出本地窗口和立即窗口，并通过运行宏、暂停和中断来实现调试和测试程序的目的。大家不妨试试。

11、【2009.1.31】代码起步——我的程序（续）

今天是新年长假的最后一天，趁着冬日暖阳自然带着家人出门走走，毕竟明天就要开始新一年的工作了。

接着昨天的话题，我们来看一个小程序。一方面熟悉下如何写代码，一方面也可以提高点学习兴趣。

程序如下：  例 2.rar (5.98 KB)

其中的代码如下：

```
Private Sub Worksheet_SelectionChange(ByVal Target As Range)
    If (Target.Column <> 1) Then
        Cells(Target.Row, 1).Select
    End If
End Sub
```

程序的目的是这样的，在这样的表格中，往往由于列数众多，在选择某一个学生的成绩的时候，会发生错行的情况，而这个程序可以使您在选择某一个成绩时，自动选中该成绩所在行的第一列的单元格。这里用到了我们之前所讲解的很多知识。大家还记得我们之前讲过的吗？请大家回忆一下。

首先，我们讲到了 VBA 的代码在哪里？请大家打开 EXCEL 文件后，按【ALT】+【F11】键，就能打开代码编辑窗口；

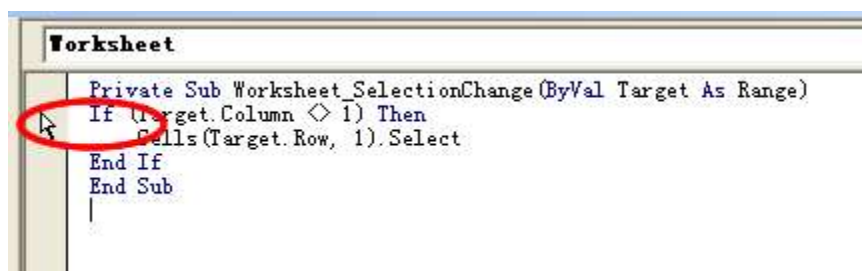
其次，我们看到代码窗口上方有两个下拉列表框，左边的是“Worksheet”，右边的是“SelectionChange”，这表示当前的代码写在 Worksheet 工作簿的 Sheet1 工作表的 SelectionChange 事件中；这个事件的触发条件是当前选中的单元格发生变化，也就是说如果之前你选中了 A1 单元格，那么现在如果你又左键单击选择了 B1 单元格，这个事件就会被触发，里面的代码就会被执行。这个事件还带了一个参数，名字叫“Target”，这个参数的类型是 Range，即单元格区域类型，所以这个参数表示的是对当前最新被选中的单元格的引用。

那么我们是怎么实现上述程序的目的的呢？让我们继续看一看程序的代码，

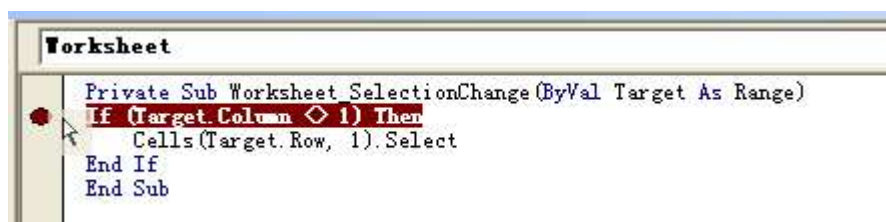
其中是一个分支判断选择结构的 IF 语句，判断的条件是：当前被选中的单元格所对应的列序号如果不是 1（也就是说当前被选中的不是 A 列的单元格），如果条件成立那么就执行其中的代码，“Cells”是对工作表所有单元格集合的引用，有两个参数，第一个是表示行序号，第二个表示列序号，如果给定了这两个参数，那么也就是指定了在 SHEET1 工作表的所有单元格集合中符合当前给定行序号和列序号的唯一单元格的引用；看下这里指定的两个参数是什么呢？行序号被指定为当前被选中单元格所在行的行序号，列序号被指定为 1，即 A 列单元格，那么现在就通过“Select”方法使由行序号和列序号两个参数指定的单元格集合中的这个单元格成为当前被选中的单元格。这样就实现了我们之前定义的这个程序的目的。大家明白了吗？

现在大家可以尝试一下选中非 A 列单元格的操作，看看是否自动变为选中了同行的行首第一个单元格。

细心的朋友可能还记得，我们之前还讲过调试程序，那么我们现在通过单步执行的方式来跟踪一下程序的执行，以便于我们加深理解。方法是：先将光标移到代码编辑窗口左侧灰色纵条区域，使鼠标变成从右下向左上的空心箭头现状，如图：



单击左键，使该行背景色变为暗红色，表示定义了一个断点。如图：



断点的意思是当程序执行到这里就停下来，以便于我们单步跟踪执行并查看各个参数的值。接下来我们就可以通过不断按【F8】来单步执行该程序，并查看各个参数的值的变化情况了。可能通过这样的跟踪执行，大家发现，在程序按照我们上述思路执行到“Cells(Target.Row, 1).Select”语句执行完毕后，突然又从头开始执行了，这个其实是正常的，因为如果你选中的不是行首的单元格，则判断条件成立，判断分支选择结构中的语句就会被执行，而恰恰这条语句的作用是

将当前选中的单元格由非行首单元格转变到同行行首的单元格了，那么这时也将再次触发这个事件，即事件发生了递归调用。那么这时由于此时选中的单元格已经是行首单元格了，所以判断条件不成立，该事件过程立即结束，从而递归收敛，程序也就执行完毕了。

请大家对照上述讲解，反复执行和跟踪程序代码的执行过程，细心体会，也许通过这个简单的例子并不能让你完全明白 VBA 的代码如何写，但应该能让你被 VBA 所吸引，从而期待接下来更为精彩的深入讲解内容吧~

12、【2009.2.1】代码起步——我的程序（续）

上班第一天，好多店家还没开门，中午单位门前整个一条街啥吃的都没有，哎，第一天就吃方便面~~

今天我们继续看一个程序，还记得昨天的程序吗？当你选择了某一行的单元格，系统会将当前选中单元格自动切换为该行的第一个单元格，这样可以了解该成绩是哪个学生的成绩。今天我们继续在昨天的 EXCEL 文件的基础上，实现一个新的功能，就是当按到某一个单元格时，除了自动切换选中单元格外，将本行内容通过一个对话框显示出来。

EXCEL 文件如下：  例 3.rar (7.52 KB)

代码如下：

```
Private Sub Worksheet_SelectionChange(ByVal Target As Range)
```

```
If (Target.Column <> 1) Then
```

```
Cells(Target.Row, 1).Select
```

```
MsgBox "姓名： " & Me.Cells(Target.Row, 1) & Chr(13) _  
    & "语文： " & Me.Cells(Target.Row, 2) & Chr(13) _  
    & "数学： " & Me.Cells(Target.Row, 3) & Chr(13) _  
    & "英语： " & Me.Cells(Target.Row, 4) & Chr(13) _  
    & "物理： " & Me.Cells(Target.Row, 5) & Chr(13) _  
    & "化学： " & Me.Cells(Target.Row, 6) & Chr(13) _  
    & "地理： " & Me.Cells(Target.Row, 7) & Chr(13) _  
    & "历史： " & Me.Cells(Target.Row, 8) & Chr(13) _  
    & "生物： " & Me.Cells(Target.Row, 9) & Chr(13) _  
    & "体育： " & Me.Cells(Target.Row, 10) & Chr(13) _  
    & "总分： " & Me.Cells(Target.Row, 11) & Chr(13) _  
    , vbOKOnly, "提示"
```

```
End If  
End Sub
```

大家对比昨天的代码就会发现，今天就加了一句代码，即上述代码中用红色标出的部分，这个就是今天要讲解的内容。

首先，我们看到这个代码使用了一条语句，Msgbox，这条语句的作用是调用一个消息对话框，这个语句带有三个参数，各个参数之间都用逗号分隔。

其次，我们看到该语句的第一个参数就占据了好几行，那么大家可能会问，不是在上一讲中提到了 VBA 识别判断是否是一条语句就凭借换行符吗？那么像这样在多行中的语句为什么是属于同一条语句呢？这个主要是因为有时候语句太长，如果在一行写可能会导致格式混乱且不容易阅读和理解，所以我们就使用下划线（即：“_”）来将未写完的代码转到下一行继续写，并告诉系统，该语句在这行没有结束，下行还是属于该语句的内容。让我们仔细看看第一个参数的内容，总体上是一个字符串，该字符串由个部分构成，每个部分占据了一行，每个部分都是由工作表中的列标题字符串组成，“&”符号表示两个字符串之间的连接符，通过该符号，后面跟了一串代码，看起来也像个函数的样子（为什么说像函数呢？也为有函数名、括号和括号里的参数），对，这个是一个集合，其中的“me”表示引用本工作表，例如这里就是指 Worksheet1 工作表，“Cells”表示本工作表的所有单元格的集合，其中的参数表示行序号和列序号，这个在昨天的讲解中讲过。其实这里省略了一个部分，完整的应该是这样写

“Me.Cells(Target.Row, 1).Value”，即这里返回的是指定单元格内的值，为啥省略呢？由于 Value 属性是 Cells 集合的默认属性，所以可以省略不写。最后，还跟了一个函数，即“Chr(13)”，这个函数的参数为一个整数值，作用是返回 ASCII 编码表中对应该整数值字符或不可打印符号的。比如这里的 13，在 ASCII 表中就对应了换行符，那么通过这个函数就返回了一个换行符，这样就可以让每个列标题的部分在对话框中占据一行的位置。

然后，我们看到第二个参数比较简单，“vbOKOnly”，该参数是一个字符串常量，由系统内部定义，这里的这个值表示该对话框只有一个 OK 按钮，类似的可以放在这里的常量见下图：

常数	值	描述
vbOKOnly	0	只显示 OK 按钮。
VbOKCancel	1	显示 OK 及 Cancel 按钮。
VbAbortRetryIgnore	2	显示 Abort、Retry 及 Ignore 按钮。
VbYesNoCancel	3	显示 Yes、No 及 Cancel 按钮。
VbYesNo	4	显示 Yes 及 No 按钮。
VbRetryCancel	5	显示 Retry 及 Cancel 按钮。
VbCritical	16	显示 Critical Message 图标。
VbQuestion	32	显示 Warning Query 图标。
VbExclamation	48	显示 Warning Message 图标。
VbInformation	64	显示 Information Message 图标。
vbDefaultButton1	0	第一个按钮是缺省值。
vbDefaultButton2	256	第二个按钮是缺省值。
vbDefaultButton3	512	第三个按钮是缺省值。
vbDefaultButton4	768	第四个按钮是缺省值。
vbApplicationModal	0	应用程序强制返回；应用程序一直被挂起，直到用户对消息框作出响应才继续工作。
vbSystemModal	4096	系统强制返回；全部应用程序都被挂起，直到用户对消息框作出响应才继续工作。
vbMsgBoxHelpButton	16384	将Help按钮添加到消息框
VbMsgBoxSetForeground	65536	指定消息框窗口作为前景窗口
vbMsgBoxRight	524288	文本为右对齐
vbMsgBoxRtlReading	1048576	指定文本应在希伯来和阿拉伯语系统中的从右到左显示

该函数如果有返回值，则可能的返回值见下图：

常数	值	描述
vbOK	1	OK
vbCancel	2	Cancel
vbAbort	3	Abort
vbRetry	4	Retry
vbIgnore	5	Ignore
vbYes	6	Yes
vbNo	7	No

为什么要用字符串常量替代该参数的数值类型的值呢？这样做主要是为了便于程序的阅读和程序代码的记忆。

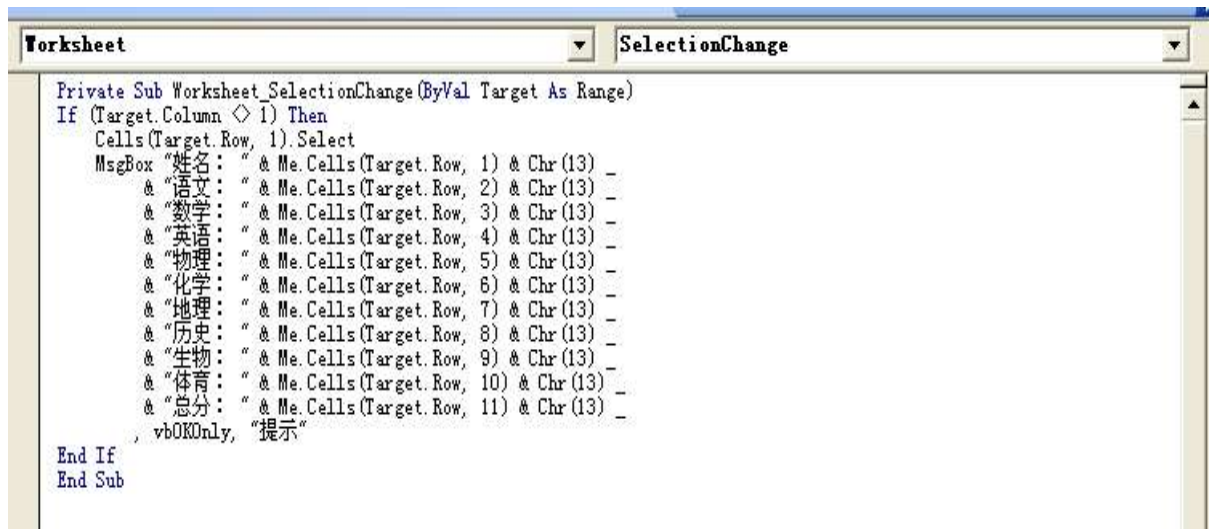
最后一个参数更简单，就是一个字符串，作用是定义该对话框的标题。

大家运行下试试吧？！是不是很有趣？

13、【2009.2.2】代码起步——我的程序（续）

这两天陆续看到朋友们回复，有的说越来越难于理解了，有的说讲解得太快，快要跟不上了，还有的说前面的差不多都快忘完了……哈哈，告诉大家，有这个感觉是很正常的，看来我们有必要停下来解解惑了~~

还记得这两天我们的代码吗？见下图：

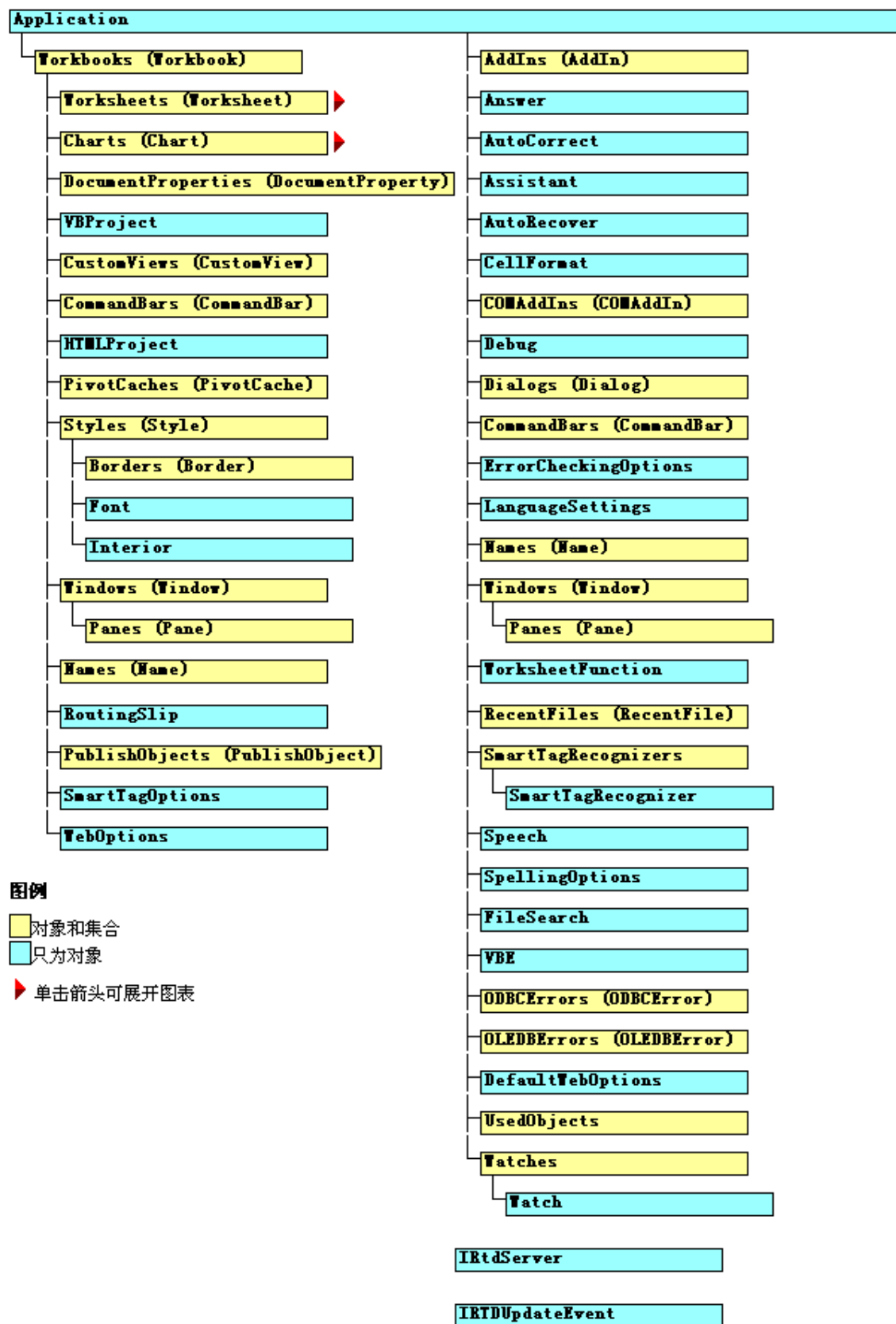


```
Private Sub Worksheet_SelectionChange(ByVal Target As Range)
If (Target.Column <> 1) Then
Cells(Target.Row, 1).Select
MsgBox "姓名: " & Me.Cells(Target.Row, 1) & Chr(13) & _
      "语文: " & Me.Cells(Target.Row, 2) & Chr(13) & _
      "数学: " & Me.Cells(Target.Row, 3) & Chr(13) & _
      "英语: " & Me.Cells(Target.Row, 4) & Chr(13) & _
      "物理: " & Me.Cells(Target.Row, 5) & Chr(13) & _
      "化学: " & Me.Cells(Target.Row, 6) & Chr(13) & _
      "地理: " & Me.Cells(Target.Row, 7) & Chr(13) & _
      "历史: " & Me.Cells(Target.Row, 8) & Chr(13) & _
      "生物: " & Me.Cells(Target.Row, 9) & Chr(13) & _
      "体育: " & Me.Cells(Target.Row, 10) & Chr(13) & _
      "总分: " & Me.Cells(Target.Row, 11) & Chr(13) & _
      , vbOKOnly, "提示"
End If
End Sub
```

我们回头再来看看 VBA 代码到底是如何与前台的 EXCEL 工作表联系起来的。工作表在 EXCEL 里是一个独立的对象（还记得啥叫对象吗？），名字叫“Worksheet”，对象有很多属性、方法和事件，当然，还包括子对象集合，在这段代码里，我们用到了工作表的一个事件，即 SelectionChange 事件，这个事件的作用我们已经在上一讲谈过了，即当被选中的单元格发生变化时触发该事件并执行其中的代码。对于其他我不想多说了，今天我就想重点谈谈子对象及集合。在 EXCEL 里有很多对象，到底这些对象是如何组织起来的呢？请看下图：

Microsoft Excel 对象

请参阅



从这张图里我们看到了很多对象，他们按照一定的级别有规律的被组织成树形排列。尽管在 Excel 的对象模型中包括了 100 多个对象，但你会发现最常用的主要集中在如下五个对象上，Application、Workbook、Worksheet、Range、Chart，这并不是说用不着使用其他对象，只是说明这五个对象最为常用。

Application 对象代表 Excel。使用 Application 对象可以控制应用程序级的设置、内置的 Excel 函数以及高级方法，例如 InputBox 方法。Workbook 对象是指 Excel 中的工作簿，即是说 Excel 文件。在 VBA 环境中，不说打开一个文件，而称为打开一个工作簿；也不说保存一个文件，而称为保存工作簿。学习 Excel 时最先了解到的其中一点就是，Workbook 中包括 Worksheet。Worksheet 是 Workbook 中独立的页，数据就保存在 Worksheet 中。Worksheet 中包括单元格（Cell）。你也许会认为不得不编写大量的代码以对单元格对象进行控制，但是，实际上没有单元格这样的对象。有单元格属性，将在后面的学习中学到。相反，你将对范围（Range）对象进行处理。范围对象是指一个或者多个单元格。大多数 Excel 用户都使用 Excel 的图表功能，所以你经常需要处理图表（Chart）对象。用“图表向导”创建图表时所做的一切都可以通过 VBA 代码做到。

那么什么是集合呢？集合就是多个对象组织在一起的一种引用方法。比如在一个工作簿 Workbook 对象中，就默认包含了三个 Worksheet 对象，那么在一起引用这三张工作表的时候就采用引用集合的方法来写代码，即 Workbook.Worksheets，这里的 Worksheets 就代表了工作表的集合。集合在使用中有一个属性经常被用到，就是 Count 属性，这个属性的作用是返回该集合中对象的个数，比如你想知道一个工作簿中当前包含了几张工作表，就可以通过 Workbook.Worksheets.Count 来得到。集合还有一个用法就是如果需要单独引用其中的一个对象时，可以通过指定索引值或该对象名称来引用，例如需要引用工作簿中名称为“考试表”的一张工作表，默认索引为 3，那么可以这样引用，即 Workbook.Worksheets(3) 或者 Workbook.Worksheets(“考试表”)，现在大家再看前面讲解的代码的时候是不是有了更清晰的理解呢？

14、【2009.2.3】代码起步——我的程序（续）

如何才能让大家更容易地理解、掌握和能运用 VBA 代码呢？这是我这几天一直在思考的问题。看到这么多朋友在关心和支持这个帖子，越发感到责任重大~~我们在前面 2 课讲解了如何让 VBA 代码从 EXCEL 工作表中取到单元格的数据，今天我们要把取到的数据计算一下，并且写回指定的单元格中。

实例文件如下： 例 4.rar (10.99 KB)

首先，为了不让每次点击单元格都弹出的对话框来烦我们，我们需要将相关

的语句去掉。直接删除当然是一件容易的事，但如果想保留下来以备不时之需，那么这里就介绍一种方法给大家，大家也可以将这个运用在添加注释或者屏蔽调试程序过程中的测试代码等方面。其实说了这么多，方法很简单，就是在该条语句前加个西文半角方式下的单引号，即“'”。添加这个符号以后，你会发现，该语句已经变成绿色字体了，表示该语句被转换为注释部分，不再执行了。

其次，为了实现计算数据的目的，我们需要添加一个按钮控件，还记得我们之前讲过的控件都在哪里吗？对，控件工具箱，如果没有这个工具栏，你可以通过【视图】菜单的【工具栏】项目的下拉列表来添加。添加按钮控件的方法是，点击控件工具箱上的【按钮】控件（如图：



), 然后在需要添加按钮的单元格区域按住鼠标左键，拖放出一个指定大小的按钮，松开鼠标，按钮就出来了。现在的按钮是不是比较难看？我们可以通过设置几个属性来“美化”一下。如果看不到属性对话框，那么可以点击控件工具箱中的【属性】按钮（如图：



) 打开属性对话框。第一个修改的是“名称”属性，这个属性表示这个按钮控件在我们的代码里的唯一名称，我们今后在代码中如果需要引用到这个按钮，就需要通过这个名称来引用，这里修改为“CMD_pj”，这个名称的被“_”符号分为前后两个部分，前面大写的部分表示控件类型是按钮（CMD=Command 简称），后面小写的部分表示该按钮的作用是评级。当然这个名称是由我们随便起的，只要有意义就可以了。第二个修改的是 Autosize 属性，这个属性有一个逻辑值，默认为 False，这个属性表示当前控件的大小尺寸是否自动调节，根据什么调节呢？当然是根据按钮上显示的文字来条件，False 表示不自动调节，True 表示自动调节，这里我们设置为 True（一会就能看到变化了）。第三个修改的是 Caption 属性，这个属性的作用是设置按钮上显示的文字内容，有的教材也把它叫做按钮的名称，我觉得会跟之前讲的“名称”属性搞混了，所以我叫它“按钮上显示的文字”。虽然冗长，但易于理解。这里我们设置为“评级”。好了，看看按钮，是否好看多了？这时你还可以用鼠标点住按钮拖动以调节下放置的位置。（放心，现在处于设计模式下，单击按钮是不会触发按钮的 Click 事件的。）

下面，我们在 L1 单元格输入“等级”两个字，表示我们将根据前面 K 列的总分，来对每个学生进行评级，并将最终结果放在 L 列对应单元格中。那么按什么标准来评级呢？我们在这里先定义一下：每门课的分都是 0-100 分，总分在 765（含）-900（含）为“优秀”，675（含）-765（不含）为“良好”，540（含）-675（不含）为“及格”，0（含）-540（不含）为“不及格”。该标准也就是按平均分的范围来定的，平均分在 0 到 60 分（不含）为“不及格”，60 分到 75 分（不含）为“及格”，75 分到 85 分（不含）为“良好”，85 分到 100 分为“优

秀”。如果我们是手工完成评级任务，那么我们只要按照上面的标准，看看 K 列的总分在哪个范围，就在 L 列对应单元格输入相应的等级就可以了。但我们现在需要用 VBA 代码完成，通过点击“评级”按钮自动填写到 L 列所有单元格中，接下来该怎么做呢？其实我们只需要做 1 件事，即添加按钮中的代码，使之具备功能就可以了。那么就动手吧！我们双击按钮，进入 VBA 代码编辑窗口，我们看到已经有了一个过程，即之前我们添加的 SelectionChange 事件的代码，我们再看看现在光标所在区域，是不是在另一个过程中，这个过程名称叫做

“Cmd_pj_Click”，大家看到代码编辑窗口顶端两个下拉列表框的内容了吗？正好是两个下拉框内容连起来，那么我们可以知道，前一部分即“Cmd_pj”，表示当前对象是按钮对象，后一部分“Click”，表示调用按钮对象的“Click”事件，即单击事件，只要鼠标单击该按钮，就会触发该事件并自动执行内部定义的代码。好了，接下来我们就来完成这里的代码吧！添加的代码及解释如下：

```
Private Sub Cmd_pj_Click() ' 评级按钮单击事件代码
Dim i As Integer ' 声明一个变量，名称为 I，类型为整形，作用为循环控制变量
Dim tmp_Total As Single ' 声明一个变量，名称为 tmp_Total，类型为单精度型，作用为保存总分
Sheet1.Range("K2").Select ' 选中 K2 单元格
For i = 2 To Sheet1.Range("K:K").End(xlDown).Row ' 开始一个循环语句，循环条件是从第 2 行至 K 列最后一个有数值的行
    tmp_Total = Sheet1.Cells(i, 11).Value ' 将某行 L 列单元格数值（总分）赋值给变量 tmp_Total
    Select Case tmp_Total ' 开始一个分支选择判断语句
        Case 0 To 539.99 ' 条件为“大于等于 0 分且小于 540 分”的情况
            Sheet1.Cells(i, 12).Value = "不及格" ' 该条件下设置为“不及格”
        Case 540 To 674.99 ' 条件为“大于等于 540 分且小于 675 分”的情况
            Sheet1.Cells(i, 12).Value = "及格" ' 该条件下设置为“及格”
        Case 675 To 764.99 ' 条件为“大于等于 675 分且小于 765 分”的情况
            Sheet1.Cells(i, 12).Value = "良好" ' 该条件下设置为“良好”
        Case 765 To 900 ' 条件为“大于等于 765 分且小于等于 900 分”的情况
            Sheet1.Cells(i, 12).Value = "优秀" ' 该条件下设置为“优秀”
        Case Else ' 除上述条件外的其他取值情况
            Sheet1.Cells(i, 12).Value = "错误" ' 其他情况下设置为“错误”
    End Select ' Select 分支选择判断结束语句
Next ' For 循环结束语句
End Sub
```

好了，我们已经做了所有需要做的工作，现在保存一下然后关闭代码窗口，我们回到熟悉的 EXCEL 窗口，点下“评级”按钮试试，咦？怎么没反应？呵呵，还记得刚才我说过的吗？注意控件工具箱里的第一个按钮状态（如图：



), 现在处于“设计”模式下，单击按钮是不会触发任何事件的，所以我们需要退出“设计”模式，方法是单击控件工具箱里的第一个按钮，使之恢复弹出的状态（区别于刚才按下的状态）。再点下按钮，试试，怎么样？结果出来了吧？别光顾着乐，好好理解和琢磨下代码吧！今天来不及详细讲解这些代码的作用了，明天将逐句讲解这段代码里用到各个知识点和技巧，敬请期待吧~~

15、【2009.2.4】代码起步——我的程序（续）

接着昨天的分析，我们今天来详细讲解一下代码中用到的各个知识点和技巧，代码列示如下：

```
Private Sub Cmd_pj_Click()' 评级按钮单击事件代码
Dim i As Integer ' 声明一个变量，名称为 I，类型为整形，作用为循环控制变量
Dim tmp_Total As Single ' 声明一个变量，名称为 tmp_Total，类型为单精度型，作用为保存总分
Sheet1.Range("K2").Select ' 选中 K2 单元格
For i = 2 To Sheet1.Range("K:K").End(xlDown).Row ' 开始一个循环语句，循环条件是从第 2 行至 K 列最后一个有数值的行
tmp_Total = Sheet1.Cells(i, 11).Value' 将某行 L 列单元格数值（总分）赋值给变量 tmp_Total
Select Case tmp_Total ' 开始一个分支选择判断语句
Case 0 To 539.99 ' 条件为“大于等于 0 分且小于 540 分”的情况
Sheet1.Cells(i, 12).Value = "不及格" ' 该条件下设置为“不及格”
Case 540 To 674.99 ' 条件为“大于等于 540 分且小于 675 分”的情况
Sheet1.Cells(i, 12).Value = "及格" ' 该条件下设置为“及格”
Case 675 To 764.99' 条件为“大于等于 675 分且小于 765 分”的情况
Sheet1.Cells(i, 12).Value = "良好" ' 该条件下设置为“良好”
Case 765 To 900 ' 条件为“大于等于 765 分且小于等于 900 分”的情况
Sheet1.Cells(i, 12).Value = "优秀" ' 该条件下设置为“优秀”
Case Else ' 除上述条件外的其他取值情况
```

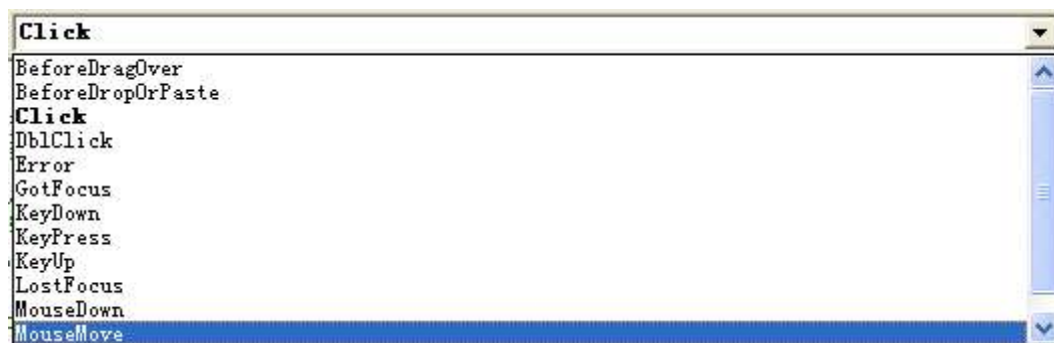
```
Sheet1.Cells(i, 12).Value = "错误" '其他情况下设置为“错误”
```

```
End Select 'Select 分支选择判断结束语句
```

```
Next 'For 循环结束语句
```

```
End Sub
```

首先，我们看到所有的代码都被放在了名为“Cmd_pj_Click”的过程中，昨天的讲解中我们知道了这个过程是 Cmd_pj 按钮的一个单击事件，只要按钮被单击，就会触发这个事件，系统会自动执行这个过程代码。通过这个例子我们知道，如果实现按钮的单击功能，我们可以通过将代码添加在这个按钮的单击事件中来实现这个目的。类似的按钮对象还有哪些事件呢？请见下图：



上图中自上而下的事件分别为：当拖放操作时、当即将在一个对象上放置或粘贴数据时、单击时、双击时、发生错误时、获得焦点时、截获键盘有键按下时、有按键时、截获键盘有键松开时、失去焦点时、鼠标键按下时、鼠标箭头移动时等。这些事件大家有兴趣，可以自行添加代码并实验。

其次，我们看到了过程中第 1 和第 2 两行声明了两个变量，一个是整形，一个是单精度型，作用已经在后面的注释里写明了。这里所声明的变量由于采用了 DIM 这个关键字，因此其生存周期为本过程执行期间，而作用范围也仅限于本过程范围内有效。

然后，接着是一个循环结构，变量 I 控制着循环的条件，即从 2 到表达式的值。这个表达式包含了几个知识点。第一，从四个点号连接符来看，这里的对象、集合和属性、参数等关系十分明确，请你结合前面讲过的内容，自己分析下，哪个是对象？哪个是集合？哪个是属性？哪个是参数？检测是否真正掌握的最好办法就是到实际使用中去检验！

在循环结构内部，第一句是一个赋值语句，通过一个变量，保存第 I 行第 L 列的单元格的值，即总分。为啥要这么做？一方面可以简化代码，不用每句条件都写一长串表达式，另一方面，也是提高程序的执行效率，否则每次都要先计算表达式的值才能进行条件判断。接下来，是一个分支选择判断语句，请大家重点注意一下这个语句的语法规则。每个条件后面都是一句赋值语句，这个就是真正将评价等级写入相应单元格的语句。

好了，今天简单解释了一下昨天的代码，也让大伙可以喘口气消化消化，明

天我们就要继续前进了，一个新的任务等着我们呢~~

【问题 1】希望通过 `range(某列).end(xlDown).value` 获得某列向下最后一个非空白单元格的值，试了下发现如果该列中有空白单元格，则获得的是该空白单元格上面的那个值而不是真正的最后一个非空白值，请问是这样么？

【答】是这样的。

【问题 2】那如果要避免这种情况，又该如何修改呢？

【答】2 种思路，一个是可以通过代码去掉空格，一个是通过其他代码实现该表达式的功能。

另外帮助里对 `range.end()` 的示例如下：

本示例选定包含单元格 B4 的区域中 B 列顶端的单元格。

Visual Basic for Applications

```
Range("B4").End(xlUp).Select
```

本示例选定包含单元格 B4 的区域中第 4 行尾端的单元格。

Visual Basic for Applications

```
Range("B4").End(xlToRight).Select
```

本示例将选定区域从单元格 B4 延伸至第四行最后一个包含数据的单元格。

Visual Basic for Applications

```
Worksheets("Sheet1").Activate
```

```
Range("B4", Range("B4").End(xlToRight)).Select
```

【问题 3】那么他这里尾端又是指什么？

【答】尾端就是你第一个问题里说的情况，到所指方向的第一个空格的前一个单元格，这个叫尾端。

【问题 4】最后个例子里区别上面尾端的地方是在于 `Worksheets("Sheet1").Activate` 吗？

【答】最后 2 个例子的区别在于，上一个选定的是 B4 单元格所在的那一行的尾端单元格（从 B4 向右遇到的第一个空值单元格的前一个单元格）；后一个选定的是从 B4 开始一直向右到那一行的尾端单元格，选中的不是一个单元格，而是一行中的一部分单元格。见下图实例：上一个实例效果：

	A	B	C	D	E	F	G	H
1	1	2	3	4	5	6	7	
2	1	2	3	4	5	6	7	
3	1	2	3	4	5	6	7	
4	1	2	3	4	5	6	7	
5	1	2	3	4	5	6	7	
6	1	2	3	4	5	6	7	
7	1	2	3	4	5	6	7	

下一个实例效果:

	A	B	C	D	E	F	G	H
1	1	2	3	4	5	6	7	
2	1	2	3	4	5	6	7	
3	1	2	3	4	5	6	7	
4	1	2	3	4	5	6	7	
5	1	2	3	4	5	6	7	
6	1	2	3	4	5	6	7	
7	1	2	3	4	5	6	7	

`Worksheets("Sheet1").Activate` 这句话的意思是使 SHEET1 工作表成为当前工作表;

`Range("B4", Range("B4").End(xlToRight)).Select` 这句话意思是选中从 B4 开始一直向右到那一行的尾端单元格。

不知道您明白了没有?

代码如何能写入新 EXCEL 表?

有个限制使用天数的代码:

```
ub Auto_Open()
```

```
Dim fs, d, s
```

```
Set fs = CreateObject("Scripting.FileSystemObject")
```

```
Set d =
```

```
fs.GetDrive(fs.GetDriveName(fs.GetAbsolutePathName(ThisWorkbook.Path)))
```

```
s = d.serialnumber ' 磁盘序列号
```

```
If s = -1111111111 Then Exit Sub ' 要使用的电脑磁盘序列号
```

```
Dim FirstDate, de, days
```

```
FirstDate = Date
```

```
de = GetSetting("XXX", "YYY", "date", "") ' 从注册表取值
```

```
If de = "" Then ' 如果取不到值
```

```
SaveSetting "XXX", "YYY", "date", FirstDate ' 把日期保存到注册表
```

```

MsgBox "本文件可使用 60 天, 今天是第 1 次使用", , "提示"
Else
    days = Date - CDate(de) ' 计算文件使用的天数
    If days > 60 Then ' 如果文件使用超过 60 天
        MsgBox "已超过使用期限, 本文件将自杀", , "警告"
        ThisWorkbook.ChangeFileAccess xlReadOnly ' 改为只读属性
        Kill ThisWorkbook.FullName ' 自杀
        ThisWorkbook.Close False ' 关闭不保存
    End If
    MsgBox "本文件已使用" & days & "天, 还有" & 60 - days & "天可使用", , "提示"
End If
End Sub

```

我把它写入新 EXCEL 表的代码窗口后, 关闭时先是提示“是否保存文件”, 点击“是”, 然后就出现“隐私问题警告: 此文档包含宏, ACTCITVEX 控件 XML 扩展包。。可能包含个人信息, 且这些东西不能通过“文档检查器”进行删除云云, 不管点确定还是取消, 重新打开表格, 没有原来文件的提示了, 写到文件里的代码也没了, 问问诸位, 代码究竟怎么写到新文件里?

引用:

原帖由 yd0209 于 2009-2-5 21:55 发表 

代码如何能写入新 EXCEL 表?

有个限制使用天数的代码:

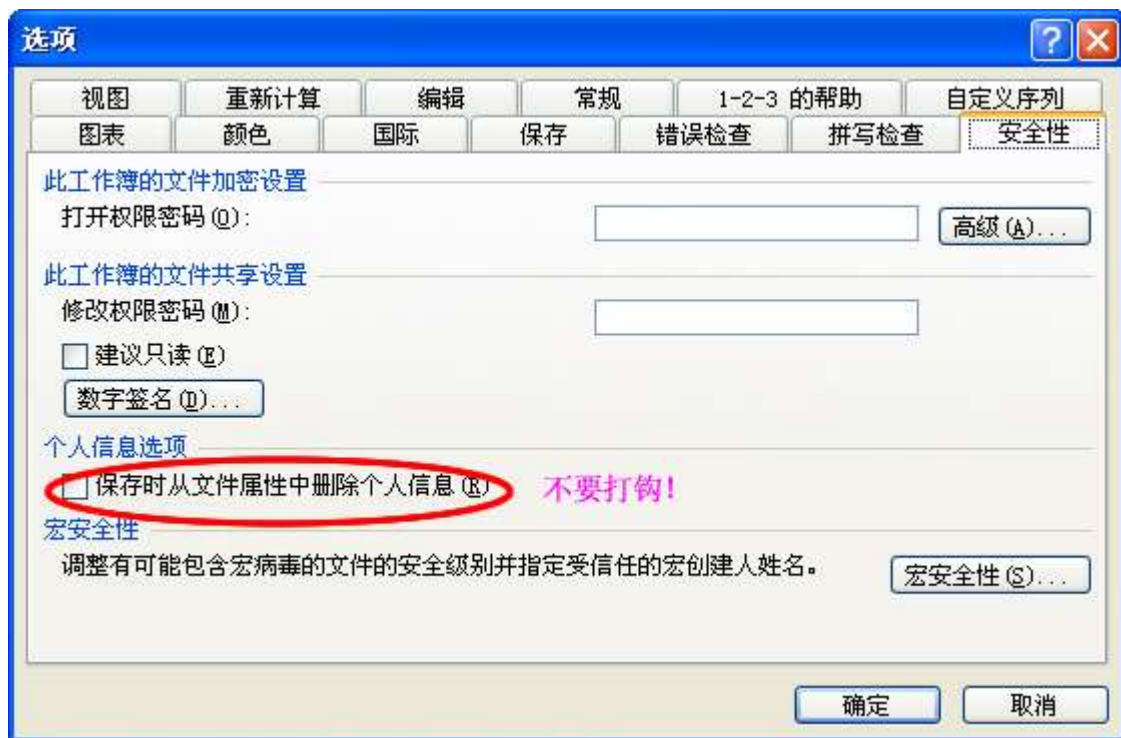
```

Sub Auto_Open()
    Dim fs, d, s
    Set fs = CreateObject("Scripting.FileSystemObject")
    Set d = fs.GetDrive(fs.GetDriveName(fs.GetAbsolutePath ...

```


【工具】》【选项】》【安全性】选项卡, 在“个人信息选项”一栏下的“保存时从文件属性中删除个人信息”前的勾去掉!! 如下图:





16、【2009.2.6】代码起步——我的程序（续）

这两天真的很忙，当你被周围的同事、朋友当做“高手”的时候，你就一定会变得这样！其实，我所提供给他们的帮助都是非常简单的应用。

停了一天，正好给大家休息一下，前面的例子还记得吗？我看到好多朋友都提出了自己的疑问，我也在帖子里都一一作答了，这使我看到了来自越来越多的朋友的关心和期望，大家的认真也让我非常感动，这也是我坚持下去的最大的动力！今天我们要进入一个新的例子，其实这个例子也来源于这里前期的一个朋友发帖的问题。实例如下：  批量数据处理.rar (25.07 KB)

这个实例的目的是进行数据的批量处理，然后将结果存放在指定位置，同时由于数据较多，处理比较慢，因此需要对处理过程进行跟踪和告知用户处理进度。那么具体要进行什么样的数据处理呢？根据要求，题目给出了从 A1 到 AF30 单元格区域的大量数据，共计 960 个数据。要求对这些数据进行分类，将出现次数依次为 1 次、2 次、3 次……的数据放在同一列中归类，并统计每一类的数据个数。

这个题目看似比较复杂，其实有很多方法可以实现，而采用 VBA 的解决方案，可以更直观、更快捷的进行处理。我们经常讲在实际开始做事前先考虑一下方法，往往会找到最好的途径，从而起到事半功倍的效果。那么现在请你先考虑下实现

这个要求的思路吧。这次，我们就用一种最简单、最易于理解、也最不经济的方法来实现，这里先讲一下思路：通过循环结构依次从 A1 到 AF30 开始逐个取数，每次取到一个数，首先统计下该数在整个 A1 到 AF30 区域内出现的次数；然后将该数与应该放置的列中已经存在的数据进行比较，以判断是否之前已经出现过该数，如果第一次出现，则记录该数，否则直接跳过，不再处理。同时，对各类数据的个数进行统计，实时修改统计单元格的数据，以达到展示处理进度的目的。

有了思路，我们就可以写出代码了，先将代码贴出来给大家看看，代码如下：

```
Private Sub CommandButton1_Click()  
Dim i, j, p, q, x, y, u As Integer  
Dim tmpaddr As String  
p = 34  
q = 34  
x = 34  
y = 34  
w = 0  
Range("A34", "D1000").Clear  
For i = 1 To Sheet2.Range("a1", "af30").Cells.Count  
    tmpaddr = Sheet2.Range("a1", "af30").Cells(i).Address  
    j = CountIf(Sheet2.Range("a1", "af30"), Sheet2.Range(tmpaddr).Value)  
    Select Case j  
    Case 1:  
        Sheet2.Range("A" & CStr(p)).Value = Sheet2.Range(tmpaddr).Value  
        p = p + 1  
        Range("g34").Value = CStr(p - 34)  
    Case 2:  
        For u = 34 To q - 1  
            If Sheet2.Range("B" & CStr(u)).Value =  
Sheet2.Range(tmpaddr).Value Then  
                GoTo exit_q  
            End If  
        Next  
        Sheet2.Range("B" & CStr(q)).Value = Sheet2.Range(tmpaddr).Value  
        q = q + 1  
        Range("h34").Value = CStr(q - 34)  
    exit_q:  
    Case 3:
```

```

        For u = 34 To x - 1
            If Sheet2.Range("C" & CStr(u)).Value =
Sheet2.Range(tmpaddr).Value Then
                GoTo exit_x
            End If
        Next
        Sheet2.Range("C" & CStr(x)).Value = Sheet2.Range(tmpaddr).Value
        x = x + 1
        Range("i34").Value = CStr(x - 34)
exit_x:
    Case Is > 3:
        For u = 34 To y - 1
            If Sheet2.Range("D" & CStr(u)).Value =
Sheet2.Range(tmpaddr).Value Then
                GoTo exit_y
            End If
        Next
        Sheet2.Range("D" & CStr(y)).Value = Sheet2.Range(tmpaddr).Value
        y = y + 1
        Range("j34").Value = CStr(y - 34)
exit_y:
    ' Case Else
    '     w = w + 1
    '     Range("k34").Value = CStr(w)
    End Select
Next
End Sub

Private Function CountIf(ByVal tmpR As Range, ByVal tmpN As String) As
Integer
Dim a, b, c As Integer
b = 0
For a = 1 To tmpR.Cells.Count
    If tmpR.Cells(a).Value = tmpN Then
        b = b + 1
    End If
Next

```

```
CountIf = b  
End Function
```

请大家通过前面讲解的知识自己先分析下代码，下次课程详细讲解代码编写方法和作用。

17、【2009.2.7】代码起步——我的程序（续）

自2月4日立春以后，这个天气是一天热过一天了，今天早晨醒来居然发现了蚊子，还是吃饱血的，哎，看来达兄的“适者生存”的理论又一次被验证了。从蚊子开始适应寒冬的气温联想到目前的经济严寒，快学好技术吧，有门技术傍身总还能有一口饭吃啊~~哈哈，说严重了。

继续昨天的实例，我们开始讲解代码。大家以后看别人写的代码可以从如下几个方面入手帮助理解。第一，当然是作者附着在代码里的注释，这个可以直接了解该变量或语句的作用；第二，先看大结构大框架，比如这里，我们看到代码总共分了两个 SUB，这个是最大的一个结构，然后在第一个 SUB 里，最大的结构是一个 FOR 循环，从大往小看，最后落实到每个最小结构里的每条语句，这样可以理清思路，忽略个别不了解作用的语句，不致于被个别语句搞得乱了阵脚；第三，在语句中，一般大的集合或者对象在语句的最左侧，由点号连接的语句中越往右侧的对象层次越低，而在最右侧的一般是属性或参数；第四，大家如果实在不了解某一部分或某几句代码的作用，不妨在该部分的第一个语句上设置一个断点，然后将程序运行到断点后逐句运行，运行一句，看一下效果（包括看对象发生的实际的变化和参数的变化等），这样或许可以帮助我们理解这些语句；最后，还有一点提示大家，看代码最好结合代码的目的去理解，这样可以帮助我们了解作者的意图。

我们今天的代码中，分了两个过程。分别是：CommandButton1_Click 和 Function CountIf，前一个我们应该很熟悉了，是一个名为 CommandButton1 的按钮的单击（Click）事件。这个过程里放的当然是单击这个按钮后作者想处理的工作的一些代码。后一个名称有点怪，怎么没有 SUB 呢？还记得我们前面讲框架类关键字的时候讲过的，有一种和过程并列的结构体吗？对了，叫函数，也就是这里的 FUNCTION，它与过程的区别是过程一般不返回结果，而函数需要返回结果。看看这两个声明：Private Sub CommandButton1_Click() 和 Private Function CountIf(ByVal tmpR As Range, ByVal tmpN As String) As Integer，是不是前者没有参数，也没有声明返回结果的类型，而后者带了2个参数，并且返回类型被声明为 Integer 类型？！

我们先来看看第一个过程，即按钮的单击事件里的代码。首先是2条声明语

句, “Dim i, j, p, q, x, y, u As Integer” 和 “Dim tmpaddr As String”。分别声明了7个整形变量和一个字符串型变量。然后是5个赋值语句, “p = 34”、“q = 34”、“x = 34”、“y = 34”、“w = 0”, 细心的朋友会发现, 怎么这里出现了一个未声明的变量 W 呢? 难道变量可以不声明就直接使用吗? 我来解释下这个问题。在 VBA 的语法规则里, 如果在一个模块的通用部分含有如下语句: Option Explicit, 那么在这个模块里的所有过程和函数中被使用的变量必须在使用前被显式的声明, 否则会发生编译时错误, 这条语句也叫强制显式声明语句。而这个模块里, 没有使用这条强制显式声明语句, 所以变量可以不经显式声明直接使用。在 VBA 中可以简单地通过一个赋值语句来隐含声明变量。所有隐含声明变量都为 Variant 类型, 而 Variant 类型变量比大多数其它类型的变量需要更多的内存资源。如果显式地声明变量为指定的数据类型, 则应用程序将更有效。显式声明所有变量可以减少命名冲突以及拼写错误的发生率。

接下来, 语句 “Range(“A34”, “D1000”).Clear” 的作用是将范围为 “A34 到 D1000” 的区域内容清空。紧接着是一个 FOR 循环, 循环控制参数 I 的范围为 1 至 A1 到 AF30 区域范围内的单元格数量。从这里, 我们学习到了一种得到某个区域范围内单元格数量的方法, 就是用这样的语句: 工作表.Range(开始单元格地址, 结束单元格地址).Cells.Count, 意思是工作表的 range 对象 (括号里参数指明了区域范围) 所引用的区域范围内的 Cell 集合的数量 (即 Count 属性)。类似这样的用法, 需要大家在学习中慢慢积累。在循环体内部, “tmpaddr = Sheet2.Range(“a1”, “af30”).Cells(i).Address” 语句的作用是将某一个 (循环控制参数 I) 单元格的地址赋值给变量 TMPADDR, “j = CountIf(Sheet2.Range(“a1”, “af30”), Sheet2.Range(tmpaddr).Value)” 语句的作用是在 A1 到 AF30 区域范围内统计与 TMPADDR 变量所指地址的单元格的值相同的单元格的数量, 并将它赋值给变量 J。接着是一个分支判断选择结构, 根据变量 J 的取值, 有四个分支, 每个分支内部分别处理对应数量的单元格的值。

那么到底是如何处理的呢? 且听明天继续讲解~

【2009.2.9】代码起步——我的程序 (续)

大家好, 今天是元宵节, 过完今天, 这个年也就结束了, 大伙也该安心上班了~然而今天也是我们这个教程的最后一讲, 在讲完这个案例之后, 相信大家已经从一个对 VBA 来说完全找不着北的门外汉, 慢慢摸到了门边上了, 虽然还是在门外, 但相信只要继续努力, 已经可以独立摸着门进去了!

还记得代码吗? 我再次贴出来:

```
Private Sub CommandButton1_Click()
```

```

Dim i, j, p, q, x, y, u As Integer
Dim tmpaddr As String
p = 34
q = 34
x = 34
y = 34
w = 0
Range("A34", "D1000").Clear
For i = 1 To Sheet2.Range("a1", "af30").Cells.Count
    tmpaddr = Sheet2.Range("a1", "af30").Cells(i).Address
    j = CountIf(Sheet2.Range("a1", "af30"), Sheet2.Range(tmpaddr).Value)
    Select Case j
    Case 1:
        Sheet2.Range("A" & CStr(p)).Value = Sheet2.Range(tmpaddr).Value
        p = p + 1
        Range("g34").Value = CStr(p - 34)
    Case 2:
        For u = 34 To q - 1
            If Sheet2.Range("B" & CStr(u)).Value =
Sheet2.Range(tmpaddr).Value Then
                GoTo exit_q
            End If
        Next
        Sheet2.Range("B" & CStr(q)).Value = Sheet2.Range(tmpaddr).Value
        q = q + 1
        Range("h34").Value = CStr(q - 34)
    exit_q:
    Case 3:
        For u = 34 To x - 1
            If Sheet2.Range("C" & CStr(u)).Value =
Sheet2.Range(tmpaddr).Value Then
                GoTo exit_x
            End If
        Next
        Sheet2.Range("C" & CStr(x)).Value = Sheet2.Range(tmpaddr).Value
        x = x + 1
    
```

```

        Range("i34").Value = CStr(x - 34)
exit_x:
    Case Is > 3:
        For u = 34 To y - 1
            If Sheet2.Range("D" & CStr(u)).Value =
Sheet2.Range(tmpaddr).Value Then
                GoTo exit_y
            End If
        Next
        Sheet2.Range("D" & CStr(y)).Value = Sheet2.Range(tmpaddr).Value
        y = y + 1
        Range("j34").Value = CStr(y - 34)
exit_y:
    ' Case Else
    '     w = w + 1
    '     Range("k34").Value = CStr(w)
    End Select
Next
End Sub
Private Function CountIf(ByVal tmpR As Range, ByVal tmpN As String) As
Integer
    Dim a, b, c As Integer
    b = 0
    For a = 1 To tmpR.Cells.Count
        If tmpR.Cells(a).Value = tmpN Then
            b = b + 1
        End If
    Next
    CountIf = b
End Function

```

上次讲到了每个分支结构的内部都是来处理对应个数数字的，比如第一个分支就是处理变量 J 等于 1 的情况的，这个从 CASE 后面跟的条件可以看出来，其他还有 J 等于 2、J 等于 3 以及 J 大于 3 这样三种情况。在每个分支内部的代码，大家仔细观察可以发现基本大同小异，因为毕竟都是做统计个数这个事情，唯一的不同是计数存放的地方不同而已。以 J 等于 2 为例，首先是一个 FOR 循环，这个循环的作用是将当前单元格的值与存放在 A1 到 AF30 区域出现 2 次的数字的队

列中的数进行比较，如果发现当前单元格的值已经被登记在这个队列里了，那么立即结束这个分支结构，GOTO exit_q 是一个无条件转移语句，目的是将程序下一步执行的目的跳转到 exit_q 这个标号表示的地方。如果通过 FOR 循环发现当前单元格的值没有被登记在该队列里，那么就将该单元格的值登记到该队列里，并将计数器 Q 加 1，同时改变记录出现 2 次的单元格的统计数字。

还有一个 FUNCTION，名字叫做 CountIf，有 2 个参数，前一个是表示一个区域，后一个是表示一个字符串，结果返回一个整形数值。该函数的作用是按照第一个参数指定的区域范围，统计某一个字符串出现的次数，并将该次数的数值作为函数结果返回给调用者。

好了，这个程序就讲解到这里。可能有的朋友会说，我还没懂呢，呵呵，没关系，从下一次课开始，我们将开始一个全新的学习旅程，我把它称为“从实例学 VBA 编程”，通过一个帖子一个实例的方式，带领大家逐步迈进 VBA 的大门，一步一个脚印，脚踏实地地打下扎实的基础！让我们为最终掌握 VBA 这门技术并为我所用而不懈努力吧！

朋友们，再见！顺祝大家元宵节快乐！

摘录：xxd36xxd