第 9 章软件架构设计

9.1 软件架构概述

9.1.1 软件架构的定义

1. 【2009 年题 35 解析】

软件架构贯穿于软件的整个生命周期，但在不同的阶段对软件架构的关注力度并不相同。

需求分析阶段主要关注问题域；

设计阶段主要将需求转换为软件架构模型；

软件实现阶段主要关注将架构设计转换为实际的代码；

软件部署阶段主要通过组装软件组件提高系统的实现效率。

其中设计与实现阶段在软件架构上的工作最多，也最重要，因此关注力度最大。

2. 【2009 年题 37 解析】

软件架构是降低成本、改进质量、按时和按需交付产品的关键因素，软件架构设计需要满足系统的质量属性，如性能、安全性和可修改性等，软件架构设计需要确定组件之间的依赖关系，支持项目计划和管理活动，软件架构能够指导设计人员和实现人员的工作。一般在设计软件架构之初，会根据用户需求，确定多个候选架构，并从中选择一个较优的架构，并随着软件的开发，对这个架构进行微调，以达到最佳效果，A 选项错误。

3. 【2009 年题 38 解析】

软件架构设计包括提出架构模型、产生架构设计和进行设计评审等活动，是一个迭代的过程，在建立软件架构的初期，一般需要选择一个合适的架构风格，将架构分析阶段已标识的构件映射到架构中，并分析这些构件之间的关系，一旦得到了详细的软件架构设计，需要邀请独立于系统开发的外部人员对系统进行评审。一般来说，软件架构设计活动将已标识构件集成到软件架构中，设计这些构件，但不予以实现，C 选项错误。

4. 【2011 年题 35 解析】

本题考查“4+1”视图。

“4+1”视图中的“4”，指的是：逻辑视图、开发视图、进程视图、物理视图，“1”指的是场景视图。

场景视图又称为用例视图，显示外部参与者观察到的系统功能。

逻辑视图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。

开发视图又称为实现视图，显示的是源代码以及实际执行代码的组织结构。

处理视图又称为进程视图，显示程序执行时并发的状态。

物理视图展示软件到硬件的映射。

【答案】A、D、C。

5. 【2012 年题 16 解析】

本题主要考查以架构为核心的软件系统开发方法。在该方法中，架构用来激发和调整设计策略，不同的视图用来表达与质量目标有关的信息。架构设计是一个迭代过程，在建立软件架构的初期，**选择一个合适的架构风格是首要的**，在此基础上，开发人员**通过架构模型**，可以**获得关于软件架构属性的理解**，为将来的架构实现与演化过程建立了目标。

【答案】C、B。

本题主要考查 ANSI/IEEE 1471-2000 标准的相关知识。在 ANSI/IEEE 1471-2000 标准中，系统是为了达成利益相关人(Stakeholder)的某些使命(Mission)，在特定环境(Enviroment)中构建的。每一个系统都有一个架构(Architecture)。

架构(Architecture)是对所有利益相关人的关注点(Concern)的响应和回答，通过架构描述(Architecture Description)来说明。每一个利益相关人都有各自的关注点。这些关注点是指对其重要的，与系统的开发、运营或其它方面相关的利益。架构描述(Architecture Description)本质上是多视图的。

每一个**视图(View)是从一个特定的视角(Viewpoint)来表述架构的某一个独立的方面**。试图用一个单一的视图来覆盖所有的关注点当然是最好的，但实际上这种表述方式将很难理解。

视角(Viewpoint)的选择，基于要解决哪些利益相关人的哪些关注点。它决定了用来创建视图的语言、符号和模型等，以及任何与创建视图相关的建模方法或者分析技术。一个视图(View)包括一个或者多个架构模型(Model)，一个模型也可能参与多个视图。模型较文本的表述的好处在于，可以更容易的可视化、检查、分析、管理和集成。

【答案】：D、C、A。

6. 【2013 年题 28 解析】

软件系统架构是关于软件系统的**结构、行为和属性**的高级抽象。在描述阶段，其对象是直接构成系统的抽象组件以及各个组件之间的连接规则，特别是相对细致地描述组件之间的**通讯**。在实现阶段，这些抽象组件被细化为实际的组件，比如具体类或者对象。软件系统架构不仅指定了软件系统的**组织结构和拓扑结构**，而且显示了系统需求和构成组件之间的对应关系，包括设计决策的基本方法和基本原理。

【答案】A、A、B。

7. 【2013 年题 30 解析】

软件架构能够在设计变更相对容易的阶段，考虑系统结构的可选方案，便于技术人员与非技术人员就软件设计进行交互，能够展现软件的结构、属性与内部交互关系。但是软件架构与用户对系统的功能性需求没有直接的对应关系。

【答案】D。

8. 【2014 年题 28 解析】

软件架构设计不能捕获需求，软件架构设计是在需求捕获并进行分析之后开展的工作。

9. 【2014 年题 29 解析】

从本质上看，需求和软件架构设计面临的是不同的对象：一个是问题空间；另一个是解空间。保持两者的可追踪性和转换，一直是软件工程领域追求的目标。从软件需求模型向 SA 模型的转换主要关注两个问题：

1、如何根据需求模型构建软件架构模型；

2、如何保证模型转换的可追踪性。

本题中选项 A 与 B 是软件架构设计阶段需要考虑的问题，而选项 D 是软件架构实现阶段中需要考虑的问题。

10. 【2015 年题 12 解析】

引入对象管理层不但不能提高性能，反而会降低系统性能。这个道理与分层模型中增加层次是一样的。

11. 【2015 年题 29 解析】

软件架构设计的一个核心问题是能否使用重复的架构模式，即能否达到架构级的软件重用。也就是说，能否在不同的软件系统中，使用同一架构。基于这个目的，学者们开始研究和实践软件架构的风格和类型问题。

软件架构风格是描述某一特定应用领域中系统组织方式的惯用模式。**它反映了领域中众多系统所共有的结构和语义特性**，并指导如何将各个模块和子系统有效地组织成一个完整的系统。按这种方式理解，软件架构风格定义了用于描述系统的术语表和一组指导构件系统的规则。

对软件架构风格的研究和实践促进了对**设计的复用**，一些经过实践证实的解决方案也可

以可靠地用于解决新的问题。架构风格的不变部分使不同的系统可以共享同一个实现代码。只要系统是使用常用的、规范的方法来组织，就可使别的设计者很容易地理解系统的架构。例如，如果某人把系统描述为"客户/服务器"模式，则不必给出设计细节，我们立刻就会明白系统是如何组织和工作的。

【答案】：A、B。

12. 【2015 年题 38 解析】

软件架构设计包括提出架构模型、产生架构设计和进行设计评审等活动，是一个迭代的过程。架构设计主要关注软件组件的结构、属性和交互作用，并通过多种视图全面描述特定系统的架构。

13. 【2016 年题 32 解析】

【答案】：A、A。

9.1.2 软件架构的重要性

1. 【2009 年题 36 解析】

软件架构设计是降低成本、改进质量、按时和按需交付产品的关键因素。架构设计能够满足系统的性能、可维护性等品质；能够使得不同的利益相关人（stakeholders）达成一致的目标；能够支持项目计划和项目管理等活动；能够有效地管理复杂性；等等。然而系统架构的给出必须建立在需求明确的基础上，架构的设计应该是在需求明确之后才能开始，有先后顺序，B 选项错误。

9.1.3 架构的模型

1. 【2012 年题 17 解析】

架构描述语言(Architecture Description Language，ADL)是一种为明确说明软件系统的概念架构和对这些概念架构建模提供功能的语言。

ADL 主要包括以下组成部分：组件、组件接口、连接件和架构配置。ADL 对连接件的重视成为区分 ADL 和其他建模语言的重要特征之一。

【答案】C

2. 【2015 年题 31 解析】

架构描述语言(Architecture Description Language，ADL)是一种为明确说明软件系统的概念架构和对这些概念架构建模提供功能的语言。

ADL 主要包括以下组成部分：组件、组件接口、连接件和架构配置。ADL 对连接件的重视成为区分 ADL 和其他建模语言的重要特征之一。

【答案】C

9.2 架构需求与软件质量属性

9.2.1 软件质量属性

1. 【2010 年题 46 解析】

本题主要考查质量属性以及实现质量属性的一般策略，不同策略主要针对一个或多个软件质量属性，其中 Ping/Echo 主要提高系统的可用性；限制访问主要提高系统的安全性；运行时注册主要提高系统的可修改性；接口-实现分离主要提高系统的可修改性；主动冗余提高系统的可靠性；队列调度主要提高系统的性能；信息隐藏主要提高系统的可修改性；记录

-回放主要提高系统的可测试性，等等。

【答案】A、D、A。

2. 【2010 年题 47 解析】

本题主要考查软件质量属性的判断与应用。“系统出错后不能在要求的时间内恢复到正常状态”，这是对系统错误恢复能力的描述，属于系统可用性的范畴。“对系统进行二次开发时总要超过半年的时间”，这是对系统进行调整和维护方面能力的描述，属于系统可修改性的范畴。

3. 【2011 年题 43 解析】

题干中描述“当车库门正常下降时，如果发现下面有障碍物，则系统停止下降的时间需要控制在 0.1 秒内”这是对系统响应时间的要求，属于性能质量属性；“系统需要为部署在远程 PC 机上的智能家居系统留有控制接口，并支持在智能家居系统中对该系统进行远程错误诊断与调试”，这是对系统测试和调试方面的描述，属于系统的可测试性质量属性。

4. 【2011 年题 44 解析】

本题考查提高质量属性的常见手段。

提高可用性的手段包括：命令/响应机制、心跳机制、异常处理机制、冗余机制等。

提高性能的手段包括：引入并发、维持数据或计算的多个副本、增加可用资源、控制采样频度、限制执行时间、固定优先级调度等。

提高安全性的手段包括：身份认证、限制访问、检测攻击、维护完整性等。

【答案】A、D、C。

5. 【2012 年题 10 解析】

本题主要考查考生对质量属性的理解和质量属性实现策略的掌握。

对于题干描述：“当系统面临断电故障后，需要在 1 小时内切换至备份站点并恢复正常运行”主要与**可用性**质量属性相关，通常可采用**心跳、Ping/Echo、主动冗余、被动冗余、选举**等架构策略实现该属性；“在并发用户数量为 1000 人时，用户的交易请求需要在 0.5 秒内得到响应”，主要与**性能**这一质量属性相关，实现该属性的常见架构策略包括：**增加计算资源、减少计算开销、引入并发机制、采用资源调度**等。“系统应该能够抵挡恶意用户的入侵行为，并进行报警和记录”主要与**安全性**质量属性相关，通常可采用**入侵检测、用户认证、用户授权、追踪审计**等架构策略实现该属性。

6. 【2014 年题 36 解析】

本题主要考查考生对质量属性的理解和质量属性实现策略的掌握。

对于题干描述：“当系统面临断电故障后，需要在 1 小时内切换至备份站点并恢复正常运行”主要与可用性质量属性相关，通常可采用心跳、Ping/Echo、主动冗余、被动冗余、选举等架构策略实现该属性；“在并发用户数量为 1000 人时，用户的交易请求需要在 0.5 秒内得到响应”，主要与性能这一质量属性相关，实现该属性的常见架构策略包括：增加计算资源、减少计算开销、引入并发机制、采用资源调度等。“对系统的小熊中间件进行替换时，替换工作需要在 5 人/月内完成”主要与可修改性质量属性相关，通常可采用接口-实现分离、抽象、信息隐藏等架构策略实现该属性。

7. 【2015 年题 30 解析】

软件架构是降低成本、改进质量、按时和按需交付产品的关键因素，**软件架构设计需要满足系统的质量属性，如性能、安全性和可修改性等**，软件架构设计需要确定组件之间的依赖关系，支持项目计划和管理活动，软件架构能够指导设计人员和实现人员的工作。一般在设计软件架构之初，会根据用户需求，确定多个候选架构，并从中选择一个较优的架构，并随着软件的开发，对这个架构进行微调，以达到最佳效果。

8. 【2015 年题 40 解析】

“系统主站断电后，能够在 2 分钟内自动切换到备用站点，并恢复正常运行”，表达的

是在出问题后的恢复能力，属于可用性范畴。主动冗余是提高可用性的有效手段。

“在并发用户数不超过 1000 人时，用户的交易请求应该在 0.5s 内完成”，这是对性能的量化指标，属于性能的范畴。有效的资源调度能提升性能。

“系统应该能够抵挡恶意用户的入侵行为，并进行报警和记录”，这是安全方面的要求，在系统中，一般会用日志记录相关信息，然后通过对日志进行的审计能了解相关情况。

【答案】： B 、A、C、B、B、C 。

9. 【2016 年题 30 解析】

可修改性包含四个方面。

可维护性(maintainability)、可扩展性(extendibility)、结构重组(reassemble）、可移植性(portability)。

10. 【2017 年题 37 解析】

“机器人系统主电源断电后，能够在 10 秒内自动启动备用电源并进行切换，恢复正常运行”属于可用性，因为场景描述的是故障恢复问题。主动冗余是可用性的常见策略。

“对机器人的远程控制命令应该进行加密，从而能够抵挡恶意的入侵破坏行为，并对攻击进行报警和记录”属于安全性，常见的策略是追踪审计。

答案：A、C、D、A、B、B。

11. 【2017 年题 39 解析】

“网站正常运行时，用户发起的交易请求应该在 3 秒内完成”属于性能，常见架构策略包括：增加计算资源、减少计算开销、引入并发机制、采用资源调度等。

“在线交易主站宕机后，能够在 3 秒内自动切换到备用站点并恢复正常运行”属于可用性，因为场景描述的是故障恢复问题。通常可采用心跳、Ping/Echo、主动冗余、被动冗余、选举。

“系统应该具备一定的安全保护措施，从而能够抵挡恶意的入侵破坏行为，并对所有针对网站的攻击行为进行报警和记录”属于安全性，常见的策略是追踪审计。

12. 【2018 年题 37 解析】

能够在 15 秒内自动切换至备用系统并恢复正常运行”主要与**可用性（58 题）**质量属性相关。通常可采用心跳、Ping/Echo、主动冗余、被动冗余、选举等**（59 题）**架构策略实现该属性。

“系统正常运行时，人员信息查询请求应该在 2 秒内返回结果”主要与**性能（60 题）**质量属性相关，实现该属性的常见架构策略包括：增加计算资源、减少计算开销、引入并发机制、采用**资源调度（61 题）**等。

“系统应该能够抵挡恶意用户的入侵行为，并进行报警和记录”主要与**安全性（62 题）**质量属性相关，通常可采用入侵检测、用户认证、用户授权、**追踪审计（63 题）**等架构策略实现该属性。

答案 A、C、D、D、B、A

9.3 软件架构风格

1. 【2009 年题 41 解析】

Windows 操作系统在图形用户界面处理方面采用的是典型的“事件驱动”的架构风格。首先注册事件处理的是回调函数，当某个界面事件发生时（例如键盘敲击、鼠标移动等），系统会查找并选择合适的回调函数处理该事件。Java 语言是一种解释型语言，在 Java 虚拟机上运行，这从架构风格上看是典型的“虚拟机”风格，即通过虚拟机架构屏蔽不同的硬件环境。

2. 【2009 年题 42 解析】

根据题干描述，Web 服务器服务端的核心功能是数据处理，由于 Web 服务在数据传输

方面具有协议分层的特征，即底层协议会包装上层协议（HTTP 协议体中包含整个 SOAP 消息内容），因此需要数据内容的逐步分解与分阶段处理。比较选项中的架构风格，由于管道-过滤器的架构风格支持分阶段数据处理，因此特别适合该服务端处理软件的要求。

3. 【2009 年题 43 解析】

根据题干描述，调试器在设置端点时，其本质是在断点处设置一个事件监听函数，当程序执行到断点位置时，会触发并调用该事件监听函数，监听函数负责进行自动卷屏、刷新变量数值等动作。这是一个典型的回调机制，属于隐式调用的架构风格。

4. 【2009 年题 44 解析】

采用闭环结构的软件通常由几个协作构件共同构成，且其中的主要构件彼此分开，能够进行替换与重用，但**闭环结构通常适用于处理简单任务**（如机器装配等），**并不适用于复杂任务**。分层结构的特点是通过引入抽象层，在较低层次不确定的实现细节在较高层次会变得确定，并能够组织层间构件的协作，系统结构更加清晰。

【答案】A。

5. 【2009 年题 45 解析】

一个软件的架构设计是随着技术的不断进步而不断变化的。以编译器为例，其主流架构经历了管道-过滤器到数据共享为中心的转变过程。早期的编译器采用管道-过滤器架构风格，以文本形式输入的代码被逐步转化为各种形式，最终生成可执行代码。

早期的编译器采用管道-过滤器架构风格，并且大多数编译器在词法分析时创造独立的符号表，在其后的阶段会不断修改符号表，因此符号表并不是程序数据的一部分。

现代的编译器采用以数据共享为中心的架构风格，主要关心编译过程中程序的中间表示。

现代的编译器采用以数据共享为中心的架构风格，分析树是在语法分析阶段结束后才产生作为语义分析的输入，**分析树是数据中心中重要的共享数据**，为后续的语义分析提供了帮助。

【答案】D。

6. 【2009 年题 46 解析】

架构模式是软件设计中的高层决策，例如 C/S 结构就属于架构模式，**架构模式反映了**开发软件系统过程中所作的**基本设计决策**；设计模式主要关注软件系统的设计，与具体的实现语言无关：**惯用法**则是实现时通过某种特定的程序设计语言来描述构件与构件之间的关系，例如引用-计数就是 C++语言中的一种惯用法。

【答案】A、B、B。

7. 【2010 年题 33 解析】

分布式系统开发分为五个逻辑计算层：表示层实现用户界面；表示逻辑层为了生成数据表示而必须进行的处理任务，如输入数据编辑等；**应用逻辑层包括为支持实际业务应用和规则所需的应用逻辑和处理过程，如信用检查、数据计算和分析等**；数据处理层包括存储和访问数据库中的数据所需的应用逻辑和命令，如查询语句和存储过程等；数据层是数据库中实际存储的业务数据。

8. 【2010 年题 34 解析】

客户机/服务器系统开发时可以采用不同的分布式计算架构：

①分布式表示架构是将表示层和表示逻辑层迁移到客户机，应用逻辑层、数据处理层和数据层仍保留在服务器上；

②**分布式数据架构**是将**数据层和数据处理层**放置于**服务器**，**应用逻辑层、表示逻辑层和表示层**放置于**客户机**；

③**分布式数据和应用架构**是将**数据层和数据处理层**放置在**数据服务器**上，**应用逻辑层**放置在**应用服务器**上，表示**逻辑层和表示层**放置在**客户机**上。

【答案】D。

9. 【2010 年题 40 解析】

本题主要考查软件架构设计策略与架构风格问题。根据题干描述，该软件系统特别强调用户定义系统中对象的关系和行为这一特性，这需要在软件架构层面提供一种运行时的系统行为定义与改变的能力，根据常见架构风格的特点和适用环境，可以知道最合适的架构设计风格应该是解释器风格。

10. 【2010 年题 41 解析】

本题主要考查架构评审和软件架构设计的应用。根据图中示波器的功能描述，结合示波器常见的功能和使用方式，可以看出图中系统设计最大的缺陷在于没有建模系统与外界，特别是用户之间的交互方式。而与用户的交互无疑是示波器的一个十分重要的功能。

【答案】C。

11. 【2010 年题 42 解析】

本题主要考查架构风格与架构设计策略。根据题目描述，调温器需要实时获取外界的温度信息，并与用户定义的温度进行比较并做出动作。根据该系统的应用领域和实际需求，可以看出这是一个典型的过程控制架构风格的应用场景。

12. 【2010 年题 43 解析】

本题主要考查架构风格与架构设计策略。本题出题本就不严谨，从描述来看多种架构风格均合适：过程控制，虚拟机，隐式调用。当次考试参考答案为 C，**但从此后的同类问题来看，答案修改为“虚拟机（解释器，规则系统）”，所以再次出现该类问题，建议首选虚拟机类风格**。

【答案】A。

13. 【2010 年题 44 解析】

本题主要考查架构风格与架构设计策略。根据题目描述，语音识别系统是一个十分典型的专家系统，其特点是求解的正确结果不止一个，求解过程比较复杂，需要通过专家知识和反馈逐步得到正确结果。因此对比 4 个候选项，黑板结构特别适合求解这类问题，语音识别是黑板架构风格的典型应用。

14. 【2011 年题 34 解析】

本题主要考查对软件架构风格和设计模式两个概念的掌握与区分。

架构风格描述了一类软件架构的特征，它独立于实际问题，强调软件系统中通用的组织结构选择。

垃圾回收机制是 Java 语言管理内存资源时常用的一种设计模式。

15. 【2011 年题 37 解析】

本题考查经典架构风格。其实从应用的角度来看，这些经典的架构风格提得越来越少了，但这些架构风格有一些经典的应用是要求掌握的。

例如：管道-过滤器风格常常用于实现编译器。以规则为中心的虚拟机系统适合于实现专家系统。黑板风格适合于自然语言处理、语音处理、模式识别、图像处理。

【答案】D。

16. 【2011 年题 38 解析】

解释器是指在程序语言定义的计算和有效硬件操作确定的计算之间建立对应的联系。完成信息识别和转换工作。题目中的场景需要用到信息的识别和转换，所以可以用解释器风格。

17. 【2011 年题 39 解析】

根据题干描述，现代编译器主要关注编译过程和程序的中间表示，围绕程序的各种形态进行转化与处理。这种情况下，可以针对程序的各种形态构建数据库，通过中心数据库进行转换与处理。根据上述分析，选项中列举的架构风格中，数据共享风格最符合要求。

18. 【2012 年题 33 解析】

本题主要考查对软件架构风格与系统性能之间关系的理解。

对于采用层次化架构风格的系统，划分的层次越多，系统完成某项功能需要的中间调用操作越多，其性能越差。

对于采用管道-过滤器架构风格的系统，可以通过引入过滤器的数据并发处理可以有效提高系统性能。

对于采用面向对象架构风格的系统，可以通过减少功能调用层次提高系统性能。

对于采用过程调用架构风格的系统，将显式调用策略替换为隐式调用策略能够提高系统的灵活性，**但会降低系统的性能**。

【答案】D。

19. 【2013 年题 29 解析】

软件架构风格是描述某一特定应用领域中系统组织方式的惯用模式。架构风格定义一个系统家族，即一个架构定义一个词汇表和一组约束。

20. 【2013 年题 32 解析】

传统的编译器一般采用数据流架构风格，在这种架构中，每个构件都有一组输入和输出，数据输入构件，经过内部处理，然后产生数据输出。编译处理过程中，会分步将源代码一次一次的处理，最终形成目标代码，这与数据流架构风格相当吻合。但选项中有两个数据流风格的架构供选择，即：“管道-过滤器”和“顺序批处理”，这就需要进一步分析哪个更合适，由于题目中提到“程序源代码作为一个整体，依次在不同模块中进行传递”，而顺序批处理是强调把数据整体处理的，所以应选用顺序批处理风格。

IDE 是一种集成式的开发环境，在这种环境中，多种工具是围绕同一数据进行处理，这种情况适合用数据共享架构风格。

在题目中提到 IDE 环境是一种交互式编程，用户在修改程序代码后，会同时触发语法高亮显示、语法错误提示、程序结构更新等多种功能的调用与结果呈现。在做一件事情时，同时触发一系列的行为，这是典型的隐式调用风格(事件驱动系统)

“使 IDE 能够生成符合新操作系统要求的运行代码”，这一要求是可以通过适配策略满足的，像设计模式中的适配器模式便是采用适配的方式，形成一致的接口。“模拟新操作系统的运行环境”是典型的虚拟机架构风格的特长。

【答案】：B、C、A、B、D。

21. 【2014 年题 34 解析】

软件架构风格是描述某一特定应用领域中系统组织方式的惯用模式。架构风格定义一个系统家族，即一个架构定义一个词汇表和一组约束。词汇表中包含一些构件和连接件类型，而这组约束指出系统是如何将这些构件和连接件组合起来的。架构风格反映了领域中众多系统所共有的**结构和语义特性**，并指导如何将各个模块和子系统有效地组织成一个完整的系统。对软件架构风格的研究和实践促进对设计的重用，一些经过实践证实的解决方案也可以可靠地用于解决新的问题。

对于语音识别、知识推理等问题复杂、解空间很大、求解过程不确定的这一类软件系统，是黑板风格的经典应用。

22. 【2015 年题 33 解析】

根据题目描述，轿车巡航定速系统是一个十分典型的控制系统，其特点是不断采集系统当前状态，与系统中的设定状态进行对比，并通过将当前状态与设定状态进行对比从而进行控制。因此对比 4 个候选项，过程控制特别适合求解这类问题。

23. 【2015 年题 34 解析】

规则系统属于虚拟机风格的一种，在本题中要求机器人的控制者首先定义清洁任务和任务之间的关系，然后由机器人执行，这说明机器人能对自定义的一些逻辑进行解析，这是虚拟机风格的一大特色。

24. 【2015 年题 35 解析】

语音识别的处理是黑板风格的经典应用实例。

25. 【2015 年题 36 解析】

依据题目要求拟开发的在线游戏需要自定义对象之间的交互,这样必须有机制能支持系统对新定义的规则进行解析，这需要用到虚拟机风格，构造一个虚拟机对规则进行解析，所以在此应选择归属于虚拟机风格的解释器。

26. 【2015 年题 37 解析】

现代编译器的集成开发环境一般采用数据仓储（即以数据为中心的架构风格）架构风格进行开发，其中心数据就是程序的语法树。

27. 【2016 年题 28 解析】

C2 体系结构风格可以概括为:通过连接件绑定在一起的按照一组规则运作的并行构件网络。

28. 【2016 年题 31 解析】

“每个阶段产生的结果作为下一个阶段的输入”是典型的数据流架构风格的特点,选项中仅有管道-过滤器属于这种风格。

29. 【2016 年题 33 解析】

在本题所述的应用环境中，强调了自定义流程，然后按自定义流程来执行，这属于虚拟机风格的特征，备选答案中，仅有 C 选项属于虚拟机风格。

30. 【2016 年题 34 解析】

在本题所述的应用环境中，强调了自定义流程，然后按自定义流程来执行，这属于虚拟机风格的特征，备选答案中，仅有 C 选项属于虚拟机风格。

31. 【2016 年题 35 解析】

语音识别是黑板风格的经典应用。

32. 【2017 年题 34 解析】

体系结构风格反映了领域中众多系统所共有的**结构和语义**特性,并指导如何将各个模块和子系统有效地组织成一个完整的系统。

语音识别是黑板风格的经典应用场景。

输入某个构件，经过内部处理，产生数据输出的系统，正是数据流架构风格，选项中属于数据流风格的只有管道-过滤器。

【答案】B、C、C。

33. 【2017 年题 35 解析】

根据题目的意思，拟开发的 VIP 管理系统中 VIP 会员审核标准要能随时改变，灵活定义。在这方面虚拟机风格最为擅长，而属于虚拟机风格的只有 A 选项。

34. 【2017 年题 36 解析】

根据题目的意思，用户会注册自己的兴趣，然后系统也会把新闻按兴趣分类，如果某个新闻事件发生，可以通过事件来触发推送动作，将新闻推送给对其感兴趣的用户。这是典型的事件驱动系统应用场景。

35. 【2017 年题 37 解析】

C2 体系结构风格可以概括为：通过连接件绑定在一起按照一组规则运作的并行构件网络。C2 风格中的系统组织规则如下。

【答案】C。

36. 【2018 年题 34 解析】

在仓库风格中,有两种不同的构件:**中央数据结构说明当前状态，独立构件在中央数据存**

贮上执行。

答案 BA。

37. 【2018 年题 35 解析】

本题是极为经典的考题。题目中提及“支持玩家自行创建战役地图”这说明系统要能应对“自定义”内容的解析，这需要用到**解释器风格**。“并发用户数量 10000 人时用户请求要在 1 秒内得到响应”属于典型的**性能属性**，“对游戏系统进行二次开发的时间不超过 3 个月”属于**可修改性属性**。

答案 BAD。

9.4 层次系统架构风格

9.4.3 MVC 架构风格

1. 【2009 年题 28 解析】

在一个典型的基于 MVC（Model View Controlle）的 J2EE 应用中，系统的界面由 JSP 构件实现，分发客户请求、有效组织其他构件为客户端提供服务的控件器由 Servlet 构件实现，数据库相关操作由 Entity Bean 构件实现，系统核心业务逻辑由 Session Bean 构件实现。

9.8 软件架构评估

1. 【2009 年题 50 解析】

加密子系统的加密级别会对安全性和性能产生影响，一般而言，加密程度越高，安全性越好，但是其性能会降低；而加密程度越低，安全性越差，但性能一般会提高。因此该子系统将在安全性和性能两个方面产生冲突，所以该子系统一定属于权衡点和敏感点。

2. 【2010 年题 51 解析】

本题主要考查软件架构评价的理解和应用。正确识别风险点、非风险点、敏感点和权衡点是进行软件架构评价的关键步骤。其中敏感点是实现一个特定质量属性的关键特征，该特征为一个或多个软件构件所共有。系统权衡点会影响一个或多个属性，并对于多个属性来说都是敏感点。基于该定义，可以看出“改变加密的级别可能会对安全性和性能都产生显著的影响”正是一个对系统权衡点的描述。

3. 【2011 年题 45 解析】

ATAM 是评价软件构架的一种综合全面的方法。这种方法不仅可以揭示出构架满足特定质量目标的情况，而且(因为它认识到了构架决策会影响多个质量属性)可以使我们更清楚地认识到质量目标之间的联系——即如何权衡诸多质量目标。

ATAM 是针对软件架构的评估方法，其层次较高，不会涉及具体代码质量的评估，所以 A 选项不正确。而对于软件系统需求的正确性评价，应是需求验证的主要工作，也非 ATAM 所关注的内容。集成测试是在软件开发的测试阶段需要完成的任务，此时，架构设计、架构评审(即用 ATAM，SAAM 进行软件架构评审)、软件详细设计、编码、单元测试工作都已完成，所以该工作，也非 ATAM 所关注的内容。只有 D 选项的属性优先级排序是 ATAM 所要做的。

4. 【2011 年题 46 解析】

风险点与非风险点不是以标准专业术语形式出现的，只是一个常规概念，即可能引起风险的因素，可称为风险点。

敏感点是一个或多个构件(和/或构件之间的关系)的特性。研究敏感点可使设计人员或分析员明确在搞清楚如何实现质量目标时应注意什么。

权衡点是影响多个质量属性的特性，是多个质量属性的敏感点。例如，改变加密级别可能会对安全性和性能产生非常重要的影响。提高加密级别可以提高安全性，但可能要耗费更多的处理时间，影响系统性能。如果某个机密消息的处理有严格的时间延迟要求，则加密级别可能就会成为一个权衡点。

【答案】A、B。

5. 【2013 年题 34 解析】

架构权衡分析方法是一种系统架构评估方法，主要在系统开发之前，针对**性能、可用性、安全性和可修改性**等质量属性进行评价和折中。

ATAM 可以分为 4 个主要的活动阶段，包括**需求收集、架构视图描述、属性模型构造和分析、架构决策与折中**，整个评估过程强调以**属性作为架构评估的核心概念**。

题目中提到“某软件公司采用 ATAM 进行软件架构评估，在评估过程中识别出了多个关于质量属性的描述。其中，系统在进行文件保存操作时，应该与 Windows 系统的操作方式保持一致。”与用户所熟悉的操作方式，操作界面保持一致，这是一种减轻用户记忆负担，降低学习成本的做法，这有利于提高系统的易用性。

“系统应该提供一个开放的 API 接口，支持远程对系统的行为进行控制与调试”，在此处，我们注意到描述的核心落在“支持远程对系统的行为进行控制与调试”上了，而调试是在测试之后精确定位系统错误的一种机制，所以这种做法有利于提高系统的可测试性。

最后的两空也是考概念：在识别出上述描述后，通常采用效用树对质量属性的描述进行刻画与排序。在评估过程中，权衡点是一个会影响多个质量属性的架构设计决策。

【答案】：C、A、C、D、A、D、C。

6. 【2014 年题 35 解析】

架构复审一词来自于 ABSD(基于架构的软件设计)。在 ABSD 中，架构设计、文档化和复审是一个迭代过程。从这个方面来说，在一个主版本的软件架构分析之后，要安排一次**由外部人员（用户代表和领域专家）参加的复审**。

复审的目的是标识潜在的风险，及早发现架构设计中的缺陷和错误，包括架构能否满足需求、质量需求是否在设计中得到体现、层次是否清晰、构件的划分是否合理、文档表达是否明确、构件的设计是否满足功能与性能的要求等等。

由外部人员进行复审的目的是保证架构的设计能够公正地进行检验，使组织的管理者能够决定正式实现架构。

7. 【2014 年题 37 解析】

风险点：架构设计中潜在的、存在问题的架构决策所带来的隐患。

敏感点：为了实现某种特定的质量属性，一个或多个构件所具有的特性。

权衡点：影响多个质量属性的特性，是多个质量属性的敏感点。

风险点与非风险点不是以标准专业术语形式出现的，只是一个常规概念，即可能引起风险的因素，可称为风险点。某个做法如果有隐患，有可能导致一些问题，则为风险点；而如果某件事是可行的可接受的，则为非风险点。

8. 【2015 年题 41 解析】

本题主要考查考生对架构权衡分析方法(Architecture Tradeoff Analysis Method，ATAM)的掌握和理解。 ATAM 是在基于场景的架构分析方法(Scenarios-based Architecture Analysis Method，SAAM)基础之上发展起来的。

主要包括场景和需求收集、架构视图和场景实现、属性模型构造和分析、属性模型折中等 4 个阶段。ATAM 方法要求在系统开发之前，首先对这些质量属性进行评价和折中。

9. 【2018 年题 33 解析】

包含 4 个主要的活动领域，分别是场景和需求收集、体系结构视图和场景实现、**属性模型构造和分析** 、折中。

SAAM 的主要输入问题是问题描述、需求声明和**体系结构描述**。

答案 CC。

9.8.1 软件架构评估的方法

1. 【2009 年题 49 解析】

ATAM 是软件体系结构评估中的一种方法，主要对软件体系结构的设计结果进行评估。评估是软件系统详细设计、实现和测试之前的阶段工作，因此评估不涉及系统的实现代码和测试，因为评估是考查软件体系结构是否能够合适地解决软件系统的需求，并不对软件需求自身是否准确进行核实，而软件需求是否准确是需求评审阶段的工作。**ATAM 并不是一种精确的评估方法**，该方法表现的主要形式是评审会议。

【答案】D。

2. 【2012 年题 9 解析】

本题主要考查考生对基于场景的架构分析方法(Scenarios-based Architecture Analysis Method，SAAM)的掌握和理解。SAAM 是卡耐基梅隆大学软件工程研究所的 Kazman 等人于 1983 年提出的一种非功能质量属性的架构分析方法，是最早形成文档并得到广泛应用的软件架构分析方法。SAAM 的主要输入是**问题描述、需求说明和架构描述**，其分析过程主要包括**场景开发、架构描述、单个场景评估、场景交互和总体评估**。

3. 【2014 年题 38 解析】

本题主要考查考生对基于场景的架构分析方法（Scenarios-based Architecture Analysis Method，SAAM）的掌握和理解。SAAM 是卡耐基梅隆大学软件工程研究所的 Kazman 等人于 1983 年提出的一种非功能质量属性的架构分析分析方法，是最早形成文档并得到广泛应用的软件架构分析方法。SAAM 的主要输入是**问题描述、需求说明和架构描述**，其分析过程主要包括场景开发、架构描述、单个场景评估、场景交互和总体评估。

【答案】C、B。

9.9 构件及其复用

1. 【2009 年题 28 解析】

软件构件是软件系统中具有一定意义的、相对独立的可重用单元。与对象相比，构件可以基于对象实现，也可以不作为对象实现。构件需要在容器中管理并获取容器提供的服务；客户程序可以在运行状态下利用接口动态确定构件所支持的功能并调用。

【答案】C。

2. 【2016 年题 22 解析】

“面向构件的编程需要下列基本的支持：

多态性（可替代性）、模块封装性（高层次信息的隐藏）、后期的绑定和装载（部署独立性）、安全性（类型和模块安全性）。

3. 【2016 年题 24 解析】

【答案】A。

4. 【2018 年题 26 解析】

答案：ABD。

5. 【2018 年题 27 解析】

构件组装成软件系统的过程可以分为三个不同的层次定制、集成和扩展。答案 C。

6. 【2018 年题 28 解析】

伺服对象（Servant）：CORBA 对象的真正实现，负责完成客户端请求。

对象适配器（Object Adapter）：用于屏蔽 ORB 内核的实现细节，为服务器对象的实现

者提供抽象接口，以便他们使用 ORB 内部的某些功能。

对象请求代理（Object Request Broker）：解释调用并负责查找实现该请求的对象，将参数传给找到的对象，并调用方法返回结果。客户方不需要了解服务对象的位置、通信方式、实现、激活或存储机制。

答案 A。

7. 【2018 年题 29 解析】

答案：D。

9.10 产品线及系统演化

9.10.4 特定领域软件架构

1. 【2010 年题 45 解析】

参与 DSSA 的人员可以划分为四种角色：领域专家、领域分析师、领域设计人员和领域实现人员。

领域专家

领域专家可能包括该领域中系统的有经验的用户、从事该领域中系统的需求分析、设计、实现以及项目管理的有经验的软件工程师等。

领域专家的主要任务包括提供关于领域中系统的需求规约和实现的知识，帮助组织规范的、一致的领域字典，帮助选择样本系统作为领域工程的依据，复审领域模型、DSSA 等领域工程产品，等等。

领域专家应该熟悉该领域中系统的软件设计和实现、硬件限制、未来的用户需求及技术走向等。

领域分析人员

领域分析人员应由具有知识工程背景的有经验的系统分析员来担任。

领域分析人员的主要任务包括控制整个领域分析过程，进行知识获取，将获取的知识组织到领域模型中，根据现有系统、标准规范等验证领域模型的准确性和一致性，维护领域模型。

领域分析人员应熟悉软件重用和领域分析方法；熟悉进行知识获取和知识表示所需的技术、语言和工具；应具有一定的该领域的经验，以便于分析领域中的问题及与领域专家进行交互；应具有较高的进行抽象、关联和类比的能力；应具有较高的与他人交互和合作的能力。

领域设计人员

领域设计人员应由有经验的软件设计人员来担任。

领域设计人员的主要任务包括控制整个软件设计过程，根据领域模型和现有的系统开发出 DSSA，对 DSSA 的准确性和一致性进行验证，建立领域模型和 DSSA 之间的联系。

领域设计人员应熟悉软件重用和领域设计方法；熟悉软件设计方法；应有一定的该领域的经验，以便于分析领域中的问题及与领域专家进行交互。

领域实现人员

领域实现人员应由有经验的程序设计人员来担任。

领域实现人员的主要任务包括根据领域模型和 DSSA，或者从头开发可重用构件，或者利用再工程的技术从现有系统中提取可重用构件，对可重用构件进行验证，建立 DSSA 与可重用构件间的联系。

领域实现人员应熟悉软件重用、领域实现及软件再工程技术；熟悉程序设计；具有一定的该领域的经验。

【答案】C、A。

2. 【2012 年题 11 解析】

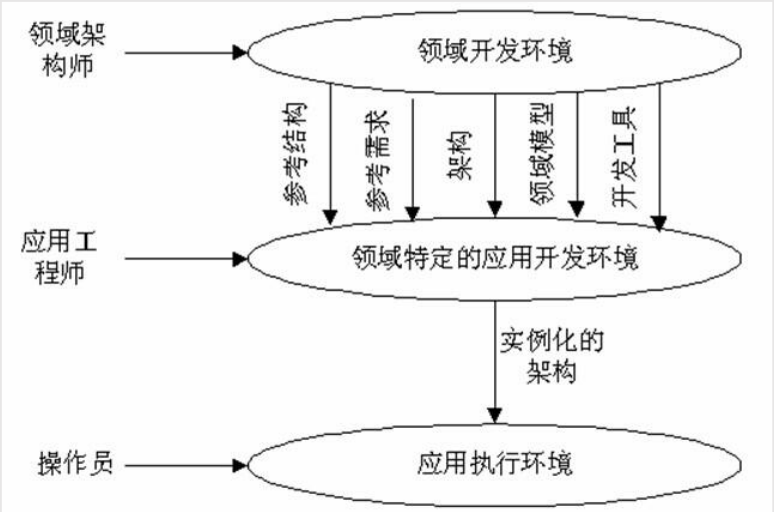
特定领域软件架构(Domain Specific Software Architecture，DSSA)以一个特定问题领域为对象，形成由领域**参考模型、参考需求、参考架构**等组成的开发基础架构，其目标是支持一

个特定领域中多个应用的生成。DSSA 的基本活动包括**领域分析、领域设计和领域实现**。其中**领域分析的主要目的是获得领域模型**，领域模型描述领域中系统之间共同的需求，即领域需求；**领域设计的主要目标是获得 DSSA**，DSSA 描述领域模型中表示需求的解决方案；**领域实现的主要目标是依据领域模型和 DSSA 开发和组织可重用信息**，并对基础软件架构进行实现。

【答案】D、B。

3. 【2013 年题 31 解析】

DSSA 通常是一个具有三个层次的系统模型，包括领域开发环境、领域特定应用开发环境和应用执行环境。



【答案】：B、C。

4. 【2014 年题 32 解析】

参与 DSSA 的人员可以划分为四种角色：领域专家、领域分析师、领域设计人员和领域实现人员。

1、领域专家

领域专家可能包括该领域中系统的有经验的用户、从事该领域中系统的需求分析、设计、实现以及项目管理的有经验的软件工程师等。

领域专家的主要任务包括提供关于领域中系统的需求规约和实现的知识，帮助组织规范的、一致的领域字典，帮助选择样本系统作为领域工程的依据，复审领域模型、DSSA 等领域工程产品，等等。

领域专家应该熟悉该领域中系统的软件设计和实现、硬件限制、未来的用户需求及技术走向等。

2、领域分析人员

领域分析人员应由具有知识工程背景的有经验的系统分析员来担任。

领域分析人员的主要任务包括控制整个领域分析过程，进行知识获取，将获取的知识组织到领域模型中，根据现有系统、标准规范等验证领域模型的准确性和一致性，维护领域模型。

领域分析人员应熟悉软件重用和领域分析方法；熟悉进行知识获取和知识表示所需的技术、语言和工具；应具有一定的该领域的经验，以便于分析领域中的问题及与领域专家进行交互；应具有较高的进行抽象、关联和类比的能力；应具有较高的与他人交互和合作的能力。

3、领域设计人员

领域设计人员应由有经验的软件设计人员来担任。

领域设计人员的主要任务包括控制整个软件设计过程，根据领域模型和现有的系统开发出 DSSA，对 DSSA 的准确性和一致性进行验证，建立领域模型和 DSSA 之间的联系。

领域设计人员应熟悉软件重用和领域设计方法；熟悉软件设计方法；应有一定的该领域的经验，以便于分析领域中的问题及与领域专家进行交互。

4、领域实现人员

领域实现人员应由有经验的程序设计人员来担任。

领域实现人员的主要任务包括根据领域模型和 DSSA，或者从头开发可重用构件，或者利用再工程的技术从现有系统中提取可重用构件，对可重用构件进行验证，建立 DSSA 与可重用构件间的联系。

领域实现人员应熟悉软件重用、领域实现及软件再工程技术；熟悉程序设计；具有一定的该领域的经验。

【答案】B、C。

5. 【2015 年题 39 解析】

特定领域软件架构(Domain Specific Software Architecture，DSSA)以一个特定问题领域为对象，形成由领域参考模型、参考需求、参考架构等组成的开发基础架构，其目标是支持一个特定领域中多个应用的生成。

DSSA 的基本活动包括领域分析、领域设计和领域实现。其中领域分析的主要目的是获得领域模型，领域模型描述领域中系统之间共同的需求，即领域需求；领域设计的主要目标是获得 DSSA，DSSA 描述领域模型中表示需求的解决方案；领域实现的主要目标是依据领域模型和 DSSA 开发和组织可重用信息，并对基础软件架构进行实现。

【答案】：C、D、B。

6. 【2016 年题 29 解析】

【答案】：C、C。

7. 【2018 年题 32 解析】

特定领域软件架构(DSSA)是一个特定的问题领域中由领域模型、参考需求及参考架构等组成的开发基础架构，其目标就是支持一个特定领域中多个应用的生成。

DSSA 的基本活动包括领域分析、领域设计和领域实现。领域分析的主要目的是获得领域模型，领域模型描述领域中系统之间共同的需求，即领域需求；**领域设计的主要目标是获得 DSSA**，DSSA 描述领域模型中表示需求的解决方案；**领域实现的主要目标是依据领域模型和 DSSA 开发并组织可重用信息**。

答案 BC。

9.11 其他

1. 【2010 年题 30 解析】

在基于构件的开发中，构件包含并扩展了模块化程序设计中子程序、面向对象系统中对象或类和系统模型中包的思想，它是系统设计、实现和维护的基础。构件定义为通过接口访问服务的一个独立可交付的功能单元。

【答案】：C。

2. 【2010 年题 32 解析】

对象管理组织（OMG）基于 CORBA 基础设施定义了四种构件标准。实体（Entity）构件需要长期持久化并主要用于事务性行为，由容器管理其持久化。加工（Process）构件同样需要容器管理其持久化，但没有客户端可访问的主键。**会话（Session）构件不需要容器管理其持久化，其状态信息必须由构件自己管理**。服务（Service）构件是无状态的。

【答案】D。

3. 【2016 年题 23 解析】

【答案】D、C。

4. 【2016 年题 25 解析】

UDDI 用于 Web 服务注册和服务查找；

WSDL 用于描述 Web 服务的接口和操作功能；

SOAP 为建立 Web 服务和服务请求之间的通信提供支持。

BPEL 翻译成中文的意思是面向 Web 服务的业务流程执行语言，也有的文献简写成 BPEL4WS，它是一种使用 Web 服务定义和执行业务流程的语言。使用 BPEL，用户可以通过组合、编排和协调 Web 服务自上而下地实现面向服务的体系结构（SOA）。BPEL 提供了一种相对简单易懂的方法，可将多个 Web 服务组合到一个新的复合服务（称作业务流程）中。

【答案】：C、D。

5. 【2016 年题 26 解析】

JCA 标准化连接子是由 J2EE 1.3 首先提出的，它位于 J2EE 应用服务器和企业信息系统（EIS）之间，比如数据库管理、企业资源规划（ERP）、企业资产管理（EAM）和客户关系管理（CRM）系统。不是用 Java 开发的企业应用或者在 J2EE 框架内的应用都可以通过 JCA 连接。

JMS 是 Java 对消息系统的访问机制，但它本身并不实现消息。JMS 支持点对点分发的消息队列，也支持多个目标订阅的消息主题。当消息发布给一个主题的适合，消息就会发送给所有那个主题的订阅者。JMS 支持各种消息类型（二进制、流、名－值表、序列化的对象和文本）。通过声明与 SQL 的 WHERE 相近的句段，可以建立消息的过滤器。

Java IDL 即 idltojava 编译器就是一个 ORB，可用来在 Java 语言中定义、实现和访问 CORBA 对象。Java IDL 支持的是一个瞬间的 CORBA（Common Object Request Broker Architecture,公共对象请求代理体系结构，通用对象请求代理体系结构）对象，即在对象服务器处理过程中有效。实际上，Java IDL 的 ORB 是一个类库而已，并不是一个完整的平台软件，但它对 Java IDL 应用系统和其他 CORBA 应用系统之间提供了很好的底层通信支持，实现了 OMG 定义的 ORB 基本功能。

【答案】：D。

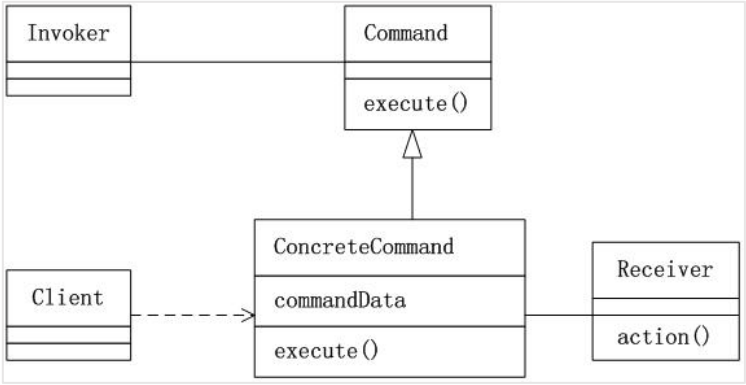
第 10 章设计模式

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。



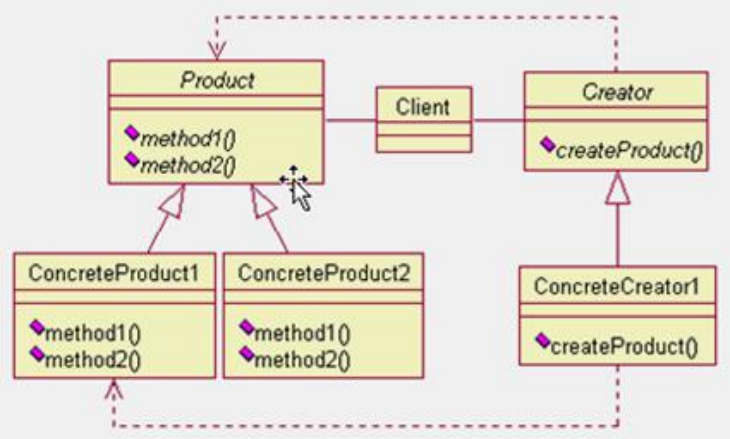
对于题目所给出的图，与“Invoker”角色相对应的类是 MenuItem，与“Concrete Command”角色相对应的类是 Open。

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】

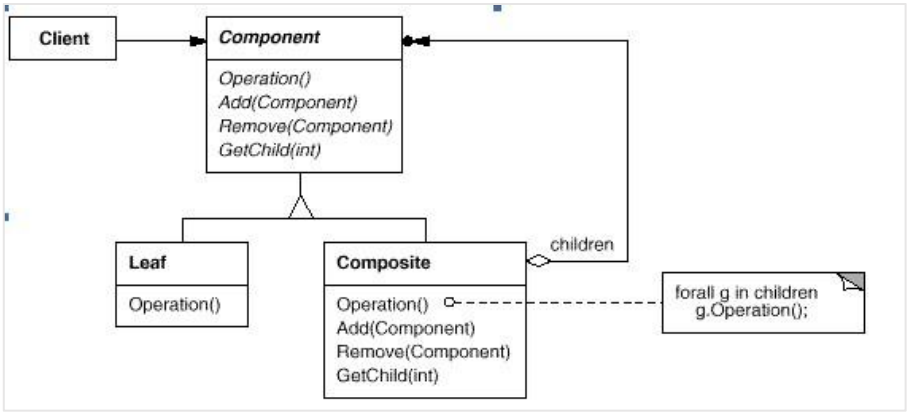
本题主要考查设计模式的理解与应用。根据题干描述，应用系统需要使用某公司开发的类库，该应用系统是一组窗格组成，应用需要协调窗格之间的行为，并且不能引用窗格自身，在这种要求下，对比 4 个候选项，其中中介者模式用一个中介对象封装一系列的对象交互。中介者使用各对象不需要显式的相互调用，从而使其耦合松散。可以看出该模式最符合需求。

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，以及支持可撤消的操作**。

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(façade)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。

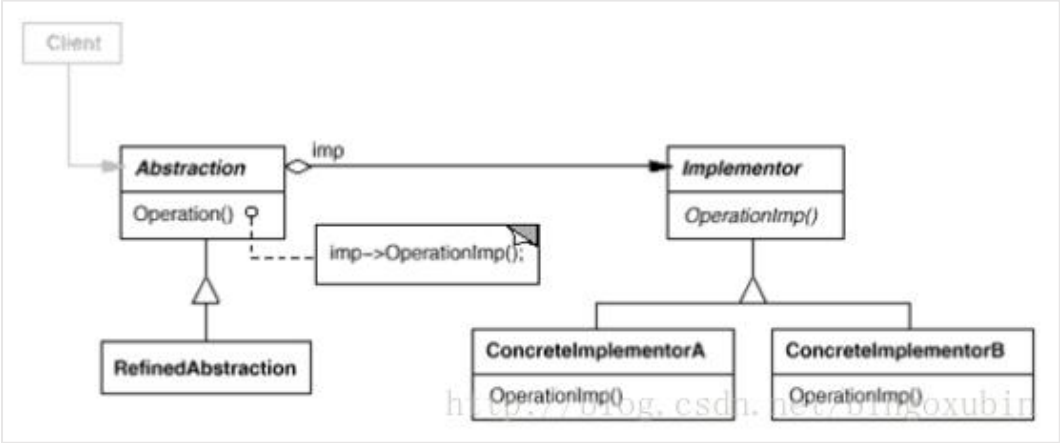
15. 【2014 年题 33 解析】

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 Builder Prototype Singleton	Adapter Bridge Composite Decorator Facade Flyweight Proxy	Chain of Responsibility Command Iterator Mediator Memento Observer 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。