

# 第 8 章系统分析与设计方法

## 8.2 需求分析与软件设计

### 8.2.1 需求分析的任务与过程

1. 【2009 年题 20】以下关于需求管理的叙述中, 正确的是( )。
  - A. 需求管理是一个对系统需求及其变更进行了解和控制的过程
  - B. 为了获得项目, 开发人员可以先向客户做出某些承诺
  - C. 需求管理的重点在于收集和分析项目需求
  - D. 软件开发过程是独立于需求管理的活动
2. 【2011 年题 19】通常有两种常用的需求定义方法: 严格定义方法和原型方法。下述的各种假设条件中, ( )不适合使用严格定义方法进行需求定义。
  - A. 所有需求都能够被预先定义
  - B. 开发人员与用户之间能够准确而清晰地交流
  - C. 需求不能在系统开发前被完全准确地说明
  - D. 采用图形(或文字)充分体现最终系统
3. 【2011 年题 20】下列关于软件需求管理或需求开发的叙述中, 正确的是( )。
  - A. 所谓需求管理是指对需求开发的管理
  - B. 需求管理包括: 需求获取、需求分析、需求定义和需求验证
  - C. 需求开发是将用户需求转化为应用系统成果的过程
  - D. 在需求管理中, 要求维持对用户原始需求和所有产品构件需求的双向跟踪
4. 【2012 年题 24】基于 UML 的需求分析过程的基本步骤为: 利用( )表示需求; 利用( )表示目标软件系统的总体架构。
  - (1)A.用例及用例图
  - B.包图及类图
  - C.剧情及序列图
  - D.组件图及部署图
  - (2)A.用例及用例图
  - B.包图及类图
  - C.剧情及序列图
  - D.组件图及部署图
5. 【2014 年题 16】系统建议方案中不应该包含的内容是( )。
  - A. 问题陈述
  - B. 项目范围
  - C. 候选方案及其可行性分析
  - D. 系统详细设计方案

### 8.2.2 如何进行系统设计

1. 【2018 年题 9】嵌入式系统设计一般要考虑低功耗，软件设计也要考虑低功耗设计，软件低功耗设计一般采用（ ）。
  - A. 结构优化、编译优化和代码优化
  - B. 软硬件协同设计、开发过程优化和环境设计优化
  - C. 轻量级操作系统、算法优化和仿真实验
  - D. 编译优化技术、软硬件协同设计和算法优化
2. 【2018 年题 23】软件概要设计包括设计软件的结构、确定系统功能模块及其相互关系，主要采用（ ）描述程序的结构。
  - A. 程序流程图、PAD 图和伪代码
  - B. 模块结构图、数据流图和盒图
  - C. 模块结构图、层次图和 HIPO 图
  - D. 程序流程图、数据流图和层次图
3. 【2018 年题 24】软件设计包括了四个既独立又相互联系的活动：高质量的（32）将改善程序结构和模块划分，降低过程复杂性；（33）的主要目标是开发一个模块化的程序结构，并表示出模块间的控制关系；（34）描述了软件与用户之间的交互关系。
  - (32) A. 程序设计
  - B. 数据设计
  - C. 算法设计
  - D. 过程设计
  - (33) A. 软件结构设计
  - B. 数据结构设计
  - C. 数据流设计
  - D. 分布式设计
  - (34) A. 数据架构设计
  - B. 模块化设计
  - C. 性能设计
  - D. 人机界面设计

## 8.3 结构化分析与设计

### 8.3.2 结构化设计

1. 【2015 年题 22】处理流程设计是系统设计的重要内容。以下关于处理流程设计工具的叙述中，不正确的是（ ）。
  - A. 程序流程图（PFD）用于描述系统中每个模块的输入，输出和数据加工
  - B. N-S 图容易表示嵌套关系和层次关系，并具有强烈的结构化特征
  - C. IPO 图的主体是处理过程说明，可以采用流程图、判定树/表等来进行描述
  - D. 问题分析图（PAD）包含 5 种基本控制结构，并允许递归使用
2. 【2015 年题 24】结构化程序设计采用自顶向下、逐步求精及模块化的程序设计方法，通过（ ）三种基本的控制结构可以构造出任何单入口单出口的程序。
  - A. 顺序、选择和嵌套
  - B. 顺序、分支和循环

- C. 分支、并发和循环
- D. 跳转、选择和并发

## 8.4 面向对象的分析与设计

### 8.4.1 面向对象的基本概念

1. 【2011 年题 22】在面向对象设计中, 用于描述目标软件与外部环境之间交互的类被称为( ), 它可以 ( )。
  - A. 实体类
  - B. 边界类
  - C. 模型类
  - D. 控制类
  - A. 表示目标软件系统中具有持久意义的信息项及其操作
  - B. 协调、控制其他类完成用例规定的功能或行为
  - C. 实现目标软件系统与外部系统或外部设备之间的信息交流和互操作
  - D. 分解任务并把子任务分派给适当的辅助类
2. 【2013 年题 22】在面向对象设计中, ( )可以实现界面控制、外部接口和环境隔离。( )作为完成用例业务的责任承担者, 协调、控制其他类共同完成用例规定的功能或行为。
  - A. 实体类
  - B. 控制
  - C. 边界类
  - D. 交互类
  - A. 实体类
  - B. 控制
  - C. 边界类
  - D. 交互类
3. 【2017 年题 25】面向对象的分析模型主要由顶层架构图、用例与用例图和( )构成; 设计模型则包含以( )表示的软件体系机构图、以交互图表示的用例实现图、完整精确的类图、描述复杂对象的( )和用以描述流程化处理过程的活动图等。
  - (1) A. 数据流模型
  - B. 领域概念模型
  - C. 功能分解图
  - D. 功能需求模型
  - (2) A. 模型试图控制器
  - B. 组件图
  - C. 包图
  - D. 2 层、3 层或 N 层
  - (3) A. 序列图
  - B. 协作图
  - C. 流程图
  - D. 状态图

### 8.4.3 统一建模语言

1. 【2009 年题 25】用例(use case)用来描述系统对事件做出响应时所采取的行动。用例之间是具有相关性的。在一个“订单输入子系统”中, 创建新订单和更新订单都需要核查用户帐号是否正确。用例“创建新订单”、“更新订单”与用例“核查客户帐号”之间是( )关系。
  - A. 包含(include)
  - B. 扩展(exclude)
  - C. 分类(classification)

- D. 聚集( aggregation )
2. 【2009 年题 26】面向对象的设计模型包含以( )表示的软件体系结构图, 以( )表示的用例实现图, 完整精确的类图, 针对复杂对象的状态图和用以描述流程化处理的活动图等。
- A. 部署图      B. 包图      C. 协同图      D. 交互图
- A. 部署图      B. 包图      C. 协同图      D. 交互图
3. 【2010 年题 24】在 RUP 中采用“4+1”视图模型来描述软件系统的体系结构。在该模型中, 最终用户侧重于( ), 系统工程师侧重于( )。
- (1)A. 实现视图  
B. 进程视图  
C. 逻辑视图  
D. 部署视图
- (2)A. 实现视图  
B. 进程视图  
C. 逻辑视图  
D. 部署视图
4. 【2014 年题 24】在 UML 提供的系统视图中, ( )是逻辑视图的一次执行实例, 描述了并发与同步结构; ( )是最基本的需求分析模型。
- A. 进程视图    B. 实现视图    C. 部署视图    D. 用例视图
- A. 进程视图    B. 实现视图    C. 部署视图    D. 用例视图
5. 【2014 年题 31】“4+1”视图主要用于描述系统逻辑架构, 最早由 Philippe Kruchten 于 1995 年提出。其中( )视图用于描述对象模型, 并说明系统应该为用户提供哪些服务。当采用面向对象的设计方法描述对象模型时, 通常使用( )表达类的内部属性和行为, 以及类集合之间的交互关系; 采用( )定义对象的内部行为。
- A. 逻辑    B. 过程    C. 开发    D. 物理
- A. 对象图    B. 活动图    C. 状态图    D. 类图
- A. 对象图    B. 活动图    C. 状态图    D. 类图
6. 【2015 年题 23】用例 (use case) 用来描述系统对事件做出响应时所采取的行动。用例之间是具有相关性的。在一个会员管理系统中, 会员注册时可以采用电话和邮件两种方式。用例“会员注册”和“电话注册”、“邮件注册”之间是( )关系。
- A. 包含 (include)    B. 扩展 (extend)    C. 泛化 (generalize)    D. 依赖 (depends on)
7. 【2018 年题 31】体系结构模型的多视图表示是从不同的视角描述特定系统的体系结构。著名的 4+1 模型支持从( )描述系统体系结构。
- A. 逻辑视图、开发视图、物理视图、进程视图、统一的场景
- B. 逻辑视图、开发视图、物理视图、模块视图、统一的场景
- C. 逻辑视图、开发视图、构件视图、进程视图、统一的场景
- D. 领域视图、开发视图、构件视图、进程视图、统一的场景

#### 8.4.4 其他

1. 【2011 年题 23】最少知识原则(也称为迪米特法则)是面向对象设计原则之一,是指一个软件实体应当尽可能少地与其他实体发生相互作用。这样,当一个实体被修改时,就会尽可能少地影响其他的实体。下列叙述中,“( )”不符合最少知识原则。
  - A. 在类的划分上,应当尽量创建松耦合的类
  - B. 在类的设计上,只要有可能,一个类型应当设计成不变类
  - C. 在类的结构设计上,每个类都应当尽可能提高对其属性和方法的访问权限
  - D. 在对其他类的引用上,一个对象对其他对象的引用应当降到最低
2. 【2012 年题 20】对于违反里氏替换原则的两个类 A 和 B,可以来用的候选解决方案中,正确的是( )。
  - A. 尽量将一些需要扩展的类或者存在变化的类设计为抽象类或者接口,并将其作为基类,在程序中尽量使用基类对象进行编程
  - B. 创建一个新的抽象类 C,作为两个具体类的超类,将 A 和 B 共同的行为移动到 C 中,从而解决 A 和 B 行为不完全一致的问题
  - C. 将 B 到 A 的继承关系改成组合关系
  - D. 区分是“Is-a”还是“Has-a”。如果是 Is-a,可以使用继承关系,如果是 Has-a,应该改成组合或聚合关系
3. 【2015 年题 26】在面向对象设计的原则中、( )原则是指抽象不应该依赖于细节,细节应该依赖于抽象,即应针对接口编程,而不是针对实现编程。
  - A. 开闭
  - B. 里氏替换
  - C. 最少知识
  - D. 依赖倒置
4. 【2016 年题 21】面向对象的分析模型主要由( )、用例与用例图、领域概念模型构成;设计模型则包含以包图表示的软件体系结构图、以交互图表示的( )、完整精确的类图、针对复杂对象的状态图和描述流程化处理过程的( )等。
  - A. 业务活动图
  - B. 顶层架构图
  - C. 数据流模型
  - D. 实体联系图
  - A. 功能分解图
  - B. 时序关系图
  - C. 用例实现图
  - D. 软件部署图
  - A. 序列图
  - B. 协作图
  - C. 流程图
  - D. 活动图

### 8.5 用户界面设计

#### 8.5.1 用户界面设计的原则

1. 【2009 年题 32】系统输入设计中应尽可能考虑人的因素,以下关于输入设计的一般原理中,错误的是( )。
  - A. 只让用户输入变化的数据
  - B. 使用创新的模式吸引用户的眼球
  - C. 表格中各个数据项应有提示信息
  - D. 尽可能使用选择而不是键盘输入的方式获取数据
2. 【2014 年题 20】下列关于用户界面设计的叙述中,错误的是( )。
  - A. 界面交互模型应经常进行修改
  - B. 界面的视觉布局应该尽量与真实世界保持一致
  - C. 所有可视信息的组织需要按照统一的设计标准
  - D. 确保用户界面操作和使用的一致性

### 8.5.3 其他

1. 【2010 年题 35】系统输入设计中,采用内部控制方式以确保输入系统数据的有效性,( )用于验证数据是否位于合法的取值范围。
  - A. 数据类型检查
  - B. 自检位
  - C. 域检查
  - D. 格式检查

### 8.10 其他

1. 【2014 年题 17】下列关于联合需求计划(Joint Requirement Planning, JRP)的叙述中, 不正确的是( )。
  - A. 在 JRP 实施之前, 应制定详细的议程, 并严格遵照议程进行
  - B. 在讨论期间尽量避免使用专业术语
  - C. JRP 是一种相对来说成本较高但十分有效的需求获取方法
  - D. JRP 的主要目的是对需求进行分析和验证