

程序员指南丛书



ASP.NET 编程基础

- ASP.NET 是继 ASP 3.0 之后推出的全新动态网站设计技术与程序框架，将为动态网站的设计带来全新的技术和概念。
- ASP.NET 是可编译执行的 Web 服务器端开发工具，采用 C# 语言作为默认开发工具。
- 本书以原理和组件相结合的方式，并辅以大量实例，全面深入地介绍了 ASP.NET 技术及编程方法与技巧。

肖建 等编著



清华大学出版社

<http://www.tup.tsinghua.edu.cn>

ASP.NET 编程基础

肖建等编著

清华大学出版社

(京)新登字 158 号

内 容 简 介

本书以原理和组件相结合的方式,并辅以大量实例,全面深入地介绍了 ASP.NET 技术及编程方法与技巧。首先介绍了 ASP.NET 特性以及 HTML 语法和 C#语言基础;然后分别介绍了 ASP.NET 最基本的组件、数据验证机制及其组件、数据绑定组件、数据操作组件以及其他重要组件;最后综合前几章的知识,以一个 E-Commerce 购物系统的构建过程为例,向读者演示了 ASP.NET 的开发过程。

本书图文并茂、结构清晰、通俗易懂,适合于 ASP.NET 初学者阅读参考,也是很好的培训用书。

版权所有,翻印必究。

本书封面贴有清华大学出版社激光防伪标签,无标签者不得销售。

图书在版编目(CIP)数据

ASP.NET 编程基础/肖建等编著. —北京:清华大学出版社, 2002. 5

(程序员指南丛书)

ISBN 7-302-05395-2

I. A... II. 肖... III. 主页制作—程序设计 IV. TP393.092

中国版本图书馆 CIP 数据核字(2002)第 019509 号

出 版 者: 清华大学出版社(北京清华大学学研大厦, 邮编: 100084)

<http://www.tup.tsinghua.edu.cn>

责任编辑: 胡先福

印 刷 者: 北京市清华园胶印厂

发 行 者: 新华书店总店北京发行所

开 本: 787 × 1092 1/16 印张: 19.25 字数: 463 千字

版 次: 2002 年 5 月第 1 版 2002 年 5 月第 1 次印刷

书 号: ISBN 7-302-05395-2/TP · 3174

印 数: 0001 ~ 5000

定 价: 27.00 元

前 言

ASP. NET 是继 Microsoft 公司的 ASP 3.0 之后推出的全新动态网站设计技术与程序框架。它已经不再仅仅是 ASP 的一个升级版本了,而是作为 Microsoft 公司的 NGWS(Next Generation Web Services)的一个重要组成部分,将为动态网站的设计带来全新的技术和概念。它带给网站开发人员的是性能的提升和开发效率的提高。

ASP. NET 是可编译执行的 Web 服务器端开发工具。与 JSP 一样,ASP. NET 也是预先编译为一个类文件,当用户访问 ASP. NET 文件时,直接执行这个类文件而不是原先的类文件。在编译前,ASP. NET 源文件被翻译成一个 C#(读作 C Sharp)或 Visual Basic 的源代码文件。

C#也是一种全新的编程语言,ASP. NET 采用该语言作为默认开发工具。因此本书在介绍完网页基本语言 HTML 之后,紧接着向读者介绍了 C#语言的大部分基础知识。本书的所有实例都以 C#作为开发语言,并且大多数采用 Microsoft Visual Studio. NET 作为开发环境。

本书采用原理与组件应用相结合的方式介绍,并辅以大量开发实例,力图比较全面深入地介绍 ASP. NET 技术。

全书内容安排如下:

第 1 章概述了 ASP. NET 的特性,并采用对比的方式比较了 ASP. NET 所支持的 3 种语言的语法格式,最后介绍了本书所使用开发环境的构建。

第 2、3 章分别介绍了 HTML 语法和 C#语言基础,这两章是全书的基础,对这两种语言不太熟悉的读者最好能够详细阅读。

第 4~8 章是本书的重点:第 4 章介绍了 ASP. NET 几种最基本的组件;第 5 章介绍了 ASP. NET 的数据验证机制及其组件;第 6 章介绍了数据绑定组件;第 7 章介绍了数据操作组件;第 8 章则介绍了前面几章没有包含的其他重要组件。

第 9 章全面综合了本书前几章的知识,以一个 E-Commerce 购物系统的构建过程为例,向读者演示了 ASP. NET 的开发过程。

由于 C#和 ASP. NET 都是新的开发工具,许多技术上的问题、观点等都还处于争论阶段,加上作者水平有限,书中难免会有不成熟之处,恳请广大读者批评指正。

对本书的写作、整理、出版等做出帮助的有胡大惟、葛文全、何琳、张思明、吴迪、曹立勋、吴清萍、安浦、方克志、包利群、李国文、汪浩、周银昆、李高云、张明翔等,在此表示感谢。

作 者
2002 年 3 月

目 录

第 1 章 ASP.NET 入门	1
1.1 ASP.NET 概述	1
1.1.1 ASP.NET 的优越性	1
1.1.2 ASP.NET 动态网站开发技术的特点	3
1.2 用 VB、C#和 JScript 实现 ASP.NET	4
1.2.1 变量	4
1.2.2 语句	5
1.2.3 注释语句	5
1.2.4 获得 URL 传递的变量	5
1.2.5 声明属性	6
1.2.6 数组	6
1.2.7 变量初始化	7
1.2.8 选择语句	8
1.2.9 分支语句	8
1.2.10 For 循环语句	9
1.2.11 While 循环语句	10
1.2.12 字符串连接	10
1.2.13 声明事件	11
1.2.14 声明对象	11
1.2.15 数据类型转换	11
1.2.16 名称空间的声明	12
1.2.17 类的定义和继承	13
1.3 搭建 ASP.NET 平台	14
1.4 本章小结	18
第 2 章 HTML 基本语法	19
2.1 认识 HTML	19
2.1.1 HTML 标记	19
2.1.2 组件的概念	20
2.1.3 HTML 的阶层概念	20
2.1.4 超级链接及其他	21
2.2 页面标记	21

2.2.1	HTML 文件结构	21
2.2.2	语言字符集的信息	22
2.2.3	背景色彩和文字色彩	22
2.2.4	页面空白	22
2.2.5	链接	23
2.2.6	开一个新的窗口	24
2.2.7	标尺线	25
2.3	字体标记	26
2.3.1	标题字体	26
2.3.2	正文字体	27
2.3.3	物理字体	28
2.3.4	逻辑字体	29
2.3.5	字体标记的组合使用	30
2.3.6	字体颜色	31
2.3.7	客户端字体	31
2.4	窗体标记	33
2.4.1	基本语法	33
2.4.2	文字输入和密码输入	33
2.4.3	复选框和单选按钮	35
2.4.4	图像坐标	36
2.4.5	列表框	38
2.4.6	文本框	40
2.5	多窗口页面	41
2.5.1	基本语法	41
2.5.2	各窗口的尺寸设置	42
2.5.3	窗口的外观	43
2.6	本章小节	48
第3章	C#语言基础	49
3.1	NGWS Runtime 技术基础	49
3.1.1	NGWS Runtime 技术简介	49
3.1.2	虚拟对象系统	53
3.2	C#的变量和类型	53
3.2.1	数值类型	54
3.2.2	引用类型	59
3.2.3	加框和消框	63
3.3	类与结构	63
3.3.1	类声明	64

3.3.2	构造函数与析构函数	65
3.3.3	方法	66
3.3.4	虚方法	72
3.3.5	重载方法	75
3.3.6	方法屏蔽	79
3.3.7	类的属性	81
3.3.8	结构	83
3.4	控制语句	85
3.4.1	选择语句	85
3.4.2	循环语句	88
3.5	异常处理	91
3.5.1	checked 和 unchecked 语句	92
3.5.2	异常处理语句	93
3.6	本章小结	96
第 4 章	ASP.NET 服务器端组件	97
4.1	ASP.NET Web 窗体	97
4.1.1	简单的 ASP.NET Web 窗体网页	97
4.1.2	ASP.NET 服务器端组件概述	100
4.2	Web 窗体语法	103
4.2.1	ASP.NET Web 窗体语法元素	103
4.2.2	ASP.NET 服务器端组件语法	105
4.2.3	ASP.NET HTML 服务器端组件语法	106
4.2.4	数据绑定语法	106
4.2.5	服务器端注释语法	109
4.2.6	服务器端包括语法	110
4.3	ASP.NET 服务器端组件起步	111
4.3.1	服务器端组件的声明和属性设定	111
4.3.2	响应组件动作	114
4.3.3	用 Visual Studio.NET 实现 ASP.NET Web 程序开发	116
4.4	基本的 ASP.NET 服务器端组件	126
4.4.1	Label 组件	126
4.4.2	TextBox 组件	127
4.4.3	Button、LinkButton 和 ImageButton 组件	127
4.4.4	CheckBox 和 CheckBoxList 组件	128
4.4.5	RadioButton 和 RadioButtonList 组件	130
4.4.6	Image 组件	130
4.4.7	HyperLink 组件	132

4.4.8	Table、TableCell 和 TableRow 组件	136
4.4.9	DropDownList 组件	138
4.5	本章小结	142
第5章	数据验证	143
5.1	数据验证机制	143
5.1.1	概述	143
5.1.2	实例演示 ASP. NET 的数据验证机制	144
5.2	数据验证组件	147
5.2.1	RequiredFieldValidator 组件	147
5.2.2	CustomValidator 组件	152
5.2.3	ValidationSummary 组件	154
5.2.4	CompareValidator 组件	157
5.2.5	RangeValidator 组件	162
5.2.6	RegularExpressionValidation 组件	164
5.3	本章小结	170
第6章	数据绑定	171
6.1	数据绑定基础	171
6.2	基于变量的数据绑定	172
6.3	基于集合的绑定	177
6.4	基于表达式的绑定	179
6.5	基于 DataBinder. Eval 方法的数据绑定	180
6.6	列表绑定组件	183
6.6.1	列表绑定组件的通用属性	183
6.6.2	使用列表绑定组件	185
6.7	本章小结	190
第7章	ADO. NET 与数据操作	191
7.1	ADO. NET 概述	191
7.1.1	目前的 ADO. NET	191
7.1.2	连接一个数据库	192
7.1.3	数据库查询	192
7.1.4	数据的显示	193
7.2	数据库操作	195
7.2.1	DataGrid 组件	195
7.2.2	使用参数查询数据	200
7.2.3	插入数据	202

7.2.4	更新数据	207
7.2.5	数据验证	211
7.2.6	主/细 (Master/Detail) 关系	218
7.2.7	使用存储过程	223
7.3	本章小结	226
第 8 章	其他 ASP.NET 组件	227
8.1	AdRotator 组件	227
8.1.1	XML 文件	227
8.1.2	使用 AdRotator 组件	234
8.2	Calendar 组件	236
8.3	Panel 组件	243
8.4	Repeater 组件	255
8.5	Timer 组件	268
8.6	本章小结	269
第 9 章	构建 E-Commerce 购物系统	270
9.1	整体设计	270
9.2	实现购物系统	273
9.3	本章小结	296

第 1 章 ASP.NET 入门

ASP.NET 是由 Microsoft 公司提出的一个程序框架，它与以前的网页开发技术相比有了很大的进步。本章将首先对 ASP.NET 进行概述，让读者对这种新的编程体系有一定的认识，通过对几种 ASP.NET 能够支持的语言进行比较。

1.1 ASP.NET 概述

动态网站开发技术发展到现在，可以说经历了具有历史性的 4 个阶段：

- CGI 阶段。在这个阶段里，人们使用 C、Perl、VB 开发动态网站，动态网站的开发技术在当时仅仅只有少数程序设计人员掌握。
- -SAPI 阶段。这个阶段的代表是 NSAPI 和 ISAPI，从开发者的角度讲，这种开发方式并没有带来开发上的方便。
- 脚本语言阶段。这个阶段涌现出许多杰出的脚本语言：ASP、PHP、嵌入式 Perl 和 JSP。脚本语言的出现大大简化了动态网站开发的难度，特别是 ASP 和 PHP，学习简单、功能强大，成为许多网站开发者的首选。
- 组件技术阶段。ASP.NET 和 J2EE 技术是这个时代的代表，本书就是以 ASP.NET 为基准，对这个新一代的开发技术进行详细的讲解。

1.1.1 ASP.NET 的优越性

ASP.NET 是编译执行的 Web 服务器端开发工具，和 JSP 一样，ASP.NET 也是预先编译为一个类文件，当用户访问 ASP.NET 文件的时候，直接执行这个类文件而不是原先的源文件，在编译前可能将 ASP.NET 源文件翻译成一个 C#(读作 C Sharp)或 VB 的源代码文件。

ASP.NET 是继 Microsoft 公司的 ASP 3.0 之后推出的全新的动态网站设计技术与程序框架，它已经不再是 ASP 的一个升级版本了，它作为 Microsoft 公司的 NGWS(Next Generation Web Services)的一个重要组成部分，将为动态网站的设计带来全新的技术和概念，它带给网站开发人员的是性能的提升和开发效率的提高。

ASP.NET 与以前的网页开发技术相比有了很大的进步，表现在以下几个方面：

- 更强大的性能。ASP.NET 是运行在服务器端的通用语言运行环境，和以前的解释运行方式不同，ASP.NET 会通过即时编译来提高性能。除此之外，ASP.NET 使用了提前绑定、JIT 编译(Just-In-Time Compilers，ASP.NET 称之为 JITers，而 Java 则称之为 JIT)、本地优化、缓存(Cache)等一系列技术来达到提高运行效率的目的。

- 更强大的开发工具支持。虽然和 ASP 一样，只要服务器操作和服务器软件（ASP.NET 要求至少 Windows 2000 Server 和 IIS 5.0）支持，就可以运行脚本，而且开发 ASP.NET 的页面或者应用程序也不需要专门的开发工具，但是如果想要快速、高效地创建应用，就必须掌握至少一种开发工具。由于 ASP.NET 和 Visual Studio.NET 等良好地集成在一起，因此可以通过专门的窗口和工具来进行高效的所见即所得的应用开发。
- 良好的适应性。ASP.NET 是基于通用语言的运行环境（Common Language Runtime）的，所有一般语言所具有类库、消息和数据连接都能方便地连接到网络上。ASP.NET 同时也具有语言无关性，所以用户可以选择自己所需要的开发语言（如 C#、VB、JScript 等）。
- 简易性。ASP.NET 使得常用的网络开发变得相当方便，从简单的提交窗体和客户端验证的制作到设置和发布整个网站，都可以轻松完成。
- 可管理性。ASP.NET 的管理是基于“无本地管理”的思想，使用纯文本的设置方式，只要设置相应的文本文件就可以完成一系列的站点设置。另外，如果需要发布网站，不必在服务器上使用一些特殊的软件进行设置，只要简单地将制作好的文件复制到服务器上即可。
- 可恢复性和有效性。ASP.NET 在设计时遵循了可扩展性的思想，它对多进程环境下的应用程序进行优化，进程可以动态地被 ASP.NET 调用和跟踪。如果一个进程失败，其他的进程还可以照常运行，从而保证应用程序不会中断。
- 可定制性和可扩展性。ASP.NET 提供了可定制和重用的体系结构，用户可以在一定的层次上修改 ASP.NET 提供的程序段，甚至可以重新编写 ASP.NET 的组件，从而实现根据不同需要定制的目的。
- 安全可靠。ASP.NET 基于 Windows 验证机制，它是为每个应用进行配置的方法，确保用户的应用是安全可靠的。
- 语言支持。ASP.NET 即时地对 C#、VB 和 JScript 三种语言提供了支持。

但是，ASP.NET 与 ASP 的最大区别，是在编程思想上的转变，而不仅仅在于上面所述各种功能的增强。ASP 使用 VBScript 或 JavaScript 这样的脚本语言混合 HTML 来编程，而脚本语言属于弱类型、面向结构的编程语言，而不是面向对象的，这就容易产生以下的问题：

- 代码逻辑混乱。由于 ASP 使脚本语言混合 HTML 编程，因此很难看清楚代码的逻辑关系，并且随着程序复杂性的增加，代码的管理十分困难，甚至超出一个程序员所能达到的管理能力，从而产生许多问题。
- 代码的可重用性差。因为 ASP 采用的是面向结构的编程方式，并且混合 HTML，所以页面原型的一点点修改，就可能导致整个程序的全面修改。
- 弱类型造成潜在的出错可能。尽管弱数据类型的编程语言使用起来更方便一些，但是相对于它所造成的出错几率是远远得不偿失的。

ASP.NET 摆脱了从前 ASP 使用脚本语言来编程的缺点，理论上可以使用任何编程语言，包括 C++、Visual Basic 和 JavaScript，等等，但是最合适的编程语言还是 Microsoft

Studio 为 ASP.NET Framework(框架)专门推出的 C#。尽管 Microsoft 公司称 C#内核中更多的像 Visual C++，但实际上它和 Java 更像一些。C#是面向对象的编程语言，而不是一种脚本，所以它具有面向对象编程语言的一切特性，比如封装性、继承性和多态性等。这就解决了刚才所谈到的 ASP 所具有的那些弱点：封装性使得代码逻辑清晰、易于管理，并且应用到 ASP.NET 上就可以使业务逻辑和 HTML 页面分离，这样无论页面原型如何改变，业务逻辑代码都不必作任何改动；继承性和多态性使得代码的可重用性大大提高，程序员可以通过继承已有的对象最大限度地减少必要的工作，并且 C#和 C++、Java 一样提供了完善的调试/纠错体系。

1.1.2 ASP.NET 动态网站开发技术的特点

ASP.NET 作为一种全新的动态网站开发技术，它具有自己的开发特点：

- 大量使用组件技术。
- 将 Web 浏览器和 Web 服务器之间的网络通信完全地包装起来。

组件技术在 ASP.NET 中并不能算是一个新的概念，在 ASP 中就利用组件技术实现了诸如数据库连接、文件处理等一系列的功能，但是 ASP 中核心的技术还是 JavaScript 和 VBScript 这样的脚本语言技术。到了 ASP.NET，组件技术则成为了整个开发技术的核心，而作为程序语言出现的 C#和 VB，则成为了为组件服务的次一级技术。实际上，在 ASP.NET 中，使用哪一种开发语言根本就不重要。

更重要的是，在动态网站开发的 CGI 阶段就开始的“利用程序输出 HTML 代码”的动态网站开发技术到 ASP.NET 完全发生了改变。在 ASP.NET 中，动态网站开发技术已经变为利用组件搭建“机遇 Web 的应用程序”了。开发人员现在已经可以基本上不考虑如何使用 HTML 代码去实现自己需要的效果，而是考虑如何利用已有的 ASP.NET 组件去实现。当然，那些最基本的 HTML 语句还是需要的，我们在下一章就会进行简要的介绍。

举例来说，假设我们想要构建一个网络书屋的购物系统，那么就需要下面这些组件：

- ADO + 中的数据库连接组件
- DataSet 数据集组件
- DataGrid 数据显示组件
- 其他一些数据验证组件

有 VB 或者 Delphi 基础的读者应该可以很容易理解上面这些组件的作用。当然，即使读者现在还不能理解这些组件的具体作用和使用方法，我们仍然可以解释整个 ASP.NET 页面的工作机制：

- 数据库连接组件用于处理数据库连接；
- 数据集组件相当于数据库的一个抽象，可以通过数据库连接组件完成数据查询、删除、更新、添加等操作；
- 数据显示组件用于显示数据和更新数据，当然这一切都是通过数据集组件完成。

当客户端访问这个 ASP.NET 页面的时候，首先得到的是由显示书本列表的 DataGrid 组件通过 DataSet 组件经过数据库连接组件从数据库中读出的书本列表，当用户要求购买

某一本书的时候，就通过作为用户购物车的 DataGrid 组件通过数据集组件经过数据库连接组件向数据库中添加一条购物信息。

从上面的解析读者应该可以发现，就这么几个简单的 ASP.NET 服务器端组件，便完成了一个相当实用的网络书屋，从中也可以看出组件技术的威力所在。

在 ASP.NET 中，开发人员根本就不关心服务器和客户端之间的数据传递，就像开发普通的应用程序一样，可以直接将某个组件的事件和一个自定义的方法关联起来，尽管这个事件是从客户端产生的，但是开发人员却可以完全不考虑这些，这正是组件的封装性使然。

1.2 用 VB、C#和 JScript 实现 ASP.NET

ASP.NET 支持多种语言，它的默认语言是 Visual Basic(VB)而不是 VBScript，这意味着可以摆脱 VBScript 的语言限制，代码将是编译后运行的，而不是原来的解释执行。

ASP.NET 可以全部用成熟的 VB 语言来书写。不过，VB 有了一些改动，许多变动是有实际意义的，也是有明显进步的。但是，由于 C#为 Microsoft 所强力推荐，预言在企业计算领域，C#将会变成为用于编写“下一代窗口服务”(Next Generation Windows Services, NGWS)应用的主要语言，并大力推广这项技术。因此，C#虽然是新生的语言，但是一定会越来越优秀的，所以本书将主要采用 C#为实现 ASP.NET 的语言，一部分 ASP.NET 应用我们也同时给出用 VB 实现的代码，给读者提供更多的信息。

下面我们通过语法规则来比较这 3 种语言的不同点。

1.2.1 变量

C#的语法：

```
int i;
String s;
String s1,s2;
Object o;
Object obj = new Object();
Public String name;
```

VB 的语法：

```
Dim i As Integer
Dim s As String
Dim s1,s2 As String
Dim o 'Implicity Object
Dim obj As New Object()
Public name As String
```

JScript 的语法：

```
var i:int;  
var s:String;  
var s1:String,s2:String;  
var o;  
var obj:Object = new Object ();  
var name:String;
```

1.2.2 语句

C#的语法:

```
Response.Write("Hello World!");
```

VB 的语法:

```
Response.Write("Hello World!")
```

JScript 的语法:

```
Response.Write("Hello World!");
```

1.2.3 注释语句

C#的语法:

```
//This is only a comment
```

或者

```
/* This is only  
a comment */
```

VB 的语法:

```
'This is only a comment
```

JScript 的语法:

```
//This is only a comment
```

1.2.4 获得 URL 传递的变量

C#的语法:

```
String s = Request.QueryString["Name"];  
String value = Request.Cookies["key"];
```

VB 的语法:

```
Dim s,value As String  
s = Request.QueryString["Name"]
```

```
value = Request.Cookies["key"].Value
```

JScript 的语法:

```
var s:String = Request.QueryString["Name "];  
var value:String = Request.Cookies["key"];
```

1.2.5 声明属性

C#的语法:

```
Public String name(  
    get {  
        ...  
        return ...;  
    }  
    set {  
        ... = value;  
    }  
}
```

VB 的语法:

```
Public Property name As String  
Get  
    ...  
    Return ...;  
End Get  
Set  
    ... = value;  
End Set  
End Property
```

JScript 的语法:

```
function get name():String  
{  
    ...  
    return  
    ...;}  
function set name(value:String)  
{  
    ... = value;  
}
```

1.2.6 数组

C#的语法:

```
//一维数组
```

```
String[] a = new String[5];
a[0] = "1";
a[1] = "2";
a[2] = "3";
a[3] = "4";
a[4] = "5";
//二维数组
String[][] a = new String[2][2];
a[0][0] = "1";
a[0][1] = "2";
a[1][0] = "3";
a[1][1] = "4";
```

VB 的语法:

```
'一维数组
Dim a(3) As String
a(0) = "1"
a(1) = "2"
a(2) = "3"
a(3) = "4"
a(4) = "5"
'二维数组
Dim a(3,3) As String
a(0,0) = "1"
a(0,1) = "2"
a(1,0) = "3"
a(1,1) = "4"
```

JScript 的语法:

```
//一维数组
var a:String = new String[5];
a[0] = "1";
a[1] = "2";
a[2] = "3";
a[3] = "4";
a[4] = "5";
//二维数组
var a:String[][] = new String[2][2];
a[0][0] = "1";
a[0][1] = "2";
a[1][0] = "3";
a[1][1] = "4";
```

1.2.7 变量初始化

C#的语法:

```
string s = "Hello World";
```



```
int i = 1;
double[] a = {1.01, 2.02, 3.03};
```

VB 的语法:

```
Dim s As String = "Hello World"
Dim I As Integer = 1
Dim a() As Double = {1.01, 2.02, 3.03}
```

JScript 的语法:

```
var s:String = "Hello World";
var i:int = 1;
var a:double[] = [1.01, 2.02, 3.03];
```

1.2.8 选择语句

C#的语法:

```
if (Request.QueryString != null)
{
    ...
}
```

VB 的语法:

```
If Not (Request.QueryString = Null)
...
End If
```

JScript 的语法:

```
if (Request.QueryString != null)
{
    ...
}
```

1.2.9 分支语句

C#的语法:

```
switch (Num)
{
    case "1":
        ...
        break;
    case "2":
        ...
        break;
    case "3":
```