

1 UML 中的交互图有两种，分别是顺序图和协作图，请分析一下两者之间的主要差别和各自的优缺点。掌握利用两种图进行的设计的方法。

2 什么是高内聚度

3 UML 提供一系列的图支持面向对象的分析与设计,其中_G__给出系统的静态设计视图;_B_对系统的行为进行组织和建模是非常重要的;_D_和_F_都是描述系统动态视图的交互图,其中_D_描述了以时间顺序组织的对象之间的交互活动,_F_强调收发消息的对象的组织结构.

4 什么是对象间的可见性?

5 领域模型是一组表示_A_, 在设计工作中广泛用来启发设计软件对象.

A.真实世界的概念类

B.虚拟世界的概念类

C.软件部件的模型

D.硬件部件的模型

7 统一过程中的四个阶段?

统一过程中有哪四个阶段，各阶段需要完成的主要工作有哪些?

统一过程中的**核心 workflow**有哪些?

8 定义大多数的需求和范围的工作是在 UP 中的__(11)__阶段完成的。

(11) A 初始阶段 B 细化阶段 C 构造阶段 D 提交阶段

9 下列__(12)__不在 UP 的初始阶段中完成的

(12) A 编制简要的愿景文档 B 粗略评估成本 C 定义大多数的需求 D 业务案例

10.领域模型是一组表示真实世界的 (16) ，不是 (17) 的模型。

11 在协作图中通过 (18) 表示出消息的时间顺序。

12 在画 SSD 图时，应该如何对待所涉及的系统:

- A.详细描述其内部结构及其功能;
- B.简单描述其内部结构,但是罗列系统所有的功能
- C.详细描述其内部结构,并不列出系统的功能
- D.不对系统的内部结构与功能进行描述.

13 UP 的核心思想有哪些?

开发过程中不断进行调整

在早期的迭代中解决高风险和高价值的主要问题

不断与用户衔接，及时得到反馈意见

早期注意构造核心的体系结构

早期进入实现和测试，不断进行质量检验

使用用例（use case）

可视化建模（用 UML）

仔细地管理需求

控制变更

14 什么是增量开发?

- 1) 对复杂的用例分多次迭代，一部分一部分地实现

2) 将所有用例按其优先级分别安排在不同的迭代中实现

15 **领域建模**的步骤?

列出候选的概念类

画出领域模型图

必要时加入概念类间的关联

必要时加入概念类的属性

16 **什么**是用例，**用例模型**，**如何描述用例**?

17 **标准建模语言 UML 的重要内容可以由哪五类图(共 9 种图形)来定义?**

18 **简述统一建模语言(UML):**

统一建模语言(UML)是一种绘制软件蓝图的标准语言。可以用 UML 对软件密集型系统的制品进行可视化详述和文档化。UML 是一种定义良好、易于表达、功能强大且普遍适用的可视化建模语言。它融入了软件工程领域的新思想、新方法和新技术。它的作用域不限于支持面向对象的分析与设计，还支持从需求分析开始的软件开发的全过程。UML 的作用就是用很多图从静态和动态方面来全面描述我们将要开发的系统。

19 **软件开发生命期**

软件开发生命期是指软件产品从考虑其概念开始，到该产品交付使用为止的整个时期。

一般包括概念阶段，需求阶段，设计阶段，实现阶段，测试阶段，安装阶段及交付阶段。这些阶段可以有重复，执行时也可以有迭代。

1、 21 下列描述中，哪个不是**建模的基本原则**（）

A.要仔细的选择模型

B.每一种模型可以在不同的精度级别上表示所要开发的系统

C.模型要与现实相联系

D.对一个重要的系统用一个模型就可以充分描述

2、 **UML 体系包括三个部分**：UML 基本构造块，（）和 UML 公共机制

A、UML 规则 B、UML 命名

C、UML 模型 D、UML 约束

3、 UML 中的**事物**包括：结构事物，分组事物，注释事物和（）

A、实体事物 B、边界事物

C、控制事物 D、动作事物

4、 （）模型的**缺点**是缺乏灵活性，特别是无法解决软件需求不明确或不准确的问题

A、瀑布模型 B、原型模型

C、增量模型 D、螺旋模型

5 下面哪个不是 UML 中的**静态视图**（）

A.状态图 B.用例图

C.对象图 D.类图

6（）技术是将一个活动图中的活动状态进行分组，每一组表示一个特定的类、人或部门，他们负责完成组内的活动。

- A、泳道 B、分叉汇合
- C、分支 D、转移

7 下列关于状态图的说法中，正确的是（ ）

- A. 状态图是 UML 中对系统的静态方面进行建模的五种图之一。
- B. 状态图是活动图的一个特例，状态图中的多数状态是活动状态
- C. 活动图和状态图是对一个对象的生命周期进行建模，描述对象随时间变化的行为。
- D. 状态图强调对有几个对象参与的活动过程建模，而活动图更强调对单个反应型对象建模

8 对**反应型对象**建模一般使用（ ）图

- A、状态图 B、顺序图
- C、活动图 D、类图

9 **类图应该画在 Rose** 的哪种（ ）视图中

- A、Use Case View B、Logic View
- C、Component View D、Deployment View

10 **类**通常可以分为实体类，（ ）和边界类

- A、父类 B、子类
- C、控制类 D、祖先类

11 **顺序图**由类角色，生命线，激活期和（ ）组成

- A、关系 B、消息
- C、用例 D、实体

12 () 是系统中遵从一组接口且提供实现的一个物理部件，通常指开发和运行时类的物理实现

- A、部署图
- B、类
- C、接口
- D、组件

13 关于**协作图**的描述，下列哪个**不正确** ()

- A.协作图作为一种交互图，强调的是参加交互的对象的组织；
- B.协作图是顺序图的一种特例
- C.协作图中有消息流的顺序号；
- D.在 ROSE 工具中，协作图可在顺序图的基础上按“F5”键自动生成；

14 **组件图**用于对系统的静态实现视图建模，这种视图主要支持系统部件的配置管理，通常可以分为四种方式来完成，下面哪种**不是**其中之一 ()

- A.对源代码建模
- B.对事物建模
- C.对物理数据库建模
- D.对可适应的系统建模

15 一个对象和另一个对象之间，通过消息来进行**通信**。消息通信在面向对象的语言中即 ()

- A、方法实现
- B、方法嵌套
- C、方法调用
- D、方法定义

16 () 是可复用的，提供明确接口完成特定功能的程序代码块。

- A、模块 B、函数
C、用例 D、软件构件

17 若将**活动状态**比作方法，那么**动作状态**即（ ）

- A、方法名 B、方法返回值
C、方法体中的每一条语句 D、方法的可见性

18 **事件**可以分为内部事件和外部事件。按下按钮和打印机的中断是（ ）事件

- A、内部事件 B、外部事件

19（ ）是用于把元素组织成组的通用机制

- A、包 B、类
C、接口 D、组件

20（ ）是一组用于描述类或组件的一个服务的操作

- A、包 B、节点
C、接口 D、组件

21（ ）是被**节点执行的事物**

- A、包 B、组件
C、接口 D、节点

1、 UML 中，对象行为是通过**交互**来实现的，是对象间为完成某一目的而进行的一系列消息交换。消息序列可用两种类来表示，分别是（ ）

- A.状态图和顺序图 B.活动图和协作图

C.状态图和活动图

D.顺序图和协作图

2、 **用例** (Use-case) 用来描述系统在事件做出响应时所采取的行动。用例之间是具有相关性的。在一个“订单输入子系统”中，创建新订单和更新订单都需要检查用户帐号是否正确。那么，用例“创建新订单”、“更新订单”与用例“检查用户帐号”之间是 () 关系

A.包含 (include)

B.扩展 (extend)

C.分类 (classification)

D.聚集 (aggregation)

3、 **瀑布模型**的**生存周期**是 ()

A. 听取客户意见→建造/修改→测试/运行

B. 计划→听取客户意见→设计→编码→测试→运行/维护

C. 计划→需求分析→设计→编码→测试→运行/维护

D. 需求分析→计划→设计→编码→测试→运行/维护

4 下列关于**状态图**的说法中，正确的是 ()

A. 状态图是 UML 中对系统的静态方面进行建模的五种图之一。

B. 状态图是活动图的一个特例，状态图中的多数状态是活动状态

C. 活动图和状态图是对一个对象的生命周期进行建模，描述对象随时间变化的行为。

D. 状态图强调对有几个对象参与的活动过程建模，而活动图更强调对单个反应型对象建模

5 () 是系统中遵从一组接口且提供实现的一个物理部件，通常指开发和运行时类的物理实现

A、部署图

B、类

C、接口

D、组件

6 **事件** (event) 表示对一个在时间和空间上占据一定位置的有意义的事情的规格说明, 下面哪个不是事件的类型 ()

A. 信号

B. 调用事件

C. 源事件

D. 时间事件

7 下面关于**正向工程与逆向工程**的描述, 哪个不正确 ()

A. 正向工程是通过到实现语言的映射而把模型转换为代码的过程;

B. 逆向工程是通过从特定实现语言的映射而把代码转换为模型的过程;

C. 正向工程是通过从特定实现语言的映射而把代码转换为模型的过程;

D. 正向工程与逆向工程可以通过 rose 支持来实现;

1、组成 UML 有三种基本的**建筑块**是: () , 事物和图

A、关系

B、类

C、用例

D、实体