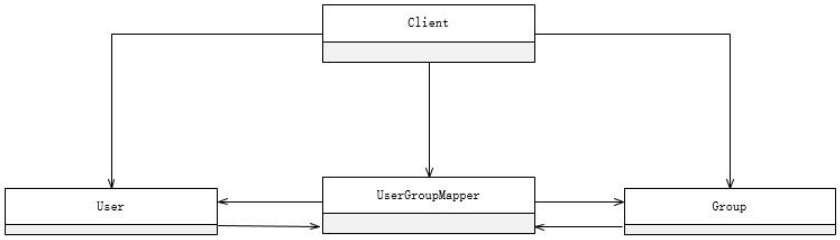
2022 知识点练习->上午->8.面向对象

1.在设计某购物中心的收银软件系统时，要求能够支持在不同时期推出打折、返利、满减等不同促销活动，则适合采用（）模式。

A.策略(Strategy)　　B.访问者(Visitor)　　C.观察者(Observer)　D.中介者(Mediator)

2.在某系统中，不同组（GROUP）访问数据的权限不同，每个用户（User）可以是一个或多个组中的成员，每个组包含零个或多个用户。现要求在用户和组之间设计映射，将用户和组之间的关系由映射进行维护，得到如下所示的类图。该设计采用(　)模式，用一个对象来封装系列的对象交互；使用户对象和组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。该模式属于(　)模式，该模式适用（　）。



问题 1：A.状态(State)　B.策略(Strategy)　C.解释器(Interpreter)　D.中介者(Mediator)

问题 2：A.创建型类　　B.创建型对象　　C.行为型对象　　D.行为型类

问题 3：

A.需要使用一个算法的不同变体

B.有一个语言需要解释执行，并且可将句子表示为一个抽象语法树

C.一个对象的行为决定于其状态且必须在运行时刻根据状态改变行为

D.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

3.某软件系统限定：用户登录失败的次数不能超过 3 次。采用如所示的 UML 状态图对用户登录状态进行建模，假设活动状态是 Logging in,那么当 Valid Entry 发生时，(　)。 其中，[tries<3]和 tries+ + 分别为(　)和(　)。

问题 1：A.状态(State)　B.策略(Strategy)　C.解释器(Interpreter)　D.中介者(Mediator)

问题 2：

A.保持在 Logging in 状态

B.若[tries<3]为 true，则 Logged in 变为下一个活动状态

C.Logged in 立刻变为下一 个活动状态

D.若 tries=3 为 true，则 Logging Denied 变为下一个活动状态

问题 3：A.状态　　B.转换　　C.监护条件　　D.转换后效果

4.某电商系统在采用面向对象方法进行设计时，识别出网店、商品、购物车、订单买家、库存、支付（微信、支付宝）等类。其中，购物车与商品之间适合采用（　）关系，网店与商品之间适合采用（　）关系。

问题 1：A.关联　　B.依赖　C.组合　　D.聚合

问题 2：A.关联　　B.依赖　C.组合　　D.聚合

5.在面向对象设计时，如果重用了包中的一个类，那么就要重用包中的所有类，这属于（）原则。

A.接口分离　　B.开放-封闭　C.共同封闭　　D.共同重用

6.面向对象设计时包含的主要活动是（　）。

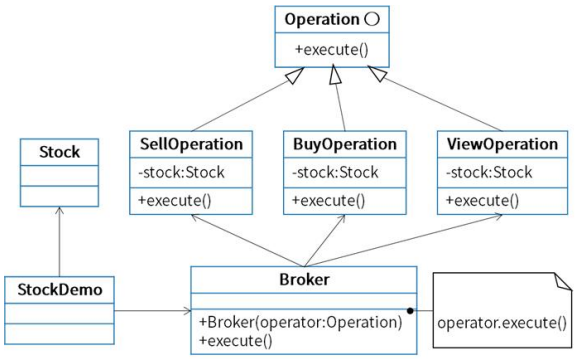
A.认定对象、组织对象、描述对象间的相互作用、确定对象的操作

B.认定对象、定义属性、组织对象、确定对象的操作

C.识别类及对象、确定对象的操作、描述对象间的相互作用、识别关系

D.识别类及对象、定义属性、定义服务、识别关系、识别包

7.股票交易中，股票代理(Broker)根据客户发出的股票操作指示进行股票的买卖操作，设计如下所示类图。该设计采用（　）模式将一个请求封装为一个对象，从而使得以用不同的请求对客户进行参数化；对请求排队或记录请求日志，以及支持可撤销的操作，其中，（　）声明执行操作的接口。该模式属于（　）模式，该模式适用于：（　）。



问题 1：A.命令（Command）　B.观察者（Observer）　C.状态（State）　D.中介者（Mediator）

问题 2：A.Operation　B.sellOperation/BayOperation/ViewOperation　C.Broker　D.Stock

问题 3：A.结构类型　　B.结构型对象　　C.创建类型　　D.行为型对象

问题 4：

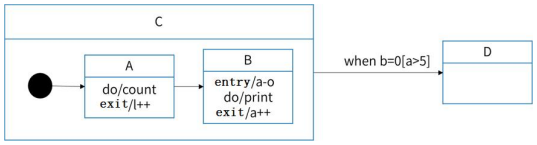
A.一个对象必须通知其他对象，而它又不能假定其他对象是谁

B.抽象出特执行的动作以参数化某对象

C.一个对象的行为决定于其状态且必须在运行时刻根据状态改变行为

D.一个对象引用其他很多对象并且直接与这些对象通信，导致难以复用该对象

8.当 UML 状态图用于对系统、类或用例的动态方面建模时，通常是对(　)建模。以下 UML 状态图中，假设活动的状态是 A，事件 b=0 发生并且 a>5，发生条件是 c 状态到 d 状态的转换条件的是（　），D 变为活动的状态，有关状态图的叙述中，不正确的是（　）。



问题 1：A.系统的词汇　　B.反应型对象　　C.活动流程　　D.对象快照

问题 2：A.一旦状态 A 的 exit 动作完成，或如果当前执行 do 动作，则终止执行

B.一旦状态 A 和 B 的所有动作完成　C.一旦正在进行的状态 A 完成　D.一旦状态 B 的 exit 动作完成

问题 3：A.动作可以在状态内执行，也可以在状态转换时执行

B.当触发转换的事件发生并且转换没有指定的监护条件时，对象将离开当前状态，并且其 do 动作终止

C.when（b=5）称为时间事件　　D.状态由事件触发

9.假设 Bird 和 Cat 是 Animal 的子类,Parrot 是 Bird 的子类，bird 是 Bird 的一个对象，cat 是 Cat 的一个对象，parrot 是 Parrot 的一个对象。以下叙述中，不正确的是（ ）。

问题 1：

A.cat 和 bird 可看作是 Animal 的对象

B.parrot 和 bird 可看作是 Animal 的对象

C.bird 可以看作是 Parrot 的对象

D.parrot 可以看作是 Bird 的对象

问题 2：A.封装 B.继承 C.消息传递 D.多态

10.采用面向对象方法进行系统设计时，不应该强迫客户依赖于他们不用的方法，接口属于客户，不属于它所在的类层次结构。即：依赖于抽象，不要依赖于具体，同时在抽象级别不应该有对于细节的依赖。这属于（ ）。

A.单一责任 B.开放-封闭 C.接口分离 D.里氏替换

11.面向对象分析时，执行的活动顺序通常是（ ）。

A.认定对象、组织对象、描述对象的相互作用、确定对象的操作

B.认定对象、定义属性、组织对象、确定对象的操作

C.认定对象、描述对象间的相互作用、确定对象的操作、识别包

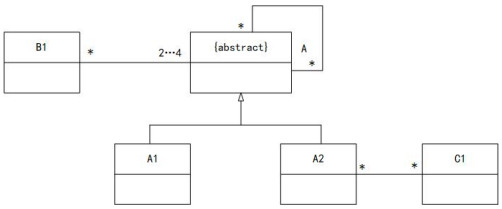
D.识别类及对象、识别关系、定义属性、确定对象的操作

12.Java 语言符合的特征有（ ）和自动的垃圾回收处理

①采用即时编译②采用静态优化编译③对象在堆空间分配④对象在栈空间分配

A.①③ B.①④ C.②③ D.②④

13.关于以下 UML 类图的叙述中，错误的是（ ）。



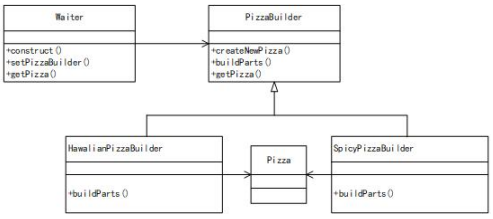
A.一个 A1 的对象可能与一个 A2 的对象关联

B.一个 A 的非直接对象可能与一个 A1 的对象关联

C.类 B1 的对象可能通过 A2 与 C1 的对象关联

D.有可能 A 的直接对象与 B1 的对象关联

14.某快餐厅主要制作并出售儿童套餐，一般包括主餐（各类比萨)、饮料和玩具，其餐品种类可能不同，但制作过程相同。前台服务员（Waiter） 调度厨师制作套餐。欲开发一软件，实现该制作过程，设计如下所示类图。该设计采用（ ）模式将一个复杂对象的构建与它的表示分离，使得同样的构建过程可以创建不同的表示。其中，（ ）构造一个使用 Builder 接口的对象。该模式属于（ ）模式，该模式适用于（ ）的情况。



问题 1：

A.生成器（Builder）　　B.抽象工厂（Abstract Factory）

C.原型（Prototype）　　D.工厂方法（Factory Method）

问题 2：A.PizzaBuilder　B.SpicyPizaBuilder　C.Waiter　D.Pizza

问题 3：A.创建型对象　B.结构型对象　C.行为型对象　D.结构型类

问题 4：

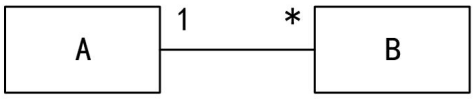
A.当一个系统应该独立于它的产品创建、构成和表示时

B.当一个类希望由它的子类来指定它所创建的对象的时候

C.当要强调一系列相关的产品对象的设计以便进行联合使用时

D.当构造过程必须允许被构造的对象有不同的表示时

15.UML 图中，对象图展现了（　），（　）所示对象图与下图所示类图不一致。



问题 1：

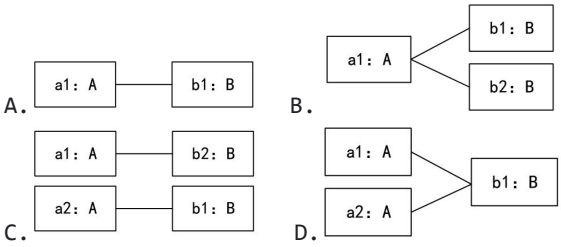
A.一组对象、接口、协作和它们之间的关系

B.一组用例、参与者以及它们之间的关系

C.某一时刻一组对象以及它们之间的关系

D.以时间顺序组织的对象之间的交互活动

问题 2：



16.多态有不同的形式，（　）的多态是指同一个名字在不同上下文中所代表的含义不同。

A.参数　　B.包含　　C.过载　　D.强制

17.进行面向对象系统设计时，针对包中的所有类对于同一类性质的变化；一个变化若对一个包产生影响，则将对该包中的所有类产生影响，而对于其他的包不造成任何影响。这属于（　）设计原则。

A.共同重用　　B.开放-封闭　　C.接口分离　　D.共同封闭

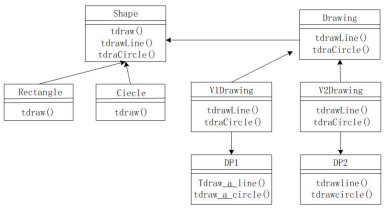
18.采用面向对象方法进行系统开发时，以下与新型冠状病毒有关的对象中，存在“一般-特殊“关系的是（　）。

A.确诊病人和治愈病人　B.确诊病人和疑似病人　C.医生和病人　D.发热病人和确诊病人

19.面向对象程序设计语言 C++、JAVA 中，关键字（ ）可以用于区分同名的对象属性和局部变量名。

A.private　　B.protected　　C.public　　D.this

20.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。该绘图软件的扩展性要求将不断扩充新的图形和新的绘图程序。以绘制直线和图形为例，得到如下图所示的类图。该设计采用（ ）模式将抽象部分与其实现部分分离，使它们都可以独立地变化。其中 （ ） 定义了实现类接口，该模式适用于 （ ） 的情况，该模式属于 （ ） 模式。



问题 1：A.适配器（Adapten）　B.装饰（Decorator）　C.桥接（Bridge）　D.组合（Composite）

问题 2：A.Shape　　B.Circle 和 Rectangle　　C.V1Drawing 和 V2Drawing　　D.Drawing

问题 3：

A.不希望在抽象和它的实现部分之间有一个固定判定关系

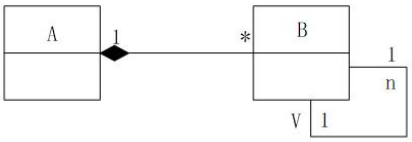
B.想表示对象的部分-整体层次结构

C.想使用一个已经存在的类，而它的接口不符合要求

D.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

问题 4：A.创建型对象　B.结构型对象　C.行为型对象　D.结构型类

21.下图所示 UML 图为（ ），有关该图的叙述中，不正确的是（ ）。



问题 1：A.对象图　　B.类图　　C.组件图　　D.部署图

问题 2：

A.如果 B 的一个实例被删除，所有包含 A 的实例都被删除

B.A 的一个实例可以与 B 的一个实例关联

C.B 的一个实例被唯一的一个 A 的实例所包含

D.B 的一个实例可与 B 的另外两个实例关联

22.以下关于 UML 状态图的叙述中，不正确的是（ ）。

A.活动可以在状态内执行，也可以在迁移时执行

B.若事件触发一个没有特定监护条件的迁移，则对象离开当前状态

C.迁移可以包含事件触发器、监护条件和状态　　D.事件触发迁移

23.（ ）绑定是指在运行时把过程调用和响应调用所需要执行的代码加以结合。

A.动态　　B.过载　　C.静态　　D.参数

24.进行面向对象系统设计时，软件实体（类、模块、函数等）应该是可以扩展但不可修改的，这属于（ ）设计原则。

A.共同重用　　B.开放封闭　　C.接口分离　　D.共同封闭

25.采用面向对象方法进行系统开发时，需要对两者之间关系创建新类的是（ ）。

A.汽车和座位　　B.主人和宠物　　C.医生和病人　　D.部门和员工

26.一个类中成员变量和成员函数有时也可以分别被称为（ ）。

A.属性和活动　　B.值和方法　　C.数据和活动　　D.属性和方法

27.观察者（Observer）模式适用于（ ）。

A.访问一个聚合对象的内容而无须暴露它的内部表示

B.减少多个对象或类之间的通信复杂性

C.将对象的状态恢复到先前的状态

D.一对多对象依赖关系，当一个对象修改后，依赖它的对象都自动得到通知

28.以下设计模式中，（ ）模式使多个对象都有机会处理请求，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为止，从而避免请求的发送者和接收者之间的耦合关系；（ ）模式提供一种方法顺序访问一个聚合对象中的各个元素， 且不需要暴露该对象的内部表示。这两种模式均为（ ）。

问题 1：

A.责任链（Chain of Responsibility）　　B.解释器（Interpreter）

C.命令（Command）　　D.迭代器（Iterator）

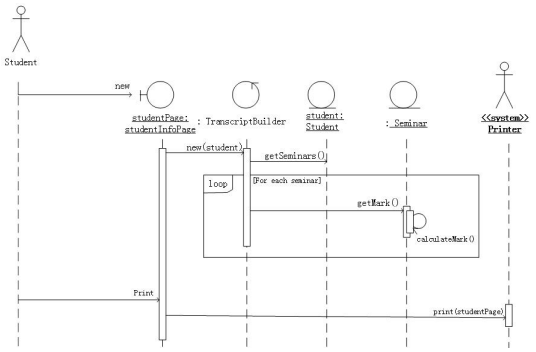
问题 2：

A.责任链（Chain of Responsibility）　　B.解释器（Interpreter）

C.命令（Command）　　D.迭代器（Iterator）

问题 3：A.创建型对象模式　　B.结构型对象模式　　C.行为型对象模式　　D.行为型类模式

29.下图所示 UML 图为（ ），用于展示系统中（ ）。



问题 1：A.用例图　　B.活动图　　C.序列图　　D.交互图

问题 2：

A.一个用例和一个对象的行为　　B.一个用例和多个对象的行为

C.多个用例和一个对象的行为　　D.多个用例和多个对象的行为

30.在 UML 图中，（ ）图用于展示所交付系统中软件组件和硬件之间的物理关系。

A.类　　B.组件　　C.通信　　D.部署

31.聚合对象是指一个对象（ ）。

A.只有静态方法　　B.只有基本类型的属性

C.包含其他对象　　D.只包含基本类型的属性和实例方法

32.进行面向对象设计时，就一个类而言，应该仅有一个引起它变化的原因，这属于（ ）设计原则。

A.单一责任　　B.开放-封闭　　C.接口分离　　D.里氏替换

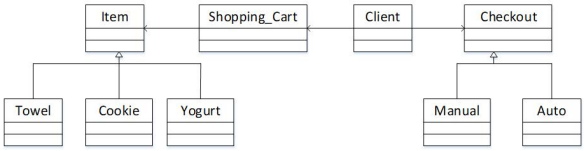
33.采用面向对象方法进行软件开发时，将汽车作为一个系统。以下（ ）之间不属于组成（Composition）关系。

A.汽车和座位　　B.汽车和车窗　　C.汽车和发动机　　D.汽车和音乐系统

34.一个类中可以拥有多个名称相同而参数表（参数类型或参数个数或参数类型顺序） 不同的方法，称为（ ）。

A.方法标记　　B.方法调用　　C.方法重载　　D.方法覆盖

35.假设现在要创建一个简单的超市销售系统，顾客将毛巾、饼干、酸奶等物品（Item）加入购物车（Shopping_Cart），在收银台（Checkout）人工（Manual）或自动（Auto）地将购物车中每个物品的价格汇总到总价格后结账。这一业务需求的类图（方法略）设计如下图所示，采用了（ ）模式。其中（ ）定义以一个 Checkout 对象为参数的 accept 操作，由子类实现此 accept 操作。此模式为（ ），适用于（ ）。



问题 1：A.观察者（Observer）　B.访问者（Visitor）　C.策略（Strategy）　D.桥接器（Bridge）

问题 2：A.Item　　B.Shopping_Cart　　C.Checkout　　D.Manual 和 Auto

问题 3：A.创建型对象模式　　B.结构型对象模式　　C.行为型类模式　　D.行为型对象模式

问题 4：

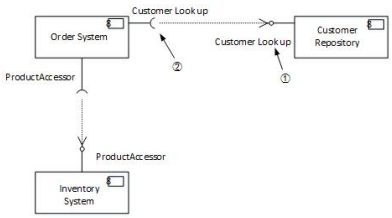
A.必须保存一个对象在某一个时刻的（部分）状态

B.想在不明确指定接收者的情况下向多个对象中的一个提交一个请求

C.需要对一个对象结构中的对象进行很多不同的并且不相关的操作

D.在不同的时刻指定、排列和执行请求

36.下图所示 UML 图为（ ），用于展示（ ）。①和②分别表示（ ）。



问题 1：A.类图　　B.组件图　　C.通信图　　D.部署图

问题 2：

A.一组对象、接口、协作和它们之间的关系

B.收发消息的对象的结构组织

C.组件之间的组织和依赖

D.面向对象系统的物理模型

问题 3：A.供接口和供接口　　B.需接口和需接口　　C.供接口和需接口　D.需接口和供接口

37.在某销售系统中，客户采用扫描二维码进行支付。若采用面向对象方法开发该销售系统，则客户类属于（ ）类，二维码类属于（ ）类。

问题 1：A.接口 B.实体 C.控制 D.状态

问题 2：A.接口 B.实体 C.控制 D.状态

38.（ ）多态是指操作（方法）具有相同的名称、且在不同的上下文中所代表的含义不同。

A.参数 B.包含 C.过载 D.强制

39.在面向对象方法中，继承用于（ ）。

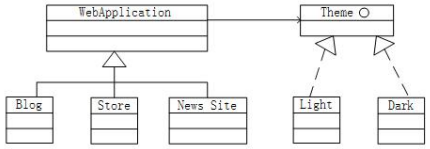
A.在已存在的类的基础上创建新类

B.在已存在的类中添加新的方法

C.在已存在的类中添加新的属性

D.在已存在的状态中添加新的状态

40.假设现在要创建一个 Web 应用框架，基于此框架能够创建不同的具体 Web 应用，比如博客、新闻网站和网上商店等；并可以为每个 Web 应用创建不同的主题样式，如浅色或深色等。这一业务需求的类图设计适合采用（ ）模式（如下图所示）。其中（ ）是客户程序使用的主要接口，维护对主题类型的引用。此模式为（ ），体现的最主要的意图是（ ）。



问题 1：A.观察者（Observer）B.访问者（Visitor）C.策略（Strategy）D.桥接（Bridge）

问题 2：A.WebApplication B.Blog C.Theme D.Light

问题 3：A.创建型对象模式 B.结构型对象模式 C.行为型类模式 D.行为型对象模式

问题 4：

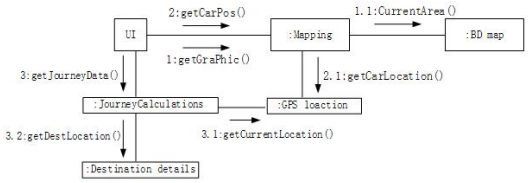
A.将抽象部分与其实现部分分离，使它们都可以独立地变化

B.动态地给一个对象添加一些额外的职责

C.为其他对象提供一种代理以控制对这个对象的访问

D.将一个类的接口转换成客户希望的另外一个接口

41.如下所示的图为 UML 的（ ），用于展示某汽车导航系统中（ ）。Mapping 对象获取汽车当前位置（GPS Location）的消息为（ ）。



问题 1： A.类图 B.组件图 C.通信图 D.部署图

问题 2：

A.对象之间的消息流及其顺序 B.完成任务所进行的活动流

C.对象的状态转换及其事件顺序 D.对象之间消息的时间顺序

问题 3：

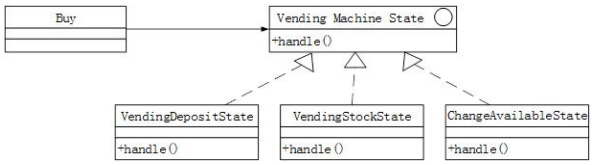
A.1: getGraphic() B.2: getCarPos() C.1.1: CurrentArea() D.2. 1: getCarLocation()

42.同一消息可以调用多种不同类的对象的方法，这些类有某个相同的超类，这种现象是（ ）。
A.类型转换 B.映射 C.单态 D.多态

43.在下列机制中，（ ）是指过程调用和响应调用所需执行的代码在运行时加以结合；而（ ）是过程调用和响应调用所需执行的代码在编译时加以结合。
问题 1：A.消息传递 B.类型检查 C.静态绑定 D.动态绑定
问题 2：A.消息传递 B.类型检查 C.静态绑定 D.动态绑定

44.对象的（ ）标识了该对象的所有属性（通常是静态的）以及每个属性的当前值（通常是动态的）。
A.状态 B.唯一 ID C.行为 D.语义

45.自动售货机根据库存、存放货币量、找零能力、所选项目等不同，在货币存入并进行选择时具有如下行为：交付产品不找零 ；交付产品并找零；存入货币不足而不提供任何产品；库存不足而不提供任何产品。这一业务需求适合采用（ ）模式设计实现，其类图如下图所示，其中（ ）是客户程序使用的主要接口，可用状态来对其进行配置。此模式为（ ），体现的最主要的意图是（ ）。



问题 1：A.观察者（Observer）B.状态（State）C.策略（Strategy）D.访问者（Visitor）
问题 2：A.Vending Machine State B.Buy C.Vending Deposit State D.Vending Stock State
问题 3：A.创建型对象模式 B.结构型对象模式 C.行为型类模式 D.行为型对象模式
问题 4：
A.当一个对象状态改变时所有依赖它的对象得到通知并自动更新
B.在不破坏封装性的前提下，捕获对象的内部状态并在对象之外保存
C.一个对象在其内部状态改变时改变其行为
D.将请求封装为对象从而可以使用不同的请求对客户进行参数化

46.如下所示的 UML 类图中，Shop 和 Magazine 之间为（ ）关系，Magazine 和 Page 之间为（ ）关系。UML 类图通常不用于对（ ）进行建模。

问题 1：A.关联 B.依赖 C.组合 D.继承
问题 2：A.关联 B.依赖 C.组合 D.继承
问题 3：A.系统的词汇 B.简单的协作 C.逻辑数据库模式 D.对象快照

47.面向对象分析过程中，从给定需求描述中选择（ ）来识别对象。
A.动词短语 B.名词短语 C.形容词 D.副词

48.采用继承机制创建子类时，子类中（ ）。
A.只能有父类中的属性 B.只能有父类中的行为
C.只能新增行为 D.可以有新的属性和行为

49.在面向对象方法中，将逻辑上相关的数据以及行为绑定在一起，使信息对使用者隐蔽称为（ ）。当类中的属性或方法被设计为 private 时，（ ）可以对其进行访问。

问题 1：A.抽象 B.继承 C.封装 D.多态

问题 2：

A.应用程序中所有方法 B.只有此类中定义的方法

C.只有此类中定义的 public 方法 D.同一个包中的类中定义的方法

50.在面向对象方法中，将逻辑上相关的数据以及行为绑定在一起，使信息对使用者隐蔽称为（ ）。当类中的属性或方法被设计为 private 时，（ ）可以对其进行访问。

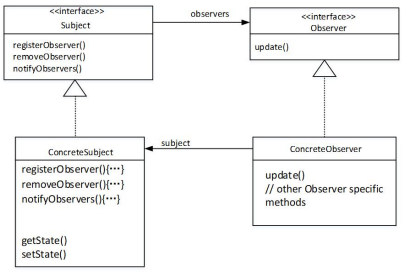
问题 1：A.抽象 B.继承 C.封装 D.多态

问题 2：

A.应用程序中所有方法 B.只有此类中定义的方法

C.只有此类中定义的 public 方法 D.同一个包中的类中定义的方法

51.下图所示为观察者（Observer）模式的抽象示意图，其中（ ）知道其观察者，可以有任何多个观察者观察同一个目标；提供注册和删除观察者对象的接口。此模式体现的最主要的特征是（ ）。



问题 1：A.Subject B.Observer C.Concrete Subject D.Concrete Observer

问题 2：

A.类应该对扩展开放，对修改关闭

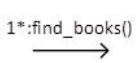
B.使所要交互的对象尽量松耦合

C.组合优先于继承使用

D.仅与直接关联类交互

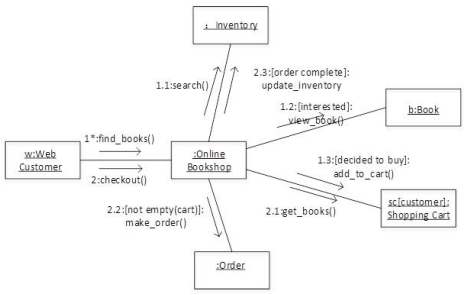
52.以下 UML 图是（ ），图中 

和

表示（ ）。 

和 

表示（ ）。



问题 1：A.序列图 B.状态图 C.通信图 D.活动图

问题 2：A.类 B.对象 C.流名称 D.消息

问题 3：.类 B.对象 C.流名称 D.消息

53.在面向对象方法中，多态指的是（ ）。

A.客户类无需知道所调用方法的特定子类的实现

B.对象动态地修改类

C.一个对象对应多张数据库表

D.子类只能够覆盖父类中非抽象的方法

54.采用面向对象方法进行软件开发，在分析阶段，架构师主要关注系统的（ ）。

A.技术 B.部署 C.实现 D.行为

55.在面向对象方法中，两个及以上的类作为一个类的超类时，称为（ ），使用它可能造成子类中存在（ ）的成员。

问题 1：A.多重继承 B.多态 C.封装 D.层次继承

问题 2：A.动态 B.私有 C.公共 D.二义性

56.（ ）模式将一个复杂对象的构建与其表示分离，使得同样的构建过程可以创建不同的表示。以下（ ）情况适合选用该模式。

①抽象复杂对象的构建步骤

②基于构建过程的具体实现构建复杂对象的不同表示

③一个类仅有一个实例

④一个类的实例只能有几个不同状态组合中的一种

问题 1:A.生成器（Builder）B.工厂方法（Factory Method）C.原型（Prototype）D.单例（ Singleton）

问题 2：A.①② B.②③ C.③④ D.①④

57.（ ）模式定义一系列的算法，把它们一个个封装起来，并且使它们可以相互替换，使得算法可以独立于使用它们的客户而变化。以下（ ）情况适合选用该模式。

①一个客户需要使用一组相关对象

②一个对象的改变需要改变其他对象

③需要使用一个算法的不同变体

④许多相关的类仅仅是行为有异

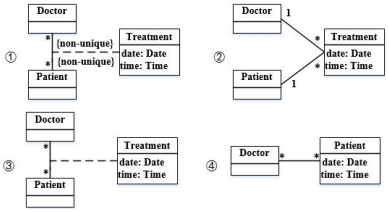
问题 1：

A.命令（Command） B.责任链（Chain of Responsibility）

C.观察者（Observer） D.策略（Strategy）

问题 2：A.①② B.②③ C.③④ D.①④

58.下图①②③④所示是 UML（ ）。现有场景：一名医生（Doctor）可以治疗多位病人（Patient），一位病人可以由多名医生治疗，一名医生可能多次治疗同一位病人。要记录哪名医生治疗哪位病人时，需要存储治疗（Treatment）的日期和时间。以下①②③④图中（ ）。是描述此场景的模型。



问题 1：A.用例图 B.对象图 C.类图 D.协作图

问题 2：A.① B.② C.③ D.④

59.以下关于 UML 状态图中转换（transition）的叙述中，不正确的是（　）。

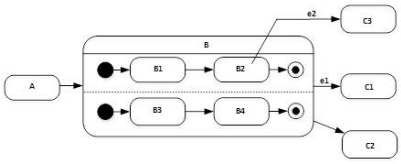
A.活动可以在转换时执行也可以在状态内执行

B.监护条件只有在相应的事件发生时才进行检查

C.一个转换可以有事件触发器、监护条件和一个状态

D.事件触发转换

60.如下所示的 UML 状态图中，（　）时，不一定会离开状态 B



A.状态 B 中的两个结束状态均达到

B.在当前状态为 B2 时，事件 e2 发生

C.事件 e2 发生

D.事件 e1 发生

61.面向对象分析的目的是为了获得对应用问题的理解，其主要活动不包括（　）。

A.认定并组织对象

B.描述对象间的相互作用

C.面向对象程序设计

D.确定基于对象的操作

62.在面向对象方法中，支持多态的是（　）。

A.静态分配　　B.动态分配　　C.静态类型　　D.动态绑定

63.在面向对象方法中，不同对象收到同一消息可以产生完全不同的结果，这一现象称为（　）。在使用时，用户可以发送一个通用的消息，而实现的细节则由接收对象自行决定。

A.接口　　B.继承　　C.覆盖　　D.多态

64.因使用大量的对象而造成很大的存储开销时，适合采用（　）模式进行对象共享，以减少对象数量从而达到较少的内存占用并提升性能。

A.组合（Composite）　　B.享元（Flyweight）

C.迭代器（Iterator）　　D.备忘录（Memento）

65.（　）设计模式最适合用于发布/订阅消息模型，即当订阅者注册一个主题后，此主题有新消息到来时订阅者就会收到通知。

A.适配器（Adapter）　　B.通知（Notifier）

C.观察者（Observer）　　D.状态（State）

66.（　）设计模式将一个请求封装为一个对象，从而使得可以用不同的请求对客户进行参数化，对请求排队或记录请求日志，以及支持可撤销的操作。

A.命令（Command）　　B.责任链（Chain of Responsibility）

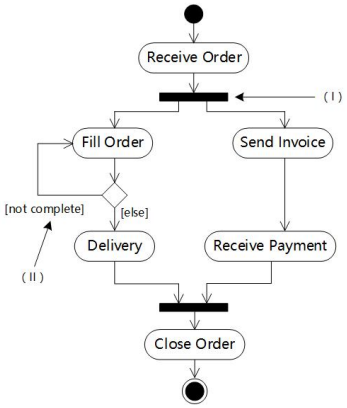
C.观察者（Observer）　　D.策略（Strategy）

67.为图形用户界面（GUI）组件定义不同平台的并行类层次结构，适合采用（ ）模式。

A.享元（Flyweight） B.抽象工厂（Abstract Factory）

C.外观（Facade）） D.装饰器（Decorator）

68.如下所示的 UML 图是（ ），图中（Ⅰ）表示（ ），（Ⅱ）表示（ ）。



问题 1：A.序列图 B.状态图 C.通信图 D.活动图

问题 2：A.合并分叉 B.分支 C.合并汇合 D.流

问题 3：A.分支条件 B.监护表达式 C.动作名 D.流名称

69.UML 中关联是一个结构关系，描述了一组链。两个类之间（ ）关联。

A.不能有多个 B.可以有多个由不同角色标识的

C.可以有任意多个 D.多个关联必须聚合成一个

70.在 UML 用例图中，参与者表示（ ）。

A.人、硬件或其他系统可以扮演的角色

B.可以完成多种动作的相同用户

C.不管角色的实际物理用户

D.带接口的物理系统或者硬件设计

71.在面向对象方法中，（ ）是父类和子类之间共享数据和方法的机制。子类在原有父类接口的基础上，用适合于自己要求的实现去置换父类中的相应实现称为（ ）。

问题 1：A.封装 B.继承 C.覆盖 D.多态

问题 2：A.封装 B.继承 C.覆盖 D.多态

72.（ ）设计模式能够动态地给一个对象添加一些额外的职责而无需修改此对象的结构；（ ）设计模式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；欲使一个后端数据模型能够被多个前端用户界面连接，采用（ ）模式最适合。

问题 1：A.组合（Composite） B.外观（Facade） C.享元（Flyweight） D.装饰器（Decorator）

问题 2：

A.工厂方法（Factory Method） B.享元（Flyweight)

C.观察者（ Observer） D.中介者（Mediator）

问题 3：

A.装饰器（Decorator）B.享元（Flyweight）

C.观察者（Observer）D.中介者（Mediator）

73.以下关于 Singleton（单例）设计模式的叙述中，不正确的是（　）。

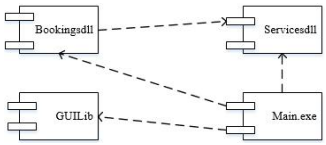
A.单例模式是创建型模式

B.单例模式保证一个类仅有一个实例

C.单例类提供一个访问唯一实例的全局访问点

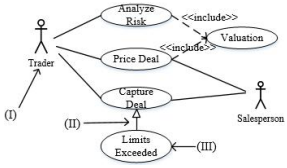
D.单例类提供一个创建一系列相关或相互依赖对象的接口

74.下图所示为 UML（　）。



A.类图　　B.部署图　　C.组件图　　D.网络图

75.如下所示的 UML 图中，（I）是（　），（Ⅱ）是（　），（Ⅲ）是（　）。

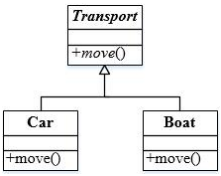


问题 1：A.参与者　B.用例　C.泛化关系　D.包含关系

问题 2：A.参与者　B.用例　C.泛化关系　D.包含关系

问题 3：A.参与者　B.用例　C.泛化关系　D.包含关系

76.如下所示的 UML 类图中，Car 和 Boat 类中的 move()方法（　）了 Transport 类中的 move()方法。



A.继承　　B.覆盖（重置）

C.重载　　D.聚合

77.在面向对象的系统中，对象是运行时实体，其组成部分不包括（　）；一个类定义了一组大体相似的对象，这些对象共享（　）。

问题 1：

A.消息　B.行为（操作）　C.对象名　D.状态

问题 2：

A.属性和状态　B.对象名和状态　C.行为和多重度　D.属性和行为

78.某些设计模式会引入总是被用作参数的对象（　）对象是一个多态 accept

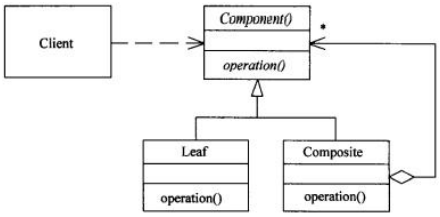
A.Visitor

B.Command

C.Memento

D.Observer

79.下图所示为（ ）设计模式，属于（ ）设计模式，适用于（ ）。



问题 1：A.代理（Proxy） B.生成器（Builder） C.组合（Composite） D.观察者（Observer）

问题 2：A.创建型 B.结构型 C.行为 D.结构型和行为

问题 3：

A.表示对象的部分一整体层次结构时

B.当一个对象必须通知其他对象，而它又不能假定其他对象是谁时

C.当创建复杂对象的算法应该独立于该对象的组成部分及其装配方式时

D.在需要比较通用和复杂的对象指针代替简单的指针时

80.UML 图中，对新开发系统的需求进行建模，规划开发什么功能或测试用例，采用（ ）最适合。而展示交付系统的软件组件和硬件之间的关系的图是（ ）

问题 1：A.类图 B.对象图 C.用例图 D.交互图

问题 2：A.类图 B.部署图 C.组件图 D.网络图

81.UML 中有 4 种关系：依赖、关联、泛化和实现。（ ）是一种结构关系，描述了一组链，链是对象之间的连接；（ ）是一种特殊/一般关系，使子元素共享其父元素的结构和行为。

问题 1：A.依赖 B.关联 C.泛化 D.实现

问题 2：A.依赖 B.关联 C.泛化 D.实现

82.一个类可以具有多个同名而参数类型列表不同的方法，被称为方法（ ）。

A.重载 B.调用 C.重置 D.标记

83.面向对象（ ）选择合适的面向对象程序设计语言，将程序组织为相互协作的对象集合，每个对象表示某个类的实例，类通过继承等关系进行组织。

A.分析 B.设计 C.程序设计 D.测试

84.对象、类、继承和消息传递是面向对象的 4 个核心概念。其中对象是封装（ ）的整体。

A.命名空间 B.要完成任务 C.一组数据 D.数据和行为

85.下图所示为（ ）设计模式，适用于（ ）。

问题 1：

A.抽象工厂（Abstract Factory） B.生成器（Builder）

C.工厂方法（Factory Method） D.原型（Prototype）

问题 2：

A.一个系统要由多个产品系列中的一个来配置时

B.当一个类希望由它的子类来指定它所创建的对象时

C.当创建复杂对象的算法应该独立于该对象的组成部分及其装配方式时

D.当一个系统应该独立于它的产品创建、构成和表示时

86.UML 图中，一张交互图显示一个交互。由一组对象及其之间的关系组成，包含它们之间可能传递的消息。（ ）不是交互图。

A.序列图 B.对象图 C.通信图 D.时序图

87.一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，用一个中介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。此模式与（ ）模式是相互竞争的模式，主要差别是：前者的中介对象封装了其他对象间的通信，而后者通过引入其他对象来分布通信。

问题 1：A.解释器(Interpreter) B.策略(Strategy) C.中介者(Mediator) D.观察者(Observer)

问题 2：A.解释器(Interpreter) B.策略(Strategy) C.中介者(Mediator) D.观察者(Observer)

88.UML 中有 4 种事物：结构事物、行为事物、分组事物和注释事物。类、接口、构件属于（ ）事物；依附于一个元素或一组元素之上对其进行约束或解释的简单符号为（ ）事物。

问题 1：A.结构 B.行为 C.分组 D.注释

问题 2：A.结构 B.行为 C.分组 D.注释

89.在面向对象程序设计语言中，对象之间通过（ ）方式进行通信。以下关于好的面向对象程序设计语言的叙述中，不正确的是（ ）。

问题 1：A.消息传递 B.继承 C.引用 D.多态

问题 2：

A.应该支持被封装的对象

B.应该支持类写实例的概念

C.应该支持通过指针进行引用

D.应该支持继承和多态

90.多态分为参数多态、包含多态、过载多态和强制多态四种不同形式，其中（ ）多态在许多语言中都存在，最常见的例子就是子类型化。

A.参数 B.包含 C.过载 D.强制

91.类（ ）之间存在着一般和特殊的关系。

A.汽车与轮船 B.交通工具与飞机 C.轮船与飞机 D.汽车与飞机

92.下列设计模式中，（ ）模式既是类结构型模式，又是对象结构型模式。此模式与（ ）模式类似的特征是，都给另一个对象提供了一定程度上的间接性，都涉及到从自身以外的一个接口向这个对象转发请求。

问题 1：

A.桥接（Bridge）

B.适配器（Adapter）

C.组成（Composite）

D.装饰器（Decorator）

问题 2：

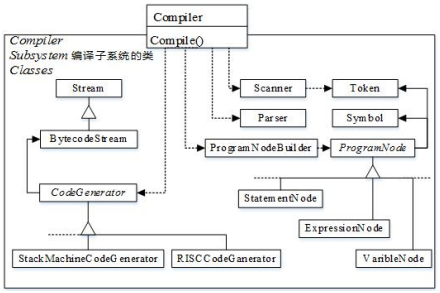
A.桥接（Bridge）

B.适配器（Adapter）

C.组成（Composite）

D.装饰器（Decorator）

93.下图所示为（ ）设计模式，适用于（ ）。



问题 1：

A.适配器（Adapter） B.责任链（Chain of Responsibility）

C.外观（Facade） D.桥接（Bridge）

问题 2：

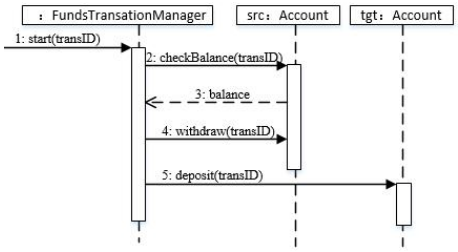
A.有多个对象可以处理一个请求，在运行时刻自动确定由哪个对象处理

B.想使用一个已经存在的类，而其接口不符合要求

C.类的抽象和其实现之间不希望有一个固定的绑定关系

D.需要为一个复杂子系统提供一个简单接口

94.如下所示的序列图中（ ）表示返回消息，Accunt 类必须实现的方法有（ ）。



问题 1：A.tansID B.balance C.withdraw D.deposit

问题 2：

A.start() B.checkBalance()和 withdraw()

C.deposit() D.checkBalance()、withdraw()和 deposit()

95.对一个复杂用例中的业务处理流程进行进一步建模的最佳工具是 UML（ ）。

A.状态图 B.顺序图 C.类图 D.活动图

96.在面向对象技术中，不同的对象在收到同一消息时可以产生完全不同的结果， 这一现象称为（ ），它由（ ）机制来支持。利用类的层次关系，把具有通用功能的消息存放在高层次，而不同的实现这一功能的行为放在较低层次，在这些低层次上生成的对象能够给通用消息以不同的响应。

问题 1：A.绑定 B.继承 C.消息 D.多态

问题 2：A.绑定 B.继承 C.消息 D.多态

97.某些程序设计语言中，在运行过程中当一个对象发送消息请求服务时，根据接收对象的具体情况将请求的操作与实现的方法进行连接，称为（ ）。

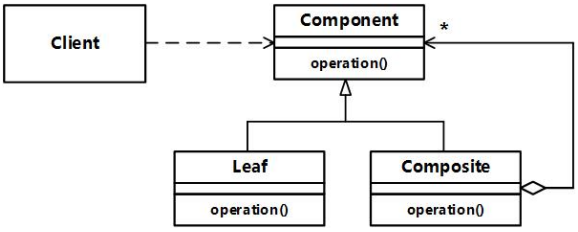
A.静态绑定 B.通用绑定

C.动态绑定 D.过载绑定

98.（ ）是一个类与它的一个或多个细化类之间的关系，即一般与特殊的关系。

A.泛化　　B.关联　　C.聚集　　D.组合

99.下图所示为（ ）设计模式，适用于（ ）。



问题 1：A.组件（Component） B.适配器（Adapter） C.组合（Composite） D.装饰器（Decorator）

问题 2：

A.表示对象的部分-整体层次结构

B.不希望在抽象和它的实现部分之间有一个固定的绑定关系

C.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.使所有接口不兼容类可以一起工作

100.在发布-订阅（Publish-Subscibe）消息模式中，订阅者订阅一个主题后，当该主题有新消息到达时，所有订阅者都会收到通知。（ ）设计模式最适合之一模式。

A.适配器（Adapter） B.通知（Notifier） C.状态（State） D.观察者（Observer）

101.（ ）设计模式定义一系列算法，把它们一个个封装起来，并且使它们可相互替换。这一模式使得算法可独立于它的客户而变化。

A.策略（Strategy） B.抽象工厂（Abstract Factory）

C.观察者（Observer） D.状态（State）

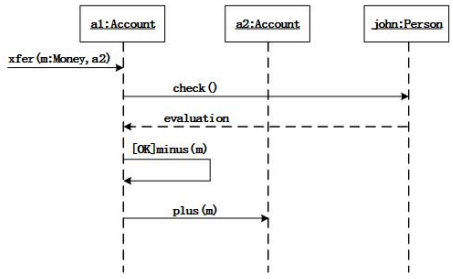
102.在面向对象技术中，（ ）定义了超类和子类之间的关系，子类中以更具体的方式实现从父类继承来的方法称为（ ），不同类的对象通过（ ）相互通信。

问题 1：A.覆盖　　B.继承　　C.消息　　D.多态

问题 2：A.覆盖　　B.继承　　C.消息　　D.多态

问题 3：A.覆盖　　B.继承　　C.消息　　D.多态

103.下图所示的 UML 序列图中，（ ）表示返回消息，Account 应该实现的方法有（ ）。



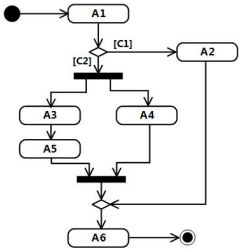
问题 1：A.xfer　　B.check　　C.evaluation　　D.minus

问题 2：

A.xfer()　　B.xfer()、plus()和 minus()

C.check()、plus()和 minus()　　D.xfer()、evaluation()、plus()和 minus()

104.在执行如下所示的 UML 活动图时，能同时运行的最大线程数为（ ）。



A.4 B.3 C.2 D.1

105.在领域类模型中不包含（ ）。

A.属性 B.操作 C.关联 D.领域对象

106.（ ）设计模式能使一个对象的状态发生改变时通知所有依赖它的监听者。（ ）设计模式限制类的实例对象只能有一个。适配器（Adapter）设计模式可以用于（ ）。用于为一个对象添加更多功能而不使用子类的是（ ）设计模式

问题 1：A.责任链 B.命令 C.抽象工厂 D.观察者

问题 2：A.原型 B.工厂方法 C.单例 D.生成器

问题 3：

A.将已有类的接口转换成和目标接口兼容

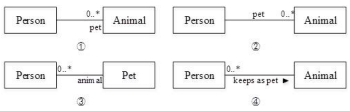
B.改进系统性能

C.将客户端代码数据转换成目标接口期望的合适的格式

D.使所有接口不兼容类可以一起工作

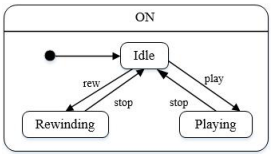
问题 4：A.桥接 B.适配器 C.组合 D.装饰器

107.描述一些人（Person）将动物（Animal）养为宠物(Pet)的是图（ ）。



A.① B.② C.③ D.④

108.以下关于 UML 状态图的叙述中，不正确的是（ ）。对下图的描述，正确的是（ ）。



问题 1：

A.用于描述一个对象在多个用例中的行为

B.用于某些具有多个状态的对象而不是系统中大多数或全部对象

C.用于描述多个对象之间的交互

D.可以用于用户界面或控制对象

问题 2：

A.ON 是一个并发状态

B.因为此状态图中没有终点(final)状态，所以此图是无效的

C.play、stop 和 rew 是动作 D.ON 是超状态

109.以下关于 UML 部署图的叙述中，正确的是（ ）。

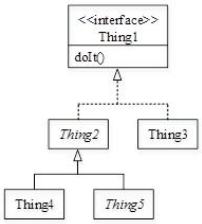
A.因为一条消息总是有某种响应，所以部署组件之间的依赖是双向的

B.部署组件之间的依赖关系类似于包依赖

C.部署图不用于描述代码的物理模块

D.部署图不用于描述系统在不同计算机系统的物理分布

110.继承是父类和子类之间共享数据和方法的机制。以下关于继承的叙述中，不正确的是（ ）。有关于图中 doIt()方法的叙述中，正确的是（ ）。



问题 1：

A.一个父类可以有多个子类，这些子类都是父类的特例

B.父类描述了这些子类的公共属性和操作

C.子类可以继承它的父类（或祖先类）中的属性和操作而不必自己定义

D.子类中可以定义自己的新操作而不能定义和父类同名的操作

问题 2：

A.doIt()必须由 Thing3 实现，同时可能由 Thing4 实现

B.doIt()必须由 Thing5 实现

C.doIt()必须由 Thing2、Thing3、Thing4、Thing5 实现

D.doIt()已经由 Thing1 实现，因为无需其它类实现

111.在多态的几种不同形式中，（ ）多态是一种特定的多态，指同一个名字在不同上下文中可代表不同的含义。

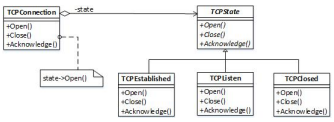
A.参数 B.包含 C.过载 D.强制

112.欲使类 A 的所有使用者都使用 A 的同一个实例，应（ ）。

A.将 A 标识为 final B.将 A 标识为 abstract

C.将单例（ Singleton）模式应用于 A D.将备忘（ Memento）模式应用于 A

113.每种设计模式都有特定的意图。（ ）模式使得一个对象在其内部状态改变时通过调用另一个类中的方法改变其行为，使这个对象看起来如同修改了它的类。下图是采用该模式的有关 TCP 连接的结构图实例。该模式的核心思想是引入抽象类（ ）来表示 TCP 连接的状态，声明不同操作状态的公共接口，其子类实现与特定状态相关的行为。当一个（ ）对象收到其它对象的请求时，它根据自身的当前状态做出不同的反应。

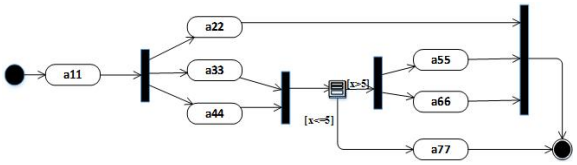


问题 1：A.适配器（ Adapter） B.命令（Command） C.观察者（ Visitor） D.状态（State）

问题 2：A.TCPConnection B.state C.TCPState D.TCPEstablished

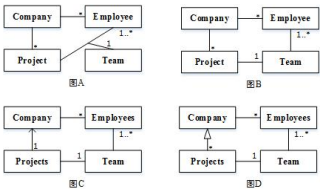
问题 3：A.TCPConnection B.state C.TCPState D.TCPEstablished

114.下列活动图中可以同时执行的活动是（ ）。



A.a44 和 a66　　B.a22，a33 和 a44　　C.a11 和 a77　　D.a66 和 a77

115.对于场景：一个公司负责多个项目，每个项目（Project）由一个员工（Employee）团队（ Team）来开发。下列 UML 概念图中，（ ）最适合描述这一场景。



A.图 A　　B.图 B　　C.图 C　　D.图 D

116.如果要表示待开发软件系统中软件组件和硬件之间的物理关系，通常采用 UML 中的（ ）。

A.组件图　　B.部署图　　C.类图　　D.网络图

117.在面向对象技术中，（ ）说明一个对象具有多种形态，（ ）定义超类与子类之间的关系。

问题 1：A.继承　　B.组合　　C.封装　　D.多态

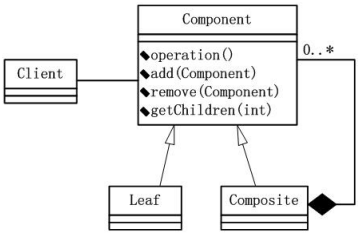
问题 2：A.继承　　B.组合　　C.封装　　D.多态

118.在面向对象技术中，对象具有以下特性：（ ）。

①清晰的边界　②良好定义的行为　③确定的位置和数量　④可扩展性

A.②④　　B.①②③④　　C.①②④　　D.①②

119.设计模式中的（ ）模式将对象组合成树形结构以表示“部分一整体”的层次结构，使得客户对单个对象和组合对象的使用具有一致性。下图为该模式的类图，其中，（ ）定义有子部件的那些部件的行为；组合部件的对象由（ ）通过 component 提供的接口操作。



问题 1：A.代理（Proxy）B.桥接器（Bridge）C.组合（ Composite） D.装饰器（Decorator）

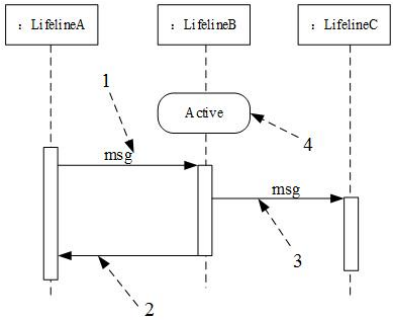
问题 2：A.Client　　B.Component　　C.Leaf　　D.Composite

问题 3：A.Client　　B.Component　　C.Leaf　　D.Composite

120.设计模式根据目的进行分类，可以分为创建型、结构型和行为型三种。其中结构型模式用于处理类和对象的组合。（ ）模式是一种结构型模式。

A.适配器（ Adapter） B.命令（Command） C.生成器（Builder） D.状态（State）

121.UML 序列图是一种交互图，描述了系统中对象之间传递消息的时间次序。其中，异步消息与同步消息不同，（ ）。下图中（ ）表示一条同步消息，（ ）表示一条异步消息，（ ）表示一条返回消息。



问题 1：

A.异步消息并不引起调用者终止执行而等待控制权的返回

B.异步消息和阻塞调用有相同的效果

C.异步消息是同步消息的响应

D.异步消息和同步消息一样等待返回消息

问题 2：A.1 B.2 C.3 D.4

问题 3：A.1 B.2 C.3 D.4

问题 4：A.1 B.2 C.3 D.4

122.在有些程序设计语言中，过程调用和响应调用需执行的代码的绑定直到运行时才进行，这种绑定称为（ ）。

A.静态绑定 B.动态绑定 C.过载绑定 D.强制绑定

123.以下关于封装在软件复用中所充当的角色的叙述，正确的是（ ）。

A.封装使得其他开发人员不需要知道一个软件组件内部如何工作

B.封装使得软件组件更有效地工作

C.封装使得软件开发人员不简要编制开发文档

D.封装使得软件组件开发更加容易

124.面向对象技术中，组合关系表示（ ）。

A.包与其中模型元素的关系 B.用例之间的一种关系

C.类与其对象的关系 D.整体与其部分之间的一种关系

125.业务用例和参与者一起描述（ ），而业务对象模型描述（ ）。

问题 1：

A.工作过程中的静态元素 B.工作过程中的动态元素

C.工作过程中的逻辑视图 D.组织支持的业务过程

问题 2：

A.业务结构 B.结构元素如何完成业务用例

C.业务结构以及结构元素如何完成业务用例 D.组织支持的业务过程

126.业务用例和参与者一起描述（ ），而业务对象模型描述（ ）。

问题 1：

A.工作过程中的静态元素

B.工作过程中的动态元素

C.工作过程中的逻辑视图

D.组织支持的业务过程

问题 2：

A.业务结构

B.结构元素如何完成业务用例

C.业务结构以及结构元素如何完成业务用例

D.组织支持的业务过程

127.采用 UML 进行面向对象开发时，部署图通常在（ ）阶段使用。

A.需求分析　　B.架构设计　　C.实现　　D.实施

128.（ ）模式通过提供与对象相同的接口来控制对这个对象的访问。

A.适配器(Adapter)　　B.代理(Proxy)

C.组合(Composite)　　D.装饰器(Decorator)

129.欲动态地给一个对象添加职责，宜采用（ ）模式。

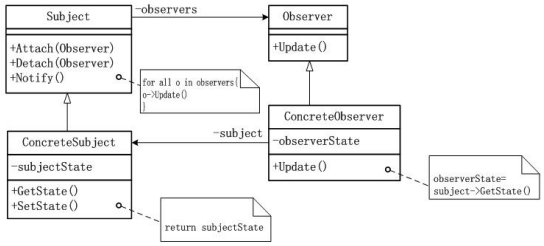
A.适配器(Adapter)　　B.桥接(Bridge)

C.组合(Composite)　　D.装饰器(Decorator)

130.在面向对象软件开发中，封装是一种（ ）技术，其目的是使对象的使用者和生产者分离。

A.接口管理　　B.信息隐藏　　C.多态　　D.聚合

131.（ ）设计模式允许一个对象在其状态改变时，通知依赖它的所有对象。该设计模式的类图如下图，其中，（ ）在其状态发生改变时，向它的各个观察者发出通知。



问题 1：

A.命令(Command)　　B.责任链(Chain of Responsibility)

C.观察者(Observer)　　D.迭代器(Iterator)

问题 2：

A.Subject

B.ConcreteSubject

C.Observer

D.ConcreteObserver

132.一个类是（ ）。在定义类时，将属性声明为 private 的目的是（ ）。

问题 1：

A.一组对象的封装　　B.表示一组对象的层次关系

C.一组对象的实例　　D.一组对象的抽象定义

问题 2：

A.实现数据隐藏，以免意外更改　　B.操作符重载

C.实现属性值不可更改　　D.实现属性值对类的所有对象共享

133.采用面向对象开发方法时，对象是系统运行时基本实体。以下关于对象的叙述中，正确的是（ ）。

A.对象只能包括数据（属性）

B.对象只能包括操作（行为）

C.对象一定有相同的属性和行为

D.对象通常由对象名、属性和操作三个部分组成

134. （ ）将一个类的接口转换成客户希望的另外一个接口，使得原本由于接口不兼容而不能一起工作的那些类可以一起工作。

A.Adapter（适配器）模式　　B.Command（命令）模式

C.Singleton（单例）模式　　D.Strategy（策略）模式

135.以下关于 Singleton（单例）模式的描述中，正确的是（ ）。

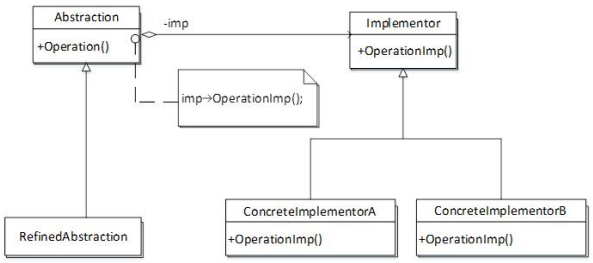
A.它描述了只有一个方法的类的集合

B.它描述了只有一个属性的类的集合

C.它能够保证一个类的方法只能被一个唯一的类调用

D.它能够保证一个类只产生唯一的一个实例

136.设计模式（ ）将抽象部分与其实现部分相分离，使它们都可以独立地变化。下图为该设计模式的类图，其中，（ ）用于定义实现部分的接口。



问题 1：A.Bridge（桥接）　B.Composite（组合）　C.Facade（外观）　D.Singleton（单例）

问题 2：

A.Abstraction　　B.ConcreteImplementorA

C.ConcreteImplementorB　　D.Implementor

137.在面向对象软件开发过程中，采用设计模式（ ）。

A.以复用成功的设计

B.以保证程序的运行速度达到最优值

C.以减少设计过程创建的类的个数

D.允许在非面向对象程序设计语言中使用面向对象的概念

138.UML 中关联的多重度是指（ ）。

A.一个类中被另一个类调用的方法个数

B.一个类的某个方法被另一个类调用的次数

C.一个类的实例能够与另一个类的多少个实例相关联

D.两个类所具有的相同的方法和属性

139.UML 的设计视图包含了类、接口和协作，其中，设计视图的静态方面由（ ）和（ ）表现；动态方面由交互图、（ ）表现。

问题 1:A.类图 B.状态图 C.活动图 D.序列图

问题 2：A.交互图 B.对象图 C.通信图 D.定时图

问题 3：

A.状态图和类图 B.类图和活动图

C.对象图和状态图 D.状态图和活动图

140.下列关于一个类的静态成员的描述中，不正确的是（ ）。

A.类的静态方法只能访问该类的静态数据成员

B.静态数据成员可被该类的所有方法访问

C.该类的对象共享其静态数据成员的值

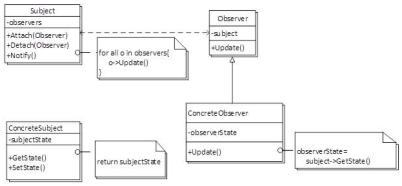
D.该类的静态数据成员的值不可修改

141.面向对象分析的第一步是（ ）。

A.定义服务 B.确定附加的系统约束

C.确定问题域 D.定义类和对象

142.下面的 UML 类图描绘的是（ ）设计模式。关于该设计模式的叙述中，错误的是（ ）。



问题 1：A.桥接 B.策略 C.抽象工厂 D.观察者

问题 2：

A.该设计模式中的 Observer 需要维护至少一个 Subject 对象

B.该设计模式中的 ConcreteObserver 可以绕过 Subject 及其子类的封装

C.该设计模式中一个 Subject 对象需要维护多个 Observer 对象

D.该设计模式中 Subject 需要通知 Observer 对象其自身的状态变化

143.（ ）是一种很强的“拥有”关系，“部分”和“整体”的生命周期通常一样。整体对象完全支配其组成部分，包括它们的创建和销毁等;（ ）同样表示“拥有”关系，但有时候“部分”对象可以在不同的“整体”对象之间共享，并且“部分”对象的生命周期也可以与“整体”对象不同，甚至“部分”对象可以脱离“整体”对象而单独存在。上述两种关系都是（ ）关系的特殊种类。

问题 1：A.聚合 B.组合 C.继承 D.关联

问题 2：A.聚合 B.组合 C.继承 D.关联

问题 3：A.聚合 B.组合 C.继承 D.关联

144.开-闭原则（Open-Closed Principle，OCP)是面向对象的可复用设计的基石。开-闭原则是指一个软件实体应当对（ ）开放，对（ ）关闭；里氏代换原则（Liskov Substitution Principle，LSP ）是指任何（)可以出现的地方，( )一定可以出现。依赖倒转原则（Dependence Inversion Principle，DIP）就是要依赖于（ ）而不依赖于（ ），或者说要针对接口编程，不要针对实现编程。

问题 1：A.修改 B.扩展 C.分析 D.设计

问题 2：A.修改 B.扩展 C.分析 D.设计

问题 3：A.变量 B.常量 C.基类对象 D.子类对象

问题 4：A.变量 B.常量 C.基类对象 D.子类对象

问题 5：A.程序设计语言 B.建模语言 C.实现 D.抽象

问题 6：A.程序设计语言 B.建模语言 C.实现 D.抽象

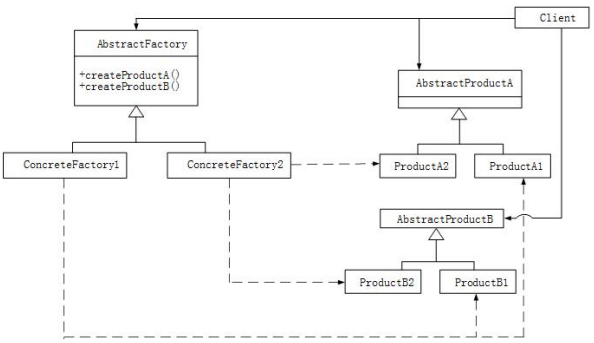
145.UML 类图中类与类之间的关系有五种：依赖、关联、聚合、组合与继承。若类 A 需要使用标准数学函数类库中提供的功能，那么类 A 与标准类库提供的类之间存在（ ）关系；若类 A 中包含了其它类的实例，且当类 A 的实例消失时，其包含的其它类的实例也消失，则类 A 和它所包含的类之间存在（ ）关系；若类 A 的实例消失时，其它类的实例仍然存在并继续工作，那么类 A 和它所包含的类之间存在（ ）关系。

问题 1：A.依赖 B.关联 C.聚合 D.组合

问题 2：A.依赖 B.关联 C.聚合 D.组合

问题 3：A.依赖 B.关联 C.聚合 D.组合

146.如下 UML 类图表示的是（ ）设计模式。以下关于该设计模式的叙述中，错误是（ ）。



问题 1： A.工厂方法 B.策略 C.抽象工厂 D.观察者

问题 2：

A.提供创建一系列相关或相互依赖的对象的接口，而无需指定这些对象所属的具体类

B.可应用于一个系统要由多个产品系列中的一个来配置的时候

C.可应用于强调一系列相关产品对象的设计以便进行联合使用的时候

D.可应用于希望使用已经存在的类，但其接口不符合需求的时候

147.以下关于面向对象设计的叙述中，错误的是（ ）。

A.面向对象设计应在面向对象分析之前，因为只有产生了设计结果才可对其进行分析

B.面向对象设计与面向对象分析是面向对象软件过程中两个重要的阶段

C.面向对象设计应该依赖于面向对象分析的结果

D.面向对象设计产生的结果在形式上可以与面向对象分析产生的结果类似，例如都可以使用 UML

148.（ ）不是面向对象分析阶段需要完成的。

A.认定对象 B.组织对象 C.实现对象及其相互关系 D.描述对象间的相互作用

149.不同的对象收到同一消息可以产生完全不同的结果，这一现象叫做（ ）。绑定是一个把过程调用和响应调用所需要执行的代码加以结合的过程。在一般的程序设计语言中，绑定在编译时进行，叫做（ ）；而（ ）则在运行时进行，即一个给定的过程调用和执行代码的结合直到调用发生时才进行。

问题 1:A.继承 B.多态 C.动态绑定 D.静态绑定

问题 2：:A.继承 B.多态 C.动态绑定 D.静态绑定

问题 3：:A.继承 B.多态 C.动态绑定 D.静态绑定

150.以下关于面向对象方法中继承的叙述中，错误的是（ ）。

A.继承是父类和子类之间共享数据和方法的机制 B.继承定义了一种类与类之间的关系

C.继承关系中的子类将拥有父类的全部属性和方法

D.继承仅仅允许单重继承，即不允许一个子类有多个父类

151.UML 中关联的多重度是指（）。

A.一个类有多少个方法被另一个类调用 B.一个类的实例能够与另一个类的多少个实例相关联

C.一个类的某个方法被另一个类调用的次数 D.两个类所具有的相同的方法和属性

152.下列关于一个类的静态成员的描述中，不正确的是（）。

A.该类的对象共享其静态成员变量的值 B.静态成员变量可被该类的所有方法访问

C.该类的静态方法只能访问该类的静态成员变量 D.该类的静态数据成员变量的值不可修改

153.以下关于单例模式（Singleton）的描述中，正确的是（）。

A.它描述了只有一个方法的类的集合 B.它能够保证一个类只产生一个唯一的实例

C.它描述了只有一个属性的类的集合 D.它能够保证一个类的方法只能被一个唯一的类调用

154.在面向对象软件开发过程中，采用设计模式（）。

A.允许在非面向对象程序设计语言中使用面向对象的概念

B.以复用成功的设计和体系结构

C.以减少设计过程创建的类的个数

D.以保证程序的运行速度达到最优值

155.已知 3 个类 O、P 和 Q，类 O 中定义了一个私有方法 Fl、一个公有方法 F2 和一个受保护的方法 F3；类 P 和类 Q 是类 O 的派生类，其继承方式如下所示：

```
class P:protected O{};
class Q:public O{…};
```

关于方法 Fl 的描述中正确的是（）；关于方法 F2 的描述中正确的是（）；关于方法 F3 的描述中正确的是（）。

问题 1：A.方法 Fl 无法被访问 B.只有在类 O 内才能访问方法 Fl

C.只有在类 P 内才能访问方法 Fl D.只有在类 Q 内才能访问方法 Fl

问题 2：

A.类 O、P 和 Q 的对象都可以访问方法 F2 B.类 P 和 Q 的对象都可以访问方法 F2

C.类 0 和 Q 的对象都可以访问方法 F2 D.只有在类 P 内才能访问方法 F2

问题 3：

A.类 O、P 和 Q 的对象都可以访问方法 F3 B.类 O、P 和 Q 的对象都不可以访问方法 F3

C.类 O 和 Q 的对象都可以访问方法 F3 D.类 P 和 Q 的对象都可以访问方法 F3

156.在 C++语言中，若类 C 中定义了一个方法 int f（int a，int b），那么方法（）不能与该方法同时存在于类 C 中。

A.int f(int x，int y)　　B.int f(float a，int b)

C.float f(int x，float y)　　D.int f(int x，float y)

157.下面给出了四种设计模式的作用：

外观（Facade）：为子系统中的一组功能调用提供一个一致的接口，这个接口使得这一子系统更加容易使用；

装饰(Decorate)：当不能采用生成子类的方法进行扩充时，动态地给一个对象添加一些额外的功能；

单件（Singleton）：保证一个类仅有一个实例，并提供一个访问它的全局访问点；

模板方法（Template Method）：在方法中定义算法的框架，而将算法中的一些操作步骤延迟到子类中实现。

请根据下面叙述的场景选用适当的设计模式。若某面向对象系统中的某些类有且只有一个实例，那么采用（）设计模式能够有效达到该目的；该系统中的某子模块需要为其它模块提供访问不同数据库系统（Oracle、SQL Server、DB2 UDB 等）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，（）设计模式可抽象出相同的数据库访问过程；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，但希望最多只增加三个类，（）设计模式可以实现该目的。

问题 1：A.外观　B.装饰　C.单件　D.模板方法

问题 2：A.外观　B.装饰　C.单件　D.模板方法

问题 3：A.外观　B.装饰　C.单件　D.模板方法

158.在进行面向对象设计时，采用设计模式能够（）。

A.复用相似问题的相同解决方案

B.改善代码的平台可移植性

C.改善代码的可理解性

D.增强软件的易安装性

159.面向对象分析需要找出软件需求中客观存在的所有实体对象（概念），然后归纳、抽象出实体类。（）是寻找实体对象的有效方法之一。

A.会议调查　B.问卷调查　C.电话调查　D.名词分析

160.面向对象分析与设计是面向对象软件开发过程中的两个重要阶段，下列活动中，（）不属于面向对象分析阶段。

A.构建分析模型

B.识别分析

C.确定接口规格

D.评估分析模型

161.在统一建模语言（UML）中，（）用于描述系统与外部系统及用户之间的交互。

A.类图　B.用例图　C.对象图　D.协作图

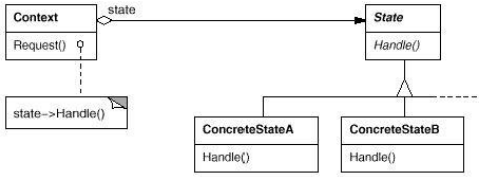
162.在 UML 的各种视图中，（）显示外部参与者观察到的系统功能；（）从系统的静态结构和动态行为角度显示系统内部如何实现系统的功能；（）显示的是源代码以及实际执行代码的组织结构。

问题 1：A.用例视图　　B.进程视图　　C.实现视图　　D.逻辑视图

问题 2：A.用例视图　　B.进程视图　　C.实现视图　　D.逻辑视图

问题 3：A.用例视图　　B.进程视图　　C.实现视图　　D.逻辑视图

163.（）设计模式允许一个对象在其内部状态改变时改变它的行为。下图为这种设计模式的类图，已知类 State 为抽象类，则类（）的实例代表了 Context 对象的状态。



问题 1：A.单件（Singleton）B.桥接（Bridge）　C.组合（Composite）D.状态（State）

问题 2：A.Context　　B.concreteStateA　　C.Handle　　D.State

164.（）是指在运行时把过程调用和响应调用所需要执行的代码加以结合。

A.绑定　　B.静态绑定

C.动态绑定　　D.继承

165.（）限制了创建类的实例数量，而（）将一个类的接口转换成客户希望的另外一个接口，使得原本由于接口不兼容而不能一起工作的那些类可以一起工作。

问题 1：

A.命令模式（Command）

B.适配器模式（Adapter）

C.策略模式（Strategy）

D.单例模式（Singleton）

问题 2：

A.命令模式（Command）

B.适配器模式（Adapter）

C.策略模式（Strategy）

D.单例模式（Singleton）

166.在选择某种面向对象语言进行软件开发时，不需要着重考虑的因素是，该语言（）。

A.将来是否能够占据市场主导地位

B.类库是否丰富

C.开发环境是否成熟

D.是否支持全局变量和全局函数的定义

167.面向对象分析与设计中的（）是指一个模块在扩展性方面应该是开放的，而在更改性方面应该是封闭的；而（）是指子类应当可以替换父类并出现在父类能够出现的任何地方。

问题 1：A.开闭原则　　B.替换原则　　C.依赖原则　　D.单一职责原则

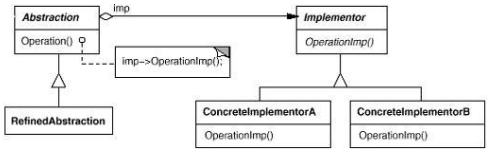
问题 2：A.开闭原则　　B.替换原则　　C.依赖原则　　D.单一职责原则

168.在面向对象系统中，用（）关系表示一个较大的“整体”类包含一个或多个较小的“部分”类。
A.泛化　　B.聚合　　C.概化　　D.合成

169.
Object-oriented analysis （OOA） is a semiformal specification technique for the object-oriented paradigm.Object-oriented analysis consists of three steps.The first step is （ ）.It determines how the various results are computed by the product and presents this information in the form of a （） and associated scenarios.The second is （） , which determines the classes and their attributes， then determines the interrelationships and interaction among the classes.The last step is （） , which determines the actions performed by or to each class or subclass and presents this information in the form of （）.
问题 1：A.use-case modeling B.class modeling C.dynamic modeling D.behavioral modeling
问题 2：A.collaboration diagram B.sequence diagram C.use-case diagram D.activity diagram
问题 3：A.use-case modeling B.class modeling C.dynamic modeling D.behavioral modeling
问题 4：A.use-case modeling B.class modeling C.dynamic modeling D.behavioral modeling
问题 5：A.activity diagram B.component diagram C.sequence diagram D.state diagram

170.在 UML 类图中，类与类之间存在依赖（Dependency）、关联（Association）、聚合（Aggregation）、组合（Composition）和继承（Inheritance）五种关系，其中，（）关系表明类之间的相互联系最弱，（）关系表明类之间的相互联系最强，聚合（Aggregation）的标准 UML 图形表示是（） 。
问题 1：A.依赖　　B.聚合　　C.组合　　D.继承
问题 2：A.依赖　　B.聚合　　C.组合　　D.继承
问题 3:A.  B.  C.  D. 

171.（）设计模式将抽象部分与它的实现部分相分离，使它们都可以独立地变化。下图为该设计模式的类图，其中，（）用于定义实现部分的接口。



问题 1：A.Singleton（单件） B.Bridge（桥接） C.Composite（组合） D.Facade（外观）
问题 2：
A.Abstraction　　B.ConcreteImplementorA
C.ConcreteImplementorB　　D.Implementor

172.已知某子系统为外界提供功能服务，但该子系统中存在很多粒度十分小的类，不便被外界系统直接使用，采用（）设计模式可以定义一个高层接口，这个接口使得这一子系统更加容易使用；当不能采用生成子类的方法进行扩充时，可采用（）设计模式动态地给一个对象添加一些额外的职责。
问题 1：A.Facade（外观） B.Singleton（单件） C.Participant（参与者） D.Decorator（装饰）
问题 2：A.Facade（外观） B.Singleton（单件） C.Participant（参与者） D.Decorator（装饰）

173.（）是指把数据以及操作数据的相关方法组合在同一个单元中，使我们可以把类作为软件中的基本复用单元，提高其内聚度，降低其耦合度。面向对象中的（）机制是对现实世界中遗传现象的模拟，通过该机制，基类的属性和方法被遗传给派生类。

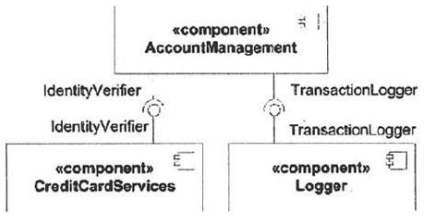
问题 1：A.封装　　B.多态　　C.继承　　D.变异

问题 2：A.封装　　B.多态　　C.继承　　D.变异

174.采用 UML 进行软件设计时，可用（）关系表示两类事物之间存在的特殊/一般关系，用聚集关系表示事物之间存在的整体/部分关系。

A.依赖　　B.聚集　　C.泛化　　D.实现

175.下图属于 UML 中的（　），其中，AccountManagement 需要（　）。



问题 1：A.组件图　　B.部署图　　C.类图　　D.对象图

问题 2：

A.实现 IdentityVerifier 接口并被 CreditCardServices 调用

B.调用 CreditCardServices 实现的 IdentityVerifier 接口

C.实现 IdentityVerifier 接口并被 Logger 调用

D.调用 Logger 实现的 IdentityVerifier 接口

176.当不适合采用生成子类的方法对已有的类进行扩充时，可以采用（　）设计模式动态地给一个对象添加一些额外的职责；当应用程序由于使用大量的对象，造成很大的存储开销时，可以采用（　）设计模式运用共享技术来有效地支持大量细粒度的对象；当想使用一个已经存在的类，但其接口不符合需求时，可以采用（　）设计模式将该类的接口转换成我们希望的接口。

问题 1：A.命令（Command）　B.适配器（Adapter）　C.装饰（Decorate）　D.享元（Flyweight）

问题 2：A.命令（Command）　B.适配器（Adapter）　C.装饰（Decorate）　D.享元（Flyweight）

问题 3：A.命令（Command）　B.适配器（Adapter）　C.装饰（Decorate）　D.享元（Flyweight）

177.若类 A 仅在其方法 Method1 中定义并使用了类 B 的一个对象，类 A 其它部分的代码都不涉及类 B，那么类 A 与类 B 的关系应为（　）；若类 A 的某个属性是类 B 的一个对象，并且类 A 对象消失时，类 B 对象也随之消失，则类 A 与类 B 的关系应为（　）。

问题 1：A.关联　　B.依赖　　C.聚合　　D.组合

问题 2：A.关联　　B.依赖　　C.聚合　　D.组合

178.在面向对象分析与设计中，（　）是应用领域中的核心类，一般用于保存系统中的信息以及提供针对这些信息的相关处理行为；（　）是系统内对象和系统外参与者的联系媒介；（　）主要是协调上述两种类对象之间的交互。

问题 1：A.控制类　　B.边界类　　C.实体类　　D.软件类

问题 2：A.控制类　　B.边界类　　C.实体类　　D.软件类

问题 3：A.控制类　　B.边界类　　C.实体类　　D.软件类

179.下面关于面向对象分析与面向对象设计的说法中，不正确的是（　）。

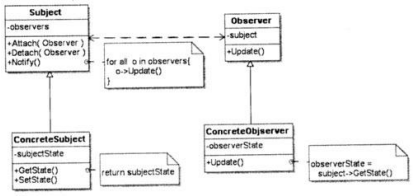
A.面向对象分析侧重于理解问题

B.面向对象设计侧重于理解解决方案

C.面向对象分析描述软件要做什么

D.面向对象设计一般不关注技术和实现层面的细节

180.下列 UML 类图表示的是（　）设计模式。该设计模式中，（　）。



问题 1：A.备忘录（Memento）　B.策略（Strategy）　C.状态（State）　D.观察者（Observer）

问题 2：

A.一个 Subject 对象可对应多个 Observer 对象

B.Subject 只能有一个 ConcreteSubject 子类

C.Observer 只能有一个 ConcreteObserver 子类

D.一个 Subject 对象必须至少对应一个 Observer 对象

181.采用（　）设计模式可保证一个类仅有一个实例：采用（　）设计模式可将对象组合成树形结构以表示“部分-整体”的层次结构，使用户对单个对象和组合对象的使用具有一致性；采用（　）设计模式可动态地给一个对象添加一些额外的职责。

问题 1：A.命令（Command）　B.单例（Singleton）　C.装饰(Decorate）　D.组合（Composite）

问题 2：A.命令（Command）　B.单例（Singleton）　C.装饰(Decorate）　D.组合（Composite）

问题 3：A.命令（Command）　B.单例（Singleton）　C.装饰(Decorate）　D.组合（Composite）

182.以下关于面向对象设计的叙述中，错误的是（　）。

A.高层模块不应该依赖于底层模块　　B.抽象不应该依赖于细节

C.细节可以依赖于抽象　　D.高层模块无法不依赖于底层模块

183.以下关于面向对象分析的叙述中，错误的是（　）。

A.面向对象分析看重分析问题域和系统责任 B.面向对象分析需要考虑系统的测试问题

C.面向对象分析忽略与系统实现有关的问题 D.面向对象分析建立独立于实现的系统分析模型

184.（　）是把对象的属性和服务结合成一个独立的系统单元，并尽可能隐藏对象的内部细节；（　）是指子类可以自动拥有父类的全部属性和服务；（　）是对象发出的服务请求，一般包含提供服务的对象标识、服务标识、输入信息和应答信息等。

问题 1：A.继承　B.多态　C.消息　D.封装

问题 2：A.继承　B.多态　C.消息　D.封装

问题 3：A.继承　B.多态　C.消息　D.封装

185.以下关于类和对象的叙述中，错误的是（　）。

A.类是具有相同属性和服务的一组对象的集合　B.类是一个对象模板，用它仅可以产生一个对象

C.在客观世界中实际存在的是类的实例，即对象 D.类为属于该类的全部对象提供了统一的抽象描述