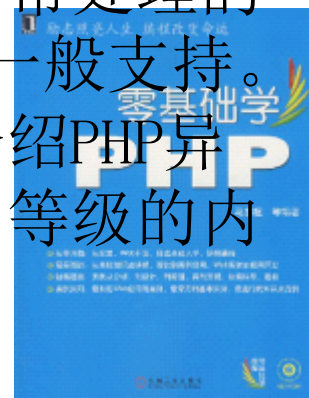


第16章 PHP程序中的错误处理

- 任何程序都由人来编写完成，因此一定会产生或多或少的错误，程序中的错误是在所难免的，开发人员不必过于为此苦恼，只要仔细编写和认真思考，很多错误都是可以避免的。一般情况销，当一个PHP程序执行出现异常或错误，可以通过在程序中设置断点、输出一下标记或变量的值，用这种方法来调试PHP程序，从而发现错误，解决问题。PHP没有象Java语言那样，提供对异常处理的丰富支持。但还是提供了对异常处理的一般支持。本章首先从常见的PHP错误讲起，然后介绍PHP异常处理类，最后介绍PHP常见错误类型、等级的内容。



16.1 重视程序中的错误处理

- 异常处理任何一门计算机语言都需要考虑的问题，也是程序员需要考虑的问题。因为程序的执行过程中总会遇到许多可预知或不可预知的错误事件，这些错误有时是由于逻辑错误造成的，有时是由于开发人员疏忽造成的，有时是系统异常造成的。比如说，某个目录下本应存在的一个文件由于某些原因不在了，或者数组越界等问题，又如，连接数据库的时候，远程数据库连接没有响应等等。
- 不管哪一种错误，都会是程序执行结果有误或者执行失败，甚至造成更严重的后果。所以一定要重视程序中的错误处理。这样，代码会更加健壮、安全、可靠。也可以减少后期维护量和维护成本。
- 在PHP程序中，通常可以在程序中设置一系列判断，来排查 PHP程序可能出现错误或异常的地方。但这样一来，程序可读性差了很多，总是有许多与真正工作无关的代码，而且也给开发人员增加了极大的工作负担。使用异常处理机制，可以把错误处理和真正的工作分开来，代码会更易组织，更清晰。最重要的是，程序更安全了，不至于由于一些小的疏忽而使程序意外崩溃。所以程序中的异常处理还是非常必要的。



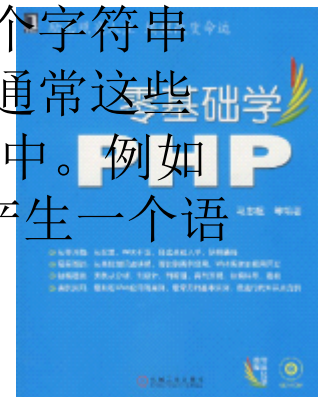
16.2 在PHP中处理程序错误

- PHP中的程序异常通常有以下3种情况。
- 语法错误。
- 运行时错误。
- 逻辑错误。
- 下面分3小节讲述这3种错误在PHP中应该如何检查、解决以及如何避免。



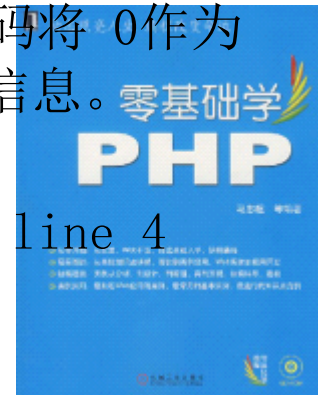
16.2.1 语法错误的处理

- 每种编程语言都有自己的一套编写规则，这些规则通常叫做语法，PHP也不例外。例如，PHP语言规定每个语句必须以分号结束，又如，定义函数时必须使用关键字function，诸如此类的都是PHP的语法内容。如果程序中的语句有不符合这些语法规则的地方，PHP分析器就会产生语法分析错误，导致程序执行失败或中断。计算机非常严格地按照PHP语法执行PHP脚本，任何语法错误都不能容忍，即便是在一个语句结束的地方少了一个分号。
- 常见的语法错误往往都是比较细小的地方，或者由粗心大意导致。比如，在一个语句末尾缺少分号，在一个字符串结束时没有单引号或双引号，括号不匹配等等。通常这些问题会在PHP脚本执行时由PHP分析器输出到页面中。例如下面一小段有语法错误的PHP代码，执行时就会产生一个语法错误。



16.2.2 运行时错误的处理

- 语法错误可以在PHP分析器执行脚本时检测到，而运行时错误就不那么简单了，下面通过实例来说明。比如有如下所示的一段PHP代码。
- `<?php`
- `$i = 10;`
- `$j = 0;`
- `$k = $i/$j;`
- `?>`
- 这段代码本身没有任何语法错误，但是，这段代码将 0 作为除数，所以会产生类似下面的警告（Warning）信息。
- Warning: Division by zero in
D:\Apache2\htdocs\webroot\ch16\test.php on line 4



16.2.3 逻辑错误的处理

- 逻辑错误是一种比较难以发现和处理的错误。这种错误发生时并没有明显错误信息报告，因为程序都正常执行，只是最终结果（或在某种情况下的结果）并不是所期望的结果。例如下面的示例代码。
- ```
<?php
```
- ```
for($i=0; $i<10; ++$i);
```
- ```
{
```
- ```
    echo 'PHP' ;
```
- ```
}
```
- ```
?>
```



16.3 PHP中的异常处理类及其用法

- PHP5增加了类似于其它语言的异常处理模块。这使得在PHP程序中排查错误、捕获异常更加简便容易，也使得PHP程序在健壮性方面有很大改善和提高。这一模块在PHP中的具体体现就是，PHP提供了一个名叫Exception的类，来完成对PHP程序异常的处理。这个类包含了一些处理异常的函数，这些函数可以捕获程序异常和错误。本节就为读者介绍PHP中的该异常处理类及类函数用法。



16.3.1 PHP中的异常处理类

- PHP5提供了内置的异常处理类——Exception，首先介绍该类的以下成员函数。
- getMessage ()，该函数可以返回对错误描述信息。
- getCode ()，该函数返回错误代码，以数字形式出现。
- getFile ()，该函数返回发生错误的文件名。
- getLine ()，该函数返回发生错误的代码行号。
- getTrace ()，该函数返回 backtrace () 数组。
- getTraceAsString ()，该函数返回已格式化成字符串的、由函数getTrace ()所产生的信息。
- __toString ()，该函数产生异常的字符串信息，它可以重载。注意，该函数最前部是两个下划线。



16.3.2 PHP程序中捕获异常的方法

- 在了解了PHP处理异常的类之后，本小节向读者介绍在 PHP 程序中使用该类捕获异常和处理异常的方法。首先介绍两种抛出异常的格式，它们如下所示。
- try...catch...：该处理异常方式的具体格式如下所示。
- try
- {
- //这里是可能出现异常或错误的代码，比如文件操作、数据库操作等
- }
- catch(Exception \$e)
- {
- //输出错误信息
- }



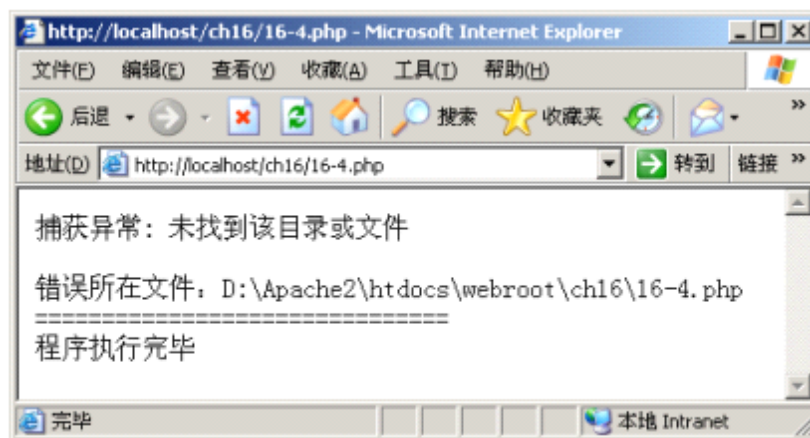
16.3.3 获取异常信息的函数

- 通过前面的学习，读者已经了解到，当一个程序遇到一个错误时，可以建立Exception类的对象来处理，就像下面这段代码一样。
- `$e = new Exception("Could not open file");`
- 此时，Exception类的构造函数将通过一错误信息创建一个该类的对象\$e。建立这个对象后，通常使用关键字throw将这个异常（或者说错误）抛出。throw将脚本的执行中止，并使相关的Exception对象对客户代码可用。



16.3.4 获取错误发生所在文件的函数

- 有了上面通过Exception类获取错误信息的程序，获取错误所发生的文件就很简单了。只需在程序中使用Exception类的成员函数getFile()即可，

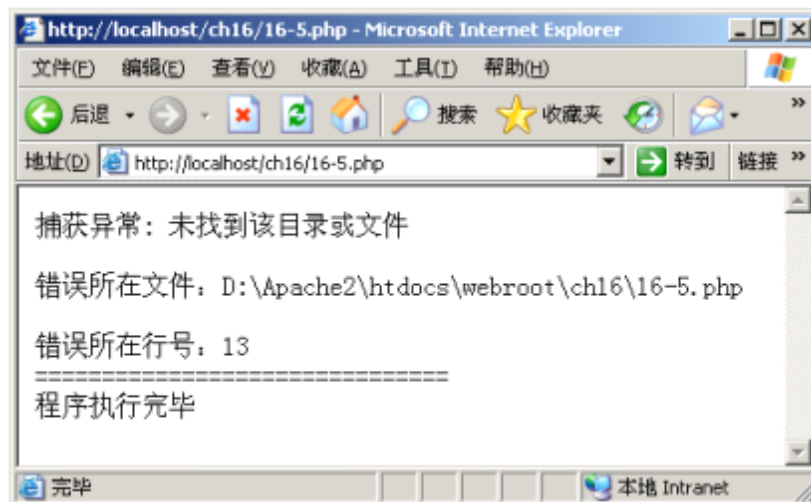


使用 Exception 类的成员函数 getFile



16.3.5 获取错误发生所在行的函数

- 类似获取错误文件名称的办法，在程序中通过调用Exception类的成员函数getLine()，可以获取错误所在的行号，



使用 Exception 类的成员函数 getLine



16.4 PHP配置中和错误处理有关的内容

- 除了通过以上几小节讲述的处理 PHP 程序错误的办法之外，通过一些对 PHP 的配置，也可以帮助开发处理 PHP 程序错误。PHP 的配置文件 `php.ini` 中，有一些和错误信息排查、错误信息处理有关的配置项，下面向读者介绍这些内容，读者通过了解这些配置项，可以了解如何增强系统安全性和程序的可控性。`php.ini` 中和错误处理有关的项及其介绍如下所示。
- `display_errors = Off`：该项决定是否显示错误信息。当其取值为 `On` 时，表示打开选项，当为 `Off` 时，表示关闭选项。如果该项设置为 `On`，错误信息会输出到用户浏览器端。当开发时，可以将该项设置为 `On`，以便快速找到程序错误所在。当正式发布 Web 站点时，应该关掉这个特性，并使用错误日志代替，因为打开这个特性可能暴露站点的一些安全信息。以下介绍的各配置项，通常都有 `On` 和 `Off` 两种设置值。



16.4 PHP配置中和错误处理有关的内容

- `log_errors = On`: 该项决定是否在日志文件中记录错误。正如上面讲述的，当正式发布站点时，该项应该设置为 `On`，这样可以通过日志文件，查找相关错误或异常信息。
- `error_log = filename`: 设定记录错误信息的日志文件。`filename`为日志文件名，如果设为“`syslog`”，错误日志将被记录到系统日志中。
- `log_errors_max_len = 1024`: 设置错误日志文件的最大长度，设为0可以允许无限长度。
- `html_errors = Off`: 如果该项设置为 `Off`，禁用出错信息中的HTML标记。
- `display_startup_errors = Off`: 该项的作用是，如果 `display_errors` 项被打开，即便关闭此参数也不显示PHP启动时的错误。通常需要将该参数设置为 `Off`，除非必须要用于调试中。
- `ignore_repeated_errors = Off`: 忽略重复的错误信息。如果关闭这个参数，错误信息必须出现在同一个文件的同一行才被忽略。如果打开这个参数，则没有这个限制。
- `ignore_repeated_source = Off`: 忽略重复的错误源，忽略错误信息的重复错误源。如果打开这个参数，将不会记录不同的出错文件和对应该行的错误信息。



16.5 PHP中自定义的错误处理

- PHP拥有内置工具，可以让开发人员捕捉程序错误，然后将它们转到自定义的错误处理器。在自定义的错误处理器中，编程显示更多关于错误的详细信息。此外，还可以将错误信息写入文件或数据库。
- 本小节，将阐述如何使用PHP的错误处理API构建用户自定义的错误处理器，并且说明如何以简单而友好的方式显示和管理程序的错误信息。
- PHP有3种最基本的错误类型：注意、警告和错误（或致命错误）。通常情况下，注意和警告不会终止程序。但是致命错误则会导致程序立即中断。例如，调用一个没有定义的函数或引用一个不存在的对象就是一个致命性的错误。在PHP中，关键字E_NOTICE表示注意类型的错误，E_WARNING表示警告类型的错误，E_ERROR表示致命类型的错误。



16.6 小结

- 本章介绍了PHP程序中处理错误和异常的基本方法，这部分内容总结如下。
- PHP程序中的语法错误。
- PHP程序中的运行时错误。
- PHP程序中的逻辑错误。
- PHP中的异常处理类及其用法。
- 最后，介绍了PHP配置文件中与错误处理有关的配置项，以及如何使用自定义的错误处理函数。

