《深入PHP 面向对象、模式与实践》学习笔记

面向对象：

1、类，对象，protected和private只能在类里面访问

2、构造函数：\_\_contruct

3、静态方法，静态属性：类::静态方法

4、对象里面用$this->，类里面用self::

5、工厂模式：接受原始数据（比如一列数据或配置信息），然后据此产生对象。

比如在类中定义一个静态方法getInstance(id,pdo)，通过根据id查询数据库活的信息，然后选择生成什么类型的对象。

20160522-P43

抽象类：不能直接被实例化。只定义子类需要的方法，子类可以继承它并且通过实现其中的抽象方法。

接口：像抽象类，但方法体为空

选择类：让一个类只有一个主要的职责，并且任务要尽可能独立。你可以把类的职责用多个词来形容，最好不超过25个词，不要用到词“且”或者“或”。如果句子太长或者有复杂的子句，应该考虑用你所描述的一部分任务来定义新类

多态：允许将子类类型的指针赋值给父类类型的指针

20160601

第8章

策略模式

1. **生成对象**

把对象的实例化放到一个静态函数里，在静态函数里再选择调用某个子类实例化对象。这样实现了对外的统一借口，并且对象的实例化封装在接口内部。比如getInstance()，在里面读取配置，然后if else判断选择生成对象。

**9.2单例模式**

不用全局变量，用单例对象

优点：改进的系统的设计，不用在系统中传递不必要的对象。

**分析：**

主要优点：

1、提供了对唯一实例的受控访问。

2、由于在系统内存中只存在一个对象，因此可以节约系统资源，对于一些需要频繁创建和销毁的对象单例模式无疑可以提高系统的性能。

3、允许可变数目的实例。

主要缺点：

1、由于单利模式中没有抽象层，因此单例类的扩展有很大的困难。

2、单例类的职责过重，在一定程度上违背了“单一职责原则”。

3、滥用单例将带来一些负面问题，如为了节省资源将数据库连接池对象设计为的单例类，可能会导致共享连接池对象的程序过多而出现连接池溢出；如果实例化的对象长时间不被利用，系统会认为是垃圾而被回收，这将导致对象状态的丢失。

**9.3工厂方法模式**

“抽象类高于实现”=尽量一般化而不是特殊化。用特定的类来处理实例化。简单工厂模式：1个工厂做所有的事，工厂方法模式：工厂的职能太多，把大工厂分解成多个小工厂，每个小工厂负责具体的产品加工。

20160603