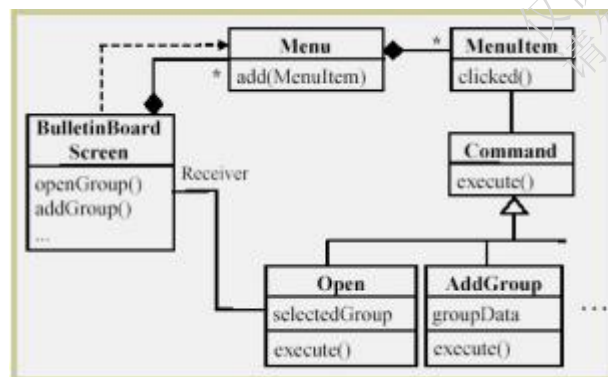


## 第 10 章 设计模式

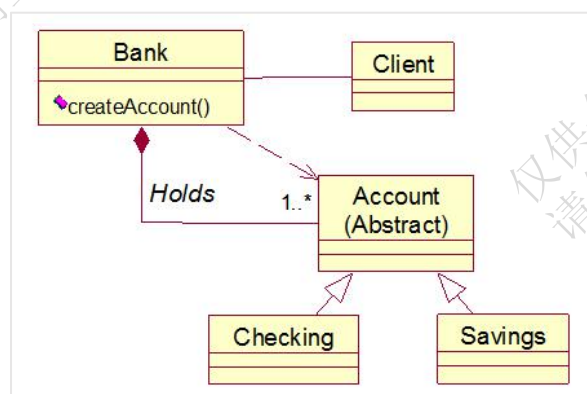
### 10.1 设计模式概述

#### 10.1.3 GoF 设计模式

1. 【2009 年题 24】某软件公司欲开发一个 Windows 平台上的公告板系统。在明确用户需求后，该公司的架构师决定采用 Command 模式实现该系统的界面显示部分，并设计 UML 类图如下图所示。图中与 Command 模式中的“Invoker”角色相对应的类是( )，与“ConcreteCommand”角色相对应的类是( )。



- A. Command                      B. MenuItem                      C. Open                      D. ButktinBoardScreen  
 A. Command                      B. MenuItem                      C. Open                      D. BulktinBoardScreen
2. 【2009 年题 27】某软件公司基于面向对象技术开发了一套图形界面显示构件库 VisualComponent。在使用该库构建某图形界面时，用户要求为界面定制一些特效显示效果，如带滚动条、能够显示艺术字体的透明窗体等。针对这种需求，公司采用( )最为灵活。
- A. 桥接模式  
 B. 命令模式  
 C. 组合模式  
 D. 装饰模式
3. 【2010 年题 29】某银行系统采用 Factory Method 方法描述其不同账户之间的关系，设计出的类图如下所示。其中与 Factory Method 中的“Creator”角色相对应的类是( )；与“Product”角色相对应的类是( )。

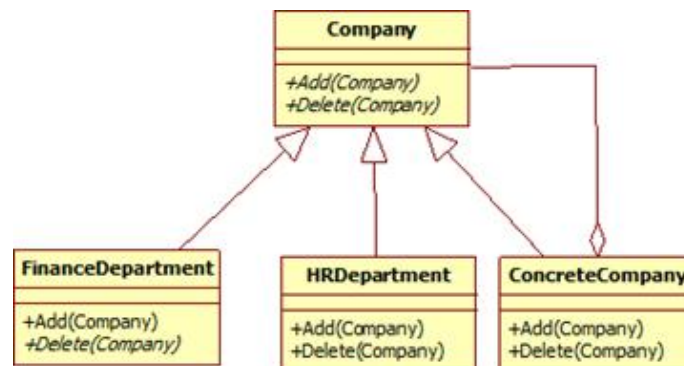


- A. Bank                      B. Account                      C. Checking                      D. Savings  
 A. Bank                      B. Account                      C. Checking                      D. Savings

4. 【2010 年题 49】某公司欲开发一套窗体图形界面类库。该类库需要包含若干预定义的窗格(Pane)对象,例如 TextPane、ListPane 等,窗格之间不允许直接引用。基于该类库的应用由一个包含一组窗格的窗口组成,并需要协调窗格之间的行为。基于该类库,在不引用窗格的前提下实现窗格之间的协作,应用开发者应采用( )最为合适。
- A. 备忘录模式  
 B. 中介者模式  
 C. 访问者模式  
 D. 迭代器模式

5. 【2010 年题 50】某公司开发一个文档编辑器,该编辑器允许在文档中直接嵌入图形对象,但开销很大。用户在系统设计之初提出编辑器在打开文档时必须十分迅速,可以暂时不显示当前页面以外的图形。针对这种需求,公司可以采用( )避免同时创建这些图形对象。
- A. 代理模式  
 B. 外观模式  
 C. 桥接模式  
 D. 组合模式

6. 【2011 年 25】某公司欲开发一门户网站,将公司的各个分公司及办事处信息进行整合。现决定采用 composite 设计模式来实现公司的组织结构关系,并设计了如图所示的 UML 类图。图中与 Composite 模式中的“Component”角色相对应的类是(1),与“Composite”角色相对应的类是(2)。



- (1) A. Company  
 B. Finance Department  
 C. HRDepartment  
 D. ConcreteCompany

- (2) A. Company  
 B. Finance Department  
 C. HRDepartment  
 D. ConcreteCompany

7. 【2011 年题 40】某软件公司正在设计一个通用的嵌入式数据处理平台, 需要支持各种数据处理芯片之间的数据传递与交换。该平台的核心功能之一要求能够屏蔽芯片之间的数据交互, 使其耦合松散, 并且可以独立改变芯片之间的交互过程。针对上述需求, 采用( )最为合适。
- A. 抽象工厂模式
  - B. 策略模式
  - C. 中介者模式
  - D. 状态模式

8. 【2011 年题 41】某软件公司正在设计一个图像处理软件, 该软件需要支持用户在图像处理过程中的撤销和重做等动作, 为了实现该功能, 采用( )最为合适。
- A. 单例模式
  - B. 命令模式
  - C. 访问者模式
  - D. 适配器模式

9. 【2011 年题 42】某互联网公司正在设计一套网络聊天系统, 为了限制用户在使用该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。针对上述功能需求, 采用\_\_\_\_\_能够灵活配置敏感词的过滤过程。
- A. 责任链模式
  - B. 工厂模式
  - C. 组合模式
  - D. 装饰模式

10. 【2012 年题 12】某软件公司欲设计一款图像处理软件, 帮助用户对拍摄的照片进行后期处理。在软件需求分析阶段, 公司的系统分析师识别出了如下 3 个关键需求。

图像处理软件需要记录用户在处理照片时所有动作, 并能够支持用户动作的撤销与重做等行为。

图像处理软件需要根据当前正在处理的照片的不同特征选择合适的处理操作, 处理操作与照片特征之间具有较为复杂的逻辑关系。

图像处理软件需要封装各种图像处理算法, 用户能够根据需要灵活选择合适的处理算法; 软件还要支持高级用户根据一定的规则添加自定义处理算法。

在系统设计阶段, 公司的架构师决定采用设计模式满足上述关键需求中对系统灵活性与扩展性的要求。具体来说, 为了支持灵活的撤销与重做等行为, 采用( )最为合适; 为了封装图像操作与照片特征之间的复杂逻辑关系, 采用( )最为合适; 为了实现图像处理算法的灵活选择与替换, 采用( )最为合适。

- |           |          |          |          |
|-----------|----------|----------|----------|
| A. 工厂模式   | B. 责任链模式 | C. 中介者模式 | D. 命令模式  |
| A. 状态模式   | B. 适配器模式 | C. 组合模式  | D. 单例模式  |
| A. 模板方法模式 | B. 访问者模式 | C. 策略模式  | D. 观察者模式 |

11. 【2012 年题 13】若系统中的某子模块需要为其他模块提供访问不同数据库系统的功能, 这些数据库系统提供的访问接口有一定的差异, 但访问过程却都是相同的, 例如, 先连接数据库, 再打开数据库, 最后对数据进行查询。针对上述需求, 可以采用( )设计模式抽象出相同的数据库访问过程, 该设计模式( )。

- (1) A. 外观      B. 装饰      C. 桥接      D. 享元

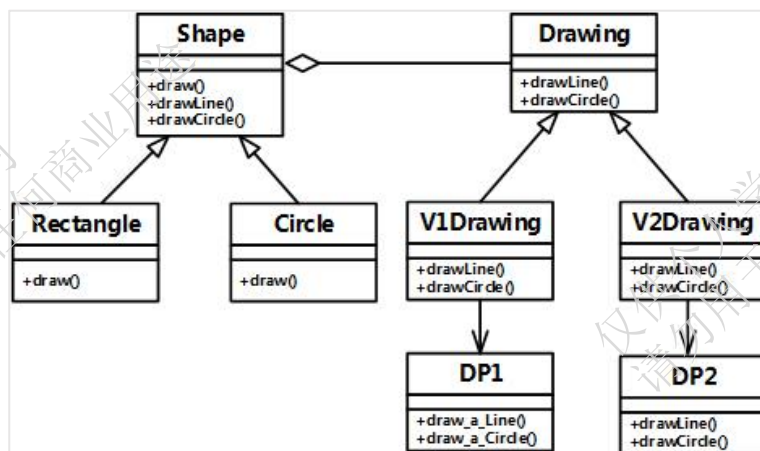
- (2) A. 可以动态、透明地给单个对象添加职责

B. 为子系统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

C. 通过运用共享技术, 有效支持大量细粒度的对象

- D.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
12. 【2013 年题 24】某系统中的文本显示类(TextView)和图片显示类(PictureView)都继承了组件类(Component),分别显示文本和图片内容,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,但希望最多只增加 3 个类。那么采用设计模式( )可实现该需求,其优点是( )。
- (1)A. 外观      B. 单体      C. 装饰      D. 模板方法
- (2)A. 比静态继承具有更大的灵活性  
B. 提高已有功能的重复使用性  
C. 可以将接口与实现相分离  
D. 为复杂系统提供了简单接口
13. 【2014 年题 22】一组对象以定义良好但是复杂的方式进行通信,产生的相互依赖关系结构混乱且难以理解。采用( )模式,用一个特定对象来封装一系列的对象交互,从而使各对象不需要显式地相互引用,使其耦合松散,而且可以独立地改变它们之间的交互。
- A. 解释器(Interpreter)  
B. 策略(Strategy)  
C. 中介者(Mediator)  
D. 迭代器(Iterator)
14. 【2014 年题 23】某广告公司的宣传产品有宣传册、文章、传单等多种形式,宣传产品的出版方式包括纸质方式、CD、DVD、在线发布等。现要求为该广告公司设计一个管理这些宣传产品的应用,采用( )设计模式较为合适,该模式( )。
- (1)A. Decorator    B. Adapter    C. Bridge    D. Facade
- (2)A. 将一系列复杂的类包装成一个简单的封闭接口  
B. 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  
C. 可在不影响其他对象的情况下,以动态、透明的方式给单个对象添加职责  
D. 将一个接口转换为客户希望的另一个接口
15. 【2014 年题 33】某公司欲开发一个用于分布式登录的服务端程序,使用面向连接的 TCP 协议并发地处理多客户端登录请求。用户要求该服务端程序运行在 Linux、Solaris 和 WindowsNT 等多种操作系统平台之上,而不同的操作系统的相关 API 函数和数据都有所不同。针对这种情况,公司的架构师决定采用“包装器外观(Wrapper Facade)”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( ),调用成功后将结果返回。使用该模式( )。
- (1)A. 客户端程序  
B. 操作系统 API 函数  
C. TCP 协议 API 函数  
D. 登录连接程序
- (2)A. 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  
B. 降低了服务端程序功能调用的灵活性,但提高了服务端程序的调用性能  
C. 降低了服务端程序的可移植性,但提高了服务端程序的可维护性  
D. 提高了系统的可复用性,但降低了系统的可配置性

16. 【2015 年题 24】某软件公司欲开发一个绘图软件，要求使用不同的绘图程序绘制不同的图形。在明确用户需求后，该公司的架构师决定采用 Bridge 模式实现该软件，并设计 UML 类图如下图所示。图中与 Bridge 模式中的“Abstraction”角色相对应的类是（ ），与“Implementor”角色相对应的类是（ ）。



- A.Shape                      B.Drawing                      C.Rectangle                      D.V2Drawing
- A.Shape                      B.Drawing                      C.Rectangle                      D.V2Drawing
17. 【2016 年题 36】设计模式基于面向对象技术，是人们在长期的开发实践中良好经验的结晶，提供了一个简单、统一的描述方法，使得人们可以复用这些软件设计办法、过程管理经验。按照设计模式的目的进行划分，现有的设计模式可以分为创建型、（ ）和行为型三种类型。其中（ ）属于创建型模式，（ ）属于行为型模式。（ ）模式可以将一个复杂的组件分成功能性抽象和内部实现两个独立的但又相关的继承层次结构，从而可以实现接口与实现分离。
- A.合成型                      B.组合型                      C.结构型                      D.聚合型
- A.Adaptor                      B.Facade                      C.Command                      D.Singleton
- A.Decorator                      B.Composite                      C.Memento                      D.Builder
- A.Prototype                      B.Flyweight                      C.Adapter                      D.Bridge
18. 【2018 年题 36】设计模式描述了一个出现在特定设计语境中的设计再现问题，并为它的解决方案提供了一个经过充分验证的通用方案，不同的设计模式关注解决不同的问题。例如，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，它是一种（54）模式；（55）模式将类的抽象部分和它的实现部分分离出来，使它们可以独立变化，它属于（56）模式；（57）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。
- （54）A. 组合型  
B. 结构型  
C. 行为型  
D. 创建型
- （55）A. Bridge  
B. Proxy  
C. Prototype  
D. Adapter
- （56）A. 组合型  
B. 结构型  
C. 行为型



- D. 创建型
- (57) A. Command  
B. Facade  
C. Memento  
D. Visitor
19. 【2019 年题 31】设计模式按照目的可以划分为三类，其中，( )模式是对对象实例化过程的抽象。例如( )模式确保一个类只有一个实例，并提供了全局访问入口；( )模式允许对象在不了解要创建对象的确切类以及如何创建等细节的情况下创建自定义对象；( )模式将复杂对象的构建与其表示分离。
- A. 创建型  
B. 结构型  
C. 行为型  
D. 功能型
- (2) A. Facade  
B. Builder  
C. Prototype  
D. Singleton
- (3) A. Facade  
B. Builder  
C. Prototype  
D. Singleton
- (4) A. Facade  
B. Builder  
C. Prototype  
D. Singleton

### 10.1.6 设计模式分类

1. 【2017 年题 38】按照设计模式的目的进行划分，现有的设计模式可以分为三类。其中创建型模式通过采用抽象类所定义的接口，封装了系统中对象如何创建、组合等信息，其代表有( )模式等；( )模式主要用于如何组合已有的类和对象以获得更大的结构，其代表有 Adapter 模式等；( )模式主要用于对象之间的职责及其提供服务的分配方式，其代表有( )模式等。

A.Decorator	B.Flyweight	C.Command	D.Singleton
A.合成型	B.组合型	C.结构型	D.聚合型
A.行为型	B.交互型	C.耦合性	D.关联型
A.Prototype	B.Facade	C.Proxy	D.Visitor

## 10.2 设计模式及实现

### 10.2.6 Observer 模式

1. 【2009 年题 48】某软件公司承接了为某 workflow 语言开发解释器的工作。该 workflow 语言由多种活动节点构成，具有类 XML 的语法结构。用户要求解释器工作时，对每个活动节点进行一系列的处理，包括执行活动、日志记录、调用外部应用程序等，并且要求处理过程具有可扩展能力。针对这种需求，公司采用( )最为恰当。
  - A. 适配器模式
  - B. 迭代器模式
  - C. 访问者模式
  - D. 观察者模式