

## 第 2 讲 SPL 和设计模式

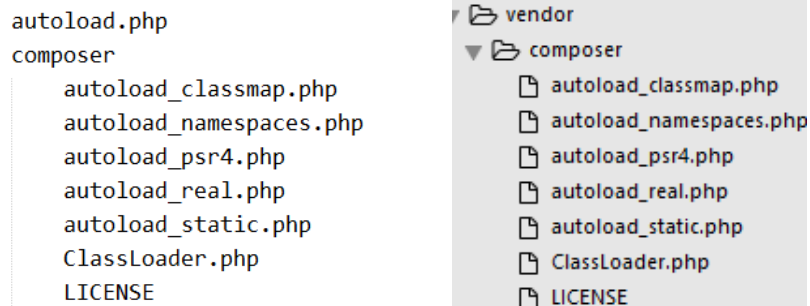
### 1. SPL 文件处理

SPL 文件操作是一个比较重要的操作，借助 SPL 提供的方法，可以很方便地显示或获取文件（或目录）的相关信息。请使用 `DirectoryIterator` 类和 `SPLFileInfo` 类，编写函数实现输出某一个目录的目录列表。

(1) 函数原型为：`function printDir($dirName, $tabNumber = 0)`，`$dirName` 表示待输出的目录名，`$tabNumber` 表示待输出的 TAB 键个数（只需设置默认值 0 即可，表示不输出 TAB 键）。

(2) 使用递归函数实现此过程。

(3) 结果示例：



### 2. SPL 接口

SPL 中有两个最基本的接口 `Countable` 接口 和 `Iterator` 接口，请编写一个类 `ArrayClass`，实现这两个接口，并编写测试代码。

- (1) 类 `ArrayClass` 表示数组元素组成的类；
- (2) 可以使用 `count($obj)` 访问 `ArrayClass` 类中数组元素的个数；
- (3) 可以使用 迭代器接口 访问数组元素。

### 3. 类的魔术方法

PHP 面向对象机制中，提供了魔术方法，借助魔术方法，可以实现一些非常方便的操作。请使用 `__call()` 和 `__callStatic()` 这两个魔术方法，实现既可以使用 对象的方式调用类方法，也可以使用 静态方式调用类方法。如 `$obj` 是 `Db` 类的实例对象，则既可以 使用 `$obj->getCharset()` 方式调用，也可以使用 `Db::getCharset()` 方式调用。

### 4. 设计模式 – 单例模式

一般使用单例模式表示一个类只能被实例化一次，请回答以下几个问题。

- (1) 单例模式的意义何在？在 PHP 网站开发中，有什么实际应用？
- (2) 一般会讨论在数据库连接时，使用单例模式，请分析这种使用方法的缺陷。
- (3) 编写一个 `Trait Singleton`，使用该 `Trait` 的类，自动实现单例模式。

### 5. 设计模式 – 观察者模式

请参考课件中的内容，实现观察者模式。