

PHP 4.x

企业级 Web 应用与开发

廖若雪 等编著



机械工业出版社
China Machine Press

Linux与自由软件资源丛书

PHP4.x企业级Web 应用与开发

廖若雪 等编著



机械工业出版社
China Machine Press

本书介绍了PHP4.x高级编程技术。主要包括：如何在PHP中实现类，组件技术的应用，XML在PHP中的使用以及如何使用PHP来开发WAP应用。此外，还涉及了有关正则表达式、模板、PHPDoc及CVS的一些知识。书中最后一章专门讲述了一个大型PHP程序的例子。

本书适合有Linux使用经验的读者阅读。

版权所有，侵权必究。

图书在版编目(CIP)数据

PHP4.x企业级Web应用与开发/廖若雪等编著.-北京：机械工业出版社，2001.4
(Linux与自由软件资源丛书)
ISBN 7-111-08798-4

I. P… II. 廖… III. PHP语言-程序设计 IV. TP312

中国版本图书馆CIP数据核字(2001)第10973号

机械工业出版社(北京市西城区百万庄大街22号 邮政编码 100037)

责任编辑：谢 昱

北京昌平奔腾印刷厂印刷 · 新华书店北京发行所发行

2001年4月第1版第1次印刷

787mm×1092mm 1/16 · 20.5印张

印数：0 001-5 000册

定价：33.00元

凡购本书，如有倒页、脱页、缺页，由本社发行部调换

前言

PHP作为一种跨平台、跨浏览器的Web开发工具，在PHP/FI 2.0以后得到了迅猛的发展。PHP3.0以其优秀的性能、跨平台的优势和开放性获得了许多人的青睐。在PHP4推出以后，凭着在类、组件技术、XML和Zend极限器方面的进步，PHP已经逐步从小型网站的开发工具和自由软件爱好者手中的“玩具”成长为了一个可以用于各种类型网站开发的强大技术。

本书将讨论各种PHP高级应用方面的知识，利用这些特性，PHP为大、中型网站的开发提供了必要的支持。这些特性包括：

- 1) PHP的面向对象技术。
- 2) PHP中可以使用的组件技术。
- 3) XML在PHP中的应用。
- 4) PHP中的文档控制和版本控制。
- 5) 正则表达式的使用。
- 6) 用PHP开发WML无线应用。
- 7) 使用模板技术来达到数据和显示的分离。

在本书的最后，还将展示一个利用PHP开发的大型项目，其中几乎用到了上面的所有知识。

本书使用的PHP版本是本书开始写作时的最新版本——PHP4.03pl1，加入了Java和Servlet两个扩充包，读者如果使用的PHP版本号小于4.03，可以到<http://www.php.net>去下载。在PHP4的每一个新的版本中，即使是最小的版本变化，也加入了许多新的内容，读者在阅读本书的时候需要注意这一点。

本书不是一本PHP的入门书籍，读者应该对PHP已经有了初步的了解，还应该具有数据库、面向对象程序设计、组件技术、XML方面的基础知识。如果读者希望利用本书的知识将PHP和Java技术结合使用（如JSP和EJB），那么还需要具有Java服务器程序设计的知识，可以参考机械工业出版社出版的《JSP高级开发》一书。

考虑到目前WML的应用情况，本书在讲解PHP的WML程序设计以前讲解了关于WML的知识。

对本书有任何看法和建议都可以到作者的主页上来发表意见，作者的主页是：www://highbook.home.chinaren.com/。

参加本书编写的还有张卉、江庆铭、张珂、顾超、马飞、王辛、沈凡、徐洋、赵小立、李星、李春起、陈宜、郑章辉、程小刚、张勇、于艳春、王勇刚、储章辉、任世岩、刘小涛、杨坚、张春生、徐丙坤、王培涛、刘志勇。

由于作者水平有限，疏漏之处在所难免，请敬读者指正。

廖若雪

2001年1月

目 录

前言	
第1章 概述	1
1.1 PHP中的类	1
1.2 PHP和组件技术	1
1.3 PHP和XML	1
1.4 使用PHP开发WAP应用	1
1.5 大型PHP程序详解	2
1.6 本书的其他内容	2
第2章 在PHP中使用类	3
2.1 PHP中类的语法	3
2.2 类的使用	6
2.2.1 Web站点中的对象	6
2.2.2 使用类和对象的优点和缺点	11
2.3 类和对象的使用技巧	11
2.3.1 对象的序列化	12
2.3.2 方法重载的替代方案	16
2.3.3 使用参数缺省值	20
2.3.4 使用类和对象相关函数	23
第3章 在PHP中使用组件技术	27
3.1 在PHP中使用COM组件	27
3.2 在PHP中使用JavaBean	29
3.3 在PHP中利用JavaBean访问EJB	34
第4章 正则表达式的使用	39
4.1 基本模式	39
4.1.1 “^” 头匹配	39
4.1.2 “\$” 尾匹配	40
4.1.3 转义序列	41
4.2 字符簇	43
4.3 “{ }” 的用途	45
4.4 Perl兼容的正则表达式	46
4.5 实例	47
第5章 分离数据与显示：使用模板	52
5.1 模板处理类的设计	52
5.2 模板处理类的实现	53
5.2.1 代码	53
5.2.2 重要属性	58
5.2.3 数据的设置和读取	59
5.2.4 文本分析	60
5.3 模板处理类的使用	60
5.3.1 最基本的例子	60
5.3.2 模板嵌套	62
5.3.3 多重数据的处理	63
5.3.4 块处理	66
第6章 XML	70
6.1 XML分析器和PHP的XML函数库	70
6.1.1 XML分析器	70
6.1.2 PHP的XML函数库	71
6.1.3 回调函数和错误处理	72
6.2 读取XML文件中的信息	74
6.2.1 分析	74
6.2.2 实现	74
6.3 处理、重新输出XML文件中的信息	77
6.3.1 分析	78
6.3.2 实现	78
6.4 XML文件的层次结构	87
6.4.1 分析	87
6.4.2 实现	87
6.5 XML文件分析器	94
6.5.1 使用OOP代码原则	94
6.5.2 XML文件分析器的实现	94
6.5.3 XML文件处理类的使用	104
第7章 轻松生成PHP文档	106
7.1 PHPDoc要求的注释格式	106
7.1.1 类注释	112

7.1.2 属性注释	113	9.2.3 手机发送Email	161
7.1.3 方法注释	115	9.2.4 数据库的使用	162
7.2 生成文档	116	第10章 PHP4使用技巧	164
7.2.1 简单的使用方案	116	10.1 变量的变量、变量的函数和函数 的变量	164
7.2.2 文档模板文件的示例	117	10.1.1 变量的变量	164
7.3 其他的选择	127	10.1.2 变量的函数	167
7.3.1 其他的PHP文档化工具	127	10.1.3 函数的变量	169
7.3.2 将HeaderDoc用于PHP	127	10.2 输出缓冲区的控制	170
7.3.3 文档化一个类	128	10.3 动态改变ini配置	171
7.3.4 文档化函数或方法	129	10.3.1 函数	171
7.3.5 文档化变量或类变量	130	10.3.2 动态选择邮件发送服务器	171
7.3.6 头信息	130	10.3.3 动态改变脚本执行时限	173
第8章 使用CVS	131	10.4 IMAP	174
8.1 CVS概述	131	10.5 发送邮件	175
8.1.1 代码集中的配置	131	10.6 RamDisk	180
8.1.2 无限制的代码检出模式	131	10.6.1 RamDisk介绍	180
8.2 CVS的使用	132	10.6.2 如何使用RamDisk	180
8.2.1 设置仓库的路径	132	10.6.3 改变RamDisk的大小	181
8.2.2 检出一个可工作目录	132	10.6.4 在Web服务器中使用RamDisk 的例子	182
8.2.3 对文件进行修改	133	10.6.5 通过RamDisk加速小型数据库的 访问速度	183
8.2.4 合并所作的修改	134	第11章 大型PHP程序实例分析	184
8.2.5 提交修改	135	11.1 项目的整体结构	184
8.2.6 检查修改	135	11.1.1 Integrated Template	184
8.2.7 增加和删除文件	138	11.1.2 Phpdoc Error	184
8.2.8 编写良好的日志记录	139	11.1.3 Phpdoc Object	184
8.2.9 处理冲突	140	11.2 生成文档的Render类族	185
第9章 无线互联网应用: WAP	142	11.2.1 代码	185
9.1 概述	142	11.2.2 分析	226
9.1.1 名词释义	143	11.3 分析源文件的Parser类族	227
9.1.2 Web服务器和WAP	143	11.3.1 代码	227
9.1.3 一个简单的WML文件	144	11.3.2 分析	314
9.1.4 WML文件的语法和特点	144		
9.2 使用PHP构建WAP应用	160		
9.2.1 配置服务器: Apache	160		
9.2.2 插入PHP脚本到WML代码中	160		

第1章 概 述

PHP作为一种跨平台、跨浏览器的Web开发工具，在PHP/FI 2.0以后得到了迅猛的发展。PHP3.0以其优秀的性能、跨平台的优势和开放性获得了许多人的青睐，在PHP4推出以后，凭着在类、组件技术、XML和Zend极限器方面的进步，PHP已经逐步从小型网站的开发工具和自由软件爱好者手中的“玩具”成长为一个可以用于各种类型网站开发的强大技术。

本章将概述本书要讨论的各种PHP高级应用方面的知识，利用这些特性，PHP为大、中型网站的开发提供了必要的支持。

1.1 PHP中的类

面向对象的程序设计方法作为一种切实有效的程序设计方法而为程序开发人员喜爱。PHP本身是一种基于函数的Web开发工具，但是也对面向对象的程序设计提供了支持，在PHP中使用类，可以最大限度地重用代码和有效地使用代码。并且，除了自己向PHP中添加函数外，使用类来扩充PHP是最佳的方法。但是，也要看到，PHP对类的支持还不是那么完善，与ASP和JSP相比，PHP中的类还仅仅具有最简单的功能，因此，在PHP中使用类就需要开发人员做出更多的控制。

许多开发人员在使用PHP的时候很不习惯使用类，而PHP的许多书籍也很少提及PHP的类。但实际上，要利用PHP开发大型的动态网站，如果希望省时省力，使用类几乎就是必须的，这是因为使用函数过程化进行PHP开发已经无法满足多人合作、含有复杂流程的大型网站的开发需要了。

本书第2章重点讲述PHP中实现类的语法和注意事项。

1.2 PHP和组件技术

组件技术的使用使得Web的开发逐步走向模块化，但是，PHP本身没有提供任何组件技术。PHP在另一个方面对组件技术提供了支持，目前PHP可以直接使用的组件有“Java Beans”和“COM”。通过使用这两种组件技术，PHP的开发终于进入了组件的世界。

本书第3章讲述组件技术的使用。

1.3 PHP和XML

在开发PHP的大型应用中，XML的使用显得至关重要，作为一种标准的数据交换格式，XML既可以作为数据源使用，也可以成为PHP应用的配置文件。总之，在开发大型的PHP应用中，XML已经成为不可缺少的部分。

本书第6章讲述XML在PHP中的使用。

1.4 使用PHP开发WAP应用

WAP（Wireless Application Protocol，无线应用协议）不仅仅是纸面上的意义，对于许多人

来说，还代表着下一轮的互联网热潮和新的机会。PHP在这个最新的技术面前依然是强大的。

本书第9章将介绍如何使用PHP来开发WAP应用。

1.5 大型PHP程序详解

学习他人的代码是尽快提高自己水平的最好办法。尽管PHP在国内已经相当流行，但是，大部分开发人员所使用的开发方法都不符合大型PHP程序开发的要求。

本书最后一章将讲述一个大型PHP程序的例子。

1.6 本书的其他内容

本书的主要目的在于讲述如何利用PHP开发大型的、企业级的Web应用，本书还包括下面的内容：

- 正则表达式——优秀的文本分析、处理函数库。
- 模板——数据显示和处理的分离。
- PHPDoc——为PHP文件建立文档的简单方法。
- CVS——多人合作开发时必不可少的源代码控制工具。

第2章 在PHP中使用类

在使用PHP开发大型的Web应用的时候，使用类是一项重要的技术。在PHP的语法中，很早就提供了对类的支持，例如，著名的PHPLIB就是使用了一整套的类来扩展PHP。但是，直到PHP3，PHP中类的执行速度仍然相当慢。到了PHP4中，这种情况大为改观，使用了Zend技术的PHP4终于为PHP中类的执行提供了满意的速度。

在PHP中使用类，使得使用PHP书写的代码具有了更好的可重用性和扩展性。如果读者熟悉C++、Java、SmallTalk或Object Pascal等面向对象的程序设计语言，那么对“类”和“对象”应该有比较深刻的认识，和这些程序设计语言中的类比较，PHP中的“类”具有以下特点：

- 类的属性和方法没有“public”、“private”和“protected”等分别。
- 不支持多重继承。
- 不支持方法重载。

如果读者对面向对象的程序设计没有任何的概念，最好首先参考其他的书籍学习一下面向对象的程序设计。

2.1 PHP中类的语法

PHP中的类语法可以通过下面的例子加以说明：

Syntax.php:

```
<?
class SyntaxSample {

    //这是一个类的属性
    var $param1;

    //这是类的构造函数
    function SyntaxSample() {
        //引用类中的属性
        $this->param1 = "I`m SyntaxSample`s param";
    }

    //这是一个类的方法
    function method($param) {
        $this->param1 = $param;
    }
}

class ExtendsSample extends SyntaxSample {

    //子类的一个属性
```

```
var $extendsParam;

//这是子类的构造函数
function ExtendsSample($param) {
    $this->SyntaxSample();
    $this->extendsParam = $param;
}

//覆盖父类中的method($param) 方法
function method($param) {
    $this->extendsParam = $param;
}

}

class DebugInfo {

    //调试信息的输出计数
    var $count;

    function DebugInfo() {
        $this->count = 0;
    }

    //输出调试信息到屏幕
    function showWorkInfoToScreen($info) {
        $outputStr = "<h4><font color=red>";
        $outputStr .= $this->count;
        $outputStr .= ", ";
        $outputStr .= $info;
        $outputStr .= "</font></h4>";
        echo $outputStr;
        $this->count++;
    }
}

$debug = new DebugInfo();

$debug->showWorkInfoToScreen("Now new a SyntaxSample!");
$test0 = new SyntaxSample();

$debug->showWorkInfoToScreen("Show param`s value:");
echo $test0->param1;

$debug->showWorkInfoToScreen("set param`s value using method");
$test0->method("I`m set by method");

$debug->showWorkInfoToScreen("Show param`s value:");
echo $test0->param1;
```

```

$debug->showWorkInfoToScreen("Now new a ExtendsSample");
$test1 = new ExtendsSample
    ("I'm ExtendsSample's extendsParam set by constructor with arguement!");

$debug->showWorkInfoToScreen("Show param's value:");
echo $test1->param1;

$debug->showWorkInfoToScreen("Show extendsParam's value:");
echo $test1->extendsParam;

$debug->showWorkInfoToScreen
    ("use <i>method</i> method to set extendsParam");
$test1->method("I'm set by ExtendsSample's method");

$debug->showWorkInfoToScreen("Show param's value:");
echo $test1->param1;

$debug->showWorkInfoToScreen("Show extendsParam's value:");
echo $test1->extendsParam;

?>

```

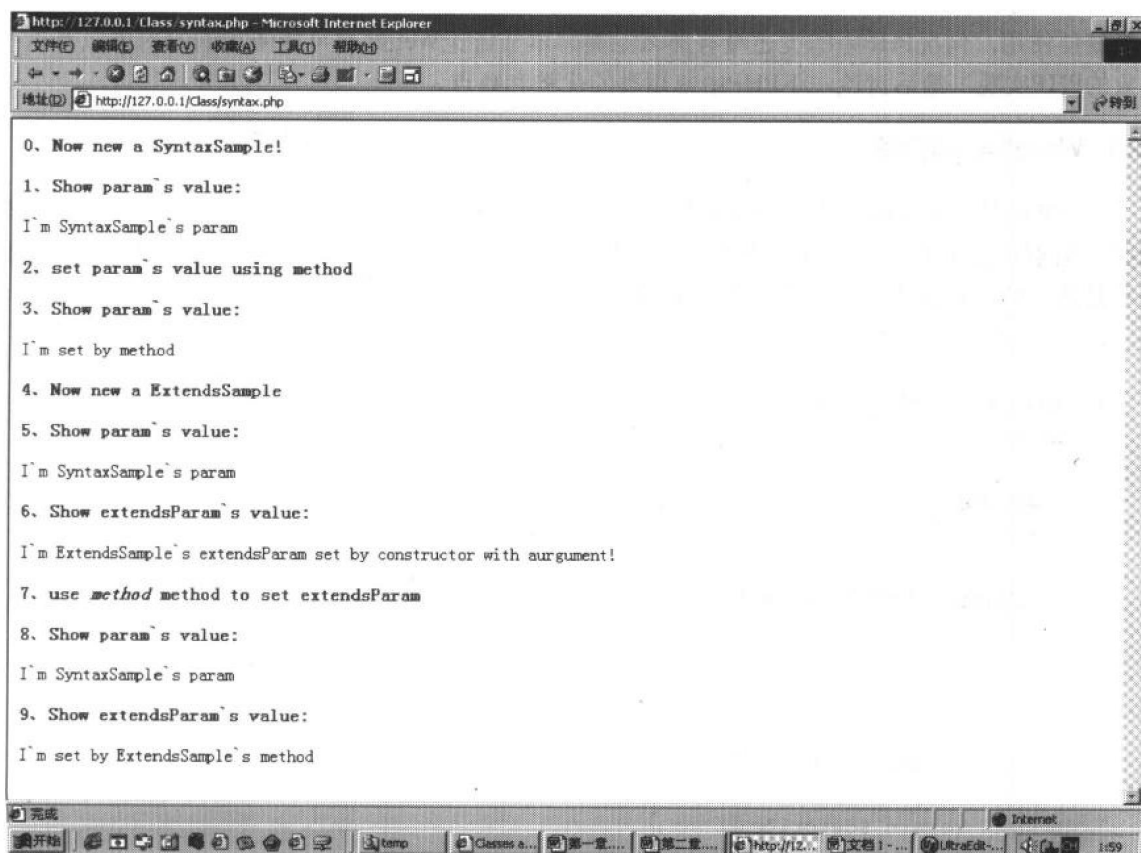


图 2-1

上面的例子演示了PHP中类的使用语法，这里归纳语法特点如下：

- 类定义以class开头。
- 使用“extends”关键字来进行继承。
- 必须使用“new”关键字来实例化对象。
- 使用var \$param 语法来声明属性。
- 使用function method(\$argument1,argument2,...) {} 语法来声明方法。
- 使用和类名相同的方法名来作为构造函数，可以包含参数。
- 子类中可以使用同名方法覆盖掉父类中的方法。
- 在类定义中需要使用“\$this->”来引用类的属性和方法。
- 在子类中需要显式地调用父类的构造函数。

上面实例的屏幕显示结果如图2-1所示。

屏幕详细输出了代码的各个执行步骤以及执行后的结果，读者可以检验一下PHP的类机制是否如前所述。

2.2 类的使用

面向对象的程序设计是一门相当有用的技术，但是，同其他任何事物一样，即使它再好也不能随便使用，何况PHP中的类还没有那么完美。本节讨论Web站点可能使用到的对象以及这些对象的PHP实现，最后讨论一下PHP中使用类的优势和缺点。

2.2.1 Web站点中的对象

一个Web站点也可以看作一个应用程序，其中可能包含各种各样的对象，例如，一个和汽车相关的站点可能需要一个描述汽车的类来储存汽车数据，而一个讲述各种人物的站点可能需要一个描述人的信息的类。下面来看看如何建立这两个类。

```
Object.php
<?
//Object类，PHP使用类的测试
class Object {

    //对象描述
    var $describe;

    //构造函数，设置类属性的初始值
    function Object() {
        $this->describe = "Not Set!";
    }

    //为Object类提供一个缺省的字符串值
    function toString() {
        return $describe;
    }
}
```

```
//设置Object的描述信息
function setDescribe($des) {
    $this->describe = $des;
}
}
?>

Vehicle.php
<?
//包含Object类的定义文件
require_once("inc/Object.php");

//Vehicle类, PHP使用类的测试, 从Object类继承
class Vehicle extends Object{

    //Vehicle的类型、重量和颜色
    var $type;
    var $weight;
    var $color;

    //构造函数, 设定缺省值
    function Vehicle() {
        $this->Object();
        $this->type = "Not Set!";
        $this->weight = 0;
        $this->color = "Not Set!";
    }

    //设置Vehicle的类型
    function setVehicleType($vehicleType) {
        $this->type = $vehicleType;
    }

    //设置Vehicle的重量
    function setVehicleWeight($vehicleWeight) {
        $this->weight = $vehicleWeight;
    }

    //设置Vehicle的颜色
    function setVehicleColor($vehicleColor) {
        $this->color = $vehicleColor;
    }

    //返回Vehicle的描述信息
    function toString() {
        $vehicleDescribe = "This is a vehicle, its type is ";
    }
}
```

```
        $vehicleDescribe .= $this->type;
        $vehicleDescribe .= " ,its weight is ";
        $vehicleDescribe .= $this->weight;
        $vehicleDescribe .= " ,its color is ";
        $vehicleDescribe .= $this->color;
        $vehicleDescribe .= ".";
        return $vehicleDescribe;
    }
}
?>
```

People.php

<?

//包含Object类的定义文件

require_once("inc/Object.php");

//Vehicle类, PHP使用类的测试, 从Object类继承

class People extends Object{

//人的性别、体重、身高、出生日期

var \$sex;

var \$weight;

var \$height;

var \$birthday;

//构造函数

function People(\$sex,\$weight,\$height,\$birthday) {

if (\$sex=="man") {

\$this->sex = 0;

}elseif (\$sex=="woman") {

\$this->sex = 1;

}else {

\$this->sex = 0;

}

\$this->weight = \$weight;

\$this->height = \$height;

\$arrayBirthday = explode("-", \$birthday);

\$this->birthday = mktime

(0,0,0,\$arrayBirthday[1],\$arrayBirthday[2],\$arrayBirthday[0]);

}

function sexSex(\$sex) {

if (\$sex=="man") {

\$this->sex = 0;

}elseif (\$sex=="woman") {

\$this->sex = 1;

}else {

\$this->sex = 0;

```
    }  
}  
  
function setWeight($weight) {  
    $this->weight = $weight;  
}  
  
function setHeight($height) {  
    $this->height = $height;  
}  
  
function setBirthday($birthday) {  
    $arrayBirthday = explode("-", $birthday);  
    $this->birthday = mktime  
        (0,0,0,$arrayBirthday[1],$arrayBirthday[2],$arrayBirthday[0]);  
}  
  
function getSex() {  
    if ($this->sex==0) {  
        return "man";  
    }elseif ($this->sex==1) {  
        return "woman";  
    }else {  
        return "man";  
    }  
}  
  
function getWeight() {  
    return $this->weight;  
}  
  
function getHeight() {  
    return $this->height;  
}  
  
function getBirthday() {  
    return date("Y-m-d", $this->birthday);  
}  
  
function toString() {  
    $peopleDescribe = "This is a people,sex is ";  
    $peopleDescribe .= $this->getSex();  
    $peopleDescribe .= " ,weight is ";  
    $peopleDescribe .= $this->weight;  
    $peopleDescribe .= " kg ,height is ";  
    $peopleDescribe .= $this->height;  
    $peopleDescribe .= " m ,birthday is ";  
    $peopleDescribe .= $this->getBirthday();  
}
```

```

    $peopleDescribe .= ".";
    return $peopleDescribe;
}
}
?>

```

上面的代码中包含了一个基础类：Object和两个子类：Vehicle和People，其中Vehicle子类包含了汽车的各种信息和这些信息的处理方法，而People子类则包含了人的各种信息和这些信息的处理方法。

Web站点中可能用到对象技术的地方是很多的，下面举几个例子：

1. 数据库连接类

在一个Web站点中，可能会涉及到许多数据库的操作，这些操作对网站有着不同的影响。在网站的成百上千个PHP文件中，可能会包含上百个需要数据库操作语句的地方，在这些地方都使用原始的PHP函数是很不好的。比如，如果需要修改数据库连接的密码，工作量就大得让人心烦。常见的解决方法是使用一个或几个函数来将数据库连接封装起来，然后放到一个文件中，在需要使用数据库连接的时候将这个文件包含进来就可以了。但最好的方法还是使用类，建立一个专门管理数据库连接的类。为什么呢？使用类的好处可以归纳如下：

- 可以隐藏各种数据库连接函数的差异，提供统一的数据库接口，为开发和转换数据库提供方便。
- 通过一系列判断，可以提供简单明了的数据库连接方式，还可以在类中自动实现数据库连接时需要的预处理。
- 隐藏错误，将错误的代码控制在类的内部。
- 通过类的继承机制，可以建立各种专用的数据库连接类。这里需要说明的是，所谓的专用的数据库连接类是指那种仅仅允许使用特定的数据库查询的类。

2. 用户信息对象

传统的获得当前用户信息的方法是使用cookie保存当前用户的标识，然后利用这个标识到数据库或其他数据源中去读数据。在PHP4中提供了内置的Session支持以后，用户信息就可以保存在Session中了。

那么使用对象来存储用户信息有什么好处呢？

首先，可以把用户数据作为一个整体的数据存储到Session中，当用户登录时就可以把所有的用户信息全部存储到对象中。以后需要使用用户信息时，就不需要再次查询数据库了，这样既加快了反应速度，又减少了数据库的负担。

其次，在类中可以为用户数据提供一系列的处理和输出形式不同的方法，方便了数据的处理工作。

3. 购物车对象

PHP的文档中对类和对象提供的例子中就包含了一个购物车的示例。实际上，购物车使用对象来实现是非常自然的一件事情，放入购物车中的物品使用购物车对象的属性类储存，而向购物车中添加数据和从购物车中读取数据就是购物车对象的方法。

实际上，如果给购物车对象提供一个可以将对象中的数据转换为可以直接存储到数据库中

的数据的方法，购物车对象就可以很容易保存到数据库中并从数据库中读取出来。

2.2.2 使用类和对象的优点和缺点

要想确定在自己的PHP程序中是否需要使用类和对象技术，首先需要看一看使用类和对象的优点和缺点。

1. 使用类和对象的优点

无论是否在PHP中，面向对象的程序设计都比传统的程序设计方法具有优势。主要表现在以下几个方面：

- 程序设计的概念清楚。
- 代码可重用性高。
- 代码灵活性好。

使用PHP开发Web应用程序的时候，如果使用面向对象的方法，当然也会具有上面所说的好处，而且，按照上一节中的叙述，Web开发中遇到的需要使用对象的地方比传统的程序设计更多，使用起来也更自然。

由于PHP开发的Web应用存在与不同的数据库连接，及在不同的系统下工作等问题，因此，使用类可以将这些需要判断的地方封装起来。在Web应用中经常会遇到需要采取特殊的检查措施来防止破坏的地方，将防止破坏的代码封装到类中，这样既方便了使用，又可以随时更新防止破坏的方法。

使用PHP开发大型Web应用的时候，使用类可以使得多人合作变得简单明了，面向对象技术加上后面讲述的CVS，就成为PHP大型Web应用开发的基石。

PHP对面向对象技术不完善的支持中最让人头疼的地方大概就是不支持方法重载了。在2.3.2节，将介绍一种替代的方案。有了方法重载的替代方案后，在PHP中使用面向对象技术的最后一个疑虑将不复存在。

2. 使用类和对象的缺点

PHP开发中使用面向对象技术也有一些明显的缺点，首先，由于PHP不支持私有和保护类型的类属性，就导致开发人员在使用PHP的类时没有任何限制。懒惰的开发人员也许会使用\$Obj->attribute的方法来使用属性，这样就很可能造成混乱。

还有一点需要引起注意的是，由于PHP是以明码形式存在的，所以无法阻止任何一名开发人员察看到PHP的源文件，这也违背了面向对象技术的封装原则。

PHP中的类和对象的速度依然是不那么快的，好在没有任何理由需要开发人员像使用Java和SmallTalk那样使用PHP。在开发PHP程序的时候，如果使用类的确可以提供好处，那么就使用类，否则就使用传统的面向过程的程序开发技术。

2.3 类和对象的使用技巧

本节讲述PHP中类和对象的几种使用技巧，通过这些技巧，可以更好地使用PHP中的类和对象。