软件架构设计

目录

[软件架构设计 1](#_Toc525331500)

[1 软件架构概述 4](#_Toc525331501)

[1.1 软件架构的定义 4](#_Toc525331502)

[1.1.1 知识点 4](#_Toc525331503)

[1.1.2 试题 5](#_Toc525331504)

[1.2 软件架构的重要性 9](#_Toc525331505)

[1.2.1 知识点 9](#_Toc525331506)

[1.2.2 试题 10](#_Toc525331507)

[1.3 架构的模型 12](#_Toc525331508)

[1.3.1 知识点 12](#_Toc525331509)

[1.3.2 试题 14](#_Toc525331510)

[2 架构需求与软件质量属性 17](#_Toc525331511)

[2.1 软件质量属性 17](#_Toc525331512)

[2.2 6个质量属性及实现 17](#_Toc525331513)

[2.2.1 知识点 17](#_Toc525331514)

[2.2.2 试题 19](#_Toc525331515)

[3 软件架构风格 32](#_Toc525331516)

[3.1 软件架构风格分类 32](#_Toc525331517)

[3.1.1 知识点 32](#_Toc525331518)

[3.1.2 试题 33](#_Toc525331519)

[4 层次系统架构风格 54](#_Toc525331520)

[4.1 二层及三层 C/S 架构风格 54](#_Toc525331521)

[4.1.1 知识点 54](#_Toc525331522)

[4.1.2 试题 55](#_Toc525331523)

[4.2 B/S 架构风格、 56](#_Toc525331524)

[4.2.1 知识点 56](#_Toc525331525)

[4.2.2 试题 57](#_Toc525331526)

[4.3 MVC 架构风格 57](#_Toc525331527)

[4.3.1 知识点 57](#_Toc525331528)

[4.3.2 试题 57](#_Toc525331529)

[4.4 MVP 架构风格 58](#_Toc525331530)

[5 面向服务的架构 58](#_Toc525331531)

[5.1 SOA 概述 58](#_Toc525331532)

[5.1.1 知识点 58](#_Toc525331533)

[5.1.2 试题 60](#_Toc525331534)

[5.2 微服务 61](#_Toc525331535)

[6 基于架构的软件开发(ABSD) 61](#_Toc525331536)

[6.1 概念 61](#_Toc525331537)

[6.1.1 知识点 61](#_Toc525331538)

[6.1.2 试题 64](#_Toc525331539)

[6.2 架构需求 69](#_Toc525331540)

[6.2.1 知识点 69](#_Toc525331541)

[6.2.2 试题 69](#_Toc525331542)

[6.3 架构设计 70](#_Toc525331543)

[6.3.1 知识点 70](#_Toc525331544)

[6.3.2 试题 70](#_Toc525331545)

[6.4 架构文档化 73](#_Toc525331546)

[6.4.1 知识点 73](#_Toc525331547)

[6.4.2 试题 73](#_Toc525331548)

[6.5 架构复审 74](#_Toc525331549)

[6.5.1 知识点 74](#_Toc525331550)

[6.5.2 试题 74](#_Toc525331551)

[6.6 架构实现 75](#_Toc525331552)

[6.6.1 知识点 75](#_Toc525331553)

[6.6.2 试题 76](#_Toc525331554)

[6.7 架构演化 76](#_Toc525331555)

[6.7.1 知识点 76](#_Toc525331556)

[6.7.2 试题 76](#_Toc525331557)

[7 软件架构评估 76](#_Toc525331558)

[7.1 软件架构评估的方法 76](#_Toc525331559)

[7.1.1 知识点 76](#_Toc525331560)

[7.1.2 试题 77](#_Toc525331561)

[7.2 架构的权衡分析法(ATAM、SAAM) 80](#_Toc525331562)

[7.2.1 知识点 80](#_Toc525331563)

[7.2.2 试题 80](#_Toc525331564)

[7.3 成本效益分析法 87](#_Toc525331565)

[8 构件及其复用 87](#_Toc525331566)

[8.1 商用构件标准规范 87](#_Toc525331567)

[8.1.1 知识点 87](#_Toc525331568)

[8.1.2 试题 88](#_Toc525331569)

[8.2 应用系统簇与构件系统 92](#_Toc525331570)

[8.2.1 知识点 92](#_Toc525331571)

[8.2.2 试题 93](#_Toc525331572)

[8.3 基于复用开发的组织结构 96](#_Toc525331573)

[9 产品线及系统演化 96](#_Toc525331574)

[9.1 复用与产品线 96](#_Toc525331575)

[9.2 基于产品线的架构 96](#_Toc525331576)

[9.3 产品线的开发模型 96](#_Toc525331577)

[9.4 特定领域软件架构DSSA 96](#_Toc525331578)

[9.4.1 知识点 96](#_Toc525331579)

[9.4.2 试题 98](#_Toc525331580)

[9.5 架构及系统演化 105](#_Toc525331581)

[10 软件架构视图 105](#_Toc525331582)

[10.1 软件视图的分类 105](#_Toc525331583)

[10.2 模块视图类型及其风格 105](#_Toc525331584)

[10.3 C&C视图类型及其风格 105](#_Toc525331585)

[10.4 分配视图类型及其风格 106](#_Toc525331586)

[10.5 各视图类型间的映射关系 106](#_Toc525331587)

# 软件架构概述

## 软件架构的定义

知识点

定义 :是具有一定形式的结构化元素，即构件的集合，包含处理构件、连接构件、数据构件

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，主要描述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的交互关系。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和拓扑结构，而且显示了系统需求和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。

发展史:

无架构阶段🡺汇编

萌芽阶段🡺程序结构设计

初级阶段🡺UML

高级阶段🡺4+1视图

软件系统设计决策:

架构模式🡺分析模式🡺设计模式🡺惯用法

架构模式是一个通用的、可重用的解决方案，用于在给定上下文中的软件体系结构中经常出现的问题。架构模式与软件设计模式类似，但具有更广泛的范围。

10种常见的体系架构模式:

1. 分层模式
2. 客户端-服务器模式
3. 主从设备模式
4. 管道-过滤器模式
5. 代理模式
6. 点对点模式
7. 事件总线模式
8. 模型-视图-控制器模式
9. 黑板模式
10. 解释器模式

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

试题

* (57)的选择是开发一个软件系统时的基本设计决策；(58)是最低层的模式，关注软件系统的设计与实现，描述了如何实现构件及构件之间的关系。引用-计数是C++ 管理动态资源时常用的一种（59)。

2009年(57)

A.架构模式

B.惯用法

C.设计模式

D.分析模式

2009年(58)

A.架构模式

B.惯用法

C.设计模式

D.分析模式

2009年(59)

A.架构模式

B.惯用法

C.设计模式

D.分析模式

【答案】A B B 【解析】本题考查软件设计中使用的架构模式、设计模式和惯用法的基本概念。

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。

* ANSI/IEEE 1471-2000是对软件密集型系统的架构进行描述的标准。在该标准中， (39)这一概念主要用于描述软件架构模型。在此基础上，通常采用（40)描述某个利益相关人（Stakeholder)所关注架构模型的某一方面。（41)则是对所有利益相关人关注点的响应和回答。

2012年(39)

A.上下文

B.架构风格

C.组件

D.视图

2012年(40)

A.环境

B.资源

C.视角

D.场景

2012年(41)

A.架构

B.系统

C.模型

D.使命

【答案】D C A 【解析】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、运营或其他方面相关的利益。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。

* 软件系统架构是关于软件系统的结构、(40)和属性的高级抽象。在描述阶段，主要描述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的（41)。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和（42)结构，而且显示了系统需求和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。

2013年(40)

A.行为

B.组织

C.性能

D.功能

2013年(41)

A.交互关系

B.实现关系

C.数据依赖

D.功能依赖

2013年(42)

A.进程

B.拓扑

C.处理

D.数据

【答案】A A B 【解析】本题主要考查软件系统架构的基础知识。

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，主要描述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的交互关系。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和拓扑结构，而且显示了系统需求和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。

* 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、（38)和架构配置。

2012年(38)

A.架构风格

B.架构实现

C.连接件

D.组件实现

【答案】C 【解析】本题主要考查架构描述语言的知识。 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

* 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、（43）和架构配置。

2015年(43)

A.架构风格

B.架构实现

C.连接件

D.组件约束

【答案】C 【解析】本题考查架构描述语言的理解与掌握。

架构描述语言(ADL)用于描述软件的体系架构。已有多种架构描述语言，如Wright (由卡内基梅隆大学开发)，Acme (由卡内基梅隆大学开发)，C2 (由UCI开发)， Darwin (由伦敦帝国学院开发)。ADL的基本构成包括组件、连接器和配置。

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。

## 软件架构的重要性

知识点

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。

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。

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，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。

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。

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。其中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。

试题

* 以下叙述，(44)不是软件架构的主要作用。

2013年(44)

A.在设计变更相对容易的阶段，考虑系统结构的可选方案

B.便于技术人员与非技术人员就软件设计进行交互

C.展现软件的结构、属性与内部交互关系

D.表达系统是否满足用户的功能性需求

【答案】D 【解析】本题主要考查软件架构基础知识。

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。

* 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。以下关于软件架构的描述，错误的是(44)。

2010年(44)

A.根据用户需求，能够确定一个最佳的软件架构，指导整个软件的开发过程

B.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

C.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

D.软件架构能够指导设计人员和实现人员的工作

【答案】A 【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，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。

* 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活动。以下关于软件架构重要性的叙述中，错误的是（46）.

2009年(46)

A.架构设计能够满足系统的性能、可维护性等品质

B.良好的架构设计能够更好地捕获并了解用户需求

C.架构设计能够使得不同的利益相关人（stakeholders)达成一致的目标

D.杂构设计能够支持项目计划和项目管理等活动

【答案】B 【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。

* 软件架构贯穿于软件的整个生命周期，但在不同阶段对软件架构的关注力度并不相同，在(45)阶段，对软件架构的关注最多。

2009年(45)

A.需求分析与设计

B.设计与实现

C.实现与测试

D.部署与变更

【答案】B 【解析】本题主要考査软件架构对软件开发的影响和在生命周期中的关注力度。

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。其中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。

## 架构的模型

知识点

|  |  |  |  |
| --- | --- | --- | --- |
| 结构模型 |  |  |  |
| 框架模型 |  |  |  |
| 动态模型 |  |  |  |
| 过程模型 |  |  |  |
| 功能模型 |  |  |  |
| 4+1视图模型 | 逻辑视图 |  |  |
| 侧重支持系统的功能需求 |  |
| 在面向对象技术中可通过UML描述 | 对象模型代表逻辑视图 |
| 类图描述逻辑视图 |
| 逻辑视图的风格为面向对象的风格 |  |
| 系统静态结构 |  |
| 开发视图 | 侧重于软件模块的组织和管理 |  |
| 在面向对象技术中可通过UML描述 | 实现视图 |
| 系统静态结构 |  |
| 进程视图 | 侧重系统的运行特性，关注一些非功能性需求 |  |
| 在面向对象技术中可通过UML描述 | 进程视图 |
| 强调 | 并发性 |
| 分布性 |
| 系统集成性 |
| 容错能力 |
| 逻辑视图中的主要抽象如何适合进程结构 |
| 系统动态结构 |  |
| 部署视图 | 如何把软件映射到硬件 |  |
| 在面向对象技术中可通过UML描述 | 部署视图 |
| 强调 | 系统拓扑结构 |
| 系统安装 |
| 通信等问题 |
| 系统动态结构 |  |
| 场景 | 重要系统活动的抽象，是最重要的需求抽象 |  |
| 在面向对象技术中可通过UML描述 | 用例视图 |

在RUP中采用“4+1”视图模型来描述软件系统的体系结构。“4+1”视图包括逻辑视图、实现视图、进程视图、部署视图和用例视图。

分析人员和测试人员关心的是系统的行为，因此会侧重于用例视图；最终用户关心的是系统的功能，因此会侧重于逻辑视图：程序员关心的是系统的配置、装配等问题，因此会侧重于实现视图：系统集成人员关心的是系统的性能、可伸缩性、吞吐率等问题，因此会侧重于进程视图；系统工程师关心的是系统的发布、安装、拓扑结构等问题，因此会侧重于部署视图。

试题

* 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(26)，系统工程师侧重于(27)。

2010年(26)

A.实现视图

B.进程视图

C.逻辑视图

D.部署视图

2010年(27)

A.实现视图

B.进程视图

C.逻辑视图

D.部署视图

【答案】C D 【解析】

在RUP中采用“4+1”视图模型来描述软件系统的体系结构。“4+1”视图包括逻辑视图、实现视图、进程视图、部署视图和用例视图。

分析人员和测试人员关心的是系统的行为，因此会侧重于用例视图；最终用户关心的是系统的功能，因此会侧重于逻辑视图：程序员关心的是系统的配置、装配等问题，因此会侧重于实现视图：系统集成人员关心的是系统的性能、可伸缩性、吞吐率等问题，因此会侧重于进程视图；系统工程师关心的是系统的发布、安装、拓扑结构等问题，因此会侧重于部署视图。

* 1995年Kruchten提出了著名的“4+1”视图，用来描述软件系统的架构。在“4+1” 视图中，(46)用来描述设计的对象模型和对象之间的关系;(47)描述了软件模 块的组织与管理；(48)描述设计的并发和同步特征。

2011年(46)

A.逻辑视图

B.用例视图

C.过程视图

D.开发视图

2011年(47)

A.逻辑视图

B.用例视图

C.过程视图

D.开发视图

2011年(48)

A.逻辑视图

B.用例视图

C.过程视图

D.开发视图

【答案】A D C 【解析】本题主要考查对“4+1”视图概念的掌握。 1995年Kruchten提出了著名的“4+1”

视图，用来描述软件系统的架构。在“4+1”视图中，逻辑视图用来描述设计的对象模型和对象之间的关系；开发视图描述了软件模块的组织与管理；过程视图描述设计的并发和同步特征。

* “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（44）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（45）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（46）定义对象的内部行为。

2014年(44)

A.逻辑

B.过程

C.开发

D.物理

2014年(45)

A.对象图

B.活动图

C.状态图

D.类图

2014年(46)

A.对象图

B.活动图

C.状态图

D.类图

【答案】A D C 【解析】本题主要考查考生对“4+1”视图的即.解与掌握。

“4+1”视图是对逻辑架构进行描述，最早由Philippe Kruchten提出，他在1995年的IEEE

Software上发表了题为The 4+1 View Model of Architecture

的论文，引起了业界的极大关注，并最终被RUP采纳，现在已经成为架构设计的结构标准。“4+1”视图主要包括： ①逻辑视图(Logical

View),设计的对象模型（使用面向对象的设计方法时)。 ②过程视图（Pmcess View)，捕捉设计的并发和同步特征。

③物理视图（Physical View),描述了软件到硬件的映射，反映了分布式特性。 ④开发视图(Development

View),描述了在开发环境中软件的静态组织结构。 ⑤架构的描述，即所做的各种决定，可以围绕着这四个视图来组织，然后由一些用例（Use

Cases)或场景（Scenarios)来说明，从而形成了第五个视图。

当采用面向对象的设计方法描述对象模型时，通常使用类图表达类的内部属性和行为，以及类集合之间的交互关系；采用状态图定义对象的内部行为。

# 架构需求与软件质量属性

## 软件质量属性

## 6个质量属性及实现

知识点

常见的六个质量属性:可用性、可修改性、性能、安全性、可测试性、易用性。

质量属性场景是一种面向特定的质量属性的需求，由6部分组成：刺激源、刺激、环境、制品、响应、响应度量。

　　以《淘宝网》为例：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 质量属性 | 质量属性场景 | | 设计策略 | 子质量点 |
| 可用性 | 场景 | 天猫双十一购物狂欢节 | Ping/Echo 主动冗余 心跳机制 被动冗余 选举 |  |
| 刺激源 | 海量用户 |  |
| 刺激 | 过多用户涌入抢购，系统出现崩溃的状态 |  |
| 制品 | 处理系统崩溃的处理器 |  |
| 环境 | 正常操作 |  |
| 响应 | 淘宝网监控系统记录，处理人员进行紧急处理 |  |
| 响应度量 | 短时间内恢复系统正常运行 |  |
| 可修改性 | 场景 | 系统进行升级 | 运行时注册 接口-实现分离 信息隐藏 |  |
| 刺激源 | 开发人员 |  |
| 刺激 | 改变页面的形态，增加少许功能、 |  |
| 制品 | 升级完后的系统 |  |
| 环境 | 设计时 |  |
| 响应 | 修改了用户的操作页面，未产生副作用 |  |
| 响应度量 | 在15分钟左右完成升级更改 |  |
| 性能 | 场景 | 天猫双十一购物狂欢节 | 队列调度 增加计算资源 减少计算开销 引入并发机制 采用资源调度 |  |
| 刺激源 | 用户 |  |
| 刺激 | 进行疯狂购物交易 |  |
| 制品 | 系统 |  |
| 环境 | 在正常操作下 |  |
| 响应 | 大量的交易同时被处理 |  |
| 响应度量 | 每个交易平均等待时间为3s |  |
| 安全性 | 场景 | 黑客想要盗窃用户信息 | 限制访问 优先级队列 用户认证 用户授权 追踪审计 |  |
| 刺激源 | 黑客 |  |
| 刺激 | 试图通过某些手段窃取用户的信息 |  |
| 制品 | 淘宝用户信息 |  |
| 环境 | 用户不在线时 |  |
| 响应 | 对访问者进行身份上的验证 |  |
| 响应度量 | 淘宝安全系统阻止黑客访问用户信息 |  |
| 可测试性 | 场景 | 一个马上要执行的系统功能 | 记录-回放 | 可维护性 |
| 刺激源 | 系统测试人员 | 可扩展性 |
| 刺激 | 对系统功能执行测试 | 结构重组 |
| 制品 | 系统的某个功能 | 可移植性 |
| 环境 | 功能要部署时 |  |
| 响应 | 提供对状态值的访问、提供所要计算的值，准备测试环境 |  |
| 响应度量 | 3个小时测试了85% |  |
| 易用性 | 场景 | 用户误将某物品移入到购物车 |  |  |
| 刺激源 | 用户 |  |  |
| 刺激 | 用户想要将物品移出 |  |  |
| 制品 | 系统 |  |  |
| 环境 | 系统运行时 |  |  |
| 响应 | 希望快速完成操作 |  |  |
| 响应度量 | 在1s内完成撤销操作 |  |  |

试题

* 某公司在对一家用车库门嵌入式软件系统进行架构设计时，识别出两个关键的质量属性场景，其中“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”与(56)质量属性相关:“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”与（57)质量属性相关。

2011年(56)

A.可用性

B.性能

C.可修改性

D.可测试性

2011年(57)

A.可用性

B.性能

C.可修改性

D.可测试性

【答案】B D 【解析】本题主要考查对质量属性的理解。

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。

* 某服务器软件系统能够正确运行并得出计算结果，但存在“系统出错后不能在要求 的时间内恢复到正常状态”和“对系统进行二次开发时总要超过半年的时间”两个问题，上述问题依次与质量属性中的(58)相关。

2010年(58)

A.可用性和性能

B.性能和可修改性

C.性能和可测试性

D.可用性和可修改性

【答案】D 【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。

* 某服务器软件系统对可用性(Availability)、性能(Performance)和可修改性 (Modification)的要求较髙，(55)设计策略能提髙该系统的可用性，(56)设计策略能够提高该系统的性能，(57)设计策略能够提髙该系统的可修改性。

2010年(55)

A.Ping/Echo

B.限制访问

C.运行时注册

D.接口-实现分离

2010年(56)

A.分层结构

B.事务机制

C.主动冗余

D.队列调度

2010年(57)

A.信息隐藏

B.记录/回放

C.任务模型

D.回滚

【答案】A D A 【解析】本题主要考査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；限制访问主要提髙系统的安全性；运行时注册主要提高系统的可修改性；接口-实现分离主要提髙系统的可修改性；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。

* 软件质量属性通常需要采用特定的设计策略实现。例如，(58)设计策略能提高该系统的可用性，(59)设计策略能够提髙该系统的性能，(60)设计策略能够提高该系统的安全性。

2011年(58)

A.心跳机制

B.数据驱动

C.关注点分离

D.信息隐藏

2011年(59)

A.引入中间层

B.事务机制

C.主动冗余

D.优先级队列

2011年(60)

A.信息隐藏

B.内置监控器

C.限制访问

D.检查点

【答案】A D C 【解析】本题主要考査对架构设计策略和质量属性的理解。

软件质量属性通常需要采用特定的设计策略实现，并且设计策略会对其他的质量属性产生影响。例如，心跳机制策略能提高该系统的可用性，优先级队列策略能够提高该系统的性能，限制访问策略能够提髙该系统的安全性。

* 某公司在对一家用车库门嵌入式软件系统进行架构设计时，识别出两个关键的质量属性场景，其中“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”与(56)质量属性相关:“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”与（57)质量属性相关。

2011年(56)

A.可用性

B.性能

C.可修改性

D.可测试性

2011年(57)

A.可用性

B.性能

C.可修改性

D.可测试性

【答案】B D 【解析】本题主要考查对质量属性的理解。

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。

* 某公司欲开发一个在线交易系统，在架构设计阶段，公司的架构师识别出3个核心质量属性场景。其中“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（56)质量属性相关,通常可采用（57)架构策略实现该属性:“当系统由于软件故障意外崩溃后，需要在0.5小时内恢复正常运行”主要与（58)质量属性相关，通常可采用（59)架构策略实现该属性;“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（60)质量属性相关，通常可采用（61)架构策略实现该属性。

2012年(56)

A.性能

B.吞吐量

C.可靠性

D.可修改性

2012年(57)

A.操作串行化

B.资源调度

C.心跳

D.内置监控器

2012年(58)

A.可测试性

B.易用性

C.可用性

D.互操作性

2012年(59)

A.主动冗余

B.信息隐藏

C.抽象接口

D.记录/回放

2012年(60)

A.可用性

B.安全性

C.可测试性

D.可修改性

2012年(61)

A.内置监控器

B.记录/回放

C.追踪审计

D.维护现有接口

【答案】A B C A B C 【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：“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。“当系统由于软件故障意外崩溃后，需要在0.5小时内恢复正常运行”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；“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。

* 某公司欲开发一个在线交易系统，在架构设计阶段，公司的架构师识别出3个核心质量属性场景。其中“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（54）质量属性相关，通常可采用（55）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（56）质量属性相关，通常可采用（57）架构策略实现该属性；“对系统的消息中间件进行替换时，替换工作需要在5人/月内完成”主要与（58）质量属性相关，通常可采用（59）架构策略实现该属性。

2014年(54)

A.性能

B.安全性

C.可用性

D.可修改性

2014年(55)

A.操作隔离

B.资源调度

C.心跳

D.内置监控器

2014年(56)

A.性能

B.易用性

C.可用性

D.互操作性

2014年(57)

A.主动冗余

B.资源调度

C.抽象接口

D.记录/回放

2014年(58)

A.可用性

B.安全性

C.可测试性

D.可修改性

2014年(59)

A.接口-实现分离

B.记录/回放

C.内置监控器

D.追踪审计

【答案】C C A B D A 【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。“对系统的消息中间件进行替换时，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-实现分类、抽象、信息隐藏等架构策略实现该属性。

* （47）不属于可修改性考虑的内容。

2016年(47)

A.可维护性

B.可扩展性

C.结构重构

D.可变性

【答案】D

【解析】 可修改性(modifiability)是指能够快速地以较高的性能价格比对系统进行变更的能力。通常以某些具体的变更为基准，通过考察这些变更的代价衡量可修改性。可修改性包含四个方面。

(1)可维护性(maintainability)。这主要体现在问题的修复上：在错误发生后“修复”软件系统。为可维护性做好准备的软件体系结构往往能做局部性的修改并能使对其他构件的负面影响最小化。

(2)可扩展性(extendibility)。这一点关注的是使用新特性来扩展软件系统，以及使用改进版本来替换构件并删除不需要或不必要的特性和构件。为了实现可扩展性，软件系统需要松散耦合的构件。其目标是实现一种体系结构，它能使开发人员在不影响构件客户的情况下替换构件。支持把新构件集成到现有的体系结构中也是必要的。

(3)结构重组(reassemble）。这一点处理的是重新组织软件系统的构件及构件间的关系，例如通过将构件移动到一个不同的子系统而改变它的位置。为了支持结构重组，软件系统需要精心设计构件之间的关系。理想情况下，它们允许开发人员在不影响实现的主体部分的情况下灵活地配置构件。

(4)可移植性(portability)。可移植性使软件系统适用于多种硬件平台、用户界面、操作系统、编程语言或编译器。为了实现可移植，需要按照硬件无关的方式组织软件系统，其他软件系统和环境被提取出。可移植性是系统能够在不同计算环境下运行的能力。这些环境可能是硬件、软件，也可能是两者的结合。在关于某个特定计算环境的所有假设都集中在一个构件中时，系统是可移植的。如果移植到新的系统需要做些更改，则可移植性就是一种特殊的可修改性。

* 某公司欲开发一个智能机器人系统，在架构设计阶段，公司的架构师识别出3个核心质量属性场景。其中“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”主要与（58）质量属性相关，通常可采用（59）架构策略实现该属性；“机器人在正常运动过程中如果发现前方2米内有人或者障碍物，应在1秒内停止并在2秒内选择一条新的运行路径”主要与（60）质量属性相关，通常可采用（61）架构策略实现该属性；“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”主要与（62）质量属性相关，通常可采用（63）架构策略实现该属性。

2016年(58)

A.可用性

B.性能

C.易用性

D.可修改性

2016年(59)

A.抽象接口

B.信息隐藏

C.主动冗余

D.记录/回放

2016年(60)

A.可测试性

B.易用性

C.互操作性

D.性能

2016年(61)

A.资源调度

B.操作串行化

C.心跳

D.内置监控器

2016年(62)

A.可用性

B.安全性

C.可测试性

D.可修改性

2016年(63)

A.内置监控器

B.追踪审计

C.记录/回放

D.维护现有接口

【答案】A C D A B B

【解析】(58、59、60、61)“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。

(62、63)“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。

* 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。软件架构设计需满足系统的（42），如性能、安全性和可修改性等，并能够指导设计人员和实现人员的工作。

2015年(42)

A.功能需求

B.性能需求

C.质量属性

D.业务属性

【答案】C 【解析】本题考查软件架构设计方面的基础知识。

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。

* 某公司欲开发一个网上商城系统，在架构设计阶段，公司的架构师识别出3个核心质量属性场景，其中“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”主要与（56）质量属性相关，通常可采用（57）架构策略实现该属性；“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”主要与（58）质量属性相关通常可采用（59）架构策略实现该属性；“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（60）质量属性相关，通常可采用（61）架构策略实现该属性。

2015年(56)

A.性能

B.可用性

C.易用性

D.可修改性

2015年(57)

A.主动冗余

B.信息隐藏

C.抽象接口

D.记录/回放

2015年(58)

A.可测试性

B.易用性

C.性能

D.互操作性

2015年(59)

A.操作窜行化

B.资源调度

C.心跳

D.内置监控器

2015年(60)

A.可用性

B.安全性

C.可测试性

D.可修改性

2015年(61)

A.内置监控器

B.记录/回放

C.追踪审计

D.维护现有接口

【答案】B A C B B C 【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：“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”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；“在并发用户数不超过1000人时，用户的交易请求应该在0.5s内完成”，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。

# 系统可靠性

## 可靠性

知识点

1. 可靠性的定义。
2. 系统可靠性定义：系统在规定的时间内及规定的环境条件下，完成规定功能的能力，就是系统无故障运行的概率
3. 系统可靠性4个子特性
4. 成熟性：成熟性是指系统避免因错误的发生而导致失效的能力；
5. 容错性：容错性是指在系统发生故障或违反指定接口的情况下，系统维持规定的性能级别的能力；
6. 易恢复性：易恢复性是指系统发生失效的情况下，重建规定的性能级别并恢复受直接影响的数据的能力；
7. 依从性：可靠性的依从性是指系统依附于与可靠性相关的标准、约定或规定的能力。
8. 提高系统可靠性一般采用以下4类技术： (1) 冗余技术； (2) 软件容错技术； (3) 双机容错技术； (4) 集群技术。
9. 软件的可靠性设计主要包括了恢复块和N版本程序设计两种方法。
10. 恢复块方法是一种反向恢复的方法，其核心原理是：对于可靠性要求高的软件，在程序运行的某时刻，将数据或程序进行备份，一旦发现主程序块有异常发生时，可将已备份的数据或程序进行恢复，保证程序的正确性。
11. N版本程序设计方法是一种静态的故障屏蔽技术，采用前向恢复的策略，其设计思想是用N个具有相同功能的程序同时执行一项计算，结果通过多数表决来选择。其中N份程序必须由不同人设计，使用不同的方法，不同的设计语言，不同的环境，目的是减少N版本软件在表决点上相关错误的概率，可在不同计算机中执行。

试题

* 试题三【说明】某宇航公司长期从事宇航装备的研制工作，嵌入式系统的可靠性分析与设计已成为该公司产品研制中的核心工作，随着宇航装备的综合化技术发展，嵌入式软件规模发生了巨大变化，代码规模已从原来的几十万扩展到上百万，从而带来了由于软件失效而引起系统可靠性降低的隐患。公司领导非常重视软件可靠性工作，决定抽调王工程师等5人组建可靠性研究团队，专门研究提高本公司宇航装备的系统可靠性和软件可靠性问题，并要求在三个月内，给出本公司在系统和软件设计方面如何考虑可靠性设计的方法和规范。可靠性研究团队很快拿出了系统及硬件的可靠性提高方案，但对于软件可靠性问题始终没有研究出一种普遍认同的方法。

2015【问题1】

请用200字以内文字说明系统可靠性的定义及包含的4个子特性，并简要指出提高系统可靠性一般采用哪些技术？

系统可靠性定义：系统在规定的时间内及规定的环境条件下，完成规定功能的能力，就是系统无故障运行的概率。

根据国家标准《软件工程产品质量第1部分：质量模型》(GB/T16260.1—2006)的规定，系统可靠性包括：成熟性、容错性、易恢复性和可靠性的依从性4个子特性。

提高系统可靠性一般采用以下4类技术： (1) 冗余技术； (2) 软件容错技术； (3) 双机容错技术； (4) 集群技术。

本题主要考查考生对软件可靠性设计知识的掌握程度。本题首先系统可靠性定义，并了解考生是否分辨出系统可靠性的4个主要子特性；其次，考查对软件可靠性与硬件可靠性的差别，了解软件可靠性的主要特征；最后，通过填空回答宇航设备中飞行控制系统中为提高可靠性所采用的恢复块方法的具体设计实现。此类题目要求考生认真阅读题目对问题的描述，通过自己对软件可靠性和嵌入式系统的知识掌握程度，采用总结、概括和分析等的方式，可从问题描述中发现问题的相关性，正确回答问题。

可靠性的定义在不同书籍中给出解释各不相同，但是，比较公认的定义是：系统可靠性是指系统在规定的时间内及规定的环境条件下，完成规定功能的能力，就是系统无故障运行的概率。通常系统可靠性可分为硬件可靠性和软件可靠性，软件可靠性是近年来国内外研究的重点。

国家标准《软件工程产品质量第1部分：质量模型》(GB/T16260.1—2006)中给出了系统可靠性的4个主要子特性：

成熟性：成熟性是指系统避免因错误的发生而导致失效的能力；

容错性：容错性是指在系统发生故障或违反指定接口的情况下，系统维持规定的性能级别的能力；

易恢复性：易恢复性是指系统发生失效的情况下，重建规定的性能级别并恢复受直接影响的数据的能力；

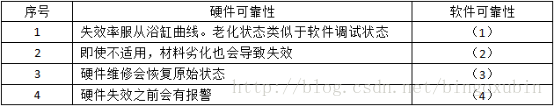
依从性：可靠性的依从性是指系统依附于与可靠性相关的标准、约定或规定的能力。

通常，提高系统可靠性采用冗余技术、软件容错技术、双机容错技术和集群技术等4类技术。

2015【问题2】

王工带领的可靠性研究团队之所以没能快速取得软件可靠性问题的技术突破，其核心原因是他们没有搞懂高可靠性软件应具备的特点。软件可靠性一般致力于系统性地减少和消除对软件程序性能有不利影响的系统故障。除非祓修改，否则软件系统不会随着时间的推移而发生退化。请根据你对软件可靠性的理解，给出表3-1所列出的硬件可靠性特征对应的软件可靠性特征之间的差异或相似之处，将答案写在答题纸上。

表3-1硬件和软件可靠性对比



(1) 不考虑软件演化的情况下，失效率在统计上是非增的 (2) 如果不使用该软件，永远不会发生失效 (3) 软件维护会创建新的软件代码(4) 软件失效之前很少会有警告

本问题主要要求考生在理解软件的特殊性基础上，深刻认识软件可靠性一般是致力于系统性地减少和消除对软件程序性能有不利影响的系统故障。除非被修改，否则软件系统不会随着时间的推移而发生退化。在掌握这一特征的基础上，可以回答硬件可靠性特征与其对应的软件可靠性特征之间的差异或相似之处。

(1)从硬件角度分析，由于硬件一旦生产完成，其可靠性指标将会随着使用时间延长而逐步老化，从而带来可靠性降低，即呈现失效率服从浴缸曲线；而软件不存在随时间延长而老化的现象，因此，在不考虑软件演化的情况下，失效率在统计上是非增的。

(2)由于硬件是由多种电子器件组成，即使不使用，材料劣化也会导致失效；而软件就不同了，软件一旦调试完成，固化到设备中，在不考虑存储介质的老化因素的前提下，即使不使用该软件，软件也永远不会发生失效。

(3)由于硬件存在可更换性，其硬件通过维修，可恢复原始状态；而对于软件而言，一旦需要维护，必然是存在需求更改、程序存在bug等现象，其维护必然会创建新的软件代码。

(4) 一般而言，硬件失效存在一个发展过程，在发生故障之前必然会有报警现象出现，而软件失效之前很少会有警告。

2015【问题3】

王工带领的可靠性研究团队在分析了大量相关资料基础上，提出软件的质量和可靠性必须在开发过程构建到软件中，也就是说，为了提高软件的可靠性，必须在需求分析、设计阶段开展软件可靠性筹划和设计。研究团队针对本公司承担的飞行控制系统制定出了一套飞控软件的可靠性设计要求。飞行控制系统是一种双余度同构型系统，输入采用了独立的两路数据通道，在系统内完成输入数据的交叉对比、表决‘制导率计算，输出数据的交叉对比、表决、一输出等功能，系统的监控模块实现对系统失效或失步的检测与宠位。其软件的可靠性设计包括恢复块方法和N版本程序设计方法。请根据恢复块方法工作原理完成图3-1，在(1)～(4)中填入恰当的内容。并比较恢复块方法与N版本程序设计方法，将比较结果(5)～(8)填入表3-2中。

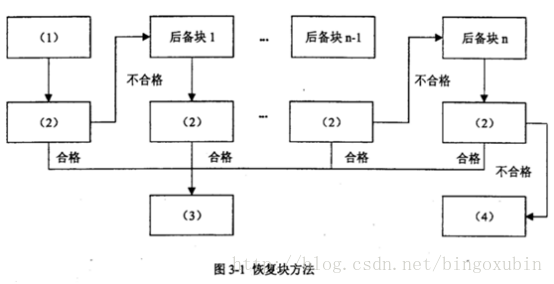
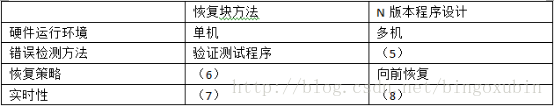


表3-2 恢复块方法与N版本程序设计的比较



1. 恢复块方法： (1) 主块 (2) 验证测试 (3) 输出正确结果 (4) 异常处理

2. 恢复块方法与N版本程序设计的比较 (5) 表决 (6) 反向恢复 (7) 差 (8) 好

软件的可靠性设计主要包括了恢复块和N版本程序设计两种方法，如果考生对这两种方法有所了解，则可很容易地进行判断。

恢复块方法是一种反向恢复的方法，其核心原理是：对于可靠性要求高的软件，在程序运行的某时刻，将数据或程序进行备份，一旦发现主程序块有异常发生时，可将已备份的数据或程序进行恢复，保证程序的正确性。基于这样的原理，显然(1)空处是主块，而(2)空处将是对正确性进行检验测试判断，一旦判断正确，(3)空处将是“输出正确结果”，以此类推，显然(4)空处是进入“异常处理”了。

如果考生能够答对第一问，说明考生已掌握恢复块与N版本两种方法，那么，填写第二问的空应该没问题的。在表3-2已经给出了两种方法适应的硬件环境，也就是说：恢复块方法的使用必然是单机环境，而N版本方法必然要使用多机环境。理解了环境需求，那么多机余度工作方式，必然采用“表决”((5)空)方式进行容错；同时不难分析出N版本的工作原理是向前恢复，恢复快则是反向恢复((6)空)；由于恢复快方法是反复寻找正确的备份块，而N版本方法则是多个机器同时计算同样内容，表决完后即可给出正确结果，这样，恢复快方法相比N版本方法显然实时性差((7)空)，而N版本方法显然好((8)空)于恢复快方法。

# 软件架构风格

## 软件架构风格分类

知识点

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。架构风格定义了一类架构所共有的特征，主要包括架构定义、架构词汇表和架构约束，反映了领域中众多系统所共有的结构和语义两个方面的特征。

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。

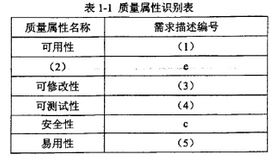
软件架构设计的一个核心问题是能否使用重复的架构模式，即能否达到架构级的软件重用

|  |  |
| --- | --- |
| **架构风格** | **说明** |
| 批处理 | 传统的编译器 |
| 管道/过滤器 | 支持分阶段数据处理 |
| 现代的编译器 |  |
| 主程序/程序 |  |
| 面向对象风格 |  |
| 层次结构 | 采用层次化架构风格的系统，划分的层次越多，系统的性能越差 |
| 过程控制 | 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 |
| 进程通信 |  |
| 隐式调用 | 回调机制 采用隐式调用架构风格的系统，可以通过处理函数的并发调用提高系统处理性能 |
| 事件系统 | 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 |
| 解释器 | 运行时的系统行为定义与改变的能力 采用解释器架构风格的系统，可以通过部分解释代码预先编译的方式提高系统性能 |
| 基于规则的系统 |  |
| 超文本系统 |  |
| 数据库系统 |  |
| 黑板 | 信号处理领域 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 |
| 虚拟机 | 通过虚拟机架构屏蔽不同的硬件环境 |
| C2体系结构风格 | 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 |
| 数据仓库 | 编程语言的集成开发环境 |

试题

* 阅读以下关于软件架构设计的叙述，在答题纸上回答问题1至问题3 。【说明】某软件公司为某品牌手机厂商开发一套手机应用程序集成开发环境，以提高开发手机应用程序的质量和效率。在项目之初，公司的系统分析师对该集成开发环境的需求进行了调研和分析，具体描述如下：a．需要同时支持该厂商自行定义的应用编程语言的编辑、界面可视化设计、编译、调试等模块，这些模块产生的模型或数据格式差异较大，集成环境应提供数据集成能力。集成开发环境还要支持以适配方式集成公司现有的应用模拟器工具。b．经过调研，手机应用开发人员更倾向于使用Windows系统，因此集成开发环境的界面需要与Windows平台上的主流开发工具的界面风格保持一致。c．支持相关开发数据在云端存储，需要保证在云端存储数据的机密性和完整性。d．支持用户通过配置界面依据自己的喜好修改界面风格，包括颜色、布局、代码高亮方式等，配置完成后无需重启环境。e．支持不同模型的自动转换。在初始需求中定义的机器性能条件下，对于一个包含50个对象的设计模型，将其转换为相应代码框架时所消耗时间不超过5秒。f．能够连续运行的时间不小于240小时，意外退出后能够在10秒之内自动重启。g．集成开发环境具有模块化结构，支持以模块为单位进行调试、测试与发布。h．支持应用开发过程中的代码调试功能：开发人员可以设置断点，启动调试，编辑器可以自动卷屏并命中断点，能通过变量监视器查看当前变量取值。在对需求进行分析后，公司的架构师小张查阅了相关的资料，认为该集成开发环境应该采用管道一过滤器(Pipe-Filter)的架构风格，公司的资深架构师王工在仔细分析后，认为应该采用数据仓储（Data Repository）的架构风格。公司经过评审，最终采用了王工的方案。

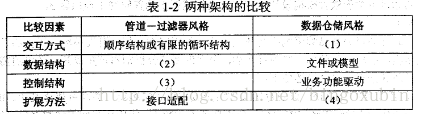
2016年【问题1】（10分） 识别软件架构质量属性是进行架构设计的重要步骤。请分析题干中的需求描述，填写表1-1中(1)～(5)处的空白。



（1）f （2）性能 （3）g （4）h （5）b

2016年【问题2】（7分）

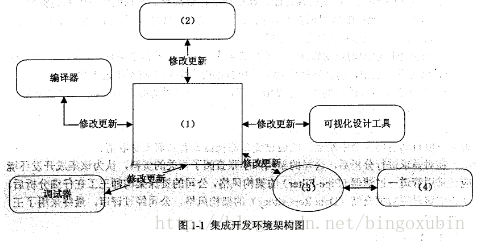
请在阅读题干需求描述的基础上，从交互方式、数据结构、控制结构和扩展方法4个方面对两种架构风格进行比较，填写表1-2中(1)～(4)处的空白。



1. 星型 （2）数据流 （3）数据流驱动 （4）模型适配

2016年【问题3】（8分）

在确定采用数据仓库架构风格后，王工给出了集成开发环境的架构图。请填写图1-1中(1)～(4)处的空白，完成该集成开发环境的架构图。



（1）语法结构树 （2）编辑器 （3）适配器 （4）应用模拟器工具

* 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。以下关于编译器架构的叙述中，错误的是（56）。

(56)

A.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

B.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

C.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

D.现代的编译器釆用以数据共享为中心的架构风格，但由于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，因此分析树不是数据中心的共享数据

【答案】D 【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。早期的编译器釆用管道-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。

* 某公司欲开发一种工业机器人，用来进行汽车零件的装配。公司的架构师经过分析与讨论，给出了该机器人控制软件的两种候选架构方案：闭环控制和分层结构。以下对于这两种候选架构的选择理由，错误的是（55）。

2009年(55)

A.应该采用闭环控制架构，因为闭环结构给出了将软件分解成几个协作构件的方法，这对于复杂任务特别适合

B.应该采用闭环控制结构，因为闭环控制架构中机器人的主要构件（监控器、传感器、发动机等）是彼此分开的，并能够独立替换

C.应该采用分层结构，因为分层结构很好地组织了用来协调机器人操作的构件，系统结构更加清晰

D.应该采用分层结构，因为抽象层的存在，满足了处理不确定性的需要：在较低层次不确定的实现细节在较髙层次会变得确定

【答案】A 【解析】

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。

* 某公司欲开发一个基于图形用户界面的集成调试器。该调试器的编辑器和变量监视器可以设置调试断点。当调试器在断点处暂停运行时，编辑程序可以自动卷屏到断点，变量监视器刷新变量数值。针对这样的功能描述，采用（54)的架构风格最为合适。

2009年(54)

A.数据共享

B.虚拟机

C.隐式调用

D.显式调用

【答案】C 【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。

* 某软件开发公司负责开发一个Web服务器服务端处理软件，其核心部分是对客户端请求消息的解析与处理，包括HTTP报头分离、SOAP报文解析等功能。该公司的架构师决定采用成熟的架构风格指导整个软件的设计，以下（53)架构风格，最适合该服务端处理软件。

2009年(53)

A.虚拟机

B.管道-过滤器

C.黑板结构

D.分层结构

【答案】B 【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。

* Windows操作系统在图形用户界面处理方面采用的核心架构风格是（51)风格。Java语言宣传的“一次编写，到处运行”的特性，从架构风格上看符合（52）风格的特点。

2009年(51)

A.虚拟机

B.管道-过滤器

C.事件驱动

D.微内核-扩展

2009年(52)

A.虚拟机

B.管道-过滤器

C.事件驱动

D.微内核-扩展

【答案】C A 【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。

* 某公司欲开发一个语音识别系统，语音识别的主要过程包括分割原始语音信号、识别音素、产生候选词、判定语法片断、提供语义解释等。每个过程都需要进行基于先验知识的条件判断并进行相应的识别动作。针对该系统的特点，采用(52)架构风格最为合适。

(52)

A.解释器

B.面向对象

C.黑板

D.隐式调用

【答案】C 【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。

* 某公司欲开发一个漫步者机器人，用来完成火星探测任务。机器人的控制者首先定义探测任务和任务之间的时序依赖性，机器人接受任务后，需要根据自身状态和外界环境进行动态调整，最终自动完成任务。针对这些需求，该机器人应该采用(51)架构风格最为合适。

2010年(51)

A.解释器

B.主程序•子程序

C.隐式调用

D.管道-过滤器

【答案】C 【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，漫步者机器人需要根据自身状态和外界环境进行自动调整，这是一个典型的根据外部事件进行响应的场景。比较4个候选项，隐式调用比较适合根据外部事件进行处理和动作的情景。

* 某公司承接了一个开发家用空调自动调温器的任务，调温器测量外部空气温度，根据设定的期望温度控制空调的开关。根据该需求，公司应采用(50)架构风格最为合适。

2010年(50)

A.解释器

B.过程控制

C.分层

D.管道-过滤器

【答案】B 【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，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。

* 某游戏公司欲开发一个大型多人即时战略游戏，游戏设计的目标之一是能够支持玩家自行创建战役地图，定义游戏对象的行为和对象之间的关系。针对该目标，公司应该采用(48)架构风格最为合适。

2010年(48)

A.管道-过滤器

B.隐式调用

C.主程序-子程序

D.解释器

【答案】D 【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。

* 编译器的主要工作过程是将以文本形式输入的代码逐步转化为各种形式，最终生成可执行代码。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。针对这种特征，现代编译器应该采用（52)架构风格最为合适。

2011年(52)

A.数据共享

B.虚拟机

C.隐式调用

D.管道-过滤器

【答案】A 【解析】本题主要考查对架构风格的理解和掌握。

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。

* 某企业内部现有的主要业务功能己经封装为Web服务。为了拓展业务范围，需要将现有的业务功能进行多种组合，形成新的业务功能。针对业务灵活组合这一要求，采用(51)架构风格最为合适。

2011年(51)

A.管道-过滤器

B.解释器

C.显式调用

D.黑板

【答案】B 【解析】本题主要考查对架构风格的理解和掌握。

根据题干描述，需要将现有的业务功能进行多种组合，形成新的业务功能。这种情况下，可以将业务功能封装成服务，并通过某种语言对业务流程进行描述，通过一个解释引擎对流程描述进行解释和执行。根据上述分析，选项中列举的架构风格中，解释器风格最符合要求。

* 某公司研发一种语音识别软件系统，需要对用户的语音指令进行音节分割、重音判断、语法分析和语义分析，最终对用户的意图进行推断。针对上述功能需求，该语音识别软件应该采用（50)架构风格最为合适。

2011年(50)

A.隐式调用

B.管道-过滤器

C.解释器

D.黑板

【答案】D 【解析】本题主要考查对架构风格的理解和掌握。

根据题干描述，语音识别软件需要对用户的语音指令进行音节分割、重音判断、语法分析和语义分析，最终对用户的意图进行推断。由于语音识别具有不确定性，需要人工智能技术的支持和专家意见的汇总和决策，并且需要支持识别过程中的推理和决策。根据上述分析，选项中列举的架构风格中，黑板风格最符合要求。

(44) 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（45)。

2011年(44)

A.架构风格

B.开发方法

C.设计模式

D.分析模式

2011年(45)

A.架构风格

B.开发方法

C.设计模式

D.分析模式

【答案】A C 【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。

* 以下关于软件架构风格与系统性能关系的叙述，错误的是（16)。

2012年(16)

A.对于采用层次化架构风格的系统，划分的层次越多，系统的性能越差

B.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 处理提高系统性能

C.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

D.对于采用过程调用架构风格的系统，可以通过将显式调用策略替换为隐式调用策略提高系统性能

【答案】D 【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。

* “编译器”是一种非常重要的基础软件，其核心功能是对源代码形态的单个或一组源程序依次进行预处理、词法分析、语法分析、语义分析、代码生成、代码优化等处理，最终生成目标机器的可执行代码。考虑以下与编译器相关的软件架构设计场景：传统的编译器设计中，上述处理过程都以独立功能模块的形式存在，程序源代码作为一个整体，依次在不同模块中进行传递，最终完成编译过程。针对这种设计思路，传统的编译器采用（47)架构风格比较合适。随着编译、链接、调试、执行等开发过程的一体化趋势发展，集成开发环境（IDE)随之出现。IDE集成了编译器、连接器、调试器等多种工具，支持代码的增量修改与处理，能够实现不同工具之间的信息交互，覆盖整个软件开发生命周期。针对这种需求，IDE采用（48)架构风格比较合适。IDE强调交互式编程，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，针对这种需求，通常采用（49)架构风格比较合适。某公司己经开发了一款针对某种嵌入式操作系统专用编程语言的IDE，随着一种新的嵌入式操作系统上市并迅速占领市场，公司决定对IDE进行适应性改造，支持釆用现有编程语言进行编程，生成符合新操作系统要求的运行代码，并能够在现有操作系统上模拟出新操作系统的运行环境，以支持代码调试工作。针对上述要求，为了使IDE能够生成符合新操作系统要求的运行代码，采用基于（50)的架构设计策略比较合适;为了模拟新操作系统的运行环境，通常采用（51)架构风格比较合适。

2013年(47)

A.管道一过滤器

B.顺序批处理

C.过程控制

D.独立进程

2013年(48)

A.规则引擎

B.解释器

C.数据共享

D.黑板

2013年(49)

A.隐式调用

B.显式调用

C.主程序一子程序

D.层次结构

2013年(50)

A.代理

B.适配

C.包装

D.模拟

2013年(51)

A.隐式调用

B.仓库结构

C.基于规则

D.虚拟机

【答案】B C A B D 【解析】本题主要考查软件架构风格的理解和掌握。

根据题干描述，传统的编译器设计中，编译处理过程都以独立功能模块的形弍存在，程序源代码作为一个整体，依次在不同模块中进行传递，最终完成编译过程。针对这种设计思路，传统的编译器采用顺序批处理架构风格比较合适，因为在顺序批处理架构风格中，数据以整体的方式在不同的处理模块之间传递，符合题目要求。

集成开发环境（IDE)需要面对不同的数据结构，不同的数据类型与形态，在这种以数据为核心的系统中，采用数据共享机制显然是最为合适的。IDE强调交互式编程，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，这一需求的核心在于根据事件进行动作响应，采用隐式调用的架构风格最为合适。

根据题干描述，公司需要对IDE进行适应性改造，支持采用现有编程语言进行编程，生成符合新操作系统要求的运行代码，并能够在现有操作系统上模拟出新操作系统的运行环境，以支持代码调试工作。针对上述要求，为了使IDE能够生成符合新操作系统要求的运行代码，应该是现有操作系统对新系统的一个适配过程，因此应该采用适配器架构设计策略比较合适，模拟新操作系统的运行模式通常会采用虚拟机架构风格。

* 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。架构风格定义了一类架构所共有的特征，主要包括架构定义、架构词汇表和架构（43).

2013年(43)

A.描述

B.组织

C.约束

D.接口

【答案】C 【解析】本题主要考查软件架构风格的定义。

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。架构风格定义了一类架构所共有的特征，主要包括架构定义、架构词汇表和架构约束。

* 软件架构风格描述某一特定领域中的系统组织方式和惯用模式，反映了领域中众多系统所共有的（51）特征。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。通常会采用（52）架构风格。

2014年(51)

A.语法和语义

B.结构和语义

C.静态和动态

D.行为和约束

2014年(52)

A.管道-过滤器

B.解释器

C.黑板

D.过程控制

【答案】B C 【解析】

软件架构风格描述某一特定领域中的系统组织方式和惯用模式，反映了领域中众多系统所共有的结构和语义两个方面的特征。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，通常会采用黑板架构风格，以知识为中心进行分析与推理。

* （44）架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。

2016年(44)

A.C2

B.黑板系统

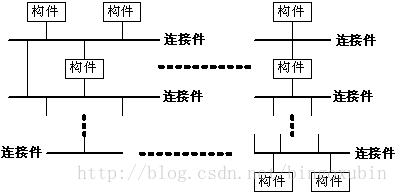
C.规则系统

D.虚拟机

【答案】A

【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。C2风格中的系统组织规则如下：(1)系统中的构件和连接件都有一个顶部和一个底部；(2)构件的顶部应连接到某连接件的底部，构件的底部则应连接到某连接件的顶部，而构件与构件之间的直接连接是不允许的；(3)一个连接件可以和任意数目的其它构件和连接件连接； (4)

当两个连接件进行直接连接时，必须由其中一个的底部到另一个的顶部。



* 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。一个体系结构定义了一个词汇表和一组（49）。架构风格反映领域中众多系统所共有的结构和（50）。

2016年(49)

A.约束

B.连接件

C.拓扑结构

D.规则

2016年(50)

A.语义特征

B.功能需求

C.质量属性

D.业务规则

【答案】A A

【解析】软件体系结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。体系结构风格定义一个系统家族，即一个体系结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。

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。对软件体系结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。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，立刻就会明白系统是如何组织和工作的。

* 以下关于软件架构风格与系统性能的关系叙述中，错误的是（16）。

2015年(16)

A.对于采用层次化架构风格的系统，划分的层次越多，系统的性能越差

B.对于采用隐式调用架构风格的系统，可以通过处理函数的并发调用提高系统处理性能

C.采用面向对象架构风格的系统，可以通过引入对象管理层提高系统性能

D.对于采用解释器架构风格的系统，可以通过部分解释代码预先编译的方式提高系统性能

【答案】C 【解析】 采用面向对象架构风格的系统，可以通过引入对象管理层提高系统性能。

抽象数据类型概念对软件系统有重要作用，目前软件界已普遍转向使用面向对象系统。这种风格建立在数据抽象和面向对象的基础上，数据的表示方法和它们的相应操作封装在一个抽象数据类型或对象中。这种风格的构件是对象，或者说是抽象数据类型的实例。对象是一种被称作管理者的构件，因为它负责保持资源的完整性。对象是通过函数和过程的调用来交互的。可以通过减少功能调用层次提高系统性能。

* 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。架构风格反映领域中众多系统所共育的结构和（40），强调对架构（41）的重用。

2015年(40)

A.语义特性

B.功能需求

C.质量属性

D.业务规则

2015年(41)

A.分析

B.设计

C.实现

D.评估

【答案】A B 【解析】本题考查软件架构风格方面的基础知识。

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。

* 某公司拟开发一个地面清洁机器人。机器人的控制者首先定义清洁任务和任务之间的关系，机器人接受任务后，需要响应外界环境中触发的一些突发事件，根据自身状态进行动态调整，最终自动完成任务。针对上述需求，该机器人应该采用（46）架构风格最为合适。

2015年(46)

A.面向对象

B.主程序-子程序

C.规则系统

D.管道-过滤器

【答案】C 【解析】本题考查架构风格与架构设计策略的理解与掌握。

根据题目描述，机器人需要根据自身状态的外界环境进行自动调整，这是一个典型的根据外部事件进行响应的场景。比较4个候选项，规则系统比较适合根据外邹事件，以自身状态为基础自动进行处理和动作的场景。

* 某公司拟开发一个语音识别系统，其语音识别的主要过程包括分割原始语音信号、识别音素、产生候选词、判定语法片断、提供语义解释等，每个过程都需要进行基于先验知识的条件判断并进行相应的识别动作。针对该系统的特点，采用（47）架构风格最为合适。

2015年(47)

A.解释器

B.面向对象

C.黑板

D.隐式调用

【答案】C 【解析】本题考查架构风格与架构设计策略的理解与掌握。

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。

* 某公司拟开发了个轿车巡航定速系统，系统需要持续测量车辆当前的实时速度，并根据设定的期望速度启动控制轿车的油门和刹车。针对上述需求，采用（48）架构风格最为合适。

2015年(48)

A.解释器

B.过程控制

C.分层

D.管道-过滤器

【答案】B 【解析】本题考查架构风格与架构设计策略的理解与掌握。

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。

* 某公司拟开发一套在线游戏系统，该系统的设计目标之一是支持用户自行定义游戏对象属性，行为和对象之间的交互关系。为了实现上述目标，公司应该采用（49）架构风格最为合适。

2015年(49)

A.管道-过滤器

B.隐式调用

C.主程序-子程序

D.解释器

【答案】D 【解析】本题主要考查软件架构设计策略与架构风格的理解与掌握。

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。

* 某公司为其研发的硬件产品设计实现了一种特定的编程语言，为了方便开发者进行软件开发，公司拟开发一套针对该编程语言的集成开发环境，包括代码编辑、语法高亮、代码编译、运行调试等功能。针对上述描述，该集成开发环境应采用（50）架构风格最为合适。

2015年(50)

A.管道-过滤器

B.数据仓储

C.主程序-子程序

D.解释器

【答案】B 【解析】本题主要考查软件架构设计策略与架构风格的理解与掌握。

根据题干描述，编程语言的集成开发环境需要提供代码编辑、语法高亮、代码编译、运行调试等功能，这些功能的特点是以软件代码为中心进行对应的编译处理与辅助操作。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数据仓库风格。

* 某公司拟为某种新型可编程机器人开发相应的编译器。该编译过程包括词法分析、语法分析、语义分析和代码生成四个阶段，每个阶段产生的结果作为下一个阶段的输入，且需独立存储。针对上述描述，该集成开发环境应采用（48）架构风格最为合适。

2016年(48)

A.管道-过滤器

B.数据仓储

C.主程序-子程序

D.解释器

【答案】A

【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。

* 某公司拟开发一个扫地机器人。机器人的控制者首先定义清洁流程和流程中任务之间的关系，机器人接受任务后，需要响应外界环境中触发的一些突发事件，根据自身状态进行动态调整，最终自动完成任务。针对上述需求，该机器人应该采用（51）架构风格最为合适。

2016年(51)

A.面向对象

B.主程序-子程序

C.规则系统

D.管道-过滤器

【答案】C

【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。

* 某企业内部现有的主要业务功能已封装成为Web服务。为了拓展业务范围，需要将现有的业务功能进行多种组合，形成新的业务功能。针对业务灵活组合这一要求，采用（52）架构风格最为合适。

2016年(52)

A.规则系统

B.面向对象

C.黑板

D.解释器

【答案】D

【解析】 依据题目要求，需要灵活组合业务，形成新的业务功能，这样虚拟机风格较为合适。但备选答案中A与D均属于虚拟机风格。

* 某公司拟开发一个语音搜索系统，其语音搜索系统的主要工作过程包括分割原始语音信号、识别音素、产生候选词、判定语法片断、提供搜索关键词等，每个过程都需要进行基于先验知识的条件判断并进行相应的识别动作。针对该系统的特点，采用（53）架构风格最为合适。

2016年(53)

A.分层系统

B.面向对象

C.黑板

D.隐式调用

【答案】C

【解析】 语音识别是黑板风格的经典应用。

# 层次系统架构风格

## 二层及三层 C/S 架构风格

知识点

1. 两层C/S架构

胖客户机，瘦服务端

服务器负责数据管理，客户端完成与用户的交互

表示层、数据层

分支主题

1. 三层C/S架构

表示层

功能层

数据层

优点

允许合理地划分三层结构，整个系统结构清晰

在处理负荷能力上与处理上有优势

各层可并行开发

1. 分布式系统5个逻辑计算层

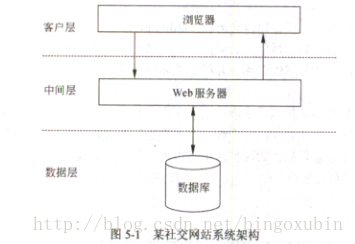
分布式系统开发分为5个逻辑计算层：

* 1. 表示层实现用户界面；
  2. 表示逻辑层包括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；
  3. 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；
  4. 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：
  5. 数据层是数据库中实际存储的业务数据

1. 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
2. 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；
3. 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；
4. 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
5. J2EE的体系结构
6. Web浏览器:也称Web客户端，以标准格式来显示从服务器传递过来的网页，它们传递给浏览器时已经是HTML或者XML格式，浏览器正确的显示给用户
7. 小应用程序（Applet）:是嵌在浏览器中的一种轻量级客户端，当web页面不能充分的表现数据或者应用界面的时候，才使用它，Applet是一种替代web页面的手段，能够使用J2SE开发Applet，Applet无法使用J2EE中的各种Service和API。需要运行在客户端安装了Java虚拟机的Web浏览器上。
8. 两大容器:EJB容器+Web容器，即业务逻辑层+表示层
   1. Web容器:管理所有的Servlet等Web组件的运行，对响应客户请求和返回结果提供了运行时的支持。
   2. EJB容器：负责所有的EJB的运行，支持EJB组件的事务处理和生命周期管理，以及Bean的查找和其他服务，支持J2EE多层架构的基础结构，是一个控制业务实现的运行期环境，并提供事务服务、持久性、安全性等重要的系统服务，让开发人员不必开发基础服务而将注意力集中在业务逻辑的实现
9. 两大组件:Web组件+Ejb组件
   1. Web组件:与基于Web的客户端进行交互，J2EE中有三类Web组件：Servlet、JSP、JavaBean，Servlet是Web服务器的功能扩展，接受Web请求，返回动态的Web页面。Web容器中的组件可以使用EJB中的组件来完成复杂的业务逻辑。值得注意的是静态的HTML页面和Applets不算是Web层组件。
   2. EJB组件:包含三种不同类型的EJB：会话Bean、消息驱动Bean、实体Bean
      1. 第一：会话Bean：着重业务逻辑的实现与控制，负责与Web层通信，给Web层提供访问业务数据的接口。当客户端完成执行过程的时候，会话Bean及相关数据会消失。
      2. 第二：实体Bean：代表持久数据，数据相当于存储在数据库表中，它负责保存业务数据，给会话Bean访问业务数据的接口。
      3. 第三：消息驱动Bean：用于接收、处理客户通过JMS发送过来的消息，允许业务组件接收衣服的JMS消息。

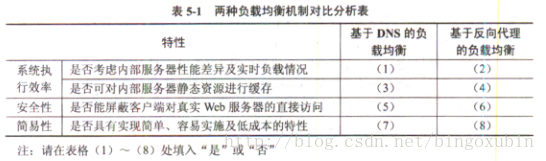
试题

* 试题五【说明】某软件公司开发运维了一个社交网站系统，该系统基于开源软件平台LAMP(Linux+Apache+MySQL+PHP)构建，运行一段时间以来，随着用户数量及访问量的增加，系统在Web服务器负载、磁盘I/O等方面出现了明显瓶颈，已不能满足大量客户端并发访问的要求，因此公司成立了专门的项目组，拟对系统架构进行调整以提高系统并发处理能力。目前系统采用了传统的三层结构，系统架构如图 5-1 所示。



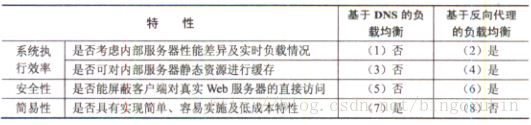
【2014问题1】

针对目前出现的Web服务器负载过大问题，项目组决定在客户端与中间层Web服务器之间引入负载均衡器，通过中间层Web服务器集群来提高Web请求的并发处理能力。在讨论拟采用的负载均衡机制时，王工提出采用基于DNS的负载均衡机制，而李工则认为应采用基于反向代理的负载均衡机制，项目组经过讨论，最终确定采用李工提出的方案。请用200字以内的文字，分别简要说明两个机制的基本原理；并从系统执行效率、安全性及简易性等方面将两种机制进行对比，将对比结果填入表5-1 中。



两种机制的基本原理： 基于DNS的负载均衡机制通过DNS服务器实现，通常通过循环复用具有同一域名的多个主机地址的服务器实现负载均衡。

反向代理负载均衡则是将来自Internet的连接请求以反向代理的方式动态转发给内部网络上的多台服务器进行处理，从而达到负载均衡的目的。



本题考查大型Web应用访问效率优化相关技术手段。 本问题考查典型负载机制相关知识。

负载均衡机制是大型Web应用解决高负荷访问和大量并发请求时常用的有效解决方法，典型的负载均衡机制包括基于DNS的负载均衡、基于反向代理的负载均衡等。

基于DNS的负载均衡机制通过DNS服务器实现，通常通过循环复用具有同一域名的多个主机地址的服务器实现负载均衡，可以看出，该机制具有实现简单、容易实施及低成本的特性。反向代理负载均衡则是将来自Internet的连接请求以反向代理的方式动态转发给内部网络上的多台服务器进行处理，从而达到负载均衡的目的。

从系统执行效率方面讲，基于DNS的负载均衡机制实现简单，但其通常不能区分服务器的差异，也不能反映服务器的当前运行状态。基于反向代理的则可以根据内部服务器的性能差异及实时负载情况进行动态负载均衡，当系统多个Web服务器性能存在明显差异或内部Web服务器出现故障时，负载均衡器可以更快做出响应，从而保证客户端的访问效率。采用基于反向代理的负载均衡机制，可在代理服务器中引入调速缓存机制，对Web服务器返回的静态页面或图片等静态资源进行缓存，由代理服务器承担对原始服务器的静态资源访问请求，从而进一步降低原始Web服务器的负载。

从安全性方面讲，采用基于反向代理的负载均衡机制，代理服务器屏蔽了客户端对真实Web服务器的直接访问，恶意用户无法对真实Web服务器进行攻击，且可以通过代理服务器为原本不安全的客户端与Web服务器之间的连接建立安全通道。因此采用基于反向代理的负载均衡机制可为系统提供更好的安全性保障。

【2014问题2】

针对并发数据库访问所带来的磁盘I/O瓶颈问题，项目组决定在数据层引入数据库扩展机制。经过调研得知系统数据库中存储的主要数据为以用户标识为索引的社交网络数据，且系统运行时发生的大部分数据库操作为查询操作。经过讨论，项目组决定引入数据库分区和MySQL主从复制两种扩展机制。数据库分区可采用水平分区和垂直分区两种方式，请用350字以内的文字说明在本系统中应采用哪种方式及其原因，并分析引入主从复制机制给系统带来的好处。

(1)

本系统中应主要使用水平分区机制。根据已知信息，系统数据库中存储的主要数据为以用户标识为索引的社交网络数据，采用水平分区机制可根据用户标识将用户数据进行水平分割，用户操作时先将请求分发到不同数据库分区，再进行具体数据库操作，以提高数据库访问效率。

(2) 引入主从复制机制所带来的好处：

①避免数据库单点故障：主服务器实时、异步复制数据到从服务器，当主数据库宕机时，可在从数据库中选择一个升级为主服务器，从而防止数据库单点故障。

②提高査询效率：根据系统数据库访问特点，可以使用主数据库进行数据的插入、删除及更新等写操作，而从数据库则专门用来进行数据査询操作，从而将査询操作分担到不同的从服务器以提高数据库访问效率。

本问题考查数据库集群相关知识。

根据已知信息，系统数据库中存储的主要数据为以用户标识为索引的社交网络数据，采用水平分区机制可根据用户标识将用户数据进行水平分割，用户操作时先将请求分发到不同数据库分区，再进行具体数据库操作，以提高数据库访问效率。因此本系统中应主要使用水平分区机制。

在系统中引入主从复制机制，首先可以利用主从复制机制的备份功能避免系统数据库单点故障；其次已知“系统运行时发生的大部分数据库操作为查询操作”。因此可以使用主数据库进行数据的插入、删除及更新等写操作，而从数据库则专门用来进行数据查询操作，从而将查询操作分担到不同的从服务器以提高数据库访问效率。

【2014问题3】 为进一步提高数据库访问效率，项目组决定在中间层与数据层之间引入缓存机制。赵工开始提出可直接使用MySQL的查询缓存(query cache)机制，但项目组经过分析好友动态显示等典型业务的操作需求，同时考虑已引入的数据库扩展机制，认为查询缓存尚不能很好地提升系统的查询操作效率，项目组最终决定在中间层与数据层之间引入Memcached分布式缓存机制。

（a）请补充下述关于引入Memcached后系统访问数据库的基本过程：系统需要读取后台数据时，先检查数据是否存在于（1）中，若存在则直接从其中读取，若不存在则从（2）中读取并保存在（3）中；当（4）中数据发生更新时，需要将更新后的内容同步到（5）实例中。（备选答案：数据库、Memcacbed缓存） （b）请结合已知信息从缓存架构、缓存有效性及缓存数据类型等方面分析使用Memcached代替数据库查询缓存的原因。

(a) (1) Memcached缓存 (2) 数据库 (3) Memcached缓存 (4) 数据库 (5) Memcached缓存

(b) 使用Memcached代替数据库查询缓存的原因： 2014年(1)

缓存架构：数据库查询缓存通常每个数据库只有一个实例，因此存储内容受数据库服务器可用内存限制，可缓存数据有限。而Memcached可采用高速分布式缓存服务器结构，不受数据库服务器约束，可扩展性更好。

(2)

缓存有效性：数据库查询缓存只要在发生写操作时就会失效，即使更新的是数据库中的其他行。而Memcached可通过键值将数据进行散列缓存，有效降低缓存的更新频率，从而提髙缓存的有效性。

(3)

缓存数据类型：数据库查询缓存只能缓存数据库行，对社交网站好友动态显示等典型业务所需要的组合数据缓存缺乏有效支持，而Memcached理论上可缓存任何内容。因此可以将分散在数据库中的关系或者列表组合后进行缓存，以提高缓存数据的针对性和效率。

本问题考查数据缓存机制相关知识。

引入Memcached后系统访问数据库的基本过程为：系统需要读取后台数据时，先检査数据是否存在于Memcached中，若存在则直接从Memcached中读取，或不存在则从数据库中读取并保存在Memcached中；当系统数据库中数据发生更新时，需要将更新后的内容同步到Memcached缓存实例中。

与MySQL查询缓存相比，使用Memcached机制存在以下优势： 2014年(1)

缓存架构：数据库查询缓存通常每个数据库只有一个实例，因此存储内容受数据库服务器可用内存限制，可缓存数据有限；而Memcached可采用高速分布式缓存服务器结构，不受数据库服务器约束，可扩展性更好。

(2)

缓存有效性：数据库查询缓存只要在发生写操作时就会失效，即使更新的是数据库中的其他行；而Memcached可通过键值将数据进行散列缓存，有效降低缓存的更新频率，从而提高缓存的有效性。

(3)

缓存数据类型：数据库杏询缓存只能缓存数据库行，对社交网站好友动态显示等典型业务所需要的组合数据缓存缺乏有效支持，而Memcached理论上可缓存任何内容，因此可以将分散在数据库中的关系或者列表组合后进行缓存，以提高缓存数据的针对性和效率。

* 试题五【说明】某信息技术公司计划开发一套在线投票系统，用于为市场调研、信息调查和销售反馈等业务提供服务。该系统计划通过大量宣传和奖品鼓励的方式快速积累用户，当用户规模扩大到一定程度时，开始联系相关企业提供信息服务，并按照信息服务种类和用户投票数量收取费用。为了降低开发成本和提高开发效率，项目组经过讨论后决定采用轻量级Java EE开发框架设计系统应用架构。在应用架构设计中，除了满足系统主要功能需求，还需要考虑的因素包括：(1)项目开发采用MySQL数据库存储数据，将来可能移植到其它数据库平台；(2)系统开发过程中尽可能降低或者消除SQL语句开发的工作量；(3)投票系统中数据之间的关系复杂，需要支持数据对象的聚合和继承等关系。项目组基于MVC模式设计出了投票系统的架构，包括表示层、业务逻辑层、数据持久层和数据层。在具体讨论数据持久层采用哪种技术方案时，老王建议采用成熟的Hibernate框架，小李则认为iBatis更加灵活，更适合作为投票系统数据持久层开发技术。

2015【问题1】

请用300以内文字说明什么是数据持久层，使用数据持久层能够为项目开发带来哪些好处？

数据持久层是根据分层思想，通过建立逻辑数据操作接口，采取一定的对象/关系映射策略，隐藏数据库访问代码细节，向业务开发人员提供透明的对象持久化操作机制。能够为项目开发带来的好处：

(1)分离业务逻辑层和数据层，降低两者之间的耦合； (2) 通过对象/关系映射向业务逻辑提供面向对象的数据访问； (3)简化数据层访问，隐藏数据库链接、数据读写命令和事务管理细节。

本题考查Web应用系统数据持久层设计相关知识。

此类题目要求考生了解轻量级JavaEE开发框架和常用的数据持久层技术，通过认真阅读题目对实际问题的描述，能够根据在线投票系统架构设计中所考虑的因素，选择合适的数据持久层技术。主流的数据持久层技术按照其实现思路可以分为4类技术方案，包括基于数据库连接UDBC封装)、命令转换(SQL Mapping)、对象关系映射(0/RMapping)和数据持久化对象(Entity Bean),考生在实际应用中，不仅要掌握技术使用方法，更重要的是能够了解其基本原理，便于在实际应用架构设计中选择合适的技术。

本问题考查数据持久层的基本概念以及在实际应用中能够为项目开发带来的好处。数据持久层主要是根据分层思想，通过建立逻辑数据操作接口，采取一定的对象/关系映射策略，隐藏数据库访问代码细节，向业务开发人员提供透明的对象持久化操作机制。具体来说，数据持久层能够带来的好处能够在三层或者多层结构设计中分离业务逻辑层和数据层，解耦两者之间的直接关联；通过对象/关系映射，将面向业务逻辑的数据处埋全部以对象形式暴露，将对对象的操作自动转换为基于关系模式的数据库访问操作；在复杂的数据访问操作中，能够利用数据持久层简化数据层的访问，隐藏数据库链接、数据读写命令和事务管理细节，有效提升系统开发效率。

2015

【问题2】

针对在线投票系统的实际应用需求和要求，项目组应选用哪种技术实现数据持久层？请用200字以内文字说明其采用该技术的原因。

项目组应该采用Hibernate框架。 原因： (1) Hibernate支持多种不同类型数据库，满足项目组数据库移植需求； (2)Hibernate相对于iBatis减少了SQL语句开发的工作量； (3)iBatis生成的P0是扁平化的，无法像Hibernate—样支持对象的继承和聚合等立体化关系。

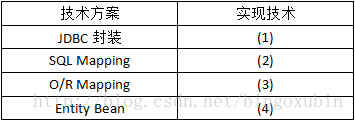
本问题考查考生对Hibernate和iBatis这两种持久层技术的掌握情况。Hibernate和iBatis是轻量级hvaEE框架中两种数据持久层技术，两者都是优秀的开源项目。iBatis相对简单易学而且更灵活，但开发工作量较大，数据之间是关联关系；Hibernate框架相对复杂，所生成的持久化对象能够表达面向对象中的继承和聚合等关系，开发工作量较小，Hibernate使用更广泛更成熟，能够适应目前所有主流的关系型数据库。根据题目中所描述的考虑因素，系统需要满足数据库的可移植性、开发过程简单、支持对象间的复杂关系等需求，所以选择Hibernate作为数据持久层技术更为合适。

2015

【问题3】

数据持久层是Web应用系统框架中重要的组成部分，主流的数据持久层技术分别基于不同的技术方案，请在表5-1中（1）-（4）处分别根据（a）~（d）所列技术的方案类别填入其序号。

表5-1 数据持久层技术分类



(a) BMP, CMP (b) iBatis/MyBatis (c) SprmgJdbcTemplate (d)TopLink,JDO,Hibernate

(1) (c) (2) (b) (3) (d) (4) (a)

本问题考查考生对数据持久层基本原理和技术方案的掌握情况。SpringJdbcTemplate通过封装JDBC操作接口实现数据库访问操作，fflatis/MyBatis是通过SQL映射将数据操作请求转换为数据库的SQL操作，Hibernate、JDO和TopLink等技术都采用了对象关系映射的思想，j2EE中的BMP和CMP及EJB3.0都是利用实体Bean对象完成数据访问操作。

* 阅读以下关于应用服务器的叙述，在答题纸上回答问题1至问题3。【说明】某电子产品制造公司，几年前开发建设了企业网站系统，实现了企业宣传、产品介绍、客服以及售后服务等基本功能。该网站技术上采用了Web服务器、动态脚本语言PHP。随着市场销售渠道变化以及企业业务的急剧拓展，该公司急需建立完善的电子商务平台。公司张工建议对原有网站系统进行扩展，增加新的功能（包括订单系统、支付系统、库存管理等），这样有利于降低成本、快速上线；而王工则认为原有网站系统在技术上存在先天不足，不能满足企业业务的快速发展，尤其是企业业务将服务全球，需要提供24小时不间断服务，系统在大负荷和长时间运行下的稳定性至关重要。建议采用应用服务器的Web开发方法，例如J2EE，为该企业重新开发新的电子商务平台。

2016年【问题1】（7分）

王工认为原有网站在技术上存在先天不足，不能满足企业业务的快速发展，根据你的理解，请用300字以内的文字说明原系统存在哪几个方面的不足。

1、PHP只能实现简单的分布式两层或三层的架构，而JAVA在这方面就比较强大，可以实现多层的网络架构。数据库层(持久化层)、应用(业务)逻辑层、表示逻辑层彼此分开，而且现在不同的层都已经有一些成熟的开发框架的支持。

2、PHP是面向过程的语言，Java是面向对象的，面向过程语言开发的程序只要业务流程发生变化，修改工作量很大，所以可修改性差，同时可复用性也差。

3、PHP语言在可靠性方面比J2EE平台差，J2EE平台有大量增强可靠性的成熟解决方案，而PHP只是一种简单的脚本语言，在可靠性方面缺乏成熟解决方案。

4、PHP对于不同的数据库采用不同的数据库访问接口，而Java通过JDBC来访问数据库，通过不同的数据库厂商提供的数据库驱动方便地访问数据库，访问数据库的接口比较统一。所以原架构在数据库连接方面修改起来工作量也是很大的。

5、PHP适合于小型项目，所以本项目中以前采用PHP是合适的，但目前大量功能需要增加，PHP在稳定性方面也达不到要求。

6、PHP比Java的可维护性差。

7、PHP比Java的扩展性差。

8、PHP比Java的安全性差。

2016年【问题2】(8分)

请简要说明应用服务器的概念，并重点说明应用服务器如何来保障系统在大负荷和长时间运行下的稳定性以及可扩展性。

【参考答案】

应用服务器是指通过各种协议把商业逻辑曝露给客户端的程序。

1、若系统负荷很大，可以布署多台应用服务，多台应用服务器分担任务，以达到性能要求。

2、应用服务器可以通过灵活的增加服务器完成扩展，所以可扩展性很好。

3、应用服务器可长时间稳定运行。因为当一台应用服务器出现故障时，可以将当前运行的事务转移至正常应用服务器上完成执行，不影响业务正常执行，从而保障高可靠性与稳定性。

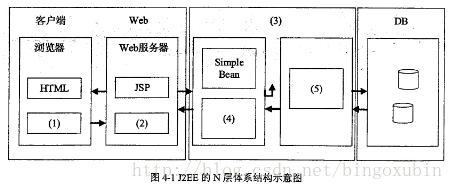
【试题解析】

应用服务器是指通过各种协议把商业逻辑曝露给客户端的程序。它提供了访问商业逻辑的途径以供客户端应用程序使用。应用服务器使用此商业逻辑就像调用对象的一个方法一样。

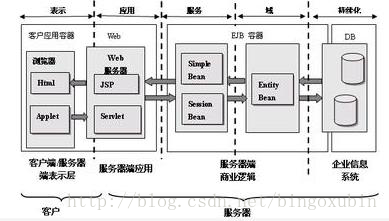
简单的说能实现动态网页技术的服务器叫做web应用服务器。

2016年【问题3】（10分）

J2EE平台采用了多层分布式应用程序模型，实现不同逻辑功能的应用程序被封装到不同的构件中，处于不同层次的构件可被分别部署到不同的机器中。请填写图4-1中(1)～(5)处的空白，完成J2EE的N层体系结构。



(1)Applet ； (2)Servlet ；(3)EJB容器 ；(4)SessionBean ； (5)EntityBean



* 分布式系统开发中，通常需要将任务分配到不同的逻辑计算层。业务数据的综合计算分析任务属于(39).

2010年(39)

A.表示逻辑层

B.应用逻辑层

C.数据处理层

D.数据层

【答案】B 【解析】

分布式系统开发分为5个逻辑计算层：表示层实现用户界面；表示逻辑层包括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。

* 在客户机/服务器系统开发中，采用(40)时，应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。

2010年(40)

A.分布式表示结构

B.分布式应用结构

C.分布式数据和应用结构

D.分布式数据结构

【答案】D 【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：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；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；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。

## B/S 架构风格

知识点

1. 三层B/S架构

浏览器

WEB服务器

数据库服务器

优点

零客户端

缺点

对动态页面的支持性差

以页面为单位提交

安全性差，扩展能力不强

试题

## MVC 架构风格

知识点

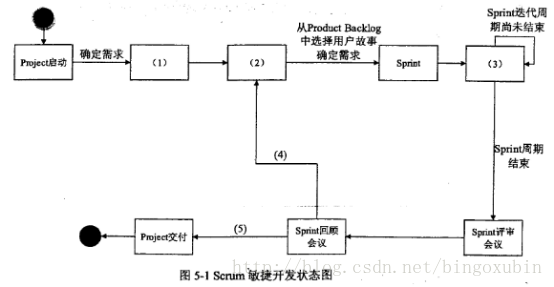
1. 典型的基于MVC (Model View Controller)的J2EE应用
2. 系统的界面由JSP构件实现，
3. 分发客户请求、有效组织其他构件为客户端提供服务的控制器由Servlet构件实现，
4. 数据库相关操作由Entity Bean构件实现，
5. 系统核心业务逻辑由Session Bean 构件实现

试题

* 阅读以下关于Scrum敏捷开发过程的叙述，在答题纸上回答问题1至问题3。【说明】Scrum是一个增量的、迭代的敏捷软件开发过程。某软件公司计划开发一个基于Web的Scrum项目管理系统，用于支持项目团队采用Scrum敏捷开发方法进行软件开发，辅助主管智能决策。此项目管理系统提供的主要服务包括项目团队的管理、敏捷开发过程管理和工件的管理。Scrum敏捷开发中，项目团队由Scrum主管、产品负责人和开发团队人员三种不同的角色组成，其开发过程由若干个Sprint（短的迭代周期，通常为2到4周）活动组成。Product Backlog是在Scrum过程初期产生的一个按照商业价值排序的需求列表，该列表条目的体现形式通常为用户故事。在每一个Sprint活动中，项目团队从Product Backlog中挑选最高优先级的用户故事进行开发。被挑选的用户故事在Sprint计划会议上经过细化分解为任务，同时初步估算每一个任务的预计完成时间，编写Sprint Backlog。在Sprint活动期间，项目团队每天早晨需举行每日站立会议，重新估算剩余任务的预计完成时间，更新Sprint Backlog、Sprint燃尽图和Release燃尽图。在每个Sprint活动结束时，项目团队召开评审会议和回顾会议，交付产品增量，总结Sprint期间的工作情况和问题。此时，如果Product Backlog中还有未完成的用户故事，则项目团队将开始筹备下一个Sprint活动迭代。为完成Scrum项目管理系统，考虑到系统的智能决策需求，公司决定使用MVC架构模式开发该项目管理系统。具体来说，系统采用轻量级J2EE架构和SSH框架进行开发，使用MySQL数据库作为底层存储。

2016年【问题1】（10分）

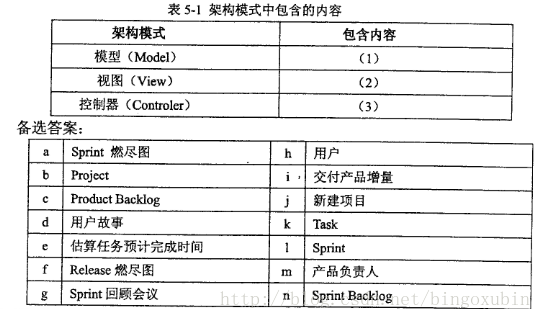
Scrum项目管理软件需真实模拟Scrum敏捷开发流程，请根据你的理解完成图5-1给出的Scrum敏捷开发状态图，填写其中(1)～(5)的内容。



(1)Product Backlog (2)Sprint计划会议 (3)每日站立会议 (4)还有未完成的用户故事 (5)交付产品增量

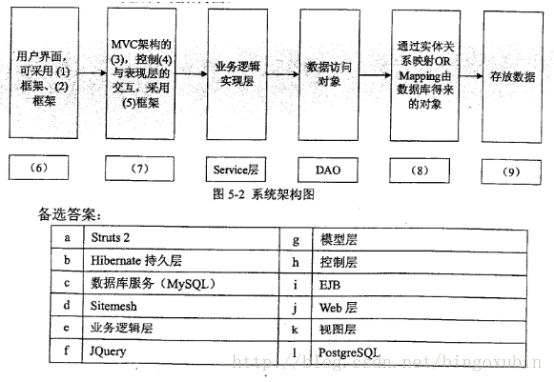
2016年【问题2】（6分）

根据题干描述，本系统采用MVC架构模式，请从各选答案a~n中分别选出属于MVC架构模型中的模型(Model)、视图(View)和控制器（Controler）的相关内容描述填入表5-1的空(1)～(3)处。



(1)c、e、n (2)a、f、j (3)g

2016年【问题3】（9分） 根据项目组给出的系统设计方案，将备选答案a~l的内容填写在图5-2中的空(1)～(9)，完成系统架构图。



(1) (2)d f (3)h (4)g (5)a (6)k (7)h (8)b (9)c

* 在一个典型的基于MVC (Model View Controller)的J2EE应用中，分发客户请求、有效组织其他构件为客户端提供服务的控制器由(39)实现。

2009年(39)

A.Entity Bean

B.Session Bean

C.Servlet

D.JSP

【答案】C 【解析】本题考查J2EE应用架构的基本知识。 在一个典型的基于MVC (Model View Controller)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制器由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。

* 试题一【说明】某软件公司欲开发一个网络设备管理系统，对管理区域内的网络设备（如路由器和交换机等）进行远程监视和控制。公司的系统分析师首先对系统进行了需求分析，识别出如下3项核心需求：（a）目前需要管理的网络设备确定为10类20种，未来还将有新类别的网络设备纳入到该设备管理系统中；（b）不同类别的网络设备，监视和控制的内容差异较大；同一类网络设备，监视和控制的内容相似，但不同厂商的实现方式（包括控制接口格式、编程语言等）差异较大；（c）网络管理员能够在一个统一的终端之上实现对这些网络设备的可视化呈现和管理操作。针对上述需求，公司研发部门的架构师对网络设备管理系统的架构进行了分析与设计，架构师王工认为该系统可以采用MVC架构风格实现，即对每种网络设备设计一个监控组件，组件通过调用网络设备厂商内置的编程接口对监控指令进行接收和处理；系统管理员通过管理模块向监控组件发送监控指令，对网络设备进行远程管理；网络状态、监控结果等信息会在控制终端上进行展示。针对不同网络设备的差异，王工认为可以对当前的20种网络设备接口进行调研与梳理，然后通过定义统一操作接口屏蔽设备差异。李工同意王工提出的MVC架构风格和定义统一操作接口的思路，但考虑到未来还会有新类别的网络设备接入，认为还需要采用扩展接口的方式支持系统开发人员扩展或修改现有操作接口。公司组织专家进行架构评审，最终同意了王工的方案和李工的改进意见。

【2014问题1】 请用300字以内的文字解释什么是MVC架构风格以及其中的组件交互关系，并根据题干描述，指出该系统中的M、V、C分别对应什么。

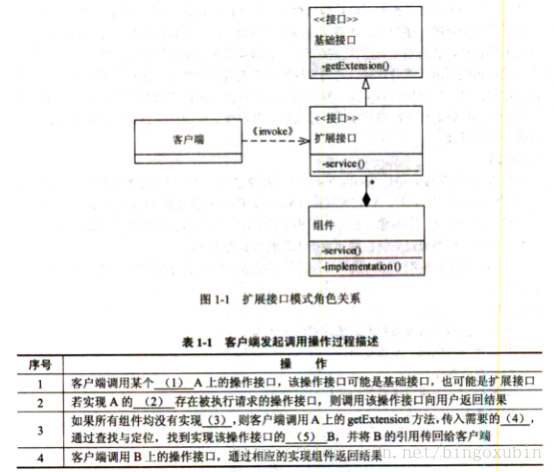
MVC架构风格最初是Smalltalk-80中用来构建用户界面时采用的架构设计风格。其中M代表模型（Model)，V代表视图（View)，C代表控制器（Controller)。在该风格中，模型表示待展示的对象，视图表示模型的展示，控制器负责把用户的动作转成针对模型的操作。模型通过更新视图的数据来反映自身的变化。

在本系统中，模型（M)代表监控组件、视图（V)代表控制终端、控制器（C)代表管理模块。

【2014问题2】

扩展接口模式结构通常包含四个角色：基础接口、组件、扩展接口和客户端，它们之间的关系如图 1-1所示。其中每个扩展接口需要通过扩展基础接口获得基本操作能力，然后加入自己特有的操作接口，并通过设置全局唯一接口ID对自身接口进行标识；每个具体的组件需要实现扩展接口完成实际操作；客户端不与组件直接交互，而需要通过与扩展接口交互提出调用请求，扩展接口根据请求查找并选择合适的实现组件响应客户端请求。请根据上图所示和题干描述，指出扩展接口模式结构中的四个角色分别对应网络设备管理系统的哪些部分；并以客户端发起调用操作这一场景为例，填写表

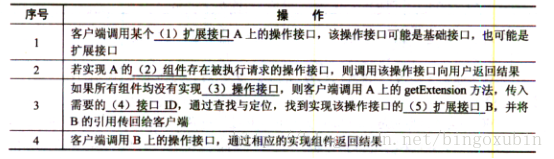
1-1 中的（1）～（5）。



备选答案：基础接口、扩展接口、操作接口、接口ID、客户端、组件。

各个角色与网络设备管理系统的对应关系为： 基础接口对应统一操作接口； 组件对应监控组件： 扩展接口对应新网络设备的操作接口；

客户端对应控制终端。 客户端发起调用操作场景下的描述如下：



扩展接口模式结构通常包含四个角色：基础接口、组件、扩展接口和客户端。其中每个扩展接口需要通过扩展基础接口获得基本操作能力，然后加入自己特有的操作接口，并通过设置全局唯一接口ID对自身接口进行标识。每个具体的组件需要实现扩展接口完成实际操作，客户端不与组件直接交互，而需要通过与扩展接口交互提出调用请求，扩展接U根据请求查找并选择合适的实现组件响应客户端请求。根据题干描述，可以看出基础接口这-角色应该对应统一操作接口，组件这一角色应该对应监控组件，扩展接口这一角色应该对应新网络设备的操作接口，客户端这一角色应该对应控制终端。

## MVP 架构风格

# 面向服务的架构

## SOA 概述

知识点

|  |  |  |  |
| --- | --- | --- | --- |
| 是一种在计算环境当中，设计、开发、部署、管理服务的一种方法 | | | |
| 结构 | 服务 | 服务接口 | 共同的封装 共同的语言格式  共同的安全和容错处理 |
| 逻辑层 |  |
| 数据访问层 |  |
| 服务总线 |  |  |
| 客户端 | 服务代理 |  |
| 优点 | 松散耦合 不依赖具体的平台，可使用多种新技术 | | |
| 服务设计原则 | 明确定义的标准服务接口 | | |
| 自包含模块化 | | |
| 粗粒度，服务间交互少但量大 | | |
| 松散耦合 | | |
| 互操作性、兼容 | | |
| 关键技术 | 发现服务层 | UDDI(统一发现描述集成) | UDDI用于Web服务注册和服务查找  UDDI数据模型:用于描述企业和服务的XML Schema UDDI API:一组用于查找或发布数据的方法，基于SOAP UDDI 注册服务:可供服务注册中心 |
| DISCO | 可定义一个文档格式和询问算法，发现给定服务器上公开的服务 通过WSDL与服务交互 |
| 描述服务层 | WSDL：Web服务描述语言 | WSDL用于描述Web服务的接口和操作功能 |
| XML Schema |  |
| 消息格式 | SOAP：简单对象访问协议 | SOAP封装  定义整体框架，用于表示消息的内容、谁处理内容及哪些内容可选 SOAP编码规则  定义了一种序列化的机制，用于交换系统所定义的数据类型的实例 SOAP RPC表示  定义一个用来表示远程过程调用和应答的协议 SOAP 绑定  定义一个使用底层传输协议来完成在节点间交换SOAP信封的约定 |
| REST：表述性状态传递（英文：Representational State Transfer） | 使用HTTP+XML基于WEB通信的技术 简单，缺少严格配置文件 只支持几个操作（POST、GET、PUT、Delete) 强调信息本身，称为资源  网络上的事物称为资源  每个资源一个ID  通过连接器接口进行操作  对资源的操作不改变资源标识  所有的操作都是无状态的 |
| 编辑模式层 | XML |  |
| 传输协议层 | HTTP、TCP/IP |  |
|  | BPEL | 面向Web 服务的业务流程执行语言 | 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 |

试题

* 面向服务系统构建过程中，（39）用于实现Web服务的远程调用，（40）用来将分散的、功能单一的Web服务组织成一个复杂的有机应用。

2016年(39)

A.UDDI（Universal Description，Discovery and Integration）

B.WSDL（Web Service Description Language)

C.SOAP（Simple Object Access Protocol）

D.BPEL（Business Process Execution Language）

2016年(40)

A.UDDI（Universal Description，Discovery and Integration）

B.WSDL（Web Service Description Language）

C.SOAP（Simple Object Access Protocol）

D.BPEL（Business Process Execution Language）

【答案】C D

【解析】 UDDI（UniversalDescription，Discovery＆Integration），UDDI用于Web服务注册和服务查找；WSDL（Web Service Description Language），WSDL用于描述Web服务的接口和操作功能；SOAP（Simple Object Access Protocol），SOAP为建立Web服务和服务请求之间的通信提供支持。 BPEL（Business Process Execution Language For WebServices）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。

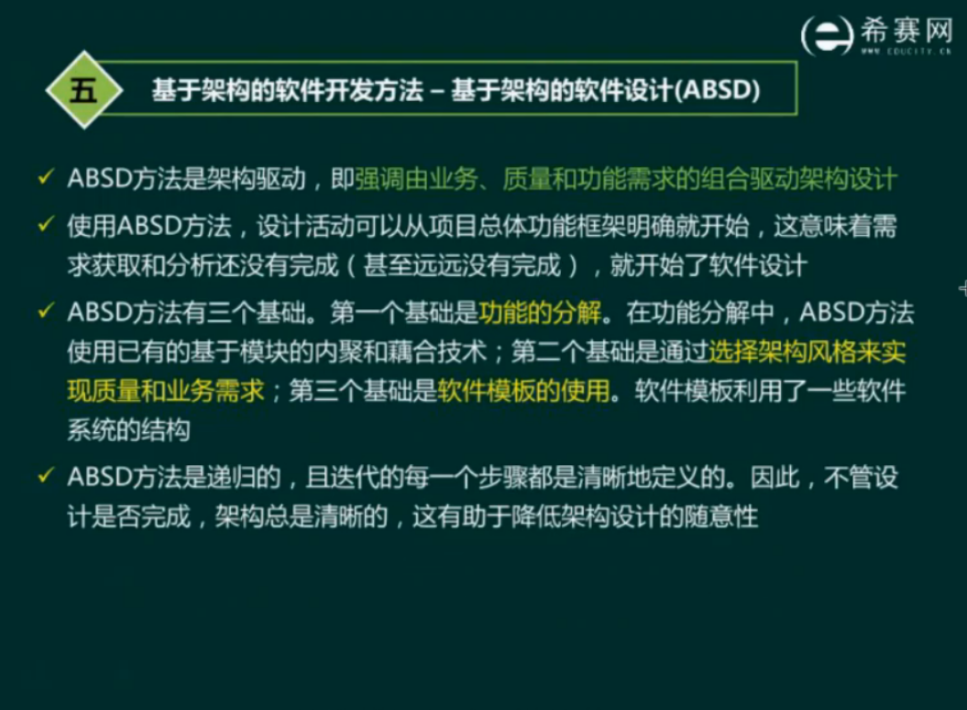
## 微服务

# 基于架构的软件开发(Architecture-Based Software Design,ABSD)

## 概念

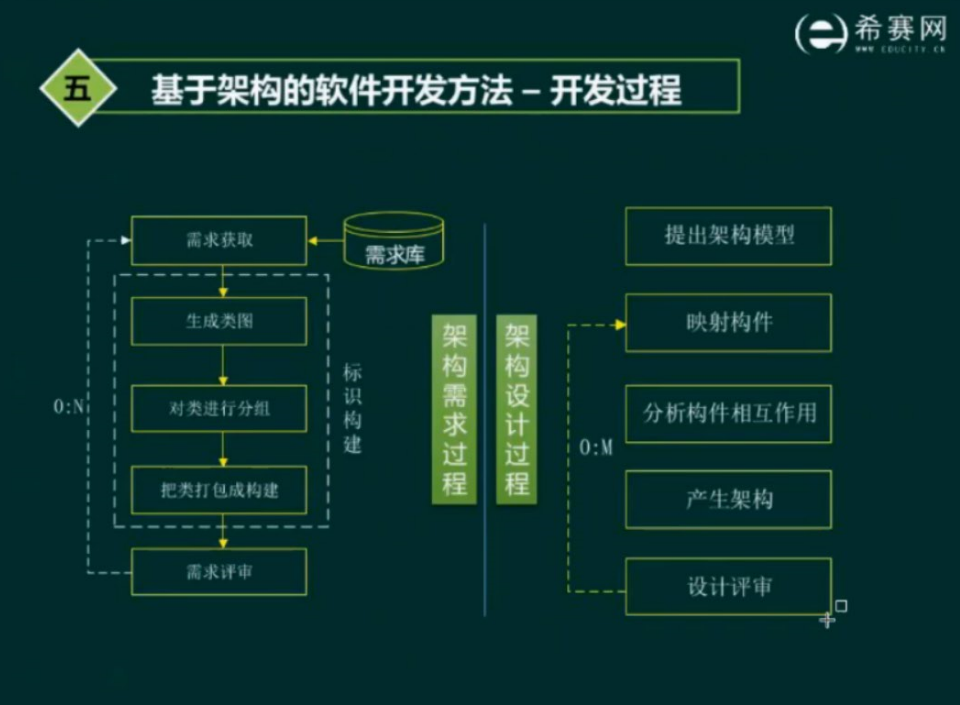
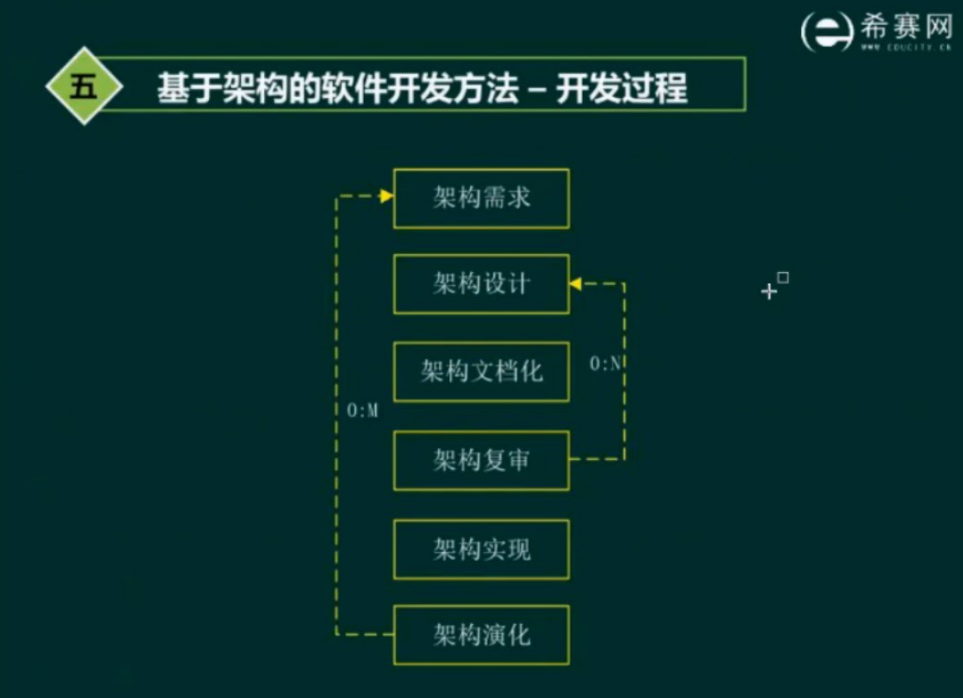
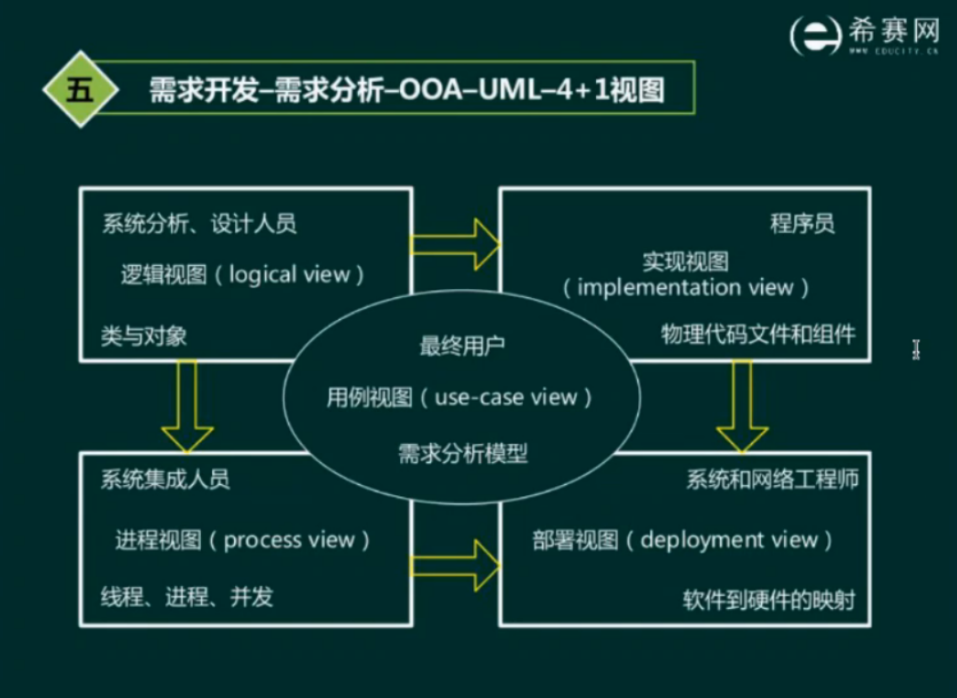
知识点

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。

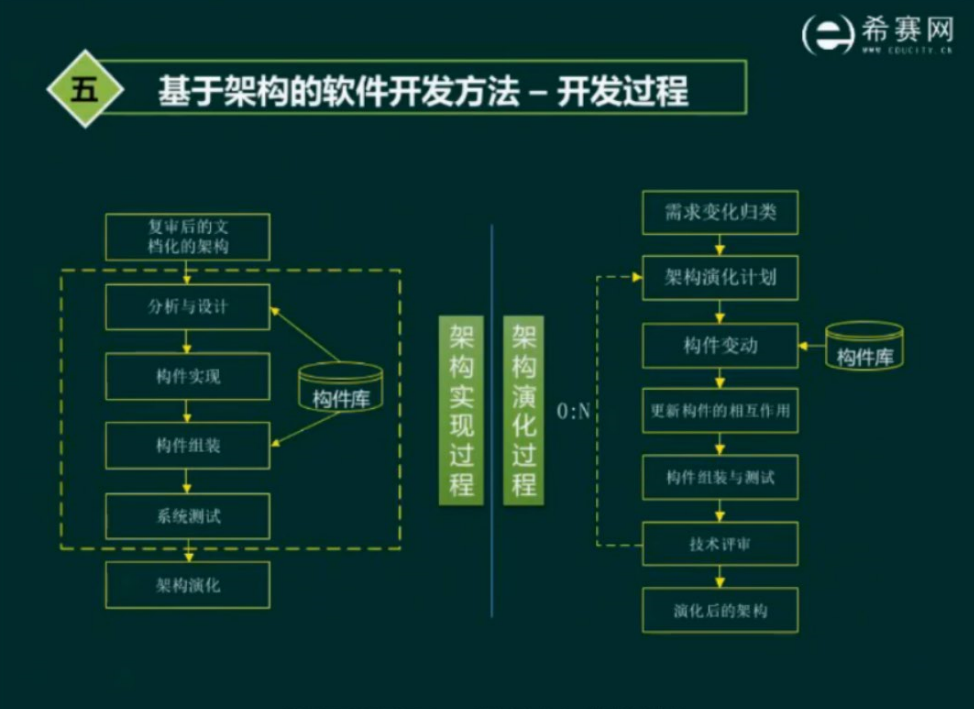


ABSD强调用视角与视图描述软件架构。

用用例与质量场景描述需求。



软件架构文档应该从使用者的角度进行书写，针对不同背景的人员采用不同的书写方式，并将文档分发给相关人员。架构文档要保持较新，但不要随时保证文档最新，要保持文档的稳定性。架构文档化的主要输出结果是架构规格说明书和架构质量说明书



试题

* 采用以架构为核心的软件开发方法，在建立软件架构的初期，首要任务是选择一个合适的（42)，在此基础上，开发人员通过架构模型，可以获得关于（43)的理解,为将来的架构实现与演化过程建立了目标。

2012年(42)

A.分析模式

B.设计模式

C.架构风格

D.架构标准

2012年(43)

A.架构需求

B.架构属性

C.架构优先级

D.架构约束

【答案】C B 【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。

* 基于架构的软件设计（ABSD)强调由商业、质量和功能需求的组合驱动软件架构设计。ABSD方法有三个基础：功能分解、(49)和软件模板的使用。

2011年(49)

A.对需求进行优先级排列

B.根据需求自行设计系统的总体架构

C.选择架构风格实现质量及商业需求

D.开发系统原型用于测试

【答案】C 【解析】本题主要考查考生对基于架构的软件设计（ABSD)的理解与掌握

ABSD以架构风格和质童属性为中心，强调由商业、质量和功能需求的组合驱动软件架构设计。ABSD方法有三个基础：功能分解、选择架构风格实现质量及商业需求和软件模板的使用。

* 基于架构的软件设计（ABSD)强调由商业、质量和功能需求的组合驱动软件架构设计。以下关于ABSD的叙述中，错误的是（48）。

2009年(48)

A.使用ABSD方法，设计活动可以从项目总体功能框架明确就开始

B.ABSD方法是一个自顶向下，递归细化的过程

C.ABSD方法有三个基础：功能分解、选择架构风格实现质量和商业需求以及软件模板的使用

D.使用ABSD方法，设计活动的开始意味着需求抽取和分析活动可以终止

【答案】D 【解析】

基于架构的软件设计（ABSD)强调由商业、质量和功能需求的组合驱动软件架构设计。使用ABSD方法，设计活动可以从项目总体功能框架明确就开始，并且设计活动的开始并不意味着需求抽取和分析活动可以终止，而是应该与设计活动并行。ABSD方法有三个基础：第一个基础是功能分解，在功能分解中使用已有的基于模块的内聚和耦合技术。第二个基础是通过选择体系结构风格来实现质量和商业需求。第三个基础是软件模板的使用。ABSD方法是一个自顶向下，递归细化的过程，软件系统的架构通过该方法得到细化，直到能产生软件构件的类。

* 基于架构的软件开发(Architecture Based Software Development，ABSD)强调由商业、质量和功能需求的组合驱动软件架构设计。它强调采用（44）描述软件架构，用（45）来描述需求。

2015年(44)

A.类图和序列图

B.视角与视图

C.构建和类图

D.构建与功能

2015年(45)

A.用例与类图

B.用例与视角

C.用例与质量场景

D.视角与质量场景

【答案】B C 【解析】本题考查基于架构的软件开发方法的基础知识。 根据定义，基于软件架构的开发(Architecture Based Software Development，ABSD)强调由商业、质量和功能需求的组合驱动软件架构设计。它强调采用视角和视图来描述软件架构，采用用例和质量属性场景来描述需求。

* 某公司采用基于架构的软件设计（Architecture-Based Software Design, ABSD)方法进行软件设计与开发。ABSD方法有三个基础，分别是对系统进行功能分解、采用(52)实现质量属性与商业需求、采用软件模板设计软件结构。ABSD方法主要包括架构需求等6个主要活动，其中（53)活动的目标是标识潜在的风险，及早发现架构设计中的缺陷和错误；(54)活动针对用户的需求变化，修改应用架构，满足新的需求。小王是该公司的一位新任架构师，在某项目中主要负责架构文档化方面的工作。小王（55)的做法不符合架构文档化的原则。架构文档化的主要输出结果是架构规格说明书和（56)。

2013年(52)

A.架构风格

B.设计模式

C.架构策略

D.架构描述

2013年(53)

A.架构设计

B.架构实现

C.架构复审

D.架构演化

2013年(54)

A.架构设计

B.架构实现

C.架构复审

D.架构演化

2013年(55)

A.从使用者的角度书写文档

B.随时保证文档都是最新的

C.将文档分发给相关人员

D.针对不同背景的人员书写文档的方式不同

2013年(56)

A.架构需求说明书

B.架构实现说明书

C.架构质量说明书

D.架构评审说明书

【答案】A C D B C 【解析】本题主要考查采用基于架构的软件设计的基础知识与应用。

基于架构的软件设计（Architecture-Based Software Design,ABSD)方法有三个基础，分别是对系统进行功能分解、采用架构风格实现质量属性与商业需求、采用软件模板设计软件结构。ABSD方法主要包括架构需求等6个主要活动，其中架构复审活动的目标是标识潜在的风险，及早发现架构设计中的缺陷和错误；架构演化活动针对用户的需求变化，修改应用架构，满足新的需求。

软件架构文档应该从使用者的角度进行书写，针对不同背景的人员采用不同的书写方式，并将文档分发给相关人员。架构文档要保持较新，但不要随时保证文档最新，要保持文档的稳定性。架构文档化的主要输出结果是架构规格说明书和架构质量说明书。

## 架构需求

知识点

1. 需求获取

架构需求获取来自三个方面，即系统的质量目标，系统的商业目标，系统开发人员的商业目标。

1. 标识构件

* 生成类图
* 对类进行分组

与其他隔离的类形成一个组，由概括关联的类组成一个附加组，由聚合或合成关联的类组成一个附加组

* 把类打包成构件

把类簇打包成构件，这些构件可以分组合并成更大的构件

1. 架构需求评审

由分析人员，客户，设计人员，测试人员组成小组，检查需求是否真实，类的分组是否合理，构件的合并是否合理

试题

* 软件架构需求是指用户对目标软件系统在功能、行为、性能、设计约束等方面的期望。以下活动中，不属于软件架构需求过程范畴的是（47）。

2009年(47)

A.设计构件

B.需求获取

C.标识构件

D.架构需求评审

【答案】A 【解析】

软件架构需求是指用户对目标软件系统在功能、行为、性能和设计约束等方面的期望。需求过程主要是获取用户需求，标识系统中所要用到的构件，并进行架构需求评审。其中标识构件又详细分为生成类图、对类图进行分组和将类打包成构件三步。软件架构需求并不应该包括设计构件的过程。

## 架构设计

知识点

1. 提出软件架构模型

选择一个合适的架构风格，该模型为将来的实现和演化建立了目标

1. 映射构件

把已标识的构件映射到架构中，将产生一个中间结构，它只包含适合架构模型的构件

1. 分析构件的相互作用
2. 产生软件架构

当决定了关键构件之间的相互作用后，就可以在中间结构的基础上进行精化，得到软件架构

1. 设计评审

邀请独立于系统开发的外部人员对架构进行评审

试题

* 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。以下关于软件架构设计活动的描述，错误的是(45)。

2010年(45)

A.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

B.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

C.软件架构设计活动将已标识构件集成到软件架构中，设计并实现这些构件

D.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

【答案】C 【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。

* 将系统需求模型转换为架构模型是软件系统需求分析阶段的一项重要工作，以下描述中，（41）是在转换过程中需要关注的问题。

2014年(41)

A.如何通过多视图模型描述软件系统的架构

B.如何确定架构模型中有哪些元素构成

C.如何采用表格或用例映射保证转换的可追踪性

D.如何通过模型转换技术，将高层架构模型逐步细化为细粒度架构模型

【答案】C 【解析】本题主要考查软件架构设计与生命周期的关系。

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。本题答案中A、B是软件架构设计阶段需要考虑的问题，D是软件架构实现阶段中需要考虑的问题。

* 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（51），并通过多种（52）全面描述特定系统的架构。

2015年(51)

A.实现方式

B.交互作用

C.设计方案

D.测试方式

2015年(52)

A.对象

B.代码

C.文档

D.视图

【答案】B D 【解析】本题主要考查软件架构设计过程的基础知识。

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。

* 以下关于软件架构设计重要性的描述，（40）是错误的。

2014年(40)

A.软件架构设计能够满足系统的性能、安全性、可维护性等品质

B.软件架构设计能够帮助项目干系入(Stakeholder)更好地理解软件结构

C.软件架构设计能够帮助架构师更好地捕获和细化系统需求

D.软件架构设计能够有效地管理系统的复杂性，并降低系统维护费用

【答案】C 【解析】本题主要考査软件架构设计的重要性。

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。软件架构设计能够满足系统的性能、安全性、可维护性等品质；软件架构设计能够帮助项目干系人（Stakeholder)更好地理解软件结构：软件架构设计能够有效地管理系统的复杂性，并降低系统维护费用；软件架构设计对系统开发具有指导性：软件架构设计为系统复用奠定的基础；软件架构设计能够支持冲突分析。需要注意的是，软件架构设计与系统需求是直交的，两者并无必然联系。

## 架构文档化

知识点

1. 主要输出结果是架构规格说明书和质量设计说明书
2. 软件架构文档遵循的原则
   1. 文档要从使用者的角度进行编写；
   2. 必须分发给所有与系统有关的开发人员；
   3. 应该保持架构文档的即时更新，
   4. 但更新不要过于频繁；
   5. 架构文档中描述应该尽量避免不必要的重复；
   6. 每次架构文档修改都应该记录进行修改的原则。

试题

* 软件架构文档是对软件架构的正式描述，能够帮助与系统有关的开发人员更好地理解软件架构。软件架构文档的写作应该遵循一定的原则。以下关于软件架构文档写作原则的叙述中，错误的是（49）。

2009年(49)

A.架构文档应该从架构设计者的角度进行编写

B.应该保持架构文档的即时更新，但更新不要过于频繁

C.架构文档中的描述应该尽量避免不必要的重复

D.每次架构文档修改，都应该记录修改的原则

【答案】A 【解析】

软件架构文档是对软件架构的一种描述，帮助程序员使用特定的程序设计语言实现软件架构。软件架构文档的写作应该遵循一定的原则，这些原则包括：文档要从使用者的角度进行编写；必须分发给所有与系统有关的开发人员；应该保持架构文档的即时更新，但更新不要过于频繁；架构文档中描述应该尽量避免不必要的重复；每次架构文档修改都应该记录进行修改的原则。

## 架构复审

知识点

1. 安排由外部人员包括用户代表和领域专家参加的复审
2. 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。
3. 架构复审过程中，通常会对一个可运行的最小化系统进行架构评估和测试。
4. 架构复审的目标是标识潜在的风险，及早发现架构设计的缺陷和错误

试题

* 在对一个软件系统的架构进行设计与确认之后，需要进行架构复审。架构复审的目的是为了标识潜在的风险，及早发现架构设计中的缺陷和错误。在架构复审过程电，主要由（53）决定架构是否满足需求、质量需求是否在设计中得到体现。

2014年(53)

A.系统分析师与架构师

B.用户代表与领域专家

C.系统拥有者与项目经理

D.系统开发与测试人员

【答案】B 【解析】

在对一个软件系统的架构进行设计与确认之后，需要进行架构复审。架构复审的目的是为了标识潜在的风险，及早发现架构设计中的缺陷和错误。在架构复审过程中，主要由用户代表与领域专家决定架构是否满足需求、质量需求是否在设计中得到体现。

* 架构复审是基于架构开发中一个重要的环节。以下关于架构复审的叙述中，错误的是（50)。

2009年(50)

A.架构复审的目标是标识潜在的风险，及早发现架构设计的缺陷和错误

B.架构复审过程中，通常会对一个可运行的最小化系统进行架构评估和测试

C.架构复审人员由系统设计与开发人员组成

D.架构设计、文档化和复审是一个迭代的过程

【答案】C 【解析】

架构复审是基于架构开发中一个重要的环节。架构设计、文档化和复审是一个迭代的过程。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。架构复审过程中，通常会对一个可运行的最小化系统进行架构评估和测试。架构复审的目标是标识潜在的风险，及早发现架构设计的缺陷和错误。

## 架构实现

知识点

1. 分析与设计

在架构说明书中，已经定义了系统中构件与构件之间的关系，构件接口约束对外唯一代表了构件，所以可以从构件库中直接查找符合接口约束的构件

1. 构件实现

必要时开发新的满足要求的构件

1. 构件组装
2. 系统测试

包括单个构件的功能性测试和被组装应用的整体功能和性能测试

试题

## 架构演化

知识点

1. 需求变化归类

对需求变化进行归类，使变化的需求与已有构件对应。对找不到对应构件的需求变动，在后续工作中将创建新的构件来对应

1. 制定架构演化计划
2. 构件变动

修改，增加或删除构件，要对修改和增加的构件进行功能性测试

1. 更新构件的相互作用
2. 构件组装与测试
3. 技术评审

试题

# 软件架构评估

## 软件架构评估的方法

知识点

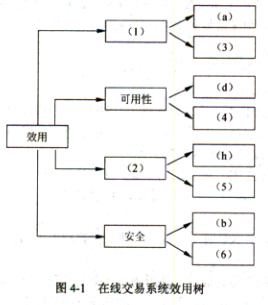
1. 识别风险、非风险、敏感点和权衡点是进行软件架构评估的重要过程
2. 风险是某个存在问题的架构设计决策，可能会导致问题;
3. 非风险与风险相对，是良好的架构设计决策；
4. 敏感点是一个或多个构件的特性；
5. 权衡点是影响多个质量属性的特性，是多个质量属性的敏感点。

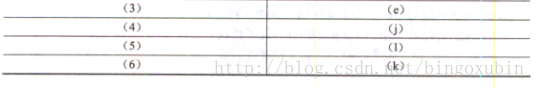
试题

* 试题四【说明】某电子商务公司拟升级目前正在使用的在线交易系统，以提高客户网上购物时在线支付环节的效率和安全性。公司研发部门在需求分析的基础上，给出了在线交易系统的架构设计。公司组织相关人员召开了针对架构设计的评估会议，会上用户提出的需求、架构师识别的关键质量属性场景和评估专家的意见等内容部分列举如下：（a）在正常负载情况下，系统必须在0.5秒内响应用户的交易请求；（b）用户的信用卡支付必须保证99.999%的安全性；（c）系统升级后用户名要求至少包含8个字符； （d）网络失效后，系统需要在2分钟内发现错误并启用备用系统；（e）在高峰负载情况下，用户发起支付请求后系统必须在10秒内完成支付功能；（f）系统拟采用新的加密算法，这会提高系统安全性，但同时会降低系统的性能；（g）对交易请求处理时间的要求将影响系统数据传输协议和交易处理过程的设计；（h）需要在30人月内为系统添加公司新购买的事务处理中间件；（i）现有架构设计中的支付部分与第三方支付平台紧耦合，当系统需要支持新的支付平台时，这种设计会导致支付部分代码的修改，影响系统的可修改性；（j）主站点断电后，需要在3秒内将访问请求重定向到备用站点；（k）用户信息数据库授权必须保证99.999%可用；（l）系统需要对Web界面风格进行修改，修改工作必须在4人月内完成；（m）系统需要为后端工程师提供远程调试接口，并支持远程调试。

【2014问题1】 在架构评估过程中，质量属性效用树(utility tree)是对系统质量属性进行识别和优先级排序的重要工具。请给出合适的质量属性，填入图 4-1

中（1）、（2）空白处；并选择题干描述的（a）～（m），填入（3）～（6）空白处，完成该系统的效用树。





本题主要考查考生对软件架构评估、软件质量属性以及架构评估中相关概念的理解与掌握。考生应该在熟记基础概念的基础上结合实际问题灵活掌握并应用这些概念。

在解答本题时，首先需要对题干中的所有软件需求描述进行分析与梳理，区分并找出其中的需求分析、软件质量属性描述，或者可能的风险、权衡点或敏感点描述。具体列举如下：

(a)在正常负载情况下，系统必须在0.5秒内响应用户的交易请求，对应性能属性。

(b)用户的信用卡支付必须保证99.999%的安全性，对应安全性属性。 (c)系统升级后用户名要求至少包含8个字符，对应软件业务需求。

(d)网络失效后，系统需要在2分钟内发现错误并启用备用系统，对应可用性属性。

(e)在高峰负载情况下，用户发起支付请求后系统必须在10秒内完成支付功能，对应性能属性。

(f)系统拟采用新的加密算法，这会提高系统安全性，但同时会降低系统的性能，系统权衡点。

(g)对交易请求处理时间的要求将影响系统数据传输协议和交易处理过程的设计，对应系统敏感点。

(h)需要在30人月内为系统添加公司新购买的事务处理中间件，对应可修改性属性。

(i)现有架构设计中的支付部分与第三方支付平台紧耦合，当系统需要支持新的支付平台时，这种设计会导致支付部分代码的修改，影响系统的可修改性，对应系统风险。

(j)主站点断电后，需要在3秒内将访问请求重定向到备用站点，对应可用性属性。

(k)用户信息数据库授权必须保证99.999%可用，对应安全性属性。

(l)系统需要对Web界面风格进行修改，修改工作必须在4人月内完成，对应可修改性属性。

(m)系统需要为后端工程师提供远程调试接口，并支持远程调试，对应可测试性属性。 在架构评估过程中，质量属性效用树（utility

tree)是对系统质量属性进行识别和优先级排序的重要工具。效用树主要关注性能、可修改性、可用性和安全4个方面，考生首先必须填入对应的质量属性名称，并根据上述分析依次填入合适的标号。

【2014问题2】

在架构评估过程中，需要正确识别系统的架构风险、敏感点和权衡点，并进行合理的架构决策。请用300字以内的文字给出系统架构风险、敏感点和权衡点的定义，并从题干（a)〜（m)中各选出1个对系统架构风险、敏感点和权衡点最为恰当的描述。

系统架构风险是指架构设计中潜在的、存在问题的架构决策所带来的隐患。 敏感点是为了实现某种特定质量属性，一个或多个系统组件所具有的特性。

权衡点是影响多个质量属性，并对多个质量属性来说都是敏感点的系统属性。

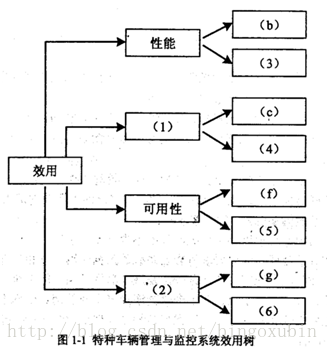
题干描述中，（i)描述的是系统架构风险；（g)描述的是敏感点；（f)描述的是权衡点。

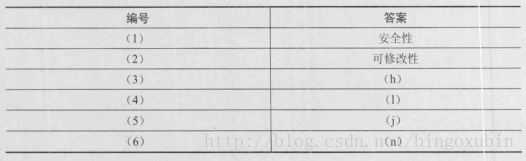
系统架构风险是指架构设计中潜在的、存在问题的架构决策所带来的隐患。敏感点是为了实现某种特定质量属性，一个或多个系统组件所具有的特性。权衡点是影响多个质量属性，并对多个质景属性来说都是敏感点的系统属性。根据上述分析可知题干描述中，（i)描述的是系统架构风险；（g)描述的是敏感点：（f)描述的是权衡点。

* 试题一【说明】某软件公司拟为某市级公安机关开发一套特种车辆管理与监控系统，以提高特种车辆管理的效率和准确性。在系统需求分析与架构设计阶段，用户提出的部分需求和关键质量属性场景如下：(a)系统用户分为管理员、分管领导和普通民警等三类；(b)正常负载情况下，系统必须在0.5秒内对用户的车辆查询请求进行响应；(c)系统能够抵御99.999%的黑客攻击；(d)系统的用户名必须以字母开头，长度不少于5个字符；(e)对查询请求处理时间的要求将影响系统的数据传输协议和处理过程的设计；(f)网络失效后，系统需要在2分钟内发现并启用备用网络系统；(g)在系统升级时，需要保证在1个月内添加一个新的消息处理中间件；(h)查询过程中涉及到的车辆实时视频传输必须保证20帧/秒的速率，且画面具有600\*480的分辨率；(i)更改系统加密的级别将对安全性和性能产生影响；(j)系统主站点断电后，需要在3秒内将请求重定向到备用站点；(k)假设每秒中用户查询请求的数量是10个，处理请求的时间为30毫秒，则“在1秒内完成用户的查询请求”这一要求是可以实现的；(l)对用户信息数据的授权访问必须保证99.999%的安全性；(m)目前对“车辆信息实时监控”业务逻辑的描述尚未达成共识，这可能导致部分业务功能模块的重复，影响系统的可修改性；(n)更改系统的Web界面接口必须在1周内完成；(o)系统需要提供远程调试接口，并支持系统的远程调试。在对系统需求和质量属性场景进行分析的基础上，系统的架构师给出了三个候选的架构设计方案。公司目前正在组织系统开发的相关人员对系统架构进行评估。

2015【问题1】

在架构评估过程中，质量属性效用树（utility tree）是对系统质量属性进行识别和优先级排序的重要工具。请给出合适的质量属性，填入图1-1中(1)、(2)空白处；并选择题干描述中的(a)～(o)，将恰当的序号填入(3)～(6)空白处，完成该系统的效用树。





本题主要考查考生对于软件质量属性的理解、掌握和应用。在解答该问题时，应认真阅读题干中给出的场景与需求描述，分析该需求描述了何种质量属性，根据质量属性描述对其归类，并需要理解架构风险、敏感点和权衡点这些概念。

质量属性效用树是对质量属性进行分类、权衡、分析的架构分析工具，主要关注系统的性能、可用性、可修改性和安全性四个方面。根据对相关质量属性的定义和含义，

其中“正常负载情况下，系统必须在0.5秒内对用户的车辆查询请求进行响应”和“查询过程中涉及到的车辆实时视频传输必须保证画面具有600×480的分辨率，20帧/秒的速率”，这描述的是系统的性能属性；“网络失效后，系统需要在2分钟内发现错误并启用备用系统”和“系统主站点断电后，需要在3秒内将请求重定向到备用站点描述的则是系统的可用性；“在系统升级时，需要保证在20人月内添加一个新的消息处理中间件”和“更改系统的Web界面接口必须在4人周内完成”描述的是系统的可修改性；“车辆信息查询功能必须保证99.999%的安全性”和“用户信息数据库授权必须保证99.999%可用”描述的是系统的安全性。

2015【问题2】

在架构评估过程中；需要正确识别系统的架构风险、敏感点和权衡点，并进行合理的架构决策。请用300字以内的文字给出系统架构风险、敏感点和权衡点的定义，并从题干描述中的(a)～(o)各选出1个属于系统架构风险、敏感点和权衡点的描述。

系统架构风险是指架构设计中潜在的、存在问题的架构决策所带来的隐患。

敏感点是指为了实现某种特定的质量属性，一个或多个系统组件所具有的特性。

权衡点是指影响多个质量属性，并对多个质量属性来说都是敏感点的系统属性。题干描述中，(m)描述的是系统架构风险；(e)描述的是敏感点；(i)描述的是权衡点。

系统的架构风险、敏感点和权衡点是对质量属性效用树进行分析的主要依据，根据相关概念，题干中“对查询请求处理时间的要求将影响系统的数据传输协议和处理过程的设计”描述的是敏感点；“目前对‘车辆信息实时监控’业务逻辑的描述尚未达成共识，这可能导致部分业务功能模块的重复，影响系统的可修改性”描述的是系统的架构风险;“更改系统加密的级别将对安全性和性能产生影响”描述的是权衡点。

* 识别风险、非风险、敏感点和权衡点是进行软件架构评估的重要过程。“改变业务数据编码方式会对系统的性能和安全性产生影响”是对（60）的描述，“假设用户请求的频率为每秒1个，业务处理时间小于30毫秒，则将请求响应时间设定为1秒钟是可以接受的”是对（61）的描述。

2014年(60)

A.风险点

B.非风险

C.敏感点

D.权衡点

2014年(61)

A.风险点

B.非风险

C.敏感点

D.权衡点

【答案】D B 【解析】本题主要考查考生对风险、非风险、敏感点和权衡点等重要评估概念的掌握和理解。

风险是某个存在问题的架构设计决策，可能会导致问题：非风险与风险相对，是良好的架构设计决策；

敏感点是一个或多个构件的特性；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。根据上述定义，可以看出“改变业务数据编码方式会对系统的性能和安全性产生影响”是对权衡点的描述，“假设用户请求的频率为每秒1个，业务处理时间小于30毫秒，则将请求响应时间设定为1秒钟是可以接受的”是对非风险的描述。

* 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(62)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。“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”，这是一个对系统(63)的描述。

(62)

A.风险点

B.非风险点

C.敏感点

D.权衡点

(63)

A.风险点

B.非风险点

C.敏感点

D.权衡点

【答案】C D 【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。

* 某公司欲为某种型号的示波器开发内置软件。该公司的架构师设计了如下图所示的软件架构。在软件架构评审时，专家认为该架构存在的问题是(49)。

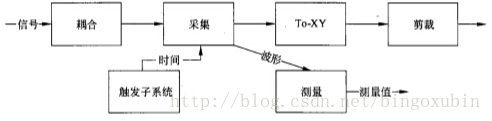
2010年(49)

A.在功能划分上将各个模块独立起来

B.在硬件构件的混合和替换方面不是很灵活

C.没有清晰地说明用户怎样与其交互

D.没有明确的层次关系，没有强调功能之间的交互试



【答案】C 【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，可以看出图中的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。

* 识别风险点、非风险点、敏感点和权衡点是软件架构评估过程中的关键步骤。针对某系统所作的架构设计中，“系统需要支持的最大并发用户数量直接影响传输协议和数据格式”描述了系统架构设计中的一个（62)：“由于系统的业务逻辑目前尚不清楚， 因此现有系统三层架构中的第二层可能会出现功能重复，这会影响系统的可修改性”描述了系统架构设计中的一个（63)。

2011年(62)

A.敏感点

B.风险点

C.非风险点

D.权衡点

2011年(63)

A.敏感点

B.风险点

C.非风险点

D.权衡点

【答案】A B 【解析】本题考查对系统风险点、非风险点、敏感点和权衡点这些架构评估概念的理解和掌握。

根据题干描述“系统需要支持的最大并发用户数量直接影响传输协议和数据格式”，

“最大并发用户数量”这一个质量属性会同时影响“传输协议和数据格式”这两个质量属性，因此其描述的是一个敏感点；“由于系统的业务逻辑目前尚不清楚，因此现有系统三层架构中的第二层可能会出现功能重复，这会影响系统的可修改性”这段话描述了由于某种问题会影响系统的某种质量属性，这是一个系统的风险。

## 架构的权衡分析法(ATAM、SAAM)

知识点

1. ATAM是一种常用的软件架构评估方法，该方法强调对软件的质量属性进行分析、分类和优先级排序等工作，在此基础上构建质量属性效用树，并对风险点、非风险点、敏感点和权衡点进行识别和分析
2. SAAM是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. 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。

试题

* 识别风险点、非风险点、敏感点和权衡点是ATAM方法中的关键步骤。已知针对某系统所做的架构设计中，提高其加密子系统的加密级别将对系统的安全性和性能都产生非常大的影响，则该子系统一定属于（63）。

2009年(63)

A.风险点和敏感点

B.权衡点和风险点

C.权衡点和敏感点

D.风险点和非风险点

【答案】C 【解析】本题考査软件体系结构中的评估方法。

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。

* Architecture Tradeoff Analysis Method(ATAM)是一种软件架构的评估方法，以下关于该方法的叙述中，正确的是（62)。

2009年(62)

A.ATAM是一种代码评估方法

B.ATAM需要评估软件的需求是否准确

C.ATAM需要对软件系统进行测试

D.ATAM不是一种精确的评估工具

【答案】D 【解析】本题考查软件体系结构中的评估方法。

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，因为评估是考査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并木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。

* 架构权衡分析方法（ATAM)是一种常用的软件架构评估方法，下列关于该方法的叙述中，正确的是（61).

2011年(61)

A.ATAM需要对代码的质量进行评估

B.ATAM需要对软件系统需求的正确性进行评价

C.ATAM需要对软件系统进行集成测试

D.ATAM需要对软件质量属性进行优先级排序

【答案】D 【解析】

ATAM是一种常用的软件架构评估方法，该方法强调对软件的质量属性进行分析、分类和优先级排序等工作，在此基础上构建质量属性效用树，并对风险点、非风险点、敏感点和权衡点进行识别和分析。

* 基于场景的架构分析方法（Scenarios-based Architecture Analysis Method，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、（62）和架构描述文档，其分析过程主要包括场景开发、(63)、单个场景评估、场景交互和总体评估。

2012年(62)

A.问题说明

B.问题建模

C.需求说明

D.需求建模

2012年(63)

A.架构需求

B.架构描述

C.架构设计

D.架构实现

【答案】C B 【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 AnalysisMethod, 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。

* 架构权衡分析方法（Architecture Tradeoff Analysis Method, ATAM)是一种系统架构评估方法，主要在系统开发之前，针对性能、(57)、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、(58)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（59）作为架构评估的核心概念。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”主要与（60）质量属性相关;“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”主要与(61)质量属性相关。在识别出上述描述后，通常采用（62）对质量属性的描述进行刻画与排序。在评估过程中，(63)是一个会影响多个质量属性的架构设计决策。

2013年(57)

A.可测试性

B.可移植性

C.可用性

D.易用性

2013年(58)

A.架构视图

B.架构排序

C.架构风格

D.架构策略

2013年(59)

A.用例

B.视图

C.属性

D.模型

2013年(60)

A.可测试性

B.互操作性

C.可移植性

D.易用性

2013年(61)

A.可测试性

B.互操作性

C.可移植性

D.易用性

2013年(62)

A.期望管理矩阵

B.决策表

C.有限队列

D.效用树

2013年(63)

A.风向点

B.决策点

C.权衡点

D.敏感点

【答案】C A C D A D C 【解析】本题主要考查架构权衡分析方法（Architecture Tradeoff AnalysisMethod,ATAM)的基础知识与应用。 架构权衡分析方法（Architecture Tradeoff Analysis Method,ATAM)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。题干描述中，“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”，讨论的是针对使用系统的用户的习惯问题，这与易用性相关。“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”这个描述与系统的可测试性相关。在识别出质量属性描述后，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。

* 架构权衡分析方法(Architecture Tradeoff Analysis Method, 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、（62），属性模型构造和分析，属性模型折中等四个阶段。ATAM方法要求在系统开发之前，首先对这些质量属性进行（63）和折中。

2015年(62)

A.架构视图和场景实现

B.架构风格和场景分析

C.架构设计和目标分析

D.架构描述和需求评估

2015年(63)

A.设计

B.实现

C.测试

D.评价

【答案】A D 【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。

## 成本效益分析法

# 构件及其复用

## 商用构件标准规范

知识点

1. 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）基础设施定义了4种构件标准
2. 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。
3. 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。
4. 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。
5. 服务(Service)构件是无状态的
6. 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。
7. 面向构件的编程（COP）关注于如何支持建立面向构件的解决方案。一个基于一般 OOP 风格的 COP定义如下（Szyperski，1995）：
8. “多态性（可替代性）；
9. 模块封装性（高层次信息的隐藏）；
10. 后期的绑定和装载（部署独立性）；
11. 安全性（类型和模块安全性）。”

试题

* 试题三【说明】构件(component)也称为组件，是一个功能相对独立的具有可复用价值的软硬件单元。近年来，构件技术正在逐步应用于大型嵌入式系统的软件设计。某公司长期从事飞行器电子设备研制工作，已积累了大量成熟软件。但是，由于当初管理和设计等原因，公司的大量软件不能被复用，严重影响了公司后续发展。公司领导层高度重视软件复用问题，明确提出了要将本公司的成熟软件进行改造，建立公司可复用的软件构件库，以提升开发效率、降低成本。公司领导层决定将此项任务交给技术部门的王工程师负责组织实施。两个月后，王工程师经过调研、梳理和实验，提交了一份实施方案。此方案得到了公司领导层的肯定，但在实施过程中遇到了许多困难，主要表现在公司软件架构的变更和构件抽取的界面等方面。

【2014问题1】 请用200字以内文字说明获取构件的方法有哪几种？开发构件通常采用哪几种策略？并列举出两种主流构件标准。

基于构件的软件开发中，可以通过不同的途径来获取构件，主要包括以下4种方法： (1)从现有构件中获得符合要求的构件，直接使用或做适应性修改，得到可复用的构件； (2) 通过遗留工程（LegacyEngineering)，将具有潜在复用价值的软件提取出来，得到可复用的构件； (3)从市场上购买现成的商业构件，BPCOTS(Commercial Off-The-Shell)构件； (4) 开发新的符合要求的构件。

开发构件通常采取3种策略： (1) 分区（partitioning)：指的是将问题情景的空间分割成几乎可以独立研究的部分； (2)抽象（abstraction)：是对在给定实践内执行指定计算的软/硬件申.元的一种抽象； (3)分割（segmentation)：是将结构引入构件的行为，支持对行为性质进行时序推理。 当前主流构件标准有： (1)CORBA：由OMG(对象管理集团）制定； (2) COM/DCOM：由Microsoft制定； (3)EJB：由SUN的Java企业Bean制定。本题考查软件构件（component)基本概念、提取构件需要采取的一般方法，通过一种简单的实例，重点考查考生对构件知识使用的掌握程度。

此类题B要求考生认真阅读题目对问题的描述，通过自己对构件知识的掌握的程度，采用总结、抽象和概括等的方式，从问题描述中发现问题的相关性，正确回答问题。

构件（component)也称为组件，是一个功能相对独立的具有可复用价值的软硬件单元。近年来，构件技术正在逐步应用于大型嵌入式系统的软件设计。从传统意义上来讲，构件就是一种可独立开发、具备独立功能的一类软件。它具备有独立性、可重用性、可组装性、可配置性等特点，构件没有大小之分，可通过将几个构件组装成一个新构件。

通常情况下，软件人员在从事开发时，在分析和论证的基础上，提炼出适合本项目需要的构件，这样可降低软件开发成本、缩短开发周期。软构件可通过多种途径获取，目前可主要归纳为以下四种方法：

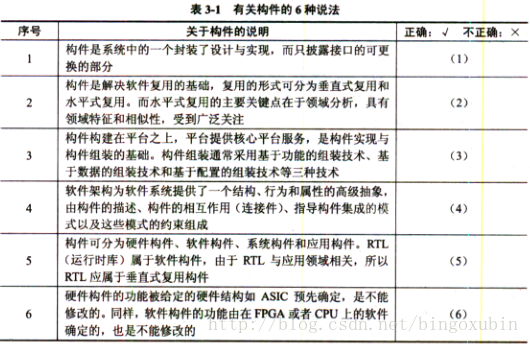
(1) 修改已有构件：从现有构件中获得符合要求的构件，直接使用或做适应性修改，得到可复用的构件： (2)封装新构件：通过遗留工程（Legacy Engineering),将具有潜在复用价值的软件提取出来，得到可复用的构件； (3)

COTS构件：从市场上购买现成的商业软件（构件），通过处理形成满足自己需要的构件，BIJCOTS(Commercial Off-The-Shell)构件； (4)新开发构件：针对项目需要，在分许、评估的基础上，开发新的符合要求的构件。

软件构件的幵发方法通常包括了分区（partitioning)、抽象（abstraction)和分割(segmentation)等三种。分区指的是将问题情景的空间分割成几乎可以独立研究的部分；抽象是对在给定实践内执行指定计算的软/硬件单元的一种抽象；分割是将结构引入构件的行为，支持对行为性质进行时序推理。通俗地说，分区就是在空间上对软件进行划分，保证构件在空间上具备独立特性，分割就是按软件程序的执行行为特征，按时间关系进行分解，保证构件在时间上具备独立特性，抽象就是按软件功能独立性进行分解和抽象。目前，基于构件的软件体系标准是由OMG(对象管理集团）制定的CORBA标准、由Microsoft公司制定COM/DCOM标准和由SUN的Java企业Bean制定EJB标准。

【2014问题2】 由于该公司已具备大量的成熟软件，王工程师此次的主要工作就是采用遗留工程 (Legacy Engineering)方法，将具有潜在复用价值的软件提取出来，得到可复用的构件。因此，在设计软件时与原开发技术人员产生了重大意见分歧，主要分歧焦点在于大家对构件概念理解上的差异。请根据你对构件的理解，判断表

3-1 给出的有关构件的说法是否正确，将答案写在答题纸上。



（1）√（2）×（3）×（4）√（5）×（6）×

本问题主要考查考生对构件基本知识的掌握程度，通过判断正确、错误的形式，考察考生对构件概念正确性理解。每个判断题正确的描述如下： 2014年(1)

“构件是系统中的一个封装了设计与实现，而只披露接口的可更换的部分”。此种描述是正确的。 2014年(2)

“构件是解决软件复用的基础，复用的形式可分为垂直式复用和水平式复用。而垂直式复用的主要关键点在于领域分析，具有领域特征和相似性，受到广泛关注”。垂直式复用是与领域特性相关的，而水平式复用是一种公用的服务，不予某个特殊领域相关。

2014年(3)

“构件构建在平台之上，平台提供核心平台服务，是构件实现与构件组装的基础。构件组装通常采用基于功能的组装技术、基于数据的组装技术和面向对象的组装技术等三种技术”。配置只是一种构件功能组合动态方法，而不是构件组装的技术。

2014年(4)

“软件架构为软件系统提供了一个结构、行为和属性的高级抽象，由构件的描述、构件的相互作用（连接件)、指导构件集成的模式以及这些模式的约束组成”。该描述是正确的。

2014年(5)

“构件可分为硬件构件、软件构件、系统构件和应用构件。RTL(运行时库）属于软件构件，由于RTL可适应多种应用领域，所以RTL与属于水平式复用构件”。RTL是C/C++语言为用户提供的一种运行时库，如数学库、stdio库等，它可服务于多种应用，而与领域需求无关，所以说RTL不属于垂直式复用构件。

2014年(6)

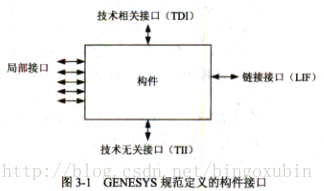
“硬件构件的功能被给定的硬件结构如ASIC预先确定，他是不能修改的。同样，软件构件的功能由在FPGA或者CPU上的软件确定的，我们将加载在软件构件上的软件称为作业。将作业分配给适当的可以执行该作业的硬件单元就创建了新的构件。软件构件的功能所以在构件的寿命期中可以修改”。

【2014问题3】

王工程师的实施方案指出：本公司的大部分产品是为用户提供标准计算平台的，而此平台中的主要开发工作是为嵌入式操作系统研制板级支持软件(BSP)。为了提高BSP软件的复用，应首先开展BSP构件的开发，且构件架构应符合国外GENESYS规范定义的嵌入式系统架构风格。图

3-1 给出了架构风格定义的构件通用接口，其中：链接接口

(LIF)是构件对外提供的功能服务接口；局部接口建立了构件和它的局部环境的连接，如传感器、作动器或人机接口；技术相关接口(TDI)提供了查看构件内部、观察构件的内部变量的手段，如诊断等；技术无关接口(TII)用来在运行时配置、复使、重启构件的接口。现需要针对BSP中常用的RS-232串行驱动程序设计一个可复用的软构件，请说明该软构件四类接口的具体功能。



RS-232驱动程序主要完成对RS-232芯片的初始化，实现RS-232数据发送、接收和控制等功能。依据GENESYS规范定义的构件接口含义，RS-232驱动程序构件的接口定义如下：

(1)链接接口

RS-232驱动构件的使用者是上层的操作系统或应用软件，本构件应该给他们提供串行接口的数据发送、数据接收服务（1分)。因此，链接接口至少包括以下功能服务：

• Send()：处理机中的程序向RS-232芯片输出数据； • Receive()：处理机中的程序通过RS-232芯片从外部接收数据； •

Init232()：初始化RS-232芯片。 (2) 局部接口

RS-232驱动构件的局部接口应该是RS-232芯片为编程提供的各类寄存器和CPU对外围芯片的支持，此接口与硬件紧密相关。就RS-232芯片而言，局部接口至少包括以下功能服务：

• Intconnect()：将RS-232芯片的外部中断向量连接到CPU； •

Rcad()：从RS-232芯片中读取寄存器的内容（包括控制、状态类寄存器和数据输入寄存器)； •

Write()：将控制信息或输出的数据写入RS-232芯片的相应寄存器。 (3) 技术相关接口

RS-232驱动构件的技术相关接口应该是RS-232驱动构件为维护人员提供的芯片诊断接口，此接口与硬件和应用紧密相关。就RS-232芯片使用而言，技术相关接口至少包括以下诊断功能服务： • Datacheck()：诊断数据传输校验的错误。 (4) 技术无关接口

RS-232驱动构件的技术无关接口应该是RS-232驱动构件为编程人员提供的芯片初始化或运行时的配置接口。技术无关接口至少包括以下功能服务：

• Reset()：RS-232芯片或驱动构件复位； • Restart()：RS-232芯片或驱动构件重启动； •

Rateconfigure()：配置、重新配置RS-232传输速率。

本问题要求考生根据自己掌握的构件基本知识，结合图3-1给出的一种构件接口要求，在分析题干的基础上，针对一个驱动实例，完成串行接口的驱动软构件的设计。

RS-232驱动程序是每个从事嵌入式软件开发人员都知道的一种软件，它主要完成对RS-232芯片的初始化，实现RS-232数据发送、接收和控制等功能。本题要求考生将这些功能（服务）有效地分解到图3-1定义的构件四类接口。如果考生具有一定实际经验，就可完成次设计。下面简要说明答题思路：

(1) 链接接口

RS-232驱动构件的使用者是上层的操作系统或应用软件，本构件应该给他们提供串行接口的数据发送、数据接收服务。因此，链接接口至少包括以下功能服务：

• Send()：处理机中的程序向RS-232芯片输出数据； • Receive()：处理机中的程序通过RS-232芯片从外部接收数据； •

Init232()：初始化RS-232芯片。 (2) 局部接口

RS-232驱动构件的局部接口应该是RS-232芯片为编程提供的各类寄存器和CPU对外围芯片的支持，此接口与硬件紧密相关。就RS-232芯片而言，局部接口至少包括以下功能服务：

• Intconnect()：将RS-232芯片的外部中断向量连接到CPU； •

Read()：从RS-232芯片中读取寄存器的内容（包括控制、状态类寄存器和数据输入寄存器)； •

Write()：将控制信息或输出的数据写入RS-232芯片的相应寄存器。 (3) 技术相关接口

RS-232驱动构件的技术相关接口应该是RS-232驱动构件为维护人员提供的芯片诊断接口，此接口与硬件和应用紧密相关。就RS-232芯片使用而言，技术相关接口至少包括以下诊断功能服务： • DatacheckO：诊断数据传输校验的错误。 (4) 技术无关接口

RS-232驱动构件的技术无关接口应该是RS-232驱动构件为编程人员提供的芯片初始化或运行时的配置接口。技术无关接口至少包括以下功能服务：

• Reset()：RS-232芯片或驱动构件复位； • Restart()：RS-232芯片或驱动构件重启动； •

Rateconfigure()：配置、重新配置RS-232传输速率。

问题3的解答内容仅供参考，每一类接口提供的服务格式、分类等可以有差异，但对于RS-232驱动构件每种功能分类一致即可。

* 对象管理组织(OMG)基于CORBA基础设施定义了4种构件标准。其中(38)的状态信息是由构件自身而不是由容器维护。

2010年(38)

A.实体构件

B.加工构件

C.服务构件

D.会话构件

【答案】D 【解析】 对象管理组织(OMG)基于CORBA基础设施定义了4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。

* (35)是一个独立可交付的功能单元，外界通过接口访问其提供的服务。

2010年(35)

A.面向对象系统中的对象(Object)

B.模块化程序设计中的子程序(Subroutine)

C.基于构件开发中的构件(Component)

D.系统模型中的包(Package)

【答案】C 【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。

* 以下关于软件中间件的叙述，错误的是（9)。

2012年(9)

A.中间件通过标准接口实现与应用程序的关联，提供特定功能的服务

B.使用中间件可以提高应用软件可移植性

C.使用中间件将增加应用软件设计的复杂度

D.使用中间件有助于提高开发效率

【答案】C 【解析】

中间件是一种独立的系统软件或服务程序，分布式应用软件借助这种软件在不同的技术之间共享资源，中间件位于客户机服务器的操作系统之上，管理计算资源和网络通信。

软件中间件的作用是为处于自己上层的应用软件提供运行与开发的环境，帮助用户开发和集成应用软件。它不仅仅要实现互连，还要实现应用之间的互操作。

* 以下关于软件构件及其接口的叙述，错误的是（38）.

2009年(38)

A.构件是软件系统中相对独立且具有一定意义的构成成分

B.构件在容器中进行管理并获取其属性或者服务

C.构件不允许外部对所支持的接口进行动态发现或调用

D.构件可以基于对象实现，也可以不基于对象实现试

【答案】C 【解析】本题考查软件构件的基本概念。

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。

* 面向构件的编程(Component Oriented Programming，COP)关注于如何支持建立面向构件的解决方案。面向构件的编程所需要的基本支持包括（35）。

2016年(35)

A.继承性、构件管理和绑定、构件标识、访问控制

B.封装性、信息隐藏、独立部署、模块安全性

C.多态性、模块封装性、后期绑定和装载、安全性

D.构件抽象、可替代性、类型安全性、事务管理

【答案】C

【解析】 面向构件的编程（COP）关注于如何支持建立面向构件的解决方案。一个基于一般 OOP 风格的 COP定义如下（Szyperski，1995）： “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）。”

* CORBA构件模型中，（36）的作用是在底层传输平台与接收调用并返回结果的对象实现之间进行协调，（37）是最终完成客户请求的服务对象实现。

2016年(36)

A.伺服对象激活器

B.适配器激活器

C.伺服对象定位器

D.可移植对象适配器POA

2016年(37)

A.CORBA对象

B.分布式对象标识

C.伺服对象Servant

D.活动对象映射表

【答案】D C

【解析】 POA是对象实现与ORB其它组件之间的中介，它将客户请求传送到伺服对象，按需创建子POA，提供管理伺服对象的策略。CORBA对象可看作是一个具有对象标识、对象接口及对象实现的抽象实体。之所以称为抽象的，是因为并没有硬性规定CORBA对象的实现机制。由于独立于程序设计语言和特定ORB产品，一个CORBA对象的引用又称可互操作的对象引用（Interoperable ObjectReference）。从客户程序的角度看，IOR中包含了对象的标识、接口类型及其他信息以查找对象实现。伺服对象（servant）是指具体程序设计语言的对象或实体，通常存在于一个服务程序进程之中。客户程序通过对象引用发出的请求经过ORB担当中介角色，转换为对特定的伺服对象的调用。在一个CORBA对象的生命期中，它可能与多个伺服对象相关联，因而对该对象的请求可能被发送到不同的伺服对象。对象标识（Object ID）是一个用于在POA中标识一个CORBA对象的字符串。它既可由程序员指派，也可由对象适配器自动分配，这两种方式都要求对象标识在创建它的对象适配器中必须具有唯一性。

* 关于构件的描述，正确的是（38）。

2016年(38)

A.构件包含了一组需要同时部署的原子构件

B.构件可以单独部署，原子构件不能被单独部署

C.一个原子构件可以同时在多个构件家族中共享

D.一个模块可以看作带有单独资源的原子构件

【答案】A

【解析】构件是一组通常需要同时部署的原子构件。构件和原子构件之间的区别在于，大多数原子构件永远都不会被单独部署，尽管它们可以被单独部署。相反，大多数原子构件都属于一个构件家族，一次部署往往涉及整个家族。一个原子构件是一个模块和一组资源。原子构件是部署、版本控制和替换的基本单位。原子构件通常成组地部署，但是它也能够被单独部署。一个模块是不带单独资源的原子构件（在这个严格定义下，Java包不是模块——在 Java 中部署的原子单元是类文件。一个单独的包被编译成多个单独的类文件——每个公共类都有一个）。模块是一组类和可能的非面向对象的结构体，比如过程或者函数。

## 应用系统簇与构件系统

知识点

1. 逻辑构件模型用功能包描述系统的抽象设计，用接口描述每个服务集合，以及功能之间如何交互以满足用户需求，它作为系统的设计蓝图以保证系统提供适当的功能。
2. 物理构件模型用技术设施产品、硬件分布和拓扑结构，以及用于绑定的网络和通信协议描述系统的物理设计，这种架构用于了解系统的性能、吞吐率等许多非功能性属性。
3. 组件模型
4. 系统交互模型
5. 架构失配
6. 构件失配主要包括由于系统对构件基础设施、控制模型和数据模型的假设存在冲突引起的失配。
7. 连接子失配包括由于系统对构件交互协议、构件连接时数据格式的假设存在冲突引起的失配。
8. 基于构件的开发模型
9. 利用模块化方法将整个系统模块化，并在一定构件模型的支持下复用构件库中的一个或多个软件构件，通过组合手段髙效率、髙质量地构造应用软件系统的过程。
10. 基于构件的开发模型融合了螺旋模型的许多特征，本质上是演化形的，开发过程是迭代的。
11. 基于构件的开发模型由软件的需求分析定义、体系结构设计、构件库建立、应用软件构建以及测试和发布5个阶段组成。

试题

* 在基于构件的软件开发中，(36)用来了解系统的性能，(37)描述系统设计蓝图以保证系统提供适当的功能: 吞吐率等非功能性属性。

2010年(36)

A.逻辑构件模型

B.物理构件模型

C.组件接口模型

D.系统交互模型

2010年(37)

A.逻辑构件模型

B.物理构件模型

C.组件接口模型

D.系统交互模型

【答案】B A 【解析】

在基于构件的软件开发中，逻辑构件模型用功能包描述系统的抽象设计，用接口描述每个服务集合，以及功能之间如何交互以满足用户需求，它作为系统的设计蓝图以保证系统提供适当的功能。物理构件模型用技术设施产品、硬件分布和拓扑结构，以及用于绑定的网络和通信协议描述系统的物理设计，这种架构用于了解系统的性能、吞吐率等许多非功能性属性。

* 在构件组装过程中需要检测并解决架构失配问题。其中（42）失配主要包括由于系统对构件基础设施、控制模型和数据模型的假设存在冲突引起的失配。（43）失配包括由于系统对构件交互协议、构件连接时数据格式的假设存在冲突引起的失配。

2014年(42)

A.构件

B.模型

C.协议

D.连接子

2014年(43)

A.构件

B.模型

C.协议

D.连接子

【答案】A D 【解析】本题主要考查构件组装过程知识。

在架构模型的指导下，可复用构件可以通过组装的方式在较高层次上实现系统，并能够提高系统实现的效率。在构件组装过程中需要检测并解决架构失配问题。其中构件失配主要包括由于系统对构件基础设施、控制模型和数据模型的假设存在冲突引起的失配。连接子失配包括由于系统对构件交互协议、构件连接时数据格式的假设存在冲突引起的失配。

* 基于构件的开发模型包括软件的需求分析定义、(35)、(36)、(37)，以及测试和发布5个顺序执行的阶段。

2009年(35)

A.构件接口设计

B.体系结构设计

C.元数据设计

D.集成环境设计

2009年(36)

A.数据库建模

B.业务过程建模

C.对象建模

D.构件库建立

2009年(37)

A.应用软件构建

B.构件配置管理

C.构件单元测试

D.构件编码实现

【答案】B D A 【解析】本题考查基于构件的软件开发模型的基础知识。

基于构件的开发模型利用模块化方法将整个系统模块化，并在一定构件模型的支持下复用构件库中的一个或多个软件构件，通过组合手段髙效率、髙质量地构造应用软件系统的过程。基于构件的开发模型融合了螺旋模型的许多特征，本质上是演化形的，开发过程是迭代的。基于构件的开发模型由软件的需求分析定义、体系结构设计、构件库建立、应用软件构建以及测试和发布5个阶段组成。

## 基于复用开发的组织结构

# 产品线及系统演化

## 复用与产品线

## 基于产品线的架构

## 产品线的开发模型

## 特定领域软件架构DSSA

知识点

1. 基本定义
2. 特定领域软件架构(DSSA)是在一个特定应用领域为一组应用提供组织结构参考的标准软件架构, 它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
3. 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
4. 基本活动:包括领域分析、领域设计和领域实现
5. 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；。。。这个阶段的主要目标是获得领域模型。领域模型描述领域中系统之间的共同的需求，即领域模型所描述的需求为领域需求。在这个阶段中首先要进行一些准备性的活动，包括定义领域的边界。从而明确分析的对象；识别信息源，整个领域工程过程中信息的来源，可能的信息源包括现存系统、技术文献、问题域和系统开发的专家、用户调查和市场分析、领域演化的历史记录等，在此基础上就可以分析领域中系统的需求，确定哪些需求是领域中的系统广泛共享的，从而建立领域模型。当领域中存在大量系统时，需要选择它们的一个子集作为样本系统。对样本系统需求的考察将显示领域需求的一个变化范围。一些需求对所有被考察的系统是共同的，一些需求是单个系统所独有的。很多需求位于这两个极端之间，即被部分系统共享
6. 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；。。。这个阶段的目标是获得DSSA。DSSA描述在领域模型中表示的需求的解决方案，它不是单个系统的表示，而是能够适应领域中多个系统的需求的一个高层次的设计。建立了领域模型之后，就可以派生出满足这些被建模的领域需求的DSSA，由于领域模型中的领域需求具有一定的变化性，DSSA也要相应地具有变化性。它可以通过表示多选一的(alternative)、可选的(optional)解决方案等来做到这一点。模型和DSSA来组织的，因此在这个阶段通过获得DSSA，也就同时形成了重用基础设施的规约
7. 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。。。这个阶段的主要目标是依据领域模型和DSSA，开发和组织可重用信息。这些可重用信息可能是从现有系统中提取得到，也可能需要通过新的开发得到。它们依据领域模型和DSSA进行组织，也就是领域模型和DSSA定义了这些可重用信息的重用时机，从而支持了系统化的软件重用。这个阶段也可以看作重用基础设施的实现阶段
8. 人员：
9. 领域专家的主要任务是提供关于领域中系统的需求规约和实现的知识；。。。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。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；
10. 领域分析者的任务是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；。。。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
11. 领域设计者的任务是根据领域模型和现有系统开发出DSSA，并对DSSA的准确性和一致性进行验证。。。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核个软件设计过程，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
12. 领域实现人员主要在领域特定应用开发环境中工作。。。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

试题

* 特定领域软件架构(DSSA)是在一个特定应用领域为一组应用提供组织结构参考的标准软件架构。实施DSSA的过程中包括一系列基本的活动，其中(53)活动的主要目的是为了获得DSSA。该活动参加人员中，(54)的主要任务是提供关于领域中系统的需求规约和实现的知识。

2010年(53)

A.领域需求

B.领域分析

C.领域设计

D.领域实现

2010年(54)

A.领域专家

B.领域分析者

C.领域设计者

D.领域实现者

【答案】C A 【解析】本题主要考查特定领域软件架构的基本定义和基本活动。

特定领域软件架构(DSSA)是在一个特定应用领域为一组应用提供组织结构参考的标准软件架构。实施DSSA的过程中包括一系列基本的活动，其中领域设计活动的主要目的是为了获得DSSA。该活动参加人员中，领域专家的主要任务是提供关于领域中系统的需求规约和实现的知识。

* 特定领域软件架构（Domain Specific Software Architecture, DSSA)是在一个特定应用领域中，为一组应用提供组织结构参考的标准软件体系结构。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（54),从而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(55),从而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开发和组织可重用信息，并对基础软件架构进行实现。

2012年(54)

A.领域边界

B.领域信息

C.领域对象

D.领域模型

2012年(55)

A.特定领域软件需求

B.特定领域软件架构

C.特定领域软件设计模型

D.特定领域软件重用模型

【答案】D B 【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。

* 特定领域软件架构（Domain Specific Software Architecture，DSSA)是在一个特定应用领域中，为一组应用提供组织结构参考的标准软件体系结构。DSSA通常是一个具有三个层次的系统模型，包括（45)环境、领域特定应用开发环境和应用执行环境，其中（46)主要在领域特定应用开发环境中工作。

2013年(45)

A.领域需求

B.领域开发

C.领域执行

D.领域应用

2013年(46)

A.操作员

B.领域架构师

C.应用工程师

D.程序员

【答案】B C 【解析】本题主要考查特定领域软件架构的基础知识。 特定领域软件架构（Domain Specific Software Architecture, DSSA)是在一个特定应用领域中，为一组应用提供组织结构参考的标准软件体系结构。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，其中应用工程师主要在领域特定应用开发环境中工作。

* 特定领域软件架构(Domain Specific Software Architecture, DSSA)是在一个特定应用领域中，为一组应用提供组织结构参考的标准软件体系结构。参加DSSA的人员可以划分为多种角色，其中（47）的任务是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；（48）的任务是根据领域模型和现有系统开发出DSSA，并对DSSA的准确性和一致性进行验证。

2014年(47)

A.领域专家

B.领域分析者

C.领域设计者

D.领域实现者

2014年(48)

A.领域专家

B.领域分析者

C.领域设计者

D.领域实现者

【答案】B C 【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描述领域模璀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。参加DSSA的人员可以划分为多种角色，其中领域分析者的任务是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；领域设计者的任务是根据领域模型和现有系统开发出DSSA，并对DSSA的准确性和一致性进行验证。

* DSSA是在一个特定应用领域中为一组应用提供组织结构参考的软件体系结构，参与DSSA的人员可以划分为4种角色，包括领域专家、领域设计人员、领域实现人员和（45），其基本活动包括领域分析、领域设计和（46）。

2016年(45)

A.领域测试人员

B.领域顾问

C.领域分析师

D.领域经理

2016年(46)

A.领域建模

B.架构设计

C.领域实现

D.领域评估

【答案】C C

【解析】 (45)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。

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核个软件设计过程，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。

(46)

DSSA的基本活动包括：领域分析、领域设计、领域实现。

1、领域分析

这个阶段的主要目标是获得领域模型。领域模型描述领域中系统之间的共同的需求，即领域模型所描述的需求为领域需求。在这个阶段中首先要进行一些准备性的活动，包括定义领域的边界。从而明确分析的对象；识别信息源，整个领域工程过程中信息的来源，可能的信息源包括现存系统、技术文献、问题域和系统开发的专家、用户调查和市场分析、领域演化的历史记录等，在此基础上就可以分析领域中系统的需求，确定哪些需求是领域中的系统广泛共享的，从而建立领域模型。当领域中存在大量系统时，需要选择它们的一个子集作为样本系统。对样本系统需求的考察将显示领域需求的一个变化范围。一些需求对所有被考察的系统是共同的，一些需求是单个系统所独有的。很多需求位于这两个极端之间，即被部分系统共享。

2、领域设计

这个阶段的目标是获得DSSA。DSSA描述在领域模型中表示的需求的解决方案，它不是单个系统的表示，而是能够适应领域中多个系统的需求的一个高层次的设计。建立了领域模型之后，就可以派生出满足这些被建模的领域需求的DSSA，由于领域模型中的领域需求具有一定的变化性，DSSA也要相应地具有变化性。它可以通过表示多选一的(alternative)、可选的(optional)解决方案等来做到这一点。模型和DSSA来组织的，因此在这个阶段通过获得DSSA，也就同时形成了重用基础设施的规约。

3、领域实现

这个阶段的主要目标是依据领域模型和DSSA开发和组织可重用信息。这些可重用信息可能是从现有系统中提取得到，也可能需要通过新的开发得到。它们依据领域模型和DSSA进行组织，也就是领域模型和DSSA定义了这些可重用信息的重用时机，从而支持了系统化的软件重用。这个阶段也可以看作重用基础设施的实现阶段。

值得注意的是，以上过程是一个反复的、逐渐求精的过程。在实施领域工程的每个阶段中，都可能返回到以前的步骤，对以前的步骤得到的结果进行修改和完善，再回到当前步骤，在新的基础上进行本阶段的活动。

* 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，（53）等组成的开发基础架构，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（54），从而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（55），从而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开发和组织可重用信息，并实现基础软件架构。

2015年(53)

A.参考设计

B.参考规约

C.参考架构

D.参考实现

2015年(54)

A.领域边界

B.领域信息

C.领域对象

D.领域模型

2015年(55)

A.特点领域软件需求

B.特定领域软件架构

C.特定领域软件设计模型

D.特定领域软件重用模型

【答案】C D B 【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。

## 架构及系统演化

# 软件架构视图

## 软件视图的分类

## 模块视图类型及其风格

## C&C视图类型及其风格

## 分配视图类型及其风格

## 各视图类型间的映射关系