

复习小测验

2018年9月12日 15:06

1.1 软件开发的本质

1. 软件开发的**本质困难**定义了不变事实
2. 两组 主要的利益相关者是**客户和开发者**
3. **增量版本并不是向软件产品增加新的功能**，增量改进非功能性的软件质量，如软件的正确性、可靠性、耐用性、性能等。
4. **COBIT是产品标准**，而ITIL，CMM，ISO9000是过程标准
5. 面向门户的集成**可以被看作一种特殊的面向信息的集成**，并不是面向接口的集成。

1.2 系统规划

1. 系统规划的**主要目标是效果**，效率其次
2. SWOT中，**具体目标来源于长远目标**，别说反了
3. SWOT中，销售和市场是一个**基础活动**
4. 根据BPR方法，过程企业 与 传统组织 之间最明显的区别是**过程所有者的存在**
5. ISA框架的5个视角：**规划者、所有者、设计者、建造者、承包者**

1.3 三级管理系统

1. 决策的**战术级**主要由数据仓库技术支持
2. 事务管理的两个主要功能：**并发控制和从故障中恢复**
3. OLAP技术中，**数据集市**的目的是支持个别部门或业务功能，仅仅存储被汇总的历史数据
4. 支撑知识处理系统的主要技术是**数据挖掘**

1.4 软件开发生命周期

1. 软件开发方法中，**结构化方法**利用了功能分解活动
2. 业务分析的另一个名字是**需求分析**
3. **体系结构设计阶段**主要负责生产交付一个**适应性**系统
4. **桩的概念与集成和部署阶段**联系在一起
5. **项目规划、度量、测试**跨越了开发生命周期，非独立生命周期阶段

1.5 开发模型与方法

1. 对软件项目增加一个新的功能是**集成**的职责。
2. **螺旋模型**对风险分析是最明确的。
3. **模型驱动的体系结构（MDA）**直接关联到形式规格说明的传统观念
4. **螺旋模型**直接关联到策划编程的概念
5. **面向方面的软件开发**直接关联的横切关注点的概念

2.1 从业务过程到解决方案的构想

1. 为了填补业务人员和IT人士之间空白的最流行的可视化业务过程建模语言是 **业务过程建模表示法BPMN**
2. BPMN中的四类建模元素：**流对象、连接对象、泳池、人工制品**
3. 一个序列流**不能**连接两个泳池，泳池**只能**通过**消息流或关联**与公共的人工制品进行通信。
4. 交付一个IT服务以解决当前业务问题或者促进未来业务创新的业务价值驱动方法是**解决方案构想**
5. 软件能力设计的主要建模输出是**能力体系结构**
6. 解决方案构想中3个不同的实现策略：**常规开发、基于包的开发、基于构件的开发**

2.2需求引导

1. 负责引导和记录领域知识需求和用例需求的专业是 **业务分析员**
2. 两种主要的需求是**功能性需求（服务陈述）**和**非功能性需求（约束陈述、补充性需求）**
3. 调查表中三种封闭式的问题是**多项选择、评价、排序问题**
4. JAD活动中的参与者有：**领导、文书、客户（用户和经理）、开发人员**
5. RAD的开发团队称为：**SWAT，拥有先进工具的专业人员**

2.3需求协商与确认

1. **环境图context diagram**是确定系统边界最好的可视化建模方法
2. 需求依赖矩阵中有**冲突的依赖和重叠的依赖**是显而易见的
3. 与开发过程中需求可能不断变化或进化的情形相联系的是**易变性风险**

2.4需求管理

1. 标识需求的技术有：**唯一标识符；在文档层次内的需求编号；在需求分类中的需求编号**
2. 负责进行变更管理的工具是**软件配置管理工具**。
3. 什么是可疑跟踪？**是需求跟踪矩阵中的一种现象，两个相互联系的需求之间的关系 本可以作为它们之间或其他相关需求变更的结果 得到验证。**

2.5需求业务模型

1. 业务用例的另一个名称：**系统特征**
2. 标识参与者与用例之间事件流的关系名称是**通信关系**
3. 在业务类中，3种主要类型的关系是**关联、泛化、聚合**
4. 在UML中可视化的业务类之间如何可选择的进行参与？**通常在关联的一端置0，在多重性定义中可见。**

2.6需求文档

1. 功能性需求可以细化分为**功能性需求和数据需求**
2. 怎样在需求文档中阐述系统范围之外的相关需求？可在 **“开放问题”** 部分列出

3.1用例视图

1. 最重要的行为建模技术是**用例图、时序图、活动图、通信图（可与时序图相互转换）**

2. 用例图与用例规格说明一样吗? **不一样, 用例规格说明包括图和用例设计行为完整文本描述**

3.2活动视图

1. 活动模型能否作为用例规格说明? **可以, 可以在不同的抽象层次完成活动建模, 包括用例计算步骤的内部规格说明。**
2. 活动图中的流 连接 动作和其他节点, 这些其他节点是**决定、分叉、交汇、合并和对象节点。**

3.3结构视图

1. 实体类与业务对象的概念**相同**
2. **重数的概念适合聚合, 聚合只是关联的一种形式。**

3.4交互视图

1. **消息是对方法的调用**

3.5状态机图

1. 对象的状态**能依赖**于对象的关联连接, 任何属性都能影响对象的状态, 包括指定其他类的属性
2. 如何区分守卫和事件? **在事件处理点上估算守卫条件来决定转换是否将被触发**

3.6实现视图

1. 子系统被建模为包的构造型还是构件的构造型? **在UML较早版本被建模为包的构造型, 在UML2.0被建模为构件的构造型**

4.1体系结构优先权

1. 要构建软件的适应性(可支持性), 必要的和最重要的条件是**遵循某个公认的体系结构框架, 优先进行体系结构设计。**
2. MVC对象中的**控制器**代表鼠标和键盘事件
3. J2EE核心层的**集成层**负责建立和维护与数据源的链接
4. PCBMER层的**资源层**负责建立和维护与数据源的链接

4.2状态规格说明

1. CRC方法(类-职责-协作) **是一种发现类的技术**
2. 角色名的作用在于**解释较复杂的关联, 特别是自身关联**(递归关联, 连接同一个类中的对象)
3. 聚合中传递性的含义是**C是B的一部分, B是A的一部分, 则C是A的一部分**
4. 子类对象是超类变量的一个合法值, 与**可替换性原则**有关

4.3行为规格说明

1. **活动图可以表示并发控制流, 用例不可表示。**

2. 活动图~~能~~在开发模型前构建，因为活动图不直接显示完成动作的对象类
3. 消息~~能~~表示对象间的异步通信，这样的消息称为signal信号
4. 时序图建模对发现类操作最有帮助

4.4状态变化规格说明

1. 状态机图~~不表示~~状态变化的顺序
2. 当已经触发了状态相关的转换，状态变化~~不一定会~~发生，可能指定了入口动作，那么在状态变化发生前，入口动作需要被满意地完成。

5.1高级类建模

1. UML中最重要的扩展机制是~~构造型~~stereotype
2. 在UML2.0中，用~~关联端名~~association end names来称呼角色名
3. Java默认可见性是~~包可见性~~
4. 具体化类refined class~~能够~~无语义semantics损失地代替关联类association class，反过来则不一定。

5.2高级泛化与继承建模

1. 什么相关原则使泛化成为有益的？~~可替换性原则~~——在代码中任何访问超类对象的地方，都可以用子类对象来替换父类对象。
2. 继承如何与封装相折中的？~~继承通过允许子类直接访问保护属性~~
3. ~~接口继承~~能用来替代多重实现继承

5.3高级聚合与委托delegation建模

1. 在典型的程序设计环境中，聚合是怎样实现的？
 - ~~聚合通过获得复合对象和构建对象之间的引用，它是用与常规关联相同的方式来实现的~~
2. ~~ExclusiveOwns聚合~~需要用{frozen}（冻结）约束来指定
3. 聚合使用~~委托~~的方法来复用构建对象的实现。

5.4高级交互建模

1. 要说明多线程执行，需要使用~~异步消息~~的交互建模概念
2. 要说明来自一个未知发送者的交互，需要使用~~发现消息~~交互建模概念
3. servlet属于~~控制器~~结构层
4. 用ref（引用）~~标签~~来标注交互使用

6.1分布式物理体系结构

1. ~~对等体系结构（同位体）~~只定义了一种单一的系统元素
2. 三层体系结构的中间层是~~业务逻辑和企业范围的业务规则~~
3. ~~通过触发器的编程手段~~在数据库中实现了企业范围的业务规则

6.2多层逻辑体系结构

1. 两种主要的、有差别的计算模型是图灵机模型和基于算法和开放的交互模型
2. 结构复杂性的计算与程序的类有关，与对象无关
3. 定义较高层接口使子系统易于使用的模型名称叫外观
4. 责任链模式使得NCP（相邻通信）原则得到增强

6.3体系结构建模

1. 泛化和依赖关系用于相联的包
2. 构件没有持久状态
3. 一个类可以被多个构件实现

6.4程序设计和复用原则

1. 类的内聚定义类的内部自主程度
2. 没有动态分类的支持造成了混合实例内聚，这是面向对象编程环境的弱点
3. 可将构件级的设计复用称为框架复用

6.5协作建模

1. 通常在何种协作模型中输入角色（具有明确定义的类型）——复合结构图
2. 协作模型不标识消息，交互模型标识消息

7.1GUI设计原则

1. GUI客户端如何分类？
 - 在桌面平台上，将GUI客户端归为可编程客户端
 - 在Web平台上，将其归为浏览器客户端
2. 主要的GUI设计指南是用户控制式
3. 反馈指南与用户控制指南最相关，补充：宽容与个性化指南相关

7.2桌面GUI设计

1. 主窗口和辅窗口之间最主要的区别：主窗口有菜单栏和工具栏
2. 属性页是标签夹中标签的窗口名称
3. 菜单项item与快捷键相关

1 What GUI component is an accelerator key related to?

快捷键

- ☐ A. Window
- ☒ B. Menu Bar 菜单栏
- ☐ C. Menu Item 菜单项 ✓
- ☐ D. Checkbox

正确答案: C 你错选为B

2 What is the main characteristic that distinguishes primary from secondary windows?

你没有填写答案

正确答案: The existence of the menu bar and toolbar in primary windows

菜单栏

工具栏

7.3WebGUI设计

1. 在Web应用系统中，**菜单项、按钮、超链接**这些GUI组件用于用户事件编程
2. 维护Web客户端和服务端之间的连接的最原始的机制是**Cookie**
3. 链接的功效是**移动到另一个网页**

7.4GUI导航建模

1. 用于展现屏幕区域的UX故事情节的名称是<<compartment>>
2. UX故事情节中，通过**标签值visible**对隐藏域建模
3. UML图中的**顺序图**可用于行为性UX协作建模

8.1业务对象和持久性

1. 实体类和持久类概念**不相同**，实体类“一定会”称为持久的，因为它在数据库中是持久表示的，但本质上它不是持久的。在把实体类的对象按对象进行存储的面向对象数据库中较难区分（关系数据库将它们存储为表中的记录）
2. **面向对象数据库模型**可以用作客户端应用程序和服务端数据源之间协同工作的对象存储API
3. 最流行的概念数据建模技术是**实体关系ER图**

8.2关系数据库模型

1. 关系数据库模型基于**集合理论（以及谓词逻辑）**的数学概念
2. 关键字的两个主要特征是**唯一且最小=unique**
3. 外键**可以是**NULL值
4. 用术语**更新异常**表示对表进行修改操作所引起的意外副作用

8.3对象关系映射

1. **多对多**的映射关系需要交叉表
2. 映射泛化关系到关系模型使如何处理多态性？**没有提，被忽略了**

8.4管理持久对象的模式

1. 什么是PEAA? **企业应用体系结构模式**
2. **数据映射**模式具有当前内存中的对象的相关信息
3. **工作单元**模式负责处理业务事务

8.5设计数据库的访问和事务

1. SQL程序设计层次的**第3层**允许一次一个记录的处理机制
2. DBMS事务管理的**两个作用**是**数据库恢复**和**并发机制**
3. **可重复读**这一隔离层次能够保证事务的可序列化执行
4. DBA**通过设置检查点出现的频率**来控制恢复时间的长短
5. 程序员**通过保存点**将对象持久地存储到用户的私有数据库中, 以此来控制长事务的回滚

9.1质量管理

1. 质量保证技术中的**评审（走查和审查）**也称为静态测试。
2. 质量保证技术中的**审查**可以生成故障日志以进行操作
3. 流行的Java应用程序测试框架是**JUnit**
4. 通过**测试脚本**来实现（详细说明）测试用例
5. **针对规格说明的测试**能发现遗漏的**功能**

9.2变更管理

1. 提出一次变更需要**以正式的变更请求提出**
2. **成本-效益分析技术**用来确定项目追踪的范围和深度
3. **捕捉-回放工具**可以自动执行测试脚本