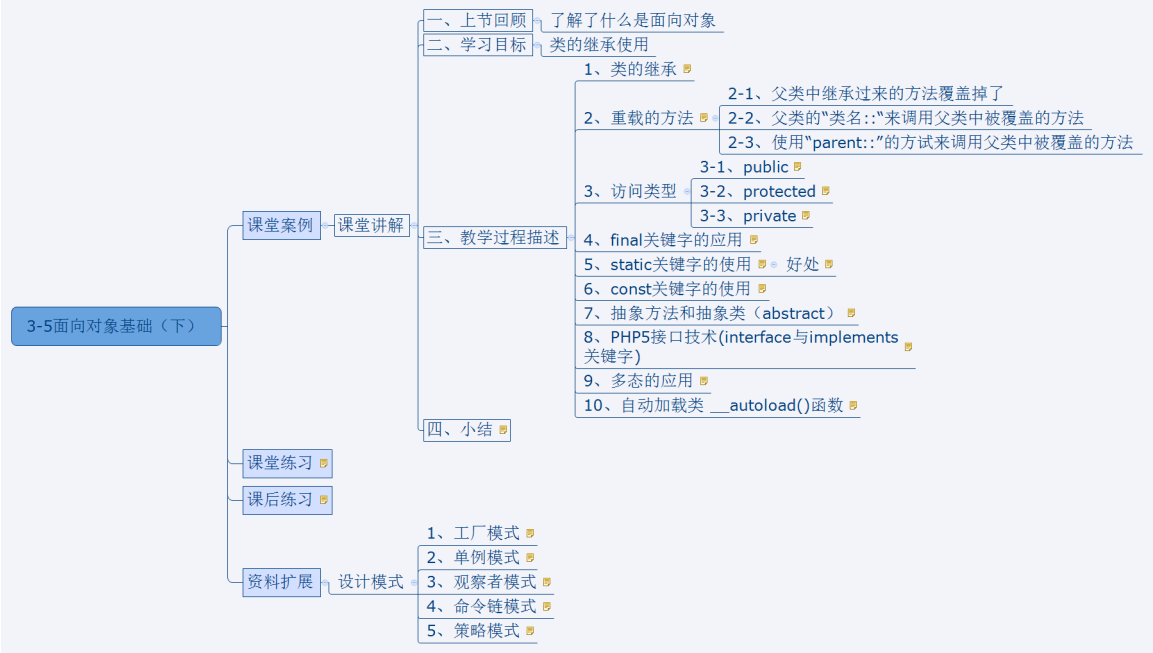
3-5面向对象基础（下）



1. 课堂案例

课堂讲解

一、上节回顾

了解了什么是面向对象

二、学习目标

类的继承使用

三、教学过程描述

1、类的继承

继承是PHP5面象对象程序设计的重要特性之一，它是指建立一个新的派生类，从一个或多个先前定义的类中继承数据和函数，而且可以重新定义或加进新数据和
函数，从而建立了类的层次或等级。说的简单点就是，继承性是子类自动共享父类数据结构和方法的机制，这是类之间的一种关系。

2、重载的方法

PHP所提供的"重载"(overloading)是指动态地"创建"类属性和方法。我们是通过魔术方法(magic methods)来实现的。

当调用当前环境下未定义或不可见的类属性或方法时，重载方法会被调用。本节后面将使用"不可访问属性(inaccessible properties)"和"不可访问方法(inaccessible methods)"来称呼这些未定义或不可见的类属性或方法。
所有的重载方法都必须被声明为 public。

2-1、父类中继承过来的方法覆盖掉了

2-2、父类的“类名::“来调用父类中被覆盖的方法

2-3、使用“parent::”的方试来调用父类中被覆盖的方法

3、访问类型

3-1、public

public
公有修饰符，类中的成员将没有访问限制，所有的外部成员都可以访问(读和写)这个类成员(包括成员属性和成员方法)，在PHP5之前的所有版本中，PHP
中类的成员都是public的，而且在PHP5中如果类的成员没有指定成员访问修饰符，将被视为public

3-2、protected

protected保护成员修饰符，被修饰为protected的成员不能被该类的外部代码访问。但是对于该类的子类有访问权限，可以进行属性、方法的读及写操作，该子类的外部代码包括其的子类都不具有访问其属性和方法的权限。

3-3、private

通过private就可以把人的成员(成员属性和成员方法)封装上了。封装上的成员就不能被类外面直接访问了，只有对象内部自己可以访问。
私有的成员是不能被外部访问的，因为私有成员只能在本对象内部自己访问

4、final关键字的应用

这个关键字只能用来定义类和定义方法，
不能使用final这个关键字来定义成员属性，因为final是常量的意思，我们在PHP里定义常量使用的是define()函数，所以不能使用final来定义成员属性。

final关键字修饰的类不能被继承，用final关键字修饰的方法不能被重写。

5、static关键字的使用

声明类属性或方法为静态，就可以不实例化类而直接访问。静态属性不能通过一个类已实例化的对象来访问（但静态方法可以）。

为了兼容 PHP 4，如果没有指定访问控制，属性和方法默认为公有。

由于静态方法不需要通过对象即可调用，所以伪变量 $this 在静态方法中不可用。

静态属性不可以由对象通过 -> 操作符来访问。

用静态方式调用一个非静态方法会导致一个 E_STRICT 级别的错误。

就像其它所有的 PHP 静态变量一样，静态属性只能被初始化为文字或常量，不能使用表达式。所以可以把静态属性初始化为整数或数组，但不能初始化为另一个变量或函数返回值，也不能指向一个对象。

自 PHP 5.3.0 起，可以用一个变量来动态调用类。但该变量的值不能为关键字 self，parent 或 static。

好处

Static关键字是在类中描述成员属性和成员方法是静态的;静态的成员好处在那里呢？前面我们声明了“Person”的人类，在”Person”
这个类里如果我们加上一个“人所属国家”的属性，这样用”Person”这个类实例化出几百个或者更多个实例对象，每个对象里面就都有“所属国家”的属性
了，如果开发的项目就是为中国人而开发的，那么每个对象里面就都有一个国家的属性是“中国“其它的属性是不同的，如果我们把“国家”的属性做成静态的成
员，这样国家的属性在内存中就只有一个，而让这几百个或更多的对象共用这一个属性，static成员能够限制外部的访问，因为static的成员是属于类
的，是不属于任何对象实例，是在类第一次被加载的时候分配的空间，其他类是无法访问的，只对类的实例共享，能一定程度对类该成员形成保护;

6、const关键字的使用

使用 const 关键字在类定义之外定义常量。一个常量一旦被定义，就不能再改变或者取消定义。
const是一个在类中定义常量的关键字，我们都知道在PHP中定义常量使用的是”define()”这个函数，但是在类里面定义常量使用的是”const”这个关键字

const只能修饰的成员属性（常量属性），其访问方式和static修饰的成员访问的方式差不多，也是使用”类名”，在方法里面使用”self”关键字。但是不用使用”$”符号，也不能使用对象来访问。

7、抽象方法和抽象类（abstract）

抽象方法

抽象方法指没有方法体的方法，具体就是在方法声明的时候没有 {}

括弧以及其中的内容，而是直接在声明时在方法名后加上分号结束。

abstract 关键字用于定义抽象方法，语法：

abstract function function_name();

抽象类

只要一个类里面有一个方法是抽象方法，那么这个类就要定义为抽象类。抽象类同样用 abstract 关键字来定义。

抽象类不能产生实例对象，通常是将抽象方法做为子类方法重载的模板使用的，且要把继承的抽象类里的方法都实现。实际上抽象类是方便继承而引入的。

8、PHP5接口技术(interface与implements关键字)

类中接口的应用

1.关键字：interface

2.关键字：implements

1.接口的介绍与创建

接口：一种成员属性全部为抽象或常量的特殊抽象类。

规则：

1.类中全部为抽象方法。

2.抽象方法钱不用加abstract。

3.接口抽象方法属性为public。

4.成员属性必须为常量。

9、多态的应用

多态是指在面向对象中能够根据使用类的上下文来重新定义或改变类的性质和行为。
PHP不支持重载实现多态，但是PHP可以变向的实现多态效果。

10、自动加载类 __autoload()函数

在编写面向对象（OOP） 程序时，很多开发者为每个类新建一个 PHP 文件。
这会带来一个烦恼：每个脚本的开头，都需要包含（include）一个长长的列表（每个类都有个文件）。

四、小结

1、类的继承
2、静态类static

2. 课堂练习

1、掌握类的继承
2、静态类
3、关键字的使用

3. 课后练习

1、创建一个类库，类库中包括静态属性与静态方法，并调用
2、创建一个类库，类库中包括抽象类与抽象方法

4. 资料扩展

设计模式

1、工厂模式

工厂模式 是一种类，它具有为您创建对象的某些方法。您可以使用工厂类创建对象，而不直接使用new。这样，如果您想要更改所创建的对象类型，只需更改该工厂即可。使用该工厂的所有代码会自动更改。

2、单例模式

某些应用程序资源是独占的，因为有且只有一个此类型的资源。例如，通过数据库句柄到数据库的连接是独占的。您希望在应用程序中共享数据库句柄，因为在保持连接打开或关闭时，它是一种开销，在获取单个页面的过程中更是如此。

单元素模式可以满足此要求。如果应用程序每次包含且仅包含一个对象，那么这个对象就是一个单元素（Singleton）

3、观察者模式

观察者模式为您提供了避免组件之间紧密耦合的另一种方法。该模式非常简单：一个对象通过添加一个方法（该方法允许另一个对象，即观察者
注册自己）使本身变得可观察。当可观察的对象更改时，它会将消息发送到已注册的观察者。这些观察者使用该信息执行的操作与可观察的对象无关。结果是对象可以相互对话，而不必了解原因。

4、命令链模式

命令链
模式以松散耦合主题为基础，发送消息、命令和请求，或通过一组处理程序发送任意内容。每个处理程序都会自行判断自己能否处理请求。如果可以，该请求被处理，进程停止。您可以为系统添加或移除处理程序，而不影响其他处理程序。

5、策略模式

我们讲述的最后一个设计模式是策略
模式。在此模式中，算法是从复杂类提取的，因而可以方便地替换。例如，如果要更改搜索引擎中排列页的方法，则策略模式是一个不错的选择。思考一下搜索引擎的几个部分——
一部分遍历页面，一部分对每页排列，另一部分基于排列的结果排序。在复杂的示例中，这些部分都在同一个类中。通过使用策略模式，您可将排列部分放入另一个类中，以便更改页排列的方式，而不影响搜索引擎的其余代码。