

Broadview®  
www.broadview.com.cn

On Software Development

# 软件开发

## —— 软件开发工具手册

# 这点事儿

邵志东 著

超值附送 55小时教学视频



电子工业出版社  
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY  
http://www.phei.com.cn

On Software Development

# 软件开发

## 这点事儿

— 软件开发工具手册

邵志东  
著

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

## 内 容 简 介

软件技术和编程语言的发展日新月异，如何能在如此快速的技术革新中把握技术发展趋势，掌握软件开发的精髓；掌握一门编程语言就是程序员了吗？如何能够成为一名优秀的程序员？作为一名程序员，职业发展路在何方……在本书中，读者将会找到这些问题的答案。

在本书的开章，读者可以了解到作为一个软件程序员需要掌握的专业和素质技能，并了解到软件程序员的职业发展方向；在本书知识篇、工具篇中，读者可以学习到软件开发的精髓：软件开发过程中所涉及的相关知识、需要掌握的相关工具，可以快速掌握这些开发工具；最后，在本书的模板篇、规范篇，读者可以惊喜地找到非常实用的软件开发模板和项目组建设所需要的规范文档。通过本书的帮助，相信广大读者能快速走上真正的软件开发之路！

本书非常适合软件程序员或有志于成为软件程序员的读者阅读。

未经许可，不得以任何方式复制或抄袭本书之部分或全部内容。  
版权所有，侵权必究。

### 图书在版编目（CIP）数据

软件开发这点事儿：软件开发工具手册 / 邵志东著.  
北京：电子工业出版社，2009.3  
ISBN 978-7-121-08072-2

I. 软… II. 邵… III. 软件工具—手册 IV. TP311.56-62

中国版本图书馆CIP数据核字（2008）第212410号

责任编辑：李 冰

印 刷：北京智力达印刷有限公司

装 订：北京中新伟业印刷有限公司

出版发行：电子工业出版社

北京市海淀区万寿路173信箱 邮编100036

开 本：787×980 1/16 印张：19.5 彩插：4 字数：402千字

印 次：2009年3月第1次印刷

印 数：4000册 定价：45.00元(含光盘1张)

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话：（010）88254888。

质量投诉请发邮件至zltz@phei.com.cn，盗版侵权举报请发邮件至dbqq@phei.com.cn。

服务热线：（010）88258888。



## 给程序员的一封信

关于软件编程的学习，有些感受和大家交流。

### 1. “软件开发学习的重点是流程，而非技术本身”

事实上，技术的发展是日新月异的。记得我第一次做项目用的编程工具是TurboC，后来用VC++5.0，再后来开始在微软中文网络广播讲VisualStudio.NET 2002，不久又讲VisualStudio.NET 2003，最近讲的是VisualStudio.NET 2005，目前VisualStudio.NET 2008已经出来了……如此快的更新速度，不由得让我们感叹：“做一个程序员真累啊！”。

显然，作为一个程序员，要求一直走在时代的前列，真的很难！那么面对如此快速的变化，我们该怎么办呢？

其实，细心的程序员分析一下，可以看到，在如此快速的技术变化后面，软件开发的流程基本没有改变！原来困扰我们的只是一个开发工具、一门编程语言而已。真正的武学大师，何必在意手中的武器到底是刀，还是剑呢？

所以，笔者认为，了解软件开发的过程，把握各个阶段需要掌握的基本知识，熟悉各个阶段所涉及的工具，这才是“以不变应万变之道”！

掌握了这些，所有的编程语言和开发工具，都是任君选择。是用Java这个“倚天剑”，还是用C#这把“屠龙刀”，就由我们的个人爱好和开发的需要来决定了！

这也是我写作本书的原因。

### 2. 学习的重点：培养一种“习惯”

我们每个人都应该把学习的重点放在能力、方法的长进上，而不仅仅是具体的某种技术、某个知识点的学习。在如今的知识爆炸时代，新的技术、新的知识层出不穷，即使我们穷其一生，仍然不可能学习掌握所有的知识和技术，而只能是在茫茫的知识海洋中占据一片孤岛。所以我们的重点应该放在培养适合于自己的一种学习方法，更确切地说，是培养一种习惯，一种学习的习惯。

我再三强调大家要有意识地自己总结一套学习新东西的方法，就是这个意思。

如果能够掌握一套得心应手的学习新东西的方法，面对茫茫的知识海洋你也会信心十足的。正如一句笑话：如果你是一个数学高手，当面对新的技术挑战时，你问一下自己：难道这个技术比高等数学还复杂？这样一想，你顿时会信心十足。虽是笑话，但也是很有道理的。

### 3. 一些思维方式的培养

生活、学习中每个问题的求解，到最后或者是归结为一种数学模型，或者是归结为一种哲学思想。因此，我建议大家在自己的自由时间能抽空学习一些哲学、数学方面的思想。这对自己是终生有益的。

人的价值往往体现在他的思想上，正如一个公司的品牌价值：技术永远是要被淘汰的；但一种思想永远有其存在的价值。因此，当对某个问题，你自己有你自己的一套完整理论时，你的身价就会上一个（几个）新的台阶。

#### 4. 何为“高手”？如何成为“高手”？

所谓的“高手”是否就是什么都懂？并不是。上面谈过，世界上是没有一个人精通所有的知识的，从古至今都没有。所谓的高手只不过是对于某些事情的处理已经养成了一种良好的习惯。举例说明，一个VC高手，是否就是熟悉MFC的每个类？了解Windows的所有SDK？显然不是。现实中可能有这样的“高手”，但寥寥无几。

只要在遇到各种问题后能很快地找到解决问题的方法，就可以算得上是一个真正的高手了。

在实际的生活、学习中，我们应该注意多总结方法、修订方法，特别是对于一些重复发生的事情（大到“学习”这个行为，小到VC编程中每次遇到问题），我们应该总结每次发生问题时自己采取的方法，并在下次发生时采用。

时间一长，就会养成一种习惯，并能时常修订自己的这种习惯。

#### 5. 求精与求广

对于知识的学习，我们或者可以“求精”，即选中一个点，精通它，因为尽管知识是无限的，但具体到某个知识点是有限的，我们有可能对某个知识点的所有知识加以了解，或者“求广”，因为人的生存，归根到底是人的竞争，你懂的东西比别人多，你就会在竞争中获胜。

相比较两种情况，后者难一些，因为在“求广”的同时，其精度不好掌握：浅了如同一无所知；深了精力有限，不太可能。前者容易些，但这个点必须选对，否则就可能被社会遗弃。比如，对于我们，知识点有：VC、数据库、网络等，选择网络可能就比选择数据库的竞争力强些（特别是在网络编程比较火的现在）。

对于我们，我感觉还是先“求精”，有了一技之长，然后在这个基础上拓展自己掌握知识的范围。

但最重要的还是要逐渐从“靠技术吃饭”转到“靠思想吃饭”。

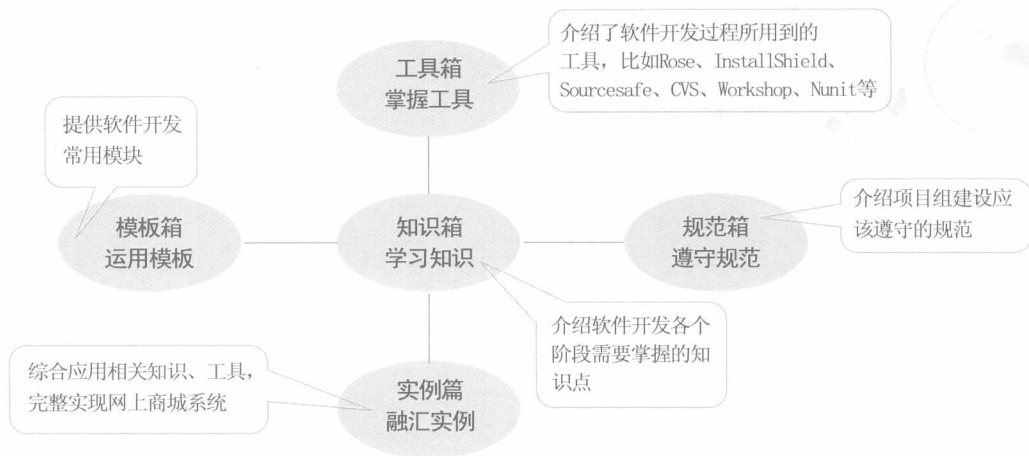
因此我们的重点应该是总结、培养自己的各种“习惯”。当一切成为一种下意识的举动时，你自己的“修为”就又上了一个新的台阶。

其实，在我看来，生活就是在培养种种新的习惯。如果把根据结果修订习惯也变为一种习惯，那就是一种真正意义上的“习惯”（比如，朋友。如果根本不相往来的话，那根本不算是朋友；但如果你和他的相处已经成为一种“习惯”后，你会发现他才是你真正的朋友，他已经成为你生活的一个部分）。

邵志东

2008年10月

# 本书导读



## 本书特色

工具、模板、示例、开发规范是本书的四大特色。

### 1. 工具

本书中介绍了许多软件开发过程所用到的11个工具，它们是Rational Rose、Power Designer、Nunit、JUnit、Install Shield、Visual Studio.NET、HLP文件制作工具、CHM帮助文件制作工具、Microsoft Visual SourceSafe、CVS、Visio等。对于这些工具介绍，都是本着能够让读者快速上手的原则。如果需要深入了解这些内容，可以参考相关资料，也可以访问本书提供的网址。

### 2. 模板

本书提供了软件开发流程各个阶段的质量模板文件，可以作为读者项目开发过程中的参考文件。

### 3. 实例

本书提供了一个“网上商城系统”的实战项目，详细介绍了从可行性分析、需求分析、系统设计、编码到安装盘等各个阶段的具体实现，并且提供了各个阶段的相关文档和具体代码。读者可以按照流程学习，体悟每个阶段的关键知识点，熟练掌握各种开发工具的使用，同时对于实际工作中的项目开发也有一定的指导意义。

### 4. 开发规范

在本书中提供了团队开发所需要的各种开发规范，包括项目组工作守则、编码规范、源代码控制规范、界面设计规范、数据库使用规范、安装盘制作规范等，会极大地方便团队开发。

## 技术支持

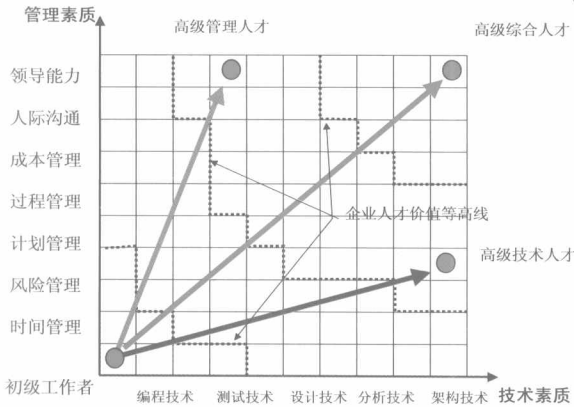
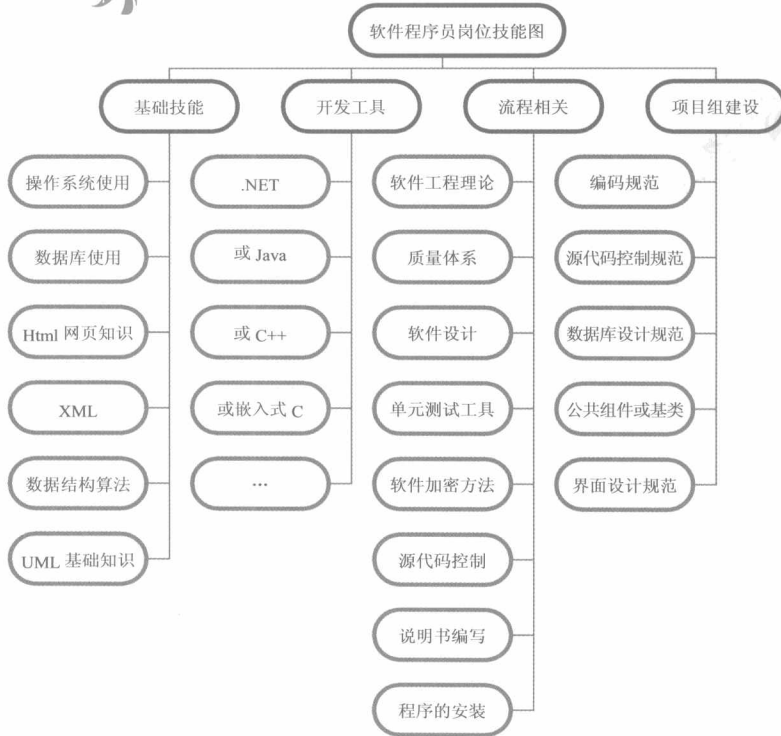
关于本书的任何问题，欢迎大家登录<http://blog.sina.com.cn/msdnwebcast2shaozhidong>进行讨论。

## 超大容量光盘介绍

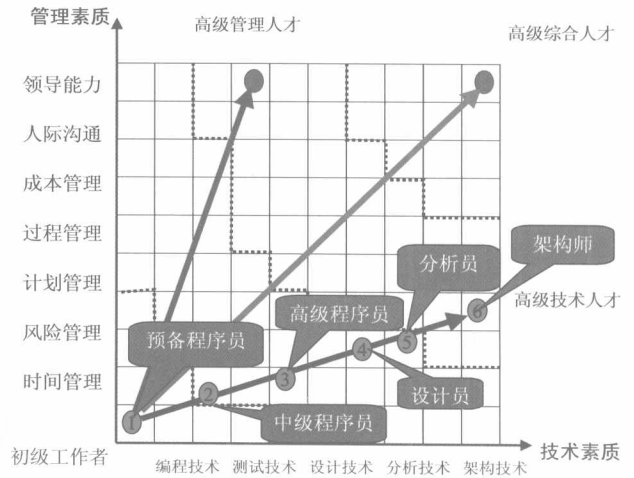
	目录名	说明	包含文件
1	ASP.NET常用代码总结	ASP.NET编程常用代码	41个常用代码段。如数据集的操作、数据库的操作、下载文件的方法等
2	ASP.NET系列视频讲座(长达55小时)	由作者主讲的长达55小时的ASP.NET系列完整视频讲座,包括知识点讲解、实例编程、实战编程、项目实战四大部分,共55讲(每讲1小时)每讲都提供PPT、Demo和视频文件	ADO.NET开发最佳实践 DataGrid使用最佳实践 DataSet和XML .NET框架类在ASP.NET中的使用 水晶报表的使用 ASP.NET程序的部署 在ASP.NET中使用用户控件 Web服务的创建和使用 ASP.NET中的错误处理和程序优化 ADO.NET数据操作 数据集的使用方法和技巧 调试和跟踪ASP.NET应用程序 ASP.NET安全性 Web服务器控件使用 ASP.NET状态管理 ASP.NET事务和异常处理 ASP.NET页面框架 ASP.NET实现在服务器端控制网页 ASP.NET中的事件处理 如何在ASP.NET中开发复合控件 ASP.NET中的XML 使用ASP.NET实现网络通信 DataSet和XML深入编程 SQLServer的内置XML支持 深入解析ASP.NET架构 深入了解DataSet 数据集和数据库的同步—DataAdapter的使用 ASP.NET网络打印 ASP.NET文件处理 ASP.NET中的Web服务 ASP.NET多线程编程 ASP.NET中的编码 在ASP.NET中进行消息处理 ASP.NET实例编程 ASP.NET实战编程 ASP.NET项目实战
3	Demo	本书各个章节所讲示例	
4	代码量统计工具	用来统计代码行	CountASP.bat CountCPP.bat CountCS.bat Ff.exe
5	开发规范	软件项目组常用规范	7个常用规范
6	开发模板	软件开发常用质量模板	12个常用模板



# 程序员岗位技能图

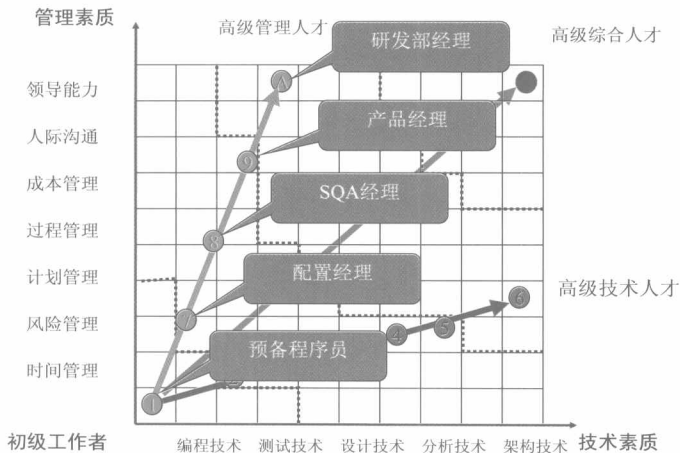


软件人员职业发展路径

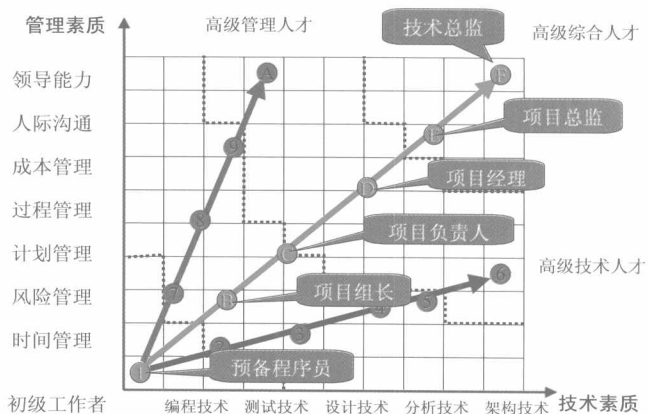


高级技术人才理想成长路径

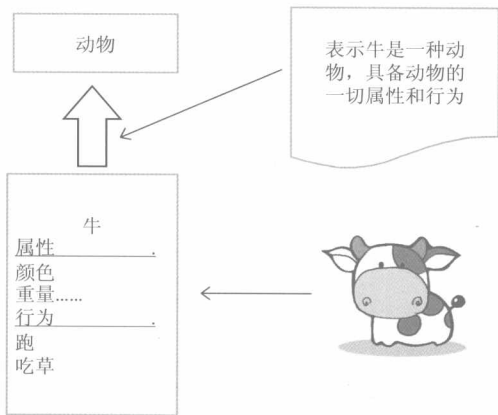




高级管理人才理想成长路径



高级综合人才理想成长路径



用面向对象的方法来认识牛

## 精品图书免费阅读!

### 《SilverLight 魔幻银灯》

苏鹏 著 ISBN 978-7-121-07468-4 2008 年 11 月出版

定价: 49.00 元 网购地址: [www.china-pub.com/129848](http://www.china-pub.com/129848)

#### 内容简介

本书通过翔实的介绍为广大开发者揭开微软 SilverLight 开发的神秘面纱, 还配有大量截图, 介绍了微软的设计工具 Blend 在设计 SilverLight 方面的使用。本书适合进行网页开发的程序员和页面设计人员, 以及对微软 SilverLight 技术有兴趣的 Web 程序设计人员阅读。

#### 本书特色

- 笔者在这本书中尽其所知介绍了 SilverLight 开发的诸多细节, 并使用了一些范例让读者能够快速了解 SilverLight 开发的种种方式和技巧。
- 本书通过笔者的收集整理并结合自身的经验, 让广大读者在开发中少走弯路, 读者通过本书的学习即可快速地了解 SilverLight 开发的全貌。
- 本书尽力介绍了各种开发技术的方式, 并介绍了其内在的运转方式, 相信聪明的读者可以在阅读中举一反三地创造出自己的应用。
- 本书配有一张光盘, 其中包括按章存放的示例程序。

#### 作者介绍

苏鹏, 现任中国网通集团北京市分公司软件开发架构设计师, 常年从事 ASP、ASP.NET 等 Web 开发工作。于 2006 年荣获微软最有价值专家 (MVP) 称号, 2007 年和 2008 年连任。目前从事的工作中开始关注 Web 表示层设计工作。

### 《SilverLight 魔幻银灯》精彩片断试读

#### SilverLight 2.0 中的动画

SilverLight 2.0 中一个非常让人激动的功能就是多媒体的播放和动画效果的展示, 这一章我们就来介绍如何实现多媒体和动画效果。

动画效果可以增强应用程序的图形互动性, 使用动态的背景颜色或者变化的前台样式可以让用户感觉非常 cool。

动画的基础实现方式说起来比较简单, 基本原理就是利用人类的视觉暂留现象, 通过快速的切换画面给人以画面运动起来的效果。人眼在某个视像消失后, 仍可使该物像在视网膜上滞留 0.1~0.4 秒左右。当我们欣赏电影时, 投影机在屏幕上每秒钟匀速播放 24 格静止的画面。一系列静态画面就会因视觉暂留作用感觉到第一格画面还未消失, 第二格画面便出现了, 使影像在观看者的视网膜上暂留, 造成一种连续的视觉印象, 观众便可看到

一组活动的画面，产生逼真的动感，于是屏幕上的影像就好像是连续的。

在实际操作中每秒绘制 24 幅图像其实是相当麻烦的，即使是现在的硬件条件，这种渲染也会耗费大量的 CPU 时间。所以通常情况下，在绘制动画的时候，SilverLight 和 Flash 一样，使用一种叫做定格绘制法的技术来实现。

这种技术起源于电影界，在没有大型计算机渲染技术的时代，绘制一个恐龙大战的场景简直是不可思议的，当然在硬件条件相对落后的时代，不可能使用电脑来实现这个渲染工作。那么这些电影人是如何完成这一艰巨工作的呢？首先，电影人制作一个模型，这个模型是固定的，不会动也不会叫，然后电影人拍一帧照片，把摄影机定格，接着手动去调整一下模型的位置，然后再拍一帧，就这样一帧一帧地完成整个动画，应该说这个过程是漫长而艰苦的。不过万幸的是，今天我们使用的 SilverLight 技术可以代替完成这个工作。下面就开始介绍一下 SilverLight 中的动画技术。

在 SilverLight 中，动画对象是通过接收其各种属性的改变来实现的，如同我们刚刚介绍的动画原理，在 SilverLight 中，设定不同动画对象的属性状态，然后由计算机代替你完成拍照片、按暂停、调动作的循环操作，并体贴地把它们自动串接在一起，让用户最后看到的是一幅流畅的画面。

### 渐变动画

需要注意的一点是，在 SilverLight 中，对象接收作为动画效果修改的属性都必须使用 Double、Color、Point 作为值的属性，也就是说，其中的 Double 是使用各种数据，而 Color 则是用来显示颜色对象，Point 则多半用于位置信息。

下面我们来介绍一个 UI 元素如何实现动画，这个效果十分简单，就是实现一个矩形对象的淡入淡出效果。

要想实现矩形对象的淡入淡出效果，需要涉及 3 个对象，首先是矩形对象的属性，所有的 UI 元素都有一个 Opacity 属性，这个属性用于设定元素的透明度，取值是 0 到 1 之间的小数，其中 1 是完全不透明，0 是完全透明（关于属性更多的相关信息请参见图形绘制章节）。在这里还要提一下，如果要隐藏一个 UI 对象，最好的办法是设置 Visible 属性而不是把透明度设置为 0，因为后者的性能会低于前者。好了，这里要设置的属性就是 Opacity 属性，它是一个 Double 类型的属性，这些信息确定了之后，要创建第二个对象，即 DoubleAnimation，DoubleAnimation 指定一个 Double 类型的属性，使其在指定的时间内由起点值到达终点值，从而形成动画效果。应用它来实现动画效果，只要简单地指定几个参数值就可以了。这个对象是创建数据值类型渐变动画效果最重要的对象，在指定的 Duration 内使用线性内插对两个目标值之间的 Double 属性值进行动画处理。

动画在一段时间内更新属性值，动画的效果可能十分微小，比如将 Shape 左右移动几个像素；也可能十分显著，比如将对象放大到其原始大小的 200 倍，同时对其进行旋转并更改其颜色。若要在 Windows Presentation Foundation (WPF) 中创建动画，请将动画与对象的属性值关联起来。

## 数据绑定控件

在 SilverLight 2.0 的诸多控件当中有一种控件非常常用，这就是数据绑定控件。

使用 DataGrid 可以方便灵活地显示数据，其单元格格式支持各种显示数据的方法，比如文本或者选择框，甚至还支持模板定义。DataGrid 控件支持一般的表格属性，比如支持单元格的背景色或者像一些表格的预置模板一样支持分行变色技术（这个技术被 ASP.NET 2.0 里的 GridView 发挥到了极致），此外还有隐藏表格、隐藏表线等多种表格常用技术。而且这个控件在设计的过程中希望完全支持用户的各种数据显示模式，所以预先内建了若干表格样式，用户可以方便地调用。下面就介绍如何在应用当中引入表格控件。

对于 DataGrid，最容易使用的方法就是通过指定它的 ItemsSource 属性来实现数据的显示。这个属性专门用于显示指定的数据，一旦指定了控件的 ItemsSource，控件就能自动生成各种表格数据用于显示。这里需要说明的一点是，一旦指定了 ItemsSource，数据的显示会自动把元数据中的数据类型转义成 String 的内容，而且这些文本在默认情况下是只读的。

如果希望使用自己定义的表格显示模式，则可以通过设置 AutoGenerateColumns 属性为 false 来限制自动生成表格内容，当然这也意味着你不得不通过使用 AutoGeneratingColumn 事件来自行处理所有的显示内容。如果希望对表格的显示内容进行排序，可以通过设置 DisplayIndex 属性来实现。

无论表格的内容是自动生成的还是编码指定的，它们都可以通过使用 DataGrid.Columns 对象来进行编程访问，可以调用它的 Add、Insert、Removeat 方法进行数据的添加、插入和删除。

## 刷子对象

在 SilverLight 中，色彩缤纷的颜色被封装成了一个对象，这个对象被用于绘制所有图形的边框、背景色以及其他填充效果。这个对象就是下面要介绍的刷子对象。

在前面的章节中我们已经见过不少刷子对象的例子，一般如同下面的样式：

```
<Ellipse Width="300" Height="100">  
<Ellipse.Fill>  
<SolidColorBrush Color="Red"/>  
</Ellipse.Fill>  
</Ellipse>
```

这种刷子对象定义了单一颜色的固体刷子，其实在 SilverLight 的刷子对象中，除了这个固体刷子之外，还有其他用于不同用途的刷子。在这一节里，我们会介绍 5 种常用的刷子对象，它们是 3 种基本图形填充刷子：

```
SolidColorBrush  
LinearGradientBrush  
RadialGradientBrush
```

以及 2 种多媒体刷子：

```
ImageBrush  
VideoBrush
```

## 精品图书免费阅读!

倾情奉献数载开发经验, 丝丝入扣讲述 ASP.NET 控件开发底层原理...

### 《庖丁解牛——纵向切入 ASP.NET 3.5 控件和组件开发技术》

电子工业出版社 【书号】9787121079795 2009年3月 出版 含光盘2张

网络购买地址: <http://www.china-pub.com/192102>

内容介绍: 本书主要介绍 ASP.NET 的控件开发, 书中通过 70 多个例子讲解了 ASP.NET 控件开发技术的各个方面, 而且剖析了很多控件中系统基类源代码, 读者从这些系统源代码可以体会设计模式思想。如果扎实地掌握了 ASP.NET 控件的运行机制, 开发一个页面级的 ASP.NET 应用程序会变得非常简单。本书宗旨就是让开发人员真正理解 ASP.NET 技术, 帮助开发人员提高 ASP.NET 开发的技术水平。学完本书后您不仅能够掌握控件开发各个方面的技术, 而且深晓 ASP.NET 的工作原理。对大部分使用 ASP.NET 技术开发两年左右的开发人员来说, 本书 75% 以上的内容可能没有接触过, 本书将通过笔者精心准备的一些生动的例子来阐述这些内容。

本书适合从事 ASP.NET 技术多年想往深度理解 ASP.NET 工作机制的开发人员阅读, 对于刚刚入门的初级开发人员, 本书的前面几章也介绍了几个入门性的应用例子, 本书还可以作为在校学生的就业预备教材。针对读者安装的开发环境不同, 笔者分别为 VS 2005 和 VS 2008 建立了解决方案, 即无论您是使用 VS 2005 还是 VS 2008 都可以成功打开本书随书光盘的代码, 只是使用 VS 2005 打开的解决方案不包含 .NET 3.5 对应章节内容的示例代码。

### 《庖丁解牛——纵向切入 ASP.NET 3.5 控件和组件开发技术》精彩片断试读

#### 客户端回传事件接口 IPostBackEventHandler

要使控件捕获回发事件, 控件必须实现 System.Web.UI.IPostBackEventHandler 接口。此接口约定允许控件在服务器上引发事件来响应来自客户端的回发。IPostBackEventHandler 接口包含一个方法。

```
/// <summary>
/// 获得本书更多内容, 请看:
/// http://blog.csdn.net/ChengKing/archive/2008/08/18/2792440.aspx
/// </summary>
public interface IPostBackEventHandler
{
    void RaisePostBackEvent(string eventArgument);
}
```

参数 eventArgument 表示要传递到事件处理程序的可选事件参数, 一般通过此参数可以确定不同的引发事件源, 进而作不同的逻辑处理。在本章最后有个例子说明 eventArgument 参数用法。回发后, 页框架就会搜索发送的内容, 并确定发送的名称是否与实现 IPostBackEventHandler 的服务器控件的 UniqueID 对应。如果对应, 页框架就会在该控件上调用 RaisePostBackEvent 方法 (在引发更改事件后)。

以下代码片段显示了在服务器上引发 Click 事件的 RaisePostBackEvent 实现:

```
/// <summary>
/// 获得本书更多内容, 请看:
/// http://blog.csdn.net/ChengKing/archive/2008/08/18/2792440.aspx
/// </summary>
public void RaisePostBackEvent(String eventArgument)
{
    ButtonEventArgs e = new ButtonEventArgs(eventArgument);
    OnClick(this, e);
}
```

在该方法中主要完成调用 OnClick(e)事件功能, 其中包含两个参数: 第一个参数为当前控件本身(即一般事件体中 sender, 类型一般为 object); 第二个参数 e 为 ButtonEventArgs 参数类型对象, 是继承于 System.EventArgs 类实现的参数类, 在该类中可以定义与代码逻辑相关的任意属性, 作为事件体的参数。

最后, RaisePostBackEvent 需要被客户端引发才能够执行, 下面是一段能够引发服务端事件的代码:

```
/// <summary>
/// 获得本书更多内容, 请看:
/// http://blog.csdn.net/ChengKing/archive/2008/08/18/2792440.aspx
/// </summary>
protected override void Render(HtmlTextWriter output)
{
    output.Write("<INPUT TYPE=submit name=" + this.UniqueID +
        " Value='Click Me' />");
}
```

这段代码输出一个 HTML 的 button 标签, 并设置为提交类型。非常重要一点是, 不要忘记设置其 name 属性, 因为当回发后, 页框架就会搜索发送的内容, 并确定发送的名称是否与实现 IPostBackEventHandler 的服务器控件的 UniqueID 对应。如果对应, 页框架就会在该控件上调用 RaisePostBackEvent 方法; 反之, 如果没有设置按钮的 name 值为 UniqueID 属性值, 当单击按钮时页框架就不会引发该控件的 RaisePostBackEvent 方法, 因为只有名称为 UniqueID (服务器控件服务端 ID) 的按钮才会被注册为具有 IPostBackEventHandler 接口功能的控件。

下面通过一个简单的完整例子, 了解事件回发处理机制, 代码如下:

```
/// <summary>
/// 获得本书更多内容, 请看:
/// http://blog.csdn.net/ChengKing/archive/2008/08/18/2792440.aspx
/// </summary>
[DefaultEvent("Click")]
[ToolboxData("<{0}:PostBackEventControl
runat=server></{0}:PostBackEventControl>")]
public class PostBackEventControl : Control, IPostBackEventHandler
{
    public event EventHandler Click;
    protected virtual void OnClick(EventArgs e)
    {
        if (Click != null)
```

```

        {
            Click(this, e);
        }
    }
    public void RaisePostBackEvent(string eventArgument)
    {
        OnClick(EventArgs.Empty);
    }
    protected override void Render(HtmlTextWriter output)
    {
        output.Write("<INPUT TYPE=submit name="+this.UniqueID+"Value='单击我'
        />");
    }
}

```

该控件内容比较简单，仅输出一个提交类型的按钮，由于这个按钮类型为 `submit`，所以当单击按钮时，其本身已经可以提交事件到服务器，但仅仅这样主控件还不能够捕捉到该按钮事件。控件能够捕捉处理该事件需要具备两个条件：第一，主控件继承了 `IPostBackEventHandler` 接口以及实现了 `RaisePostBackEvent` 方法；第二，必须有 `name` 值为 `UniqueID` 的客户端标签，页框架只认识控件的 `name` 属性。只有这两个条件同时具备才能够使控件具备捕捉并处理事件的机会。

语句 `[DefaultEvent("Click")]` 的功能是定义 `Click` 事件为默认事件。通常把最常用的一个事件定义为默认事件，如果定义了默认事件，在设计器中双击控件时系统会自动从源代码视图 (\*.aspx) 切换到后面代码 (\*.cs) 页面，并可以自动注册默认事件；否则，仅切换到后台代码，但不注册任何事件。

类代码中最上面的一句：

```
public event EventHandler Click;
```

这条语句定义了一个委托事件。`EventHandler` 是一个预定义的委托，专用于表示不生成数据的事件的事件处理程序方法。如果事件生成数据，则必须提供自定义事件数据类型，并且必须要么创建一个委托（其中第二个参数的类型为自定义类型），要么使用泛型 `EventHandler` 委托类并用自定义类型替代泛型类型参数。

在 5.1.1 节讲过事件的意义及其完成的功能，即在一个类中执行另一个功能类的功能，要想执行另一个类中的某个功能方法，必定要在当前类中保留另一个方法的引用，委托和事件就是为实现此功能而出现的。委托类似于 C++ 中的函数指针，它一般指向一个方法。这样，当调用本类中的委托就相当于调用了另一个类中的方法。

若要将事件与处理事件的方法关联，请向事件添加委托的实例和委托事件实例，如：

```
this.Button1.Click += new EventHandler(Button1_Click);
```

除非移除了该委托，否则每当发生该事件时就调用事件处理程序。

另外，委托的应用比较广泛，还有更加复杂的应用，比如定义一个委托可以同时指向多个事件，即一个委托指定一个事件列表，当调用委托时可以执行一系列的事件。这里限于篇幅就不多讲了，更多关于事件委托概念请查看微软官方文档，那里面已经讲得很详细了。

# 目 录

## 第 1 篇 知识篇

### 第 1 章 恭喜你，成为软件程序员.....2

作为一名程序员要掌握哪些专业技能？应该具备哪些素质技能？个人的职业发展道路应该如何设计？这是每个程序员都要面对的问题。在本章中，我们将就这几个方面给大家做介绍。

#### 1.1 软件程序员专业技能.....2

##### 1.1.1 专业技能检核表.....2

##### 1.1.2 软件程序员岗位技能图.....4

#### 1.2 软件程序员素质技能.....4

##### 1.2.1 软件程序员的素质技能.....4

##### 1.2.2 高级软件程序员的素质技能.....5

#### 1.3 软件程序员职业发展图.....6

##### 1.3.1 软件企业的人才类型和素质结构关系.....6

##### 1.3.2 高级技术人才理想成长路径.....7

##### 1.3.3 高级管理人才理想成长路径.....7

##### 1.3.4 高级综合人才理想成长路径.....8

### 第 2 章 软件开发流程.....9

什么是产品？什么是项目？它们的开发流程有何不同？如何基于项目进行产品开发？如何基于产品做项目开发？软件开发中，质量和效率如何平衡？这些都是每个软件开发人员面临的问题。在本章中，我们将共同探讨这些话题。

#### 2.1 项目还是产品——程序员心中的痛.....9

#### 2.2 软件开发流程.....10



2.2.1	项目开发流程	10
2.2.2	产品开发流程	11
2.3	标准还是效率——痛苦的抉择	11
2.3.1	质量/过程标准的来源	11
2.3.2	标准能做到什么和不能做到什么	12
2.3.3	质量标准和过程控制有什么用	12
2.3.4	软件开发过程中如何应用这些标准	13
<b>第3章</b>	<b>可行性研究</b>	<b>14</b>

可行性分析是要决定“做还是不做”。显然，“做还是不做”比“如何做”更重要。如果可行性分析不够充分、完整，就有可能得出错误的结论，从而导致资金的浪费、无谓的投入。因此可行性分析至关重要。在本章中我们将学习如何从市场、政策、技术、经济，以及SWOT方面做可行性分析。

3.1	可行性分析的要素	14
3.2	市场可行性分析	15
3.3	政策可行性分析	15
3.4	技术可行性分析	15
3.5	成本—收益分析	16
3.6	SWOT分析	16
3.6.1	竞争优势分析	17
3.6.2	竞争劣势分析	17
3.6.3	外部机会分析	18
3.6.4	外部威胁分析	18
3.6.5	SWOT核心问题	19
<b>第4章</b>	<b>需求分析</b>	<b>20</b>

需求分析是要决定“做什么，不做什么”。在用户导向的今天，用户需求分析变得越来越重要。在需求分析阶段，我们需要完成哪些工作？常用的需求分析工具有哪些？在本章中，这些问题会一一得到解答。

4.1	需求分析的重要性	20
4.2	需求分析为什么困难	20
4.3	需求分析基本任务	21
4.4	需求分析的结果	22