

软件工程

第一部分

2018 年上半年

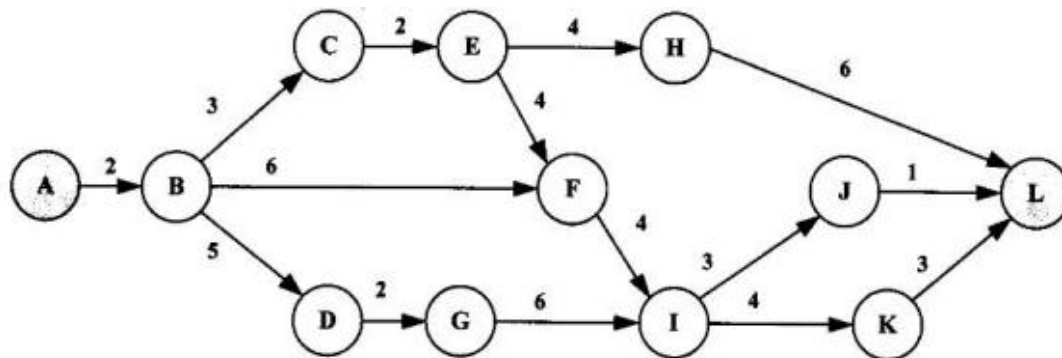
● 数据字典是结构化分析的一个重要输出。数据字典的条目不包括()。

- A. 外部实体
- B. 数据流
- C. 数据项
- D. 基本加工

● 某商店业务处理系统中, 基本加工“检查订货单”的描述为: 若订货单金额大于 5000 元, 且欠款时间超过 60 天, 则不予批准; 若订货单金额大于 5000 元, 且欠款时间不超过 60 天, 则发出批准书和发货单; 若订货单金额小于或等于 500 元, 则发出批准书和发货单, 若欠款时间超过 60 天, 则还要发催款通知书。现采用决策表表示该基本加工, 则条件取值的组合数最少是()。

- A. 2
- B. 3
- C. 4
- D. 5

● 某软件项目的活动图如下图所示, 其中顶点表示项目里程碑, 连接顶点的边表示包含的活动, 边上的数字表示活动的持续天数, 则完成该项目的最少时间为() 天。活动 EH 和 IJ 的松弛时间分别为() 天。



- A. 17
- B. 19
- C. 20
- D. 22

- A. 3 和 3
- B. 3 和 6
- C. 5 和 3
- D. 5 和 6

● 工作量估算模型 COCOMO II 的层次结构中, 估算选择不包括()。

- A. 对象点
- B. 功能点
- C. 用例数
- D. 源代码行

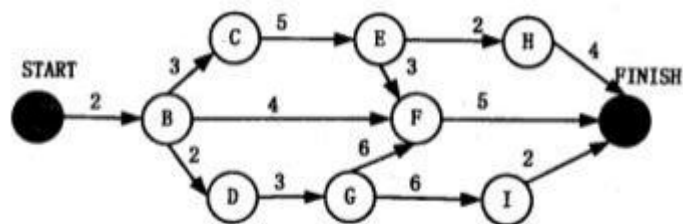
2017 年下半年

15. 某医院预约系统的部分需求为：患者可以查看医院发布的专家特长介绍及其就诊时间；系统记录患者信息，患者预约特定时间就诊。用 DFD 对其进行功能建模时，患者是（ ）；用 ERD 对其进行数据建模时，患者是（ ）。

- A. 外部实体
- B. 加工
- C. 数据流
- D. 数据存储

- A. 实体
- B. 属性
- C. 联系
- D. 弱实体

17. 某软件项目的活动图如下图所示，其中顶点表示项目里程碑，链接顶点的边表示包含的活动，变色数字表示活动的持续时间（天）。完成该项目的最少时间为（ ）天。由于某种原因，现在需要同一个开发人员完成 BC 和 BD，则完成该



项目的最少时间为（ ）天。

- | | |
|-------|-------|
| A. 11 | B. 18 |
| C. 20 | D. 21 |
| A. 11 | B. 18 |
| C. 20 | D. 21 |

19. 某企业财务系统的需求中，属于功能需求的是（ ）。

- A. 每个月特定的时间发放员工工资
- B. 系统的响应时间不超过 3 秒
- C. 系统的计算精度符合财务规则的要求
- D. 系统可以允许 100 个用户同事查询自己的工资

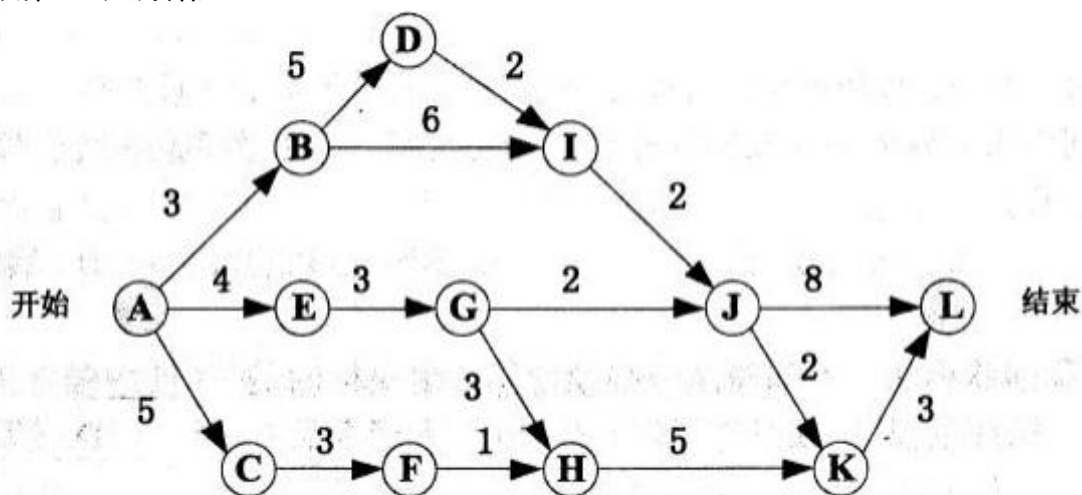
2017 年上半年

● 在采用结构化开发方法进行软件开发时，设计阶段接口设计主要依据需求分析阶段的()。接口设计的任务主要是()。

- A. 数据流图
- B. E-R 图
- C. 状态-迁移图
- D. 加工规格说明

- A. 定义软件的主要结构元素及其之间的关系
- B. 确定软件涉及的文件系统的结构及数据库的表结构
- C. 描述软件与外部环境之间的交互关系，软件内模块之间的调用关系
- D. 确定软件各个模块内部的算法和数据结构

● 某软件项目的活动图如下图所示，其中顶点表示项目里程碑，连接顶点的边表示包含的活动，边上的数字表示活动的持续时间(天)，则完成该项目的最少时间为()天。活动 BD 和 HK 最早可以从第()天开始。(活动 AB、AE 和 AC 最早从第 1 天开始)



- | | |
|-------|-------|
| A. 17 | B. 18 |
| C. 19 | D. 20 |
-
- | | |
|-----------|-----------|
| A. 3 和 10 | B. 4 和 11 |
| C. 3 和 9 | D. 4 和 10 |

● 在进行软件开发时，采用无主程序员的开发小组，成员之间相互平等；而主程序员负责制的开发小组，由一个主程序员和若干成员组成，成员之间没有沟通。在一个由 8 名开发人员构成的小组中，无主程序员组和主程序员组的沟通路径分别是__ (19) __。

- | | |
|-----------|-----------|
| A. 32 和 8 | B. 32 和 7 |
| C. 28 和 8 | D. 28 和 7 |

2016 年下半年

● 结构化开发方法中，（ ）主要包含对数据结构和算法的设计。

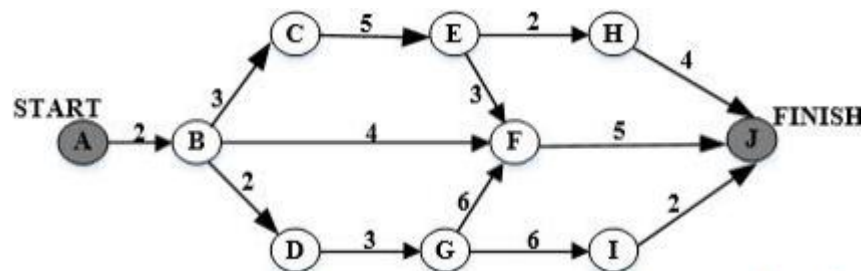
- A. 体系结构设计
- B. 数据设计
- C. 接口设计
- D. 过程设计

● 在敏捷过程的开发方法中，（ ）使用了迭代的方法，其中，把每段时间（30 天）一次的迭代称为一个“冲刺”，并按需求的优先级别来实现产品，多个自组织和自治的小组并行地递增实现产品。

- A. 极限编程 XP
- B. 水晶法
- C. 并列争球法
- D. 自适应软件开发

● 某软件项目的活动图如下图所示，其中顶点表示项目里程碑，连接顶点的边表示包含的活动，边上的数字表示相应活动的持续时间（天），则完成该项目的最少时间为（ ）天。活动 BC 和 BF 最多可以晚开始（ ）天而不会影响整个项目的进度。

- A. 11
- B. 15
- C. 16
- D. 18



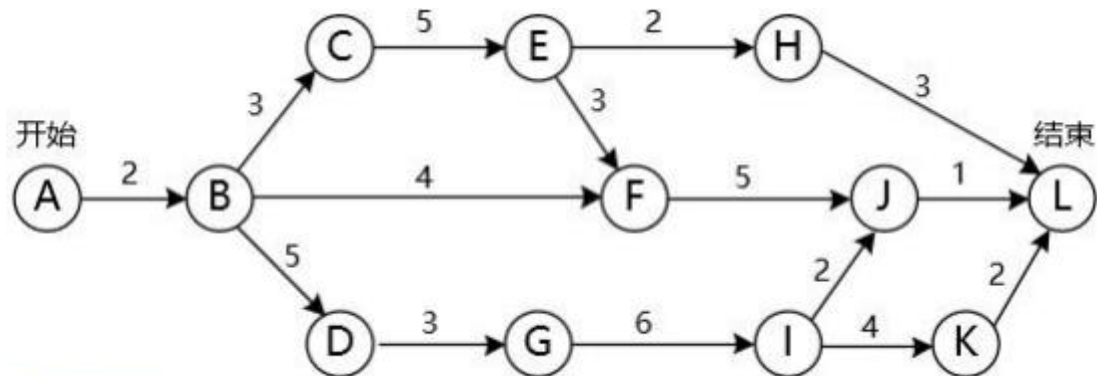
- A. 0 和 7
- B. 0 和 11
- C. 2 和 7
- D. 2 和 11

● 成本估算时，（ ）方法以规模作为成本的主要因素，考虑多个成本驱动因子。该方法包括三个阶段模型，即应用组装模型、早期设计阶段模型和体系结构阶段模型。

- A. 专家估算
- B. Wolverton
- C. COCOMO
- D. COCOMO II

2016 年上半年

● 某软件项目的活动图如下图所示，其中顶点表示项目里程碑，连接顶点的边表示包含的活动，边上的数字表示活动的持续时间（天），则完成该项目的最少时间为（15）天。活动 BD 最多可以晚开始（16）天而不会影响整个项目的进度。



- | | |
|-------|-------|
| A. 15 | B. 21 |
| C. 22 | D. 24 |
| | |
| A. 0 | B. 2 |
| C. 3 | D. 5 |

● 在结构化分析中，用数据流图描述（17）。当采用数据流图对一个图书馆管理系统进行分析时，（18）是一个外部实体。

- A. 数据对象之间的关系，用于对数据建模
- B. 数据在系统中如何被传送或变换，以及如何对数据流进行变换的功能或子功能，用于对功能建模
- C. 系统对外部事件如何响应，如何动作，用于对行为建模
- D. 数据流图中的各个组成部分

- A. 读者
- B. 图书
- C. 借书证
- D. 借阅

● 软件开发过程中，需求分析阶段的输出不包括（19）。

- A. 数据流图
- B. 实体联系图
- C. 数据字典
- D. 软件体系结构图

第二部分

2018 年上半年

- 以下关于增量模型的叙述中, 不正确的是()。
 - A. 容易理解, 管理成本低
 - B. 核心的产品往往首先开发, 因此经历最充分的“测试”
 - C. 第一个可交付版本所需要的成本低, 时间少
 - D. 即使一开始用户需求不清晰, 对开发进度和质量也没有影响

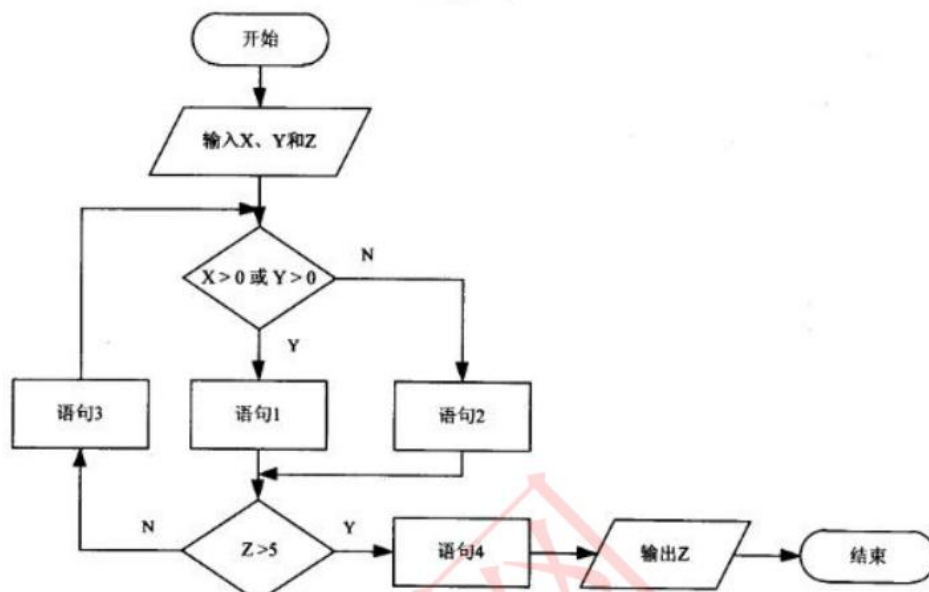
- 能力成熟度模型集成 (CMMI) 是若干过程模型的综合和改进。连续式模型和阶段式模型是 CMMI 提供的两种表示方法。连续式模型包括 6 个过程域能力等级 (Capability Level, CL), 其中()的共性目标是过程将可标识的输入工作产品转换成可标识的输出工作产品, 以实现支持过程域的特定目标。
 - A. CL1(已执行的)
 - B. CL2(已管理的)
 - C. CL3(已定义的)
 - D. CL4(定量管理的)

- 软件维护工具不包括()工具。
 - A. 版本控制
 - B. 配置管理
 - C. 文档分析
 - D. 逆向工程

- 概要设计文档的内容不包括()。
 - A. 体系结构设计
 - B. 数据库设计
 - C. 模块内算法设计
 - D. 逻辑数据结构设计

- 耦合是模块之间的相对独立性(互相连接的紧密程度)的度量。耦合程度不取决()。
 - A. 调用模块的方式
 - B. 各个模块之间接口的复杂程度
 - C. 通过接口的信息类型
 - D. 模块提供的功能数

● 对下图所示的程序流程图进行判定覆盖测试, 则至少需要 () 个测试用例。采用 McCabe 度量法计算其环路复杂度为 () 。



- A. 2
C. 4
- B. 3
D. 5

- A. 2
C. 4
- B. 3
D. 5

● 软件调试的任务就是根据测试时所发现的错误, 找出原因和具体的位置, 进行改正。其常用的方法中, () 是指从测试所暴露的问题出发, 收集所有正确或不正确的数据, 分析它们之间的关系, 提出假想的错误原因, 用这些数据来证明或反驳, 从而查出错误所在。

- A. 试探法
B. 回溯法
C. 归纳法
D. 演绎法

● 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

● 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。

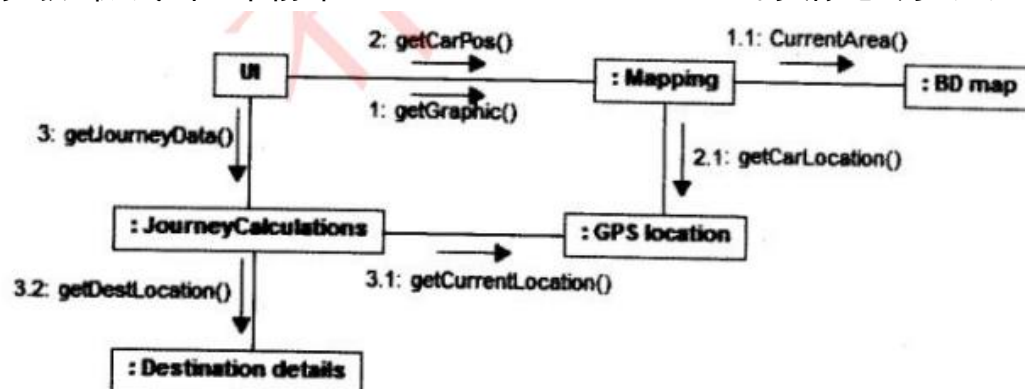
- A. 消息传递
- B. 类型检查
- C. 静态绑定
- D. 动态绑定

- A. 消息传递
- B. 类型检查
- C. 静态绑定
- D. 动态绑定

● 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 |

● 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 ()。

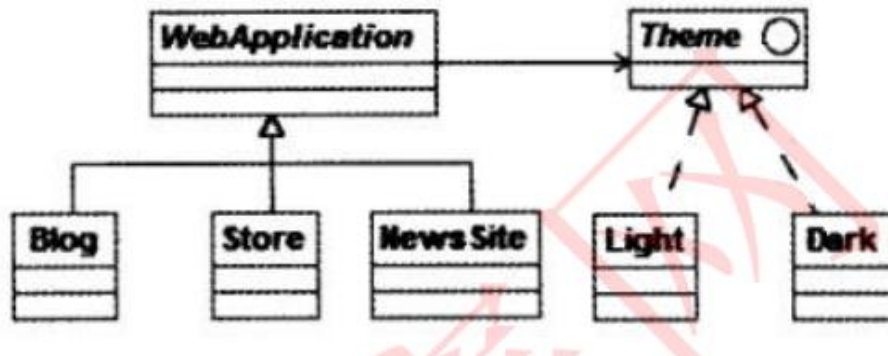


- | | |
|--------|--------|
| A. 类图 | B. 组件图 |
| C. 通信图 | D. 部署图 |

- 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

- A. 1: getGraphic()
- B. 2: getCarPos()
- C. 1.1: CurrentArea()
- D. 2.1: getCarLocation()

● 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 ()。



- | | |
|-------------------|------------------|
| A. 观察者 (Observer) | B. 访问者 (Visitor) |
| C. 策略 (Strategy) | D. 桥接 (Bridge) |

- | | |
|-------------------|----------|
| A. Webapplication | B. Blog |
| C. Theme | D. Light |

- | | |
|------------|------------|
| 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 |

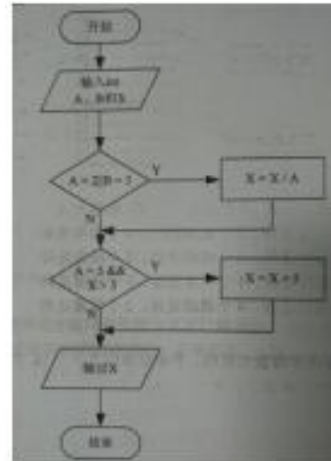
- | |
|-------------------------------|
| A. 将抽象部分与其实现部分分离, 使它们都可以独立地变化 |
| B. 动态地给一个对象添加一些额外的职责 |
| C. 为其他对象提供一种代理以控制对这个对象的访问 |
| D. 将一个类的接口转换成客户希望的另外一个接口 |

2017 年下半年

29. 用自盒测试方法对如下图所示的流程图进行测试。若要满足分支覆盖, 则至少要 () 个测试用例, 正确的测试用例对是 () (测试用例的格式为 (A, B, X; X))。

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

- A. (1, 3, 3;3) 和 (5, 2, 15;3)
- B. (1, 1, 5;5) 和 (5, 2, 20;9)
- C. (2, 3, 10;5) 和 (5, 2, 18;3)
- D. (5, 2, 16;3) 和 (5, 2, 21;9)



31. 配置管理贯穿软件开发的整个过程。以下内容中, 不属于配置管理的是 ()。

- A. 版本控制
- B. 风险管理
- C. 变更管理
- D. 配置状态报告

32. 极限编程 (XP) 的十二个最佳实践不包括 ()。

- A. 小的发布
- B. 结对编程
- C. 持续集成
- D. 精心设计

33. 以下关于管道过滤器体系结构的有点的叙述中, 不争取的是 ()。

- A. 软件构件具有良好的高内聚、低耦合的特点
- B. 支持重用
- C. 支持并行执行
- D. 提高性能

34. 模块 A 将学生信息, 即学生姓名、学号、手机号等放到一个结构体中, 传递给模块 B。模块 A 和 B 之间的藕合类型为 () 藕合。

- | | |
|-------|-------|
| A. 数据 | B. 标记 |
| C. 控制 | D. 内容 |

35. 某模块内涉及多个功能, 这些功能必须以特定的次序执行, 则该模块的内聚类型为 () 内聚。

- A. 实践
- B. 过程
- C. 信息
- D. 功能

36. 系统交付用户使用后, 为了改进系统的图形输出而对系统进行修改的维护行为属于 () 维护。

- A. 改正性
- B. 适应性
- C. 改善性
- D. 预防性

37. 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。

- | | |
|-------|-------|
| A. 抽象 | B. 继承 |
| C. 封装 | D. 多态 |

- 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

39. 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

40. 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 |

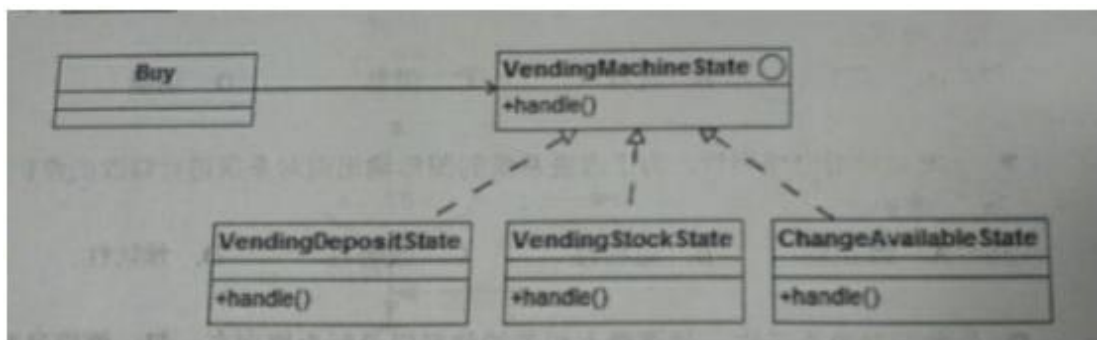
41. 如图所示的 UML 类图中, Shop 和 Magazine 之间为()关系, Magazine 和 Page 之间为()关系。UML 类图统称不用于对()进行建模。

- | | |
|-------|-------|
| A. 关联 | B. 依赖 |
| C. 组合 | D. 继承 |

- | | |
|-------|-------|
| A. 关联 | B. 依赖 |
| C. 组合 | D. 继承 |

- 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

44. 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()。



- | | |
|-------------------|------------------|
| A. 观察者 (Observer) | B. 状态 (State) |
| C. 策略 (Strategy) | D. 访问者 (Visitor) |

- | | |
|------------------------|----------------------|
| A. VendingMachineState | B. Buy |
| C. VendingDepositState | D. VendingStockState |

- 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

- 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

2017 年上半年

- 以下关于螺旋模型的叙述中，不正确的是()
 - A. 它是风险驱动的，要求开发人员必须具有丰富的风险评估知识和经验
 - B. 它可以降低过多测试或测试不足带来的风险
 - C. 它包含维护周期，因此维护和开发之间没有本质区别
 - D. 它不适用于大型软件开发
- 以下关于极限编程(XP) 中结对编程的叙述中，不正确的是()。
 - A. 支持共同代码拥有和共同对系统负责
 - B. 承担了非正式的代码审查过程
 - C. 代码质量更高
 - D. 编码速度更快
- 以下关于 C/S (客户机/服务器)体系结构的优点的叙述中，不正确的是()
 - A. 允许合理地划分三层的功能，使之在逻辑上保持相对独立性
 - B. 允许各层灵活地选用平台和软件
 - C. 各层可以选择不同的开发语言进行并行开发
 - D. 系统安装、修改和维护均只在服务器端进行
- 在设计软件的模块结构时， ()不能改进设计质量。
 - A. 尽量减少高扇出结构
 - B. 模块的大小适中
 - C. 将具有相似功能的模块合并
 - D. 完善模块的功能
- 模块 A、B 和 C 有相同的程序块，块内的语句之间没有任何联系，现把改程序块取出来，形成新的模块 D，则模块 D 的内聚类型为()内聚。以下关于该内聚类型的叙述中，不正确的是()。
 - A. 巧合
 - B. 逻辑
 - C. 时间
 - D. 过程
 - A. 具有最低的内聚性
 - B. 不易修改和维护
 - C. 不易理解
 - D. 不影响模块间的耦合关系

● 对下图所示的程序流程图进行语句覆盖测试和路劲覆盖测试，至少需要()个测试用例。采用 McCabe 度量法计算其环路复杂度为()。

- A. 2 和 3
- B. 2 和 4
- C. 2 和 5
- D. 2 和 6

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

● 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。

- A. 多重继承
- B. 多态
- C. 封装
- D. 层次继承

- A. 动态
- B. 私有
- C. 公共
- D. 二义性

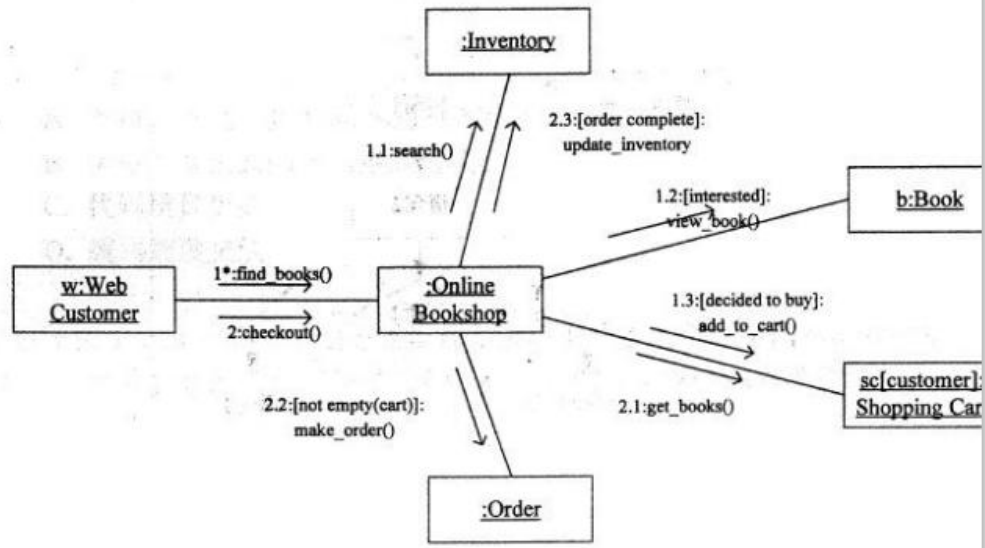
● 采用面向对象方法进行软件开发，在分析阶段，架构师主要关注系统的()。

- A. 技术
- B. 部署
- C. 实现
- D. 行为

● 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

● 以下 UML 图是 (41)，图中 :Order 和 b:Book 表示 (42)，1*:find 和 1.1:search() 表示 (43)。



- A. 序列图
- C. 通信图

- B. 状态图
- D. 活动图

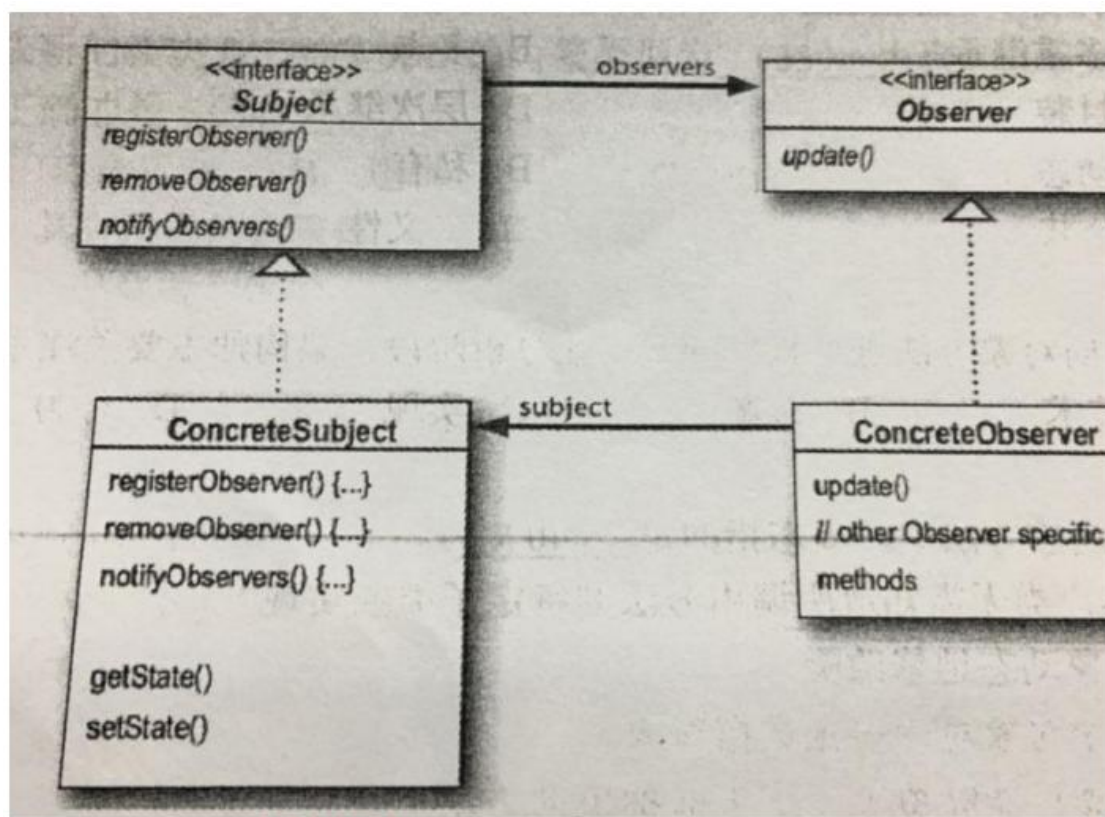
- A. 类
- C. 流名称

- B. 对象
- D. 消息

- A. 类
- C. 流名称

- B. 对象
- D. 消息

● 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, 可以有任意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 ()



- A. Subject
B. Observer
C. ConcreteSubject
D. ConcreteObserver

- 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

● 装饰器 (Decorator) 模式用于 (); 外观 (Facade) 模式用于 ()。

- ① 将一个对象加以包装以给客户提供其希望的另外一个接口
② 将一个对象加以包装以提供一些额外的行为
③ 将一个对象加以包装以控制对这个对象的访问
④ 将一系列对象加以包装以简化其接口

- A. ①
B. ②
C. ③
D. ④

- A. ①
B. ②
C. ③
D. ④

2016 年下半年

● 某开发小组欲为一公司开发一个产品控制软件, 监控产品的生产和销售过程, 从购买各种材料开始, 到产品的加工和销售进行全程跟踪。购买材料的流程、产品的加工过程以及销售过程可能会发生变化。该软件的开发最不宜采用 () 模型, 主要是因为这种模型 ()。

- A. 瀑布
- B. 原型
- C. 增量
- D. 喷泉

- A. 不能解决风险
- B. 不能快速提交软件
- C. 难以适应变化的需求
- D. 不能理解用户的需求

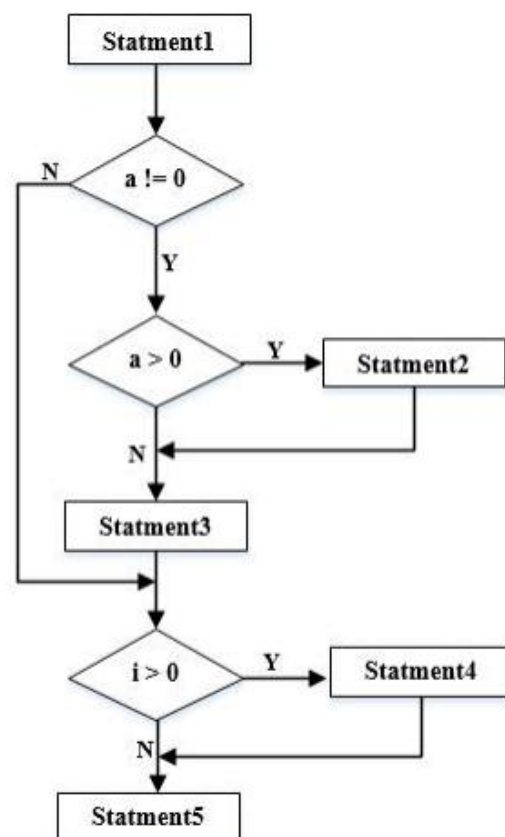
● () 不属于软件质量特性中的可移植性。

- A. 适应性
- B. 易安装性
- C. 易替换性
- D. 易理解性

● 对下图所示流程图采用白盒测试方法进行测试, 若要满足路径覆盖, 则至少需要 () 个测试用例。采用 McCabe 度量法计算该程序的环路复杂性为 ()。

- A. 3
- B. 4
- C. 6
- D. 8

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



● 计算机系统的（ ）可以用 $MTBF/(1+MTBF)$ 来度量，其中 MTBF 为平均失效间隔时间。

- A. 可靠性
- B. 可用性
- C. 可维护性
- D. 健壮性

● 以下关于软件测试的叙述中，不正确的是（ ）。

- A. 在设计测试用例时应考虑输入数据和预期输出结果
- B. 软件测试的目的是证明软件的正确性
- C. 在设计测试用例时，应该包括合理的输入条件
- D. 在设计测试用例时，应该包括不合理的输入条件

● 某模块中有两个处理 A 和 B，分别对数据结构 X 写数据和读数据，则该模块的内聚类型为（ ）内聚。

- A. 逻辑
- B. 过程
- C. 通信
- D. 内容

● 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

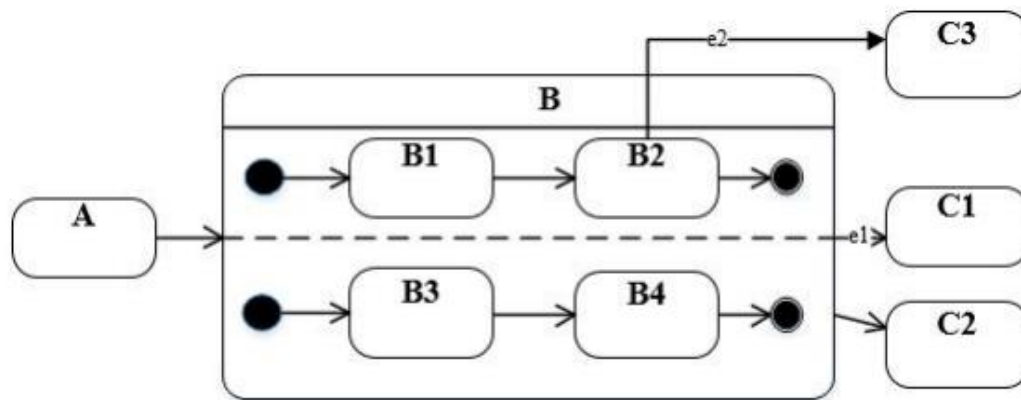
● 在面向对象方法中，支持多态的是（ ）。

- A. 静态分配
- B. 动态分配
- C. 静态类型
- D. 动态绑定

● 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（ ）。

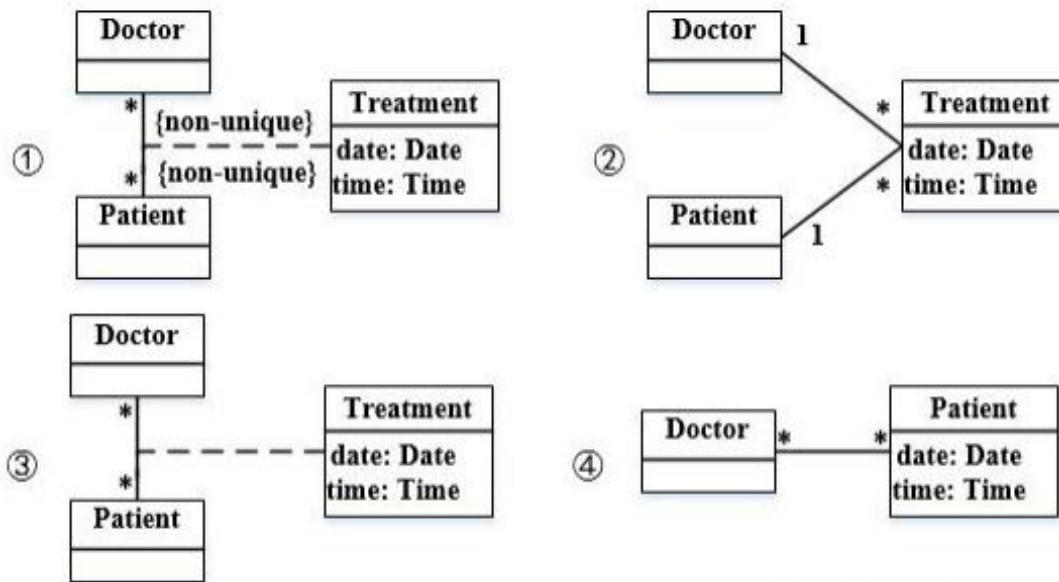
- A. 认定并组织对象
- B. 描述对象间的相互作用
- C. 面向对象程序设计
- D. 确定基于对象的操作

- 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
- 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

● 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。



- A. 用例图
B. 对象图
C. 类图
D. 协作图

- A. ①
B. ②
C. ③
D. ④

● () 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它对象
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

- A. 命令 (Command)
B. 责任链 (Chain of Responsibility)
C. 观察者 (Observer)
D. 策略 (Strategy)

- A. ①②
B. ②③
C. ③④
D. ①④

● () 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

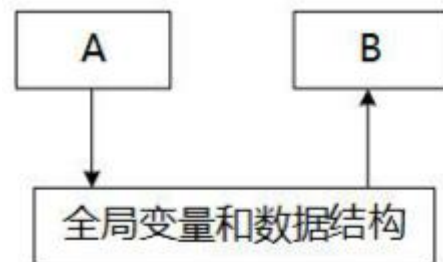
- A. 生成器 (Builder)
- B. 工厂方法 (Factory Method)
- C. 原型 (Prototype)
- D. 单例 (Singleton)

- A. ①②
- B. ②③
- C. ③④
- D. ①④

2016 年上半年

● 如下图所示，模块 A 和模块 B 都访问相同的全局变量和数据结构，则这两个模块之间的耦合类型为（29）耦合。

- A. 公共
- B. 控制
- C. 标记
- D. 数据



● 以下关于增量开发模型的叙述中，不正确的是（30）。

- A. 不必等到整个系统开发完成就可以使用
- B. 可以使用较早的增量构件作为原型，从而获得稍后的增量构件需求
- C. 优先级最高的服务先交付，这样最重要的服务接受最多的测试
- D. 有利于进行好的模块划分

● 在设计软件的模块结构时，（31）不能改进设计质量。

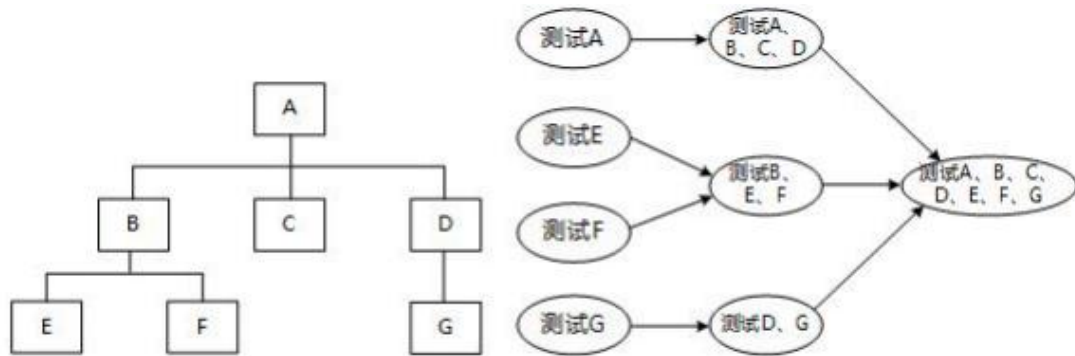
- A. 模块的作用范围应在其控制范围之内
- B. 模块的大小适中
- C. 避免或减少使用病态连接（从中部进入或访问一个模块）
- D. 模块的功能越单纯越好

● 软件体系结构的各种风格中，仓库风格包含一个数据仓库和若干个其他构件。数据仓库位于该体系结构的中心，其他构件访问该数据仓库并对其中的数据进行增、删、改等操作。以下关于该风格的叙述中，不正确的是（32）。（33）不属于仓库风格。

- A. 支持可更改性和可维护性
- B. 具有可复用的知识源
- C. 支持容错性和健壮性
- D. 测试简单

- A. 数据库系统
- B. 超文本系统
- C. 黑板系统
- D. 编译器

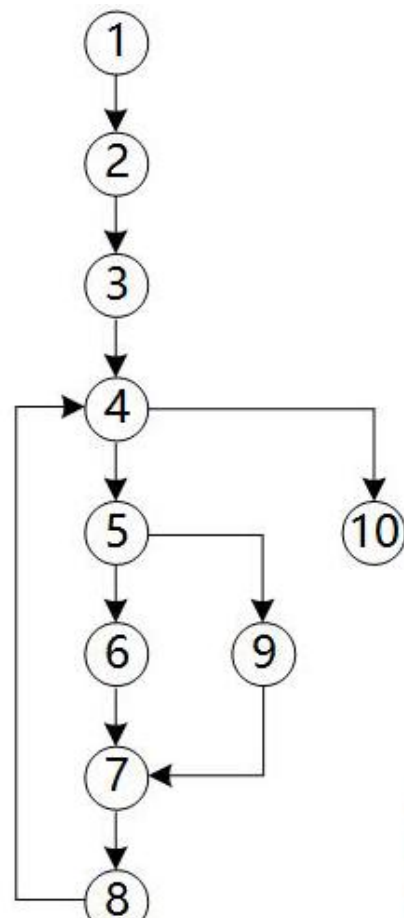
● 下图（a）所示为一个模块层次结构的例子，图（b）所示为对其进行集成测试的顺序，则此测试采用了（34）测试策略。该测试策略的优点不包括（35）。



- A. 自底向上
 - B. 自顶向下
 - C. 三明治
 - D. 一次性
-
- A. 较早地验证了主要的控制和判断点
 - B. 较早地验证了底层模块
 - C. 测试的并行程度较高
 - D. 较少的驱动模块和桩模块的编写工作量

● 采用 McCabe 度量法计算下图所示程序的环路复杂性为（36）。

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



● 在面向对象方法中，（37）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（38）。

- | | |
|-------|-------|
| A. 封装 | B. 继承 |
| C. 覆盖 | D. 多态 |
| A. 封装 | B. 继承 |
| C. 覆盖 | D. 多态 |

● 在 UML 用例图中，参与者表示（39）。

- 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

● UML 中关联是一个结构关系，描述了一组链。两个类之间（40）关联。

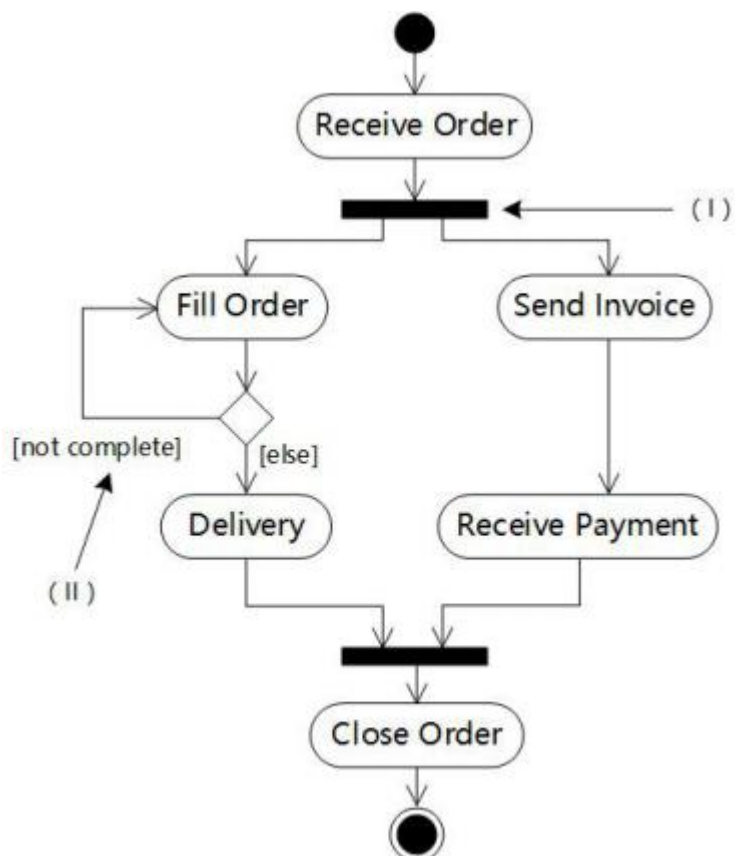
- A. 不能有多个
 B. 可以有多个由不同角色标识的
 C. 可以有任意多个
 D. 的多个关联必须聚合成一个

● 如下所示的 UML 图是（41），图中（I）表示（42），（II）表示（43）。

- A. 序列图
 B. 状态图
 C. 通信图
 D. 活动图

- A. 合并分叉
 B. 分支
 C. 合并汇合
 D. 流

- A. 分支条件
 B. 监护表达式
 C. 动作名
 D. 流名称



● 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（44）模式。

- A. 享元（Flyweight）
- B. 抽象工厂（Abstract Factory）
- C. 外观（Facade）
- D. 装饰器（Decorator）

● （45）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）

● （46）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）

● 因使用大量的对象而造成很大的存储开销时，适合采用（47）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（Memento）