1. **(2016判断)用例分析,用例设计和用例实现都是RUP分析与设计中的重要活动,三者之间的关系是:用例实现基于当前已有的用例设计结果,用例设计基于当前已有的用例分析结果.**
2. **(2016判断)识别设计元素是RUP分析与设计中的重要活动,设计类和子系统都可以是识别出来的设计元素,从实现的角度看,在通常情况下,设计类的实现要比子系统的实现简单,所以,在识别设计元素时,应该尽可能的把设计元素识别成设计类而不是子系统.**
3. **(2016判断)RUP中,包(package)是一种可以组中其他设计元素的模型元素,包之间的关系是依赖关系,由于这种依赖关系具有遗传性,所以若包A依赖于包B,包B依赖于包C,那么A的客户在使用包A是没客户不但要依赖包A,也要依赖包B和包C.**
4. **(2016简答)概述你对RUP中架构机制概念的理解,概述中,应该包括RUP定义该机制的主要意图.**
5. **(2016简答)概述你对软件体系结构(简称架构)概念的理解,并简要说明在RUP中如何从多个不同的角度吗**
6. **(2016简答)利用活动图描述RUP的分析与设计工作流.**
7. **(2016简答)RUP中,包(package)是一种常用的机制, 若某个包P内的某原色A的可见性为protected,其含义是什么?请回答什么样的客户可以访问P内的A.**
8. **(2015判断)RUP方法中,包是一种可以组织其他模型元素的,例如在识别设计元素时,可以把识别出的设计元素组织在包中,从而利用这些设计元素以包为单位进行替换和复用,另外,一个开发任务结束时,可以把包作为一种可交付的单元.**

回答:包不提供服务,不完全封装其行为,包只用于模型组织和配置管理,对于包,除非他们具有相同的公共类,否则不可能彼此替代包,所以说包不易于替换

1. **(2015判断)RUP是一种迭代是软件开发过程,一般来说,第一次迭代建立需求模型,第二次迭代进行架构分析,第三次迭代进行用例分析,第四次是识别设计元素,第五次迭代进行用例设计,随后的迭代树妖是实现测试以及验证,当然,相对于简单的系统,可以有选择的把前述不同的任务安排到同义词迭代中,从而减少迭代此时,提高软件开发效率.**

回答:RUP包含四个阶段:先启,精化,构建,产品化,而每个阶段又可以进行一次或者几次迭代,每次迭代要经历六个核心工作流:商业建模,需求,分析和设计,实现,测试,部署,和三个支持工作流:配置和变更管理,项目管理,环境,一般无法确定某次迭代的主要任务是什么,虽然除此迭代一般进行的主要任务是业务建模,需求分析等,随后逐渐变成分析设计实施,乃至部署等等,但是却不能说一般来说第几次迭代主要在干什么.

1. **(2015判断)在利用RUP方法进行用例分析时,补充用例的说明是一个重要环节,即对正在缝隙的用例进行更细化的藐视,补充更多的细节信息,对用例进行补充的描述信息用补充性文档说明进行管理和维护**

回答:补充用例说明的目的是捕获所需的额外信息,以便于理解系统中为系统客户编写的用例说明中可能缺少的内部行为,这些信息将被用作用例分析中其余步骤的输入,并用于协助分配责任,在某些情况下,我们可能会发现某些要求不正确或没有很好的理解,在这种情况下,应该更新时间的原始用例流.

补充用例说明的描述并不是用补充性文档说明来管理和维护,补充性文档说明用于包括用力中所没有的肺功能行需求和功能性需求,包含哪些不映射到特定用例的需求,包括:功能,可用性,可靠性,保障性,设计约束等方面的内容,补充性说明文档与用例模型一起使用意味着对系统范围定义的补充.

1. **(2014简答)概述你对架构机制概念以及作用的理解.**
2. **(2014简答)概述RUP方法中包与子系统的作用以及区别**
3. **(2014简答)简述你对分析类中的边界类,控制类和实体类的作用的理解**
4. **(2014简答)简述你对软件体系结构(架构),设计框架和设计概念的理解.**
5. **(2014简答)对软件系统进行分解的基本原则是高内聚低耦合,请在组件技术和RUP方法范畴内,任意列举5中支持高内聚低耦合的概念方法或技术,并简述他们支持低耦合的基本原理.**
6. **(2012简答)** **架构机制的主要作用是什么?分为哪几种?简述他们之间的主要区别.**
7. **(2012简答)** **什么是分析类?分为哪几种?简述他们之间的主要区别**
8. **(2012简答)什么是包?什么是子系统?简述他们之间的主要区别.**

回答:包是一种分组的通常手法,是可以包含其他模型元素的模型元素;子系统是一个包和一个类的混合体,子系统提供行为,完全封装其行为,易于替代,包不提供行为,不完全封装其行为,可能不易于替代.

1. **(2012简答)在描述运行时架构时,如何对进程视图进行建模?**
2. **(2011简答)列出用例设计活动的输入工件和输出工件,并简述该活动的主要工作步骤及内容.**
3. **(2010判断对错)RUP方法中识别设计元素是精化体系结构活动中的一个工作环节**
4. **(2010判断对错)在RUP方法中分析机制通常用于体系结构分析活动中,通过提供对系统复杂行为的描述来减少分析的复杂性并改善软件在各开发阶段的一致性,一般来说,分析机制所描述的内容与领域无关.**
5. **(2010判断对错)利用4+1视图的形式描述软件的体系结构时,由于每一个view可以用一个UML模型图来标识,并且,可以根据软件系统的性质,淡化省略其中的某一个view或某几个view,所以利用4+1视图描述一个软件的体系结构时,所采用的UML模型图的数量不可能超过5个.**
6. **(2010简答)简述用例分析的主要步骤**
7. **(2010简答)简述体系结构分析的目的**
8. **(2010简答)虽然分析活动和设计活动是在同一个规程”分析和设计”当中,但是这两个活动是有区别的,请简述区别**

回答:分析活动关注:问题的理解,系统的行为和结构,功能需求,期望理想化的设计和小的模型,而设计活动关注解决方案,关注具体的操作和属性,系统的性能,对象生命周期,期望接近实际的代码.

1. **(2010简答)什么是分析机制?列举4个分析机制的例子**

回答:分析机制用于在分析过程中向设计者提供复杂行为的简短表示,从而减少分析的复杂性并提高分析的一致性,

分析机制举例:

永久性机制:粒度,容量,访问机制,可靠性等,

进程间通信机制:反应时间,同步性,协议

遗留接口机制:反应时间,持续时间,访问机制,访问频率

安全机制:数据粒度,用户粒度,安全规则,授权类型,

分布机制,事务管理机制.

1. **(2009判断对错)RUP方法中,设计与实现是由用例驱动的,所以只有所有的用例均被实现,才可能产生可交付的软件.**

回答:错误,RUP是迭代式开发流程,软件产品是增量式被开发出来的,某次迭代产生了可交付的软件,但是与此同时下一回合的迭代早已开始,又有新的用例被分析提出,只要软件还在继续演进,总有未被实现的用例,但是却一直能够产出可交付的软件.

1. **(2009判断对错)在UML用例图中,任何一个actor至少与一个用例相关联,任何一个用例至少与一个actor相关联.**

回答:正确

1. **(2009简答)RUP方法中,需求活动结束后的输出是什么?**

回答:UseCase,supplement,glossary

1. **(2009简答)用例分析结束后输出是什么?**

回答:分析类,分析模型,用例实现

(用例分析的输入是:用例模型,补充说明,词汇表,项目详细指南,软件架构文档)

1. **(2008简答)简述根据用例模型获取类图和顺序图的一般方法**
2. **(2007简答)简述下列UML图的用途:用例图,类图,顺序图,活动图,状态图**
3. **词汇表是什么?**

回答:定义并解释了问题域中的专有名词,定义了系统中的核心概念可帮助领域专家与开发者进行交流,可作为一个非正式的数据字典用来记录数据的定义,包含简介(介绍)和词汇定义两部分.