电子表格应用开发入门

本章内容：

- 了解电子表格应用开发中的基本步骤
- 确定最终用户的需求
- 设计应用以满足用户的需求
- 开发并测试应用
- 记录开发过程并编写用户文档

1.1 关于电子表格应用

就本书的目的而言，电子表格应用是一种电子表格文件(或与之相关的一组文件)，用来帮助未经专门培训的非开发人员执行各种有用的操作。如果这样定义的话，已开发的绝大多数电子表格文件大概并不符合要求。在你的硬盘上，可能会有几十个或几百个电子表格文件，不过可以肯定地说，其中大部分的设计难以令人满意。

一个好的电子表格应用应该：

- 最终用户可以用来执行在其他情况下无法执行的任务。
- 给问题提供合适的解决方案(电子表格环境不可能总是最佳的方法)。
- 完成预设的目标。这一条看起来很简单，但实际上应用经常完不成预设目标。
- 生成精确的结果并摒除漏洞。
- 运用恰当有效的方法和算法来完成工作。
- 在用户被迫面对错误之前先捕获这些错误。
- 不允许用户无意或有意地删除或修改重要的组件。
- 用户界面清晰而且逻辑一致，用户可以一眼看出该如何进行操作。
- 公式、宏、用户界面元素有据可查，如有必要可进行后续的修改。
- 当用户的需求随时间发生变化时，只需要对应用进行简单修改而不必做重大改变。
- 具有简单易用的帮助系统，至少在主要操作步骤中能提供有用的信息。
- 是轻型的，可在装有适当软件(本书中指的是某个版本的 Excel)的任何系统上运行。

在日常应用开发中，我们通常需要创建很多不同使用级别的电子表格应用，这一点显而易见，可能是只需要填充空白的模板，也可能是极其复杂的应用，用了自定义的界面，看起来都已经不像是电子表格了。

1.2 应用开发的步骤

开发有效的电子表格应用并没有什么简单的万全之策。在创建这类应用时每个人都有各自的风格，另外，每个工程(project)也都不一样，因此需要适合应用自身的创建方法。在开发过程中，还需要你所服务的客户提供具体的需求和技术专长。

注意，本章后面将交替使用“应用”和“应用程序”，它们的含义是相同的。

电子表格开发人员通常进行如下活动：

- 确定用户的需求
- 对满足这些需求的应用进行规划
- 确定最合适的用户界面
- 创建电子表格、公式、宏和用户界面
- 测试和调试应用
- 使应用更安全
- 使应用更加简明美观
- 将开发成果进行归档
- 开发用户文档和帮助系统
- 给用户发布应用
- 必要时对应用进行更新

不是每个应用都需要上述所有步骤，不同工程之间上述活动的执行顺序也有所不同。下面将对每个活动进行详细描述，对于这些活动中会涉及的具体技术细节，我们将在后续章节中进行讨论。

1.3 确定用户的需求

当开发新 Excel 工程时，首先要做的就是准确识别出最终用户的需求。如果没有尽早详尽评估出用户的需要，通常在后期就不得不对应用进行调整，从而增加了额外工作量。因此，在起始阶段就应该确定到底有哪些需求。

有些工程中，你会非常熟悉最终用户——甚至有时你就是最终用户。但有些工程中(例如，如果你是为新客户开发应用的咨询顾问)，你可能会对用户或用户的情况一无所知。

那么，如何确定用户的需求呢？如果准备开发电子表格应用，最好直接接触最终用户并询问具体问题。要是把所有收集到的信息都写下来，画好流程图，关注最具体的细节，那就更好了。总之，尽量做好充足准备以确保你最终交付的产品正是客户所需要的。

下面是一些指导原则，能帮助你在这一阶段更轻松一些：

首先，不要假定你了解用户的需求。这阶段的自作聪明只会导致后续的一系列问题；

- 如有可能，直接与应用的最终用户(而非他们的管理人员或经理)对话；
- 若说最该做的事，应该是了解当前要做哪些改变以满足用户的需求。如果可以在现有的应用上进行简单修改，你就可以节省一部分工作量。看看当前解决方案至少能让你对操作更熟悉一些。
- 确定在用户的站点上有哪些可用的资源。例如，确定是否需要面对工作环境中硬件或软件的限制。
- 如有可能，确定一下将要用到的具体硬件系统。如果应用会在较慢的系统中运行，就得考虑这个因素。
- 确定将使用哪个版本的 Excel。虽然微软已经设法敦促用户将软件更新到最新版本，但大多数 Excel 用户还是没有更新。
- 了解最终用户的软件使用水平。这一信息可以帮助你更合理地设计应用。
- 确定开发应用需要的时间，以及整个工程的生命周期中是否已预料到了所有变化。这一信息对你在工程中付出的所有努力有所影响，可以帮助你有计划地应对变化。

最后，如果在完成应用开发前发现工程规范发生了改变，你也别太惊讶，这种事情很常见，如果有心理准备面对变化，而不受到变化的惊吓，这已经是件好事了。反正你能确保你的合同(如果有)已经纳入规范发生改变时的对策就行了。

1.4　对满足这些需求的应用进行规划

确定了最终用户的需求后，接下来就该直接进入 Excel 的开发工作了。要相信面临同样问题的人的忠告：试着控制自己。没有一套设计蓝图，建筑师就无法盖房，同样，没有一些计划，也没法开发电子表格应用程序。计划的形式取决于工程的范围以及你的整体工作风格，不过最好花点时间去想想接下来要做什么并拿出活动计划。

在你卷起袖子在键盘前坐定准备开工前，最好再想想还有没有其他途径解决问题，在这个计划过程中可以用到 Excel 的所有知识，最好避免钻进死胡同，不要在死胡同里盲目地行走。

如果你去问一打 Excel 专家如何去根据确切的规范来设计一个应用，大概你会得到一打能够满足这些规范的且各不相同的工程实现方案。在这些解决方案中，有些将优于另一些，因为 Excel 会为解决同一问题提供多种选项。如果对 Excel 了如指掌，就可以随心所欲地选择最合适的方法来很好地解决工程中的问题。有时一个小创意都可以产生明显优于其他方法的奇效。

因此，在列计划的开始阶段，应该考虑如下一些常见的问题：

- **文件结构**：想想准备使用一个带有多个工作表的工作簿、多个使用单工作表的工作簿还是模板文件。
- **数据结构**：应该考虑准备如何排列数据，还要确定是准备使用外部数据库文件，还是在工作表中存储数据。
- **插件或工作簿文件**：某些情况下，对于你的最终产品来讲，插件可能是最好的选择。或者可能需要为标准的工作簿使用插件。
- **Excel 的版本**：你现在只使用 Excel 2016 吗？还是也使用 Excel 2010 及其之后的版本？Excel 2003 及其更早的版本怎么样？你的应用程序也会在 Macintosh 上运行吗？这些问

题都很重要，因为 Excel 的每个新版本添加的功能都不会适用于先前的版本。在用 Excel 创建应用方面，Excel 2007 中引入的功能区界面就比旧版本功能强大很多。

- **错误处理**：错误处理是应用程序中的主要问题。你需要确定应用程序如何删除和处理错误。例如，如果你的应用程序对活动工作簿进行格式化操作，你就需要能够处理活动图表的情况。
- **具体功能的使用**：如果你的应用程序需要汇总大量数据，就可以考虑使用 Excel 的透视表功能。或者可以使用 Excel 的数据验证对数据项输入进行有效性验证。
- **性能问题**：关于提高应用的速度和效率，这一步应该在应用开发阶段就开始考虑，以免应用开发结束后用户在使用过程中发出抱怨。
- **安全级别**：Excel 提供了多种保护选项以限制访问工作簿中的特定元素。例如你可以锁定单元格以防止公式被改变，可对具体文件进行加密，以防止未授权用户查看或访问。预先确定你需要保护的内容，以及对此采用什么级别的保护措施，可使工作变得容易些。

注意

需要注意，Excel 的保护功能并非百分百有效，事实上差得远。如果你特别希望解决应用程序的安全问题，Excel 并不是最佳平台。

在该阶段，你可能需要去处理许多与工程相关的因素，尽量考虑所有的选择，而不要一想到某个方案就赶紧使用。

要记住设计时需要考虑的另一个因素，那就是对变化要有所计划。如果你将应用程序设计得尽量通用，那就当是帮自己忙了。例如，不要为一段具体范围内的单元格单独编写一个过程，而应该编写将任何范围作为一个变量的过程。这样，当需要进行不可避免的改变时，这样的设计能帮助你在进行修正时更轻松一些。而且，你将看到，做某个工程时的工作与做另一个工程时的工作类似，因此，在做工程规划时要尽量考虑可重用性。

避免让最终用户完全影响你解决问题的方法。例如，你碰到一位经理，他说部门需要一个应用程序，以便将文本文件导入到另一个应用中。别被用户对解决方案的需求所迷惑，其实用户的真实需求是共享数据。使用中间文本文件进行处理只是一种可能的办法，应该还存在更好的解决办法。换言之，不要让用户来定义用什么样的方法来解决他们的问题。确定最佳方法是你的工作。

1.5 确定最适用的用户界面

在开发供他人使用的电子表格应用时，需要额外关注用户界面问题。通过用户界面，用户就可以与应用进行交互以及执行 VBA 宏。

自从 Excel 2007 问世以来，有些用户界面方面的功能就落伍了。从实际用途来讲，定制菜单和工具栏就被淘汰了。因此，开发人员必须学习如何使用功能区。

Excel 提供了一些与用户界面设计相关的功能：

- 自定义功能区

- 自定义快捷菜单
- 创建快捷键
- 自定义对话框(UserForm)
- 可直接放在工作表上的控件(例如列表框或命令按钮)

下面将简单讨论这些功能，后续章节还会更深入地进行讲解。

1.5.1　自定义功能区

Excel 2007 引入的功能区界面是用户界面设计中的重大改变。在功能区中开发人员可以使用数量相当多的控件。虽然 Excel 允许最终用户修改功能区，但通过代码来改变用户界面也不太容易。

交叉参考

关于功能区的使用信息请参阅第 17 章。

1.5.2　自定义快捷菜单

Excel 允许 VBA 开发人员自定义右键快捷菜单。快捷菜单可以让用户方便地触发某个事件，而不需要费力地从工作区域中让光标移动太远的距离。图 1-1 展示了右击单元格时出现的自定义快捷菜单。

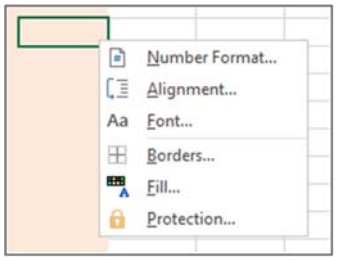

图 1-1　自定义快捷菜单

交叉参考

第 18 章描述了如何使用 VBA 创建快捷菜单，包括 Excel 2013 中引入的单文档界面带来的一些局限性。

1.5.3　创建快捷键

另一个可由你自由选择的用户界面选项是自定义快捷菜单。Excel 可以用 Ctrl(或 Shift+Ctrl)组合键来定义宏。当用户按下定义好的组合键时，就可以执行宏。

不过，需要先声明两点。第一，要让用户清楚哪些键是活动的，这些键能做什么。第二，不要再给已作他用的组合键定义新功能。你为宏设定的组合键优先级将高于内置的快捷键。例

如，Ctrl+5 是用来保存当前文件的内置 Excel 快捷键。如果你将这个组合键定义为一个宏的快捷键，就不能再用 Ctrl+5 来保存文件了，记住，快捷键是区分大小写的，所以，你可以使用像 Ctrl+Shift+S 这样的组合键。

1.5.4 创建自定义对话框

对于对话框，用过计算机的人都很熟悉。因此，在设计应用时，自定义 Excel 对话框将在用户界面中扮演重要角色。

图 1-2 就是一个自定义对话框的例子。

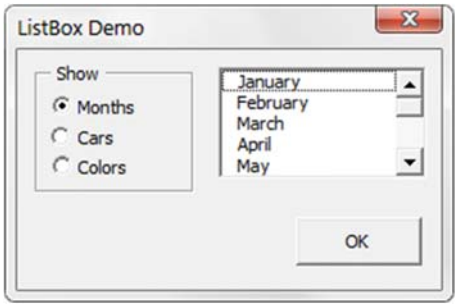

图 1-2 Excel 的 UserForm 功能中的对话框

自定义对话框即 UserForm，UserForm 可以让用户输入信息，获得用户的选择或偏好，能够指引整个应用的使用过程。组成 UserForm 的元素(按钮、下拉列表、复选框等)，我们称为控件——更确切地说，是 ActiveX 控件。Excel 提供了 ActiveX 控件的标准种类，当然，你也可以引入第三方控件。

在对话框中添加完控件后，可将其关联到工作表单元格，这样就不需要任何宏(除了用来显示对话框的简单宏)。将控件和单元格相关联很简单，但并不是从对话框中获得用户输入的最好办法。这时最常见的做法是为自定义对话框开发 VBA 宏。

交叉参考

第 III 部分将详细讨论 UserForm。

1.5.5 在工作表中使用 ActiveX 控件

Excel 同样允许你将 UserForm 的 ActiveX 控件与工作表的绘图层(位于表顶部的可以保存图像、图表和其他对象的不可见层)相关联。图 1-3 展示了一个将一些 UserForm 控件直接内嵌到工作表的简单工作表模型。这个表包含下列 ActiveX 控件：复选框、滚动条以及两组单选按钮。这个工作簿没有使用宏，而是直接将控件关联到工作表单元格上。

在线资源

这个工作表可从本书的示例文件包中找到，文件名是 worksheet controls.xlsx。

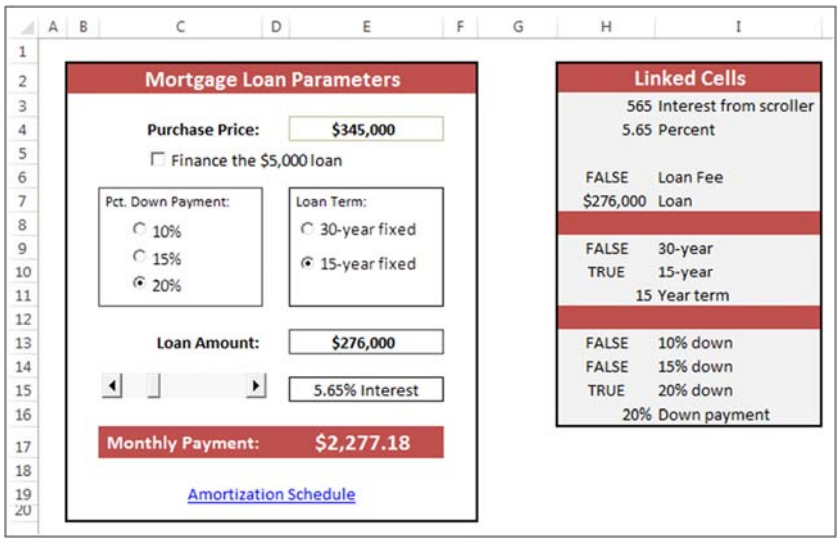

图 1-3　可将 UserForm 控件添加到工作表中，并将它们与单元格关联

最常用的控件应该是命令按钮。就它自身而言，并没有什么功能，你需要给每个命令按钮添加一个宏。

如果在工作表中直接使用对话框控件，就不再需要自定义对话框。只要向工作表中添加少量的 ActiveX 控件(或表单控件)就可以极大地简化电子表格的操作。选择这些 ActiveX 控件时，用户可通过选中熟悉的控件(而不用在单元格中输入对应条目)来完成。

通过“开发工具”|“控件”|“插入”命令(如图 1-4 所示)可以访问这些控件。如果“开发工具”选项卡不在功能区中，则可以通过使用“Excel 选项”对话框中的“自定义功能区”选项卡来添加。

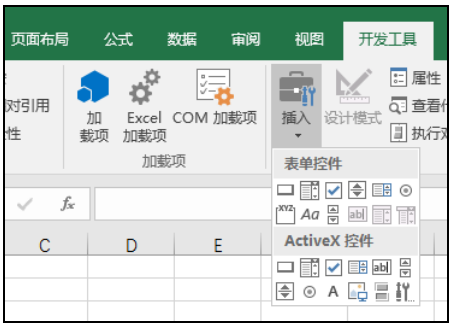

图 1-4　使用功能区给工作表添加控件

控件有两种类型：表单控件和 ActiveX 控件。两种控件有各自的优缺点。通常来讲，表单控件更易于使用，但 ActiveX 控件使用起来更灵活一些。表 1-1 总结了这两类控件。

表 1-1 ActiveX 控件与表单控件

	ActiveX 控件	表单控件
Excel 版本	97、2000、2002、2003、2007、2010、2013、2016	5、95、97、2000、2003、2007、2010、2013、2016
可用控件	复选框、文本框、命令按钮、选项按钮、列表框、组合框、切换按钮、数值调节钮、滚动条、标签、图像控件(以及其他可添加的控件)	分组框、按钮、复选框、选项按钮、列表框、组合下拉编辑框、滚动条、下拉列表
宏代码存储位置	工作表的代码模块中	任何标准 VBA 模块中
宏名称	对应控件名(例如 CommandButton1_Click)	指定的任意名
对应	UserForm 控件	Excel 97 之前版本中的 Dialog Sheet 控件
自定义	范围很广，使用“属性”对话框	很少
响应事件	是	仅在单击或改变事件时

1.5.6 开始开发工作

确认用户需求后，确定你将采用什么方法来满足这些需求，并确定在用户界面上使用哪些组件，现在该着手处理细节问题开始创建应用了。对于具体工程来讲，这一步显然要花费整个开发周期的绝大部分时间。

如何进行应用的开发由你个人的开发风格以及应用的性质决定。除了简单的填空型模板工作簿外，你的应用基本都会用到宏。在 Excel 中创建宏挺简单的，但要创建出优秀的宏并不容易。

1.6 关注最终用户

本节将讨论一些重要开发问题；在应用开发即将结束，准备打包并分发时，这些问题将浮出水面。

1.6.1 测试应用

在使用商业软件应用时，是不是碰到过很多次应用程序在关键时刻掉链子？这问题很可能是因为程序未经充分测试从而未捕获到所有 bug 引起的。所有著名软件都有 bug，但最优秀软件中的 bug 都是很隐蔽的。因此，有时你也必须解决 Excel 中的 bug 以使你的应用能正确执行。

创建完应用程序后，就该进行测试。测试是最重要的步骤之一，当然，测试和调试程序时所花费的时间并不会和创建程序一样多。实际上，在开发阶段应该做大量的测试。不过别忘了，不管是在工作表中编写 VBA 例程还是创建公式，都应该确定应用程序是在按设定好的途径工作。

像标准的编译后的应用程序一样，你所开发的电子表格应用也很容易有 bug。bug 的定义有两种：一、程序(或应用)运行时本不该发生的事情发生了；二、程序运行时本该发生的事情没

有发生。这两类 bug 都很令人生厌，你必须将一部分开发时间分配出来对应用程序在所有可能的条件下进行测试，并修复你发现的任何问题。

对你为其他人所开发的电子表格应用程序进行详尽测试是非常重要的。而且，考虑到它的最终用户，你还需要注意到应用程序的安全问题。换句话说，你要尽量预测到可能会发生的所有错误和其他乱七八糟的事情，并努力去避免——至少能通过和缓的手段来解决它们。这种预见不仅能帮助最终用户也能让你的工作更容易些，并且保护你的工作信誉。你也可以考虑进行 beta 测试，那你的最终用户就是最合适的人选了，因为他们是将要使用你产品的人(下面将解释什么是 beta 测试)。

虽然你不可能对所有的可能性都进行有效测试，但宏可以处理一些常见类型的错误。例如，如果用户在输入数值型字符时输入了文本字符串会发生什么情况？如果工作簿还没打开时用户就想运行宏会怎么样？如果用户没有进行任何选择就取消了对话框会发生什么？如果用户按下 Ctrl+F6 跳转到下一个窗口又会怎么样？如果你经验丰富，就会对这些类型的问题很熟悉，不用多想就知道该如何处理它们。

什么是 beta 测试？

对于新产品来讲，软件生产通常都会有一个严格的测试周期，经过全面的内部测试，该预发布产品通常会让一些感兴趣的用户进行 beta 测试。这一阶段通常会暴露出产品最终版发布前要解决的其他一些问题。

如果你正开发的 Excel 应用的最终用户人数不少，你就需要考虑一下 beta 测试。这个测试可以让你的预期用户在具有不同预先设置的硬件上使用你的应用。

在自行完成对应用的个人测试并觉得应用可以发布后就可以开始 beta 测试了。你需要确定一批用户来帮助你。如果将用户文档、安装程序、帮助等最终都包含在应用中的内容也发布出来，测试过程会更好些。可通过多种方法对测试进行评估，如面对面的讨论、电子邮件、调查问卷以及电话调查等。

在你准备全面发布应用之前，你大概一直要关注那些你需要解决的问题以及需要做的改进。当然，beta 测试阶段会额外花费一些时间，所以不是所有的工程都能这么奢侈地挤出时间来进行 beta 测试。

1.6.2　应用的安全问题

提到这个问题，其实会发现摧毁电子表格其实相当容易。删除一条关键公式或值所引起的错误会影响整个电子表格，甚至影响其他相关的工作表。甚至更糟的是，如果保存了已被破坏的工作表，还会覆盖掉硬盘上原本完好的备份。除非有备份的过程能进行替代，否则，应用的用户就会陷入麻烦，而你就可能会因此而被抱怨。

当用户(尤其是初学者)使用你的应用时，你显然应该已经知道为什么要给应用加一些保护了。Excel 提供了一些技术来保护整个工作表及工作表的局部：

- **锁定指定单元格：**可锁定指定的单元格(利用“单元格”的“格式”对话框中的“保护”选项卡)，这样用户就不能修改它们。仅当使用“审阅”|“更改”|“保护工作表”命令

来保护文档时锁定会生效。“保护工作表”对话框中的选项允许你指定用户可以在受保护工作表上执行哪些操作(如图 1-5 所示)。

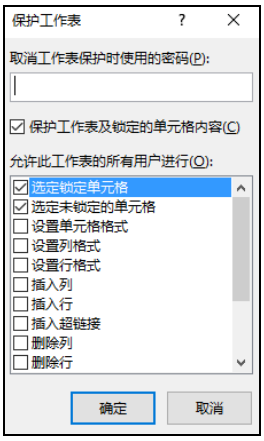

图 1-5 使用“保护工作表”对话框来指定用户能做什么和不能做什么

- **隐藏指定单元格中的公式**：你可以隐藏指定单元格中的公式(利用“单元格”的“格式”对话框中的“保护”选项卡)，这样其他用户就看不到了。同样，仅当使用“审阅”|“更改”|“保护工作表”命令来保护文档时隐藏生效。
- **保护整个工作簿**：你可以保护整个工作簿——工作簿的结构、窗口的位置和大小。使用“审阅”|“更改”|“保护工作簿”命令可以生效。
- **锁定工作表中的对象**：使用任务面板上的“属性”区域可以锁定对象(例如形状)，防止对象被移动或更改。要访问任务面板的属性区域，可以右击对象，选择“大小和属性”。仅当使用“审阅”|“更改”|“保护工作表”命令来保护文档时锁定对象会生效。默认情况下，所有对象都是被锁定的。
- **隐藏行、列、工作表和文档**：可隐藏行、列、工作表和整个工作簿。把它们隐藏起来可以让工作表看起来没那么乱，还可以防止别人的窥探。
- **将 Excel 工作簿指定为推荐的只读模式**：可将 Excel 工作簿指定为推荐的只读模式(和使用密码)，以确保文件不会被任何修改所重写。可在“常规选项”对话框中进行这种指定。在“文件”|“另存为”对话框中，单击“工具”按钮，从中就可以选择“常规选项”。
- **设置密码**：可对工作簿设置密码以防其他未授权用户打开你的文件。选择“文件”|“信息”|“保护工作簿”|“用密码进行加密”。
- **使用密码保护的加载项**：可使用经过密码保护过的加载项，这样可防止用户对它的工作表进行任何修改。

Excel 的密码并不是万无一失的

需要注意，利用一些商业的密码破解工具，可轻易绕开 Excel 的密码。Excel 2007 及后续版本的安全性看起来比先前的版本更高些，但实际上还是防不住有心破解密码的用户。所以，不要认为密码保护万无一失，对于无心的用户来讲密码是有用的，但对于有意想破解密码的人来讲，他们很可能得逞。

1.6.3　如何让应用程序看起来更简明美观

如果你用过很多不同的软件包，无疑知道很多程序用户界面设计很差，使用起来困难，外观简陋。如果你给他人开发电子表格，最好多花点精力设计一下应用的外观。

计算机程序的外观对用户有非常重大的影响，对于用 Excel 开发的应用程序来讲，也同样如此。不过，“情人眼里出西施”，所以，如果你确实更擅长于执行问题分析之类的工作，那可以考虑找些审美能力更强的人来帮你提供外观设计方面的帮助。

不过，比较值得欣慰的是，Excel 2007 及后续版本中的有些功能可以相对轻松地帮助你创建出更好看的电子表格。如果你坚持使用已经预设好的单元格样式，那基本上可以保证程序外观还不错。另外，动动鼠标，你还可以轻松地为工作簿更换新的主题样式，看起来同样美观。

最终用户都喜欢美观的用户界面，如果你多花点时间去考虑设计和审美问题，你的应用程序的外观看起来就会更美观更专业。一个让人眼前一亮的应用至少说明了它的开发者愿意花时间和精力去尽力完善这个产品。你可以参考以下建议：

- **保持一致性**。例如，在设计对话框时，尽可能去模仿 Excel 对话框的外观和感觉。在格式、字体、文字大小以及颜色方面要保持一致。
- **尽量简洁**。开发人员有个常见的误区是在单个屏幕或对话框中塞进去大量信息。较好的做法是一次只出现一块或两块信息。
- **划分界面**。如果使用输入界面来征求用户的信息，就要考虑将其分成若干个看起来不那么拥挤的界面。如果使用了复杂的对话框，就需要使用多页控件为对话框定义多个页面，创建出一个类似于选项卡的对话框。
- **不要过度使用颜色**。少用点颜色。用的颜色太多容易让界面看起来花哨且俗艳。
- **关注版式和图表**。注意数字的格式和使用一致的字体、字号和边框。

不过，对于审美的评价都是主观的，如果心有疑虑，尽量保持简洁清晰就好了。

1.6.4　创建用户帮助系统

就用户文档而言，基本上你有两种选择：纸质文档或电子文档。对 Windows 应用程序来讲提供电子帮助是标准配置。

幸运的是，Excel 也提供了帮助——而且是上下文相关的帮助。编写帮助文本内容需要花费不少额外的精力，但对于大型工程来讲，还是有必要的。

另一个需要注意的是对应用的支持。换句话说就是如果用户碰到困难了，谁来接电话解决？如果你不打算自己来处理常规问题，就需要指定专人。有些情况下，你需要做好安排以便将技术性较强或与 bug 相关的问题提交给开发人员。

交叉参考

第 19 章将讨论为应用提供帮助的一些替代方案。

1.6.5 将开发成果归档

将电子表格应用组成一个整体是一回事，想让它为其他人所知又是另一回事了。和传统的编程一样，彻底梳理并将工作归档很重要。如果你想返工的话，这样的文档可以提供很大帮助，同时，该文档也可以帮助接手你工作的其他人。

如何对工作簿应用进行归档呢？可将信息存储在工作表中或者使用其他文件。如果愿意的话，你甚至可以使用纸质文档。最简单的方式应该就是用一个单独的工作表来存储工程的注释和关键信息。对于 VBA 代码，可随意使用注释(注释符之后的 VBA 文本都会被忽略，因为该文本会被视为注释)。虽然现在你觉得一段简洁的 VBA 代码看起来非常易于理解，但如果过几个月后再来看，除非使用了 VBA 注释功能，否则会发现你对代码的理解完全模糊了。

1.6.6 给用户发布应用程序

完成工程后，就该给最终用户发布应用了。你打算如何发布呢？有很多种途径可以选择，具体选择哪种全看现实因素了。

可将应用放到 CD 或 U 盘上，写上几条指令，就可以安装了。或者你想亲自安装应用程序——但这种方法并不总是可行的。另一种选择是开发可以自动执行任务的官方安装程序。你可以用传统的编程语言编写这样的程序、购买通用的安装程序，或者用 VBA 编写该程序。

Excel 中可以让开发人员对自己的应用进行数字签名。数字签名可以帮助最终用户识别出应用的作者，以确保该工程没被改动过，而且还可以防止宏病毒的扩散或其他潜在的破坏性代码。要对工程进行数字签名，首先要从正规的证书颁发机构申请数字证书(或者通过创建你自己的数字证书对你的工程自签名)。参考帮助系统或微软的网站可获得更多信息。

1.6.7 在必要时对应用进行更新

发布完应用后，是不是就万事大吉了呢？可以轻松坐着享受，忘掉开发应用时遇到或解决掉的各种问题。在极少数情况下，你确实可以这样。不过，更常见的情况是，使用你应用的用户不会完全满意。哪怕你的应用完全遵循最初所有的规范要求，但用户在使用应用时，还是会生出其他要求希望应用能够满足。

当你更新或修正应用时，将发现在第一阶段就做好设计并且对开发过程进行归档是件非常明智的事。

1.7 其他开发问题

在开发应用时你就需要关注一些问题——尤其是你不能完全确定使用你应用的用户是谁的话。如果你所开发的应用将会被广泛应用(例如共享应用软件)，你没办法知道应用将会被如何使用，会在什么系统上运行，或者会同时运行其他什么软件。

1.7.1　用户安装的 Excel 版本

虽然 Excel 2016 已经发布了，但很多大公司还在使用早期版本的 Excel。

不过，谁都不能保证为 Excel 2010 开发的应用程序能和 Excel 的后续版本完美适应。如果你希望自己的应用适用于各种版本的 Excel，最好的办法是针对最低版本进行开发，但在各种版本下对它进行测试。

同样，你需要关注微软发布的所有服务包和安全更新。虽然这些发布中引入的变化可能很小，但有可能会让你应用中的某些组件不能再如设计时所设定的那样工作。

交叉参考　第 21 章将讨论兼容性问题。

1.7.2　语言问题

如果你的最终用户使用的都是英文版的 Excel，那你很幸运。因为非英文版的 Excel 并不能百分百地兼容，这也就意味着你还需要做些额外的测试工作。另外需要记住，两个用户虽然都使用英文版的 Excel，但 Windows 的区域设置可能并不一样。某些情况下，你需要注意到这些潜在的问题。

交叉参考　第 21 章将重点讨论语言问题。

1.7.3　系统速度

你可能是紧跟时代潮流的计算机用户，经常更新自己的硬件设备。也就是说，你的系统比所有计算机用户系统的平均水平更好。有时，你可能恰好知道你所开发的应用的最终用户用什么样的硬件，那你就很有必要在那种硬件条件下对应用进行测试。在你的机器上瞬间就能执行完毕的过程，在另一系统上很可能就需要花费好几秒。对于计算机用户来讲，数秒钟时间就已经很难接受了。

提示　如果对使用 VBA 已经有一定的经验，你会发现想达到一个目标以及尽快达成目标有多种方法。在编码时考虑到运行速度是个很好的习惯。本书中有些章节会对这一问题进行讲解。

1.7.4　显示模式

你该知道，用户的显示器设置千差万别。当前最常见的视频分辨率是 1280*1024，其次是

1024*768。分辨率设置为 800*600 的系统现在已经很少见，但还有相当一部分用户在使用。高分辨率显示器及双屏扩展显示器慢慢变得流行起来。如果你有超高分辨率的显示器，并不能假定别人的显示器都和你的一样。

如果你的应用中指定了如何在单个屏幕中显示的具体信息，那视频分辨率就是个问题。例如，你开发了一个模式为 1280*1024 才满屏的输入屏，那如果用户的计算机分辨率设置成 1024*768，就需要拖滚动条或进行缩放才能看到完整屏幕。

另外必须注意，还原后(即不是最大化也不是最小化)的工作簿应该以还原前的窗口大小和位置显示出来。在有些极端情况下，可能会出现这样的状况：在高分辨率显示器保存的窗口，在低分辨率系统上运行时，可能在屏幕上完全显示不出来。

不过，由于你不可能自动处理这种小事，因此除了显示分辨率外，其他都没什么区别。有些情况下，可缩放工作表(使用状态栏中的缩放控件)，不过这样做确实很困难。因此，如果不能确定使用你的应用的用户的视频分辨率是多少的话，在设计应用时最好还是采用大众化的选择——800*600 或者 1024*768 模式。

在本书后续章节中，可以了解到从 VBA 中调用 Windows API 就可以确定用户的视屏分辨率。有时，可能需要根据用户的视屏分辨率通过编程方式来判断某些情况。