

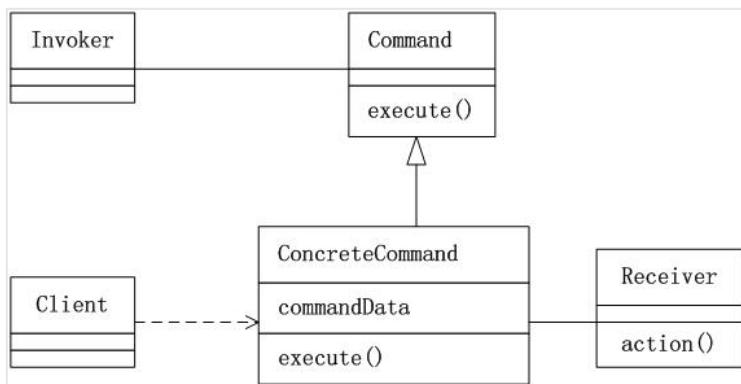
第 10 章设计模式

10.1 设计模式概述

10.1.3 GoF 设计模式

1. 【2009 年题 24 解析】

Command（命令）模式是设计模式中行为模式的一种，它将“请求”封装成对象，以便使用不同的请求、队列或者日志来参数化其他对象。Command 模式也支持可撤销的操作。Command 模式的类图如下所示。



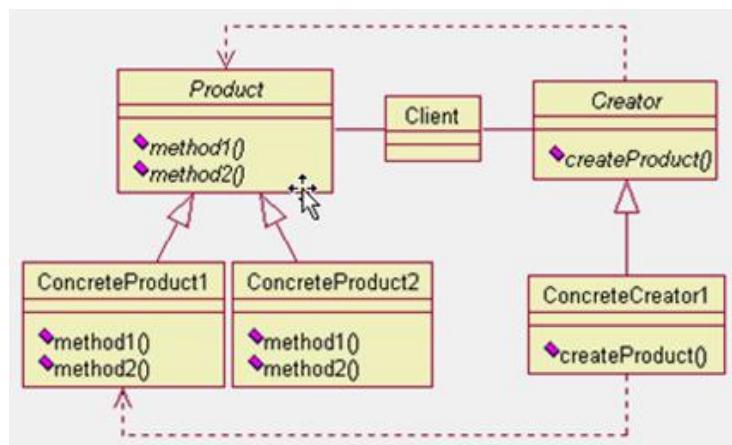
对于题目所给出的图，与“Invoker”角色相对应的类是 MenuItem，与“Concrete Command”角色相对应的类是 Open。

2. 【2009 年题 47 解析】

根据题干描述，可以看出其基础是一个图形界面，并要求为图形界面提供一些定制的特效，例如带滚动条的图形界面，能够显示艺术字体且透明的图形界面等。这要求能够动态地对一个对象进行功能上的扩展，也可以对其子类进行功能上的扩展。对照选项中的 4 种设计模式，装饰模式最符合这一要求。

3. 【2010 年题 29 解析】

Factory Method 模式的意图是，定义一个用于创建对象的接口，让子类决定实例化哪一个类。Factory Method 是一个类的实例化延迟到其子类。Factory Method 模式的类图如下图所示。



其中，类 Product 定义了 Factory Method 所创建的对象的接口；

类 ConcreteProduct 用于实现 Product 接口；

类 Creator 声明了工厂方法，该方法返回一个 Product 类型的对象。Creator 也可以定义一个工厂方法的缺省实现，它返回一个缺省的 ConcreteProduct 对象。

类 ConcreteCreator 重定义了工厂方法，以返回一个 ConcreteProduct 实例。

对照两张类图可以看出，与“Creator”角色相对应的类是 Bank；与“Product”角色相对应的类是 Accout。

【答案】A、B。

4. 【2010 年题 49 解析】

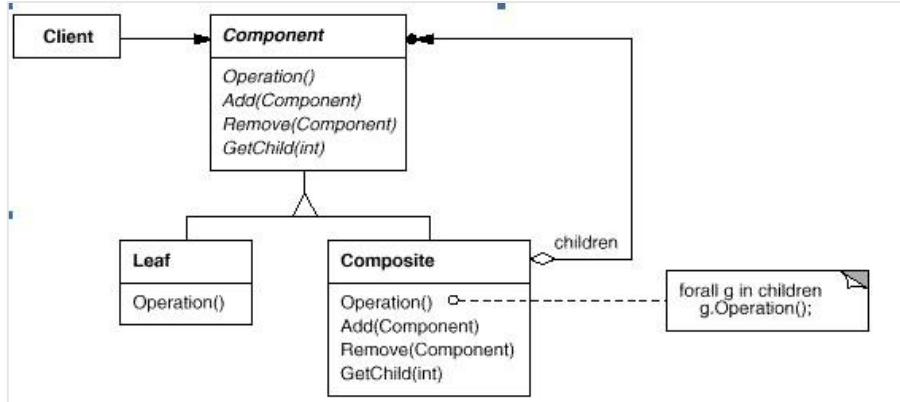
本题主要考查设计模式的理解与应用。根据题干描述，应用系统需要使用某公司开发的类库，该应用系统是一组窗格组成，应用需要协调窗格之间的行为，并且不能引用窗格自身，在这种要求下，对比 4 个候选项，其中中介者模式用一个中介对象封装一系列的对象交互。中介者使用各对象不需要显式的相互调用，从而使其耦合松散。可以看出该模式最符合需求。

5. 【2010 年题 50 解析】

本题主要考查设计模式的理解与应用。根据题干描述，该编辑器需要在文档中嵌入显示开销很大的图形对象，为了能够提高系统效率，需要避免同时创建这些图像。这对这些要求，对比候选项，可以发现代理模式可以解决直接访问对象时带来的问题，例如：要访问的对象在远程的机器上；对象创建开销很大，或者某些操作需要安全控制，或者需要进程外的访问等。因此代理模式是最为合适的设计模式。

6. 【2011 年题 25 解析】

本题考查组合模式相关的知识。下图为组合模式的 UML 图例。与题目给出的图例进行匹配可得出答案。



【答案】A、D。

7. 【2011 年题 40 解析】

本题主要考查对设计模式的理解和掌握。根据题干描述，该系统需要能够支持不同芯片之间的数据交互，并能够独立改变芯片之间的数据交互过程。这种情况下，可以引入一个中介层，通过中介层屏蔽不同芯片之间的两两交互。根据上述分析，选项中列举的设计模式中，中介者模式最符合要求。

8. 【2011 年题 41 解析】

根据题干描述，系统需要支持用户在图像处理过程中的撤销和重做的动作，因此可以将用户动作封装成对象，通过对对象之间的传递和转换实现撤销和重做等动作。根据上述分析，选项中列举的设计模式中，命令模式最符合要求。

9. 【2011 年题 42 解析】

本题考查常见设计模式的特点。

Abstract Factory(抽象工厂模式)：提供一个创建一系列相关或相互依赖对象的接口，而

无需指定它们具体的类。

Chain of Responsibility: 为解除请求的发送者和接收者之间耦合，而使多个对象都有机会处理这个请求。将这些对象连成一条链，并沿着这条链传递该请求，直到有一个对象处理它。

Composite: 将对象组合成树形结构以表示“部分-整体”的层次结构。它使得客户对单个对象和复合对象的使用具有一致性。

Decorator: 动态地给一个对象添加一些额外的职责。就扩展功能而言，它比生成子类方式更为灵活。

依据题意，需要限制用户在使用聊天系统时发表不恰当言论，需要对聊天内容进行特定敏感词的过滤，最为关键的一点是需要灵活配置过滤关键字。如果本系统采用责任链模式，即可达到这一点。

10. 【2011 年题 12 解析】

本题主要考查设计模式知识。题干描述了某软件公司一款图像处理软件的需求分析与设计过程，并明确指出采用设计模式实现关键需求对系统灵活性与扩展性的要求。

针对需求1，为了支持灵活的撤销与重做等行为，采用**命令模式**最为合适，因为**命令模式**可以将一个请求封装为一个对象，从而使你可用不同的请求对客户进行参数化，还可以对请求排队，或记录请求日志，以及支持可撤销的操作。

针对需求2，为了封装图像操作与照片特征之间的复杂逻辑关系，采用**状态模式**最为合适，因为**状态模式**将每一个条件分支放入一个独立的类中，这样就可以根据对象自身的情况将对象的状态作为一个对象，这一对象可以不依赖于其他对象而独立变化；

针对需求3，为了实现图像处理算法的灵活选择与替换，采用**策略模式**最为合适，因为**策略模式**定义一系列的算法，把它们封装起来，并且使它们可相互替换，使得算法可独立于使用它的客户而变化。

【答案】D、A、C。

11. 【2011 年题 13 解析】

外观(facade)模式是对象的结构模式，要求外部与一个子系统的通信必须通过一个统一的外观对象进行，为子系统中的一组接口提供一个一致的界面，外观模式定义了一个高层接口，这个接口使得这一子系统更加容易使用。

【答案】A、B。

扩展: 这个题本身出题有问题，这个场景最合适的，其实是模板方法，因为固定了流程但没有固定里面的内容。但给出的选项中，没有这个选项，所以已然没有最合适的了。也就这个原因才选到 A。其实如果说外观也算能行，用桥接也是可以的。把过程作为抽象，把里面要处理的内容作为实现部分。

12. 【2013 年题 24 解析】

装饰模式: 动态地给一个对象添加一些额外的职责。它提供了用子类扩展功能的一个灵活的替代，比派生一个子类更加灵活。

在本题中，“现需要构造带有滚动条或者带有黑色边框，或者既有滚动条又有黑色边框的文本显示控件和图片显示控件”，从此处可以看出需要能为构件灵活附加功能的机制，这与装饰模式的情况是吻合的。这样做比静态继承具有更大的灵活性。

13. 【2014 年题 22 解析】

解释器(interpreter)模式。解释器模式属于类的行为型模式，描述了如何为语言定义一个文法，如何在该语言中表示一个句子，以及如何解释这些句子，这里的“语言”是使用规定格式和语法规则的代码。**解释器模式主要用在编译器中，在应用系统开发中很少用到**。

策略(strategy)模式。策略模式是一种对象的行为型模式，定义一系列算法，并将每一个算法封装起来，并让它们可以相互替换。策略模式让算法独立于使用它的客户而变化，其目的是**将行为和环境分隔**，当出现新的行为时，只需要实现新的策略类。

中介者(mediator)模式。中介者模式是一种对象的行为型模式，通过一个中介对象来封

装一系列的对象交互。中介者使得各对象不需要显式地相互引用，从而使其耦合松散，而且可以独立地改变它们之间的交互。中介者对象的存在保证了对象结构上的稳定，也就是说，系统的结构不会因为新对象的引入带来大量的修改工作。

迭代器(iterator)模式。迭代器模式是一种对象的行为型模式，提供了一种方法来访问聚合对象，而不用暴露这个对象的内部表示。迭代器模式支持以不同的方式遍历一个聚合对象，复杂的聚合可用多种方法来进行遍历；允许在同一个聚合上可以有多个遍历，每个迭代器保持它自己的遍历状态，因此，可以同时进行多个遍历操作。

扩展：设计模式分类：创建型模式、结构型模式、行为型模式。

14. 【2014 年题 23 解析】

本题考点是设计模式，不同的设计模式可以应用于不同的场景，在本题题干部分提到宣传产品有多种表现形式，又有多种媒介，如果用一棵类树来表达，必然会带来“类爆炸”(题目中增加一种媒介，代码实现中需要增加多个类)的问题，所以使用桥接模式是合适的。桥接模式的最核心特点便是：将抽象部分与它的实现部分分离，使它们都可以独立地变化。

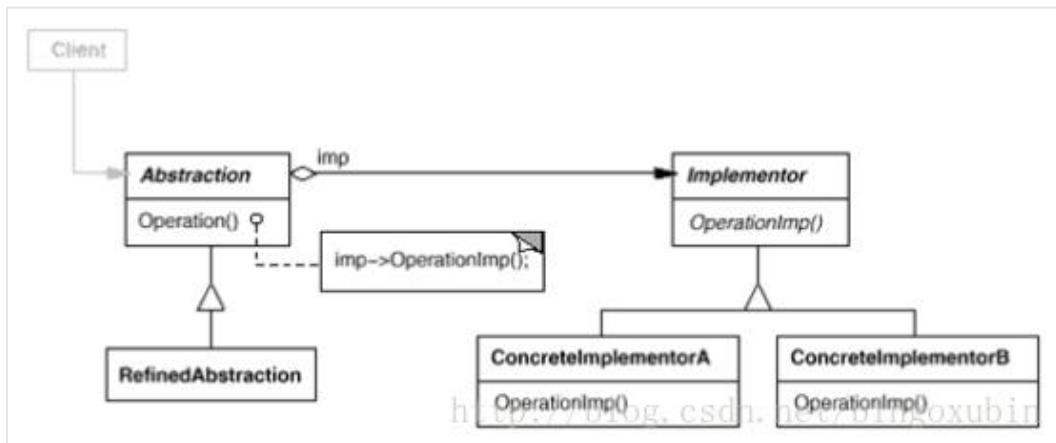
15. 【2014 年题 33 解析】

针对题目给出的情况，公司的架构师决定采用“包装器外观”架构模式解决操作系统的差异问题。具体来说，服务端程序应该在包装器外观的实例上调用需要的方法，然后将请求和请求的参数发送给操作系统 API 函数，调用成功后将结果返回。使用该模式提高了底层代码访问的一致性，但降低了服务端程序的调用性能。

16. 【2015 年题 24 解析】

桥接模式将抽象部分与它的实现部分分离，使它们都可以独立地变化。它是一种对象结构型模式，又称为柄体(Handle and Body)模式或接口(Interface)模式。桥接模式类似于多重继承方案，但是多重继承方案往往违背了类的单一职责原则，其复用性比较差，桥接模式是比多重继承方案更好的解决方法。

桥接模式的结构如下图所示，其中：



图中与 Bridge 模式中的“Abstraction”角色相对应的类是 Shape，与“Implementor”角色相对应的类是 Drawing。

【答案】：A、B。

17. 【2015 年题 36 解析】

设计模式包括：创建型、结构型、行为型三大类别。

Singleton 是单例模式，属于创建型设计模式。

Memento 是备忘录模式，属于行为型设计模式。

Bridge 是桥接模式，它的特点是实现接口与实现分离。

表 10-1 GoF 模式分类

GoF 模式				
		创建型	结构型	行为型
应用范围	应用于类	Factory Method	Adapter	Interpreter Template Method
	应用于对象	Abstract Factory	Adapter	Chain of Responsibility
		Builder	Bridge	Command
		Prototype	Composite	Iterator
		Singleton	Decorator	Mediator
			Facade	Memento
			Flyweight	Observer
			Proxy	State
				Strategy
				Visitor

【答案】：C、D、C、D。

18. 【2018 年题 36 解析】

抽象工厂模式是一种类创建型模式。

桥接模式(Bridge)-将抽象部分与它的实现部分分离,使它们都可以独立地变化,它是一种对象结构型模式。

命令 (Command) 模式将一个请求封装为一个对象, 从而可用不同的请求对客户进行参数化, 将请求排队或记录请求日志, 支持可撤销的操作。

答案 DABA。

10.1.6 设计模式分类

1. 【2015 年题 38 解析】

设计模式包括：创建型、结构型、行为型三大类别。

Singleton 是单例模式，属于创建型设计模式。

Adapter 是适配器模式，属于结构型设计模式。

Visitor 是访问者模式，属于行为型设计模式。

【答案】D、C、A、D。

10.2 设计模式及实现

10.2.6 Observer 模式

1. 【2009 年题 48 解析】

根据题干描述, 可以看出本题的核心在于对某个具有固定结构的活动节点需要多种处理能力, 且处理能力可扩展, 也就是说要求在不改变原来类结构 (活动节点) 的基础上增加新功能。对照 4 个选项, 发现访问者模式最符合要求。