

NEW PLAN

软考高级架构师

• — 段 新 征 程 •





大纲介绍

面向对象设计

面向对象相关概念

UML

用例图

类图

顺序图

设计模式

创建型

结构型

行为型

考试方式：上午选择题会有12分，下午一般会出2道大题，30分



设计模式

每一个设计模式描述了一个在我们周围不断重复发生的问题，以及该问题的解决方案的核心。这样，你就能一次又一次地使用该方案而不必做重复劳动。设计模式的核心在于提供了相关问题的解决方案，使得人们可以更加简单方便的复用成功的设计和体系结构。

设计模式的四个基本要素：模式名称、问题(应该在何时使用模式)、解决方案(设计的内容)、效果(模式应用的效果)。

分为三类，创建型模式主要是处理创建对象，结构型模式主要是处理类和对象的组合，行为型模式主要是描述类或者对象的交互行为，具体如下(红色粗体记忆关键字表示常考必须记住的):



设计模式



架构模式:软件设计中的**高层决策**，例如C/S结构就属于架构模式，架构模式反映了开发软件系统过程中所作的基本设计决策。

设计模式:每一个设计模式描述了一个在我们周围**不断重复发生的问题，以及该问题的解决方案的核心**。这样，你就能一次又一次地使用该方案而不必做重复劳动。设计模式的核心在于提供了相关问题的解决方案，使得人们可以更加简单方便的复用成功的的设计和体系结构。四个基本要素:**模式名称、问题（应该在何时使用模式）、解决方案（设计的内容）、效果（模式应用的效果）**。

惯用法:是最低层的模式，关注**软件系统的设计与实现，实现时通过某种特定的程序设计语言**来描述构件与构件之间的关系。每种编程语言都有它自己特定的模式，即语言的惯用法。例如引用一计数就是C++语言中的一种惯用法。



设计模式-创建型

工厂模式（Factory Pattern）：

工厂模式就像是一家披萨店。你告诉披萨店你想要什么类型的披萨，它会根据你的需求为你制作披萨。在这里，披萨店就是一个工厂，披萨是工厂创建的产品。

生活中的例子：你在购物时可以使用工厂模式，告诉工厂你需要哪种类型的衣服，然后工厂可以根据你的需求制作并提供相应的衣服。

单例模式（Singleton Pattern）：

单例模式确保一个类只有一个实例。这就像一座城市只有一个市长，无论多少人住在这座城市，市长都是唯一的。

生活中的例子：操作系统的任务管理器就是一个单例模式的应用。无论打开多少个任务管理器窗口，它们都指向同一个任务管理器实例。

建造者模式（Builder Pattern）：

建造者模式像是建造一座房子。你可以选择房子的类型、颜色、材料等，并由建筑工人按照你的选择来构建房子。

假设你是一位工程师，有一个木制房子构建者和一个砖瓦房子构建者。你的任务是根据客户的要求建造不同类型的房子。

客户A需要一座木制房子，你将选择木制房子构建者，然后使用它的方法来逐步构建木制房子，包括设置木质墙壁、建造屋顶和安装窗户。客户B需要一座砖瓦房子，你将选择砖瓦房子构建者，并使用它的方法来逐步构建砖瓦房子，包括砖瓦墙壁、瓦片屋顶和大窗户。

通过构建者模式，你可以根据客户的需求，逐步构建不同类型的房子，而不必亲自处理所有细节。这使得构建过程灵活且可维护，同时允许创建不同风格和材质的房子。



设计模式-创建型

原型模式 (Prototype Pattern) :

原型模式通过复制现有对象来创建新对象。这就像使用3D打印机复制一件艺术品或零件，从一个原型创建多个相同的物品。

生活中的例子： 在电影制作中，制作人可以使用原型模式来复制特效道具，以便在不同场景中使用多个相同的道具。



设计模式-创建型

速记口诀：单抽元件厂

创建型设计模式	定义	记忆关键字
Abstract Factory 抽象工厂模式	提供一个接口，可以创建一系列相关或相互依赖的对象，而无需指定它们具体的类	抽象接口
Builder 构建器模式	将一个复杂类的表示与其构造相分离，使得相同的构建过程能够得出不同的表示	类和构造分离
Factory Method 工厂方法模式	定义一个创建对象的接口，但由子类决定需要实例化哪一个类。使得子类实例化过程推迟	子类决定实例化
Prototype 原型模式	用原型实例指定创建对象的类型，并且通过拷贝这个原型来创建新的对象	原型实例，拷贝
Singleton 单例模式	保证一个类只有一个实例，并提供一个访问它的全局访问点	唯一实例

T H E E N D

功不唐捐，玉汝于成！

开 启 新 征 程

