**第一章**

**1、封装：**是指把对象的**属性和操作**结合在一起，组成一个独立的对象。是一种2、**信息隐蔽**技术，目的是使对象的生产者和使用者分离，使对象的定义和实现分开。

**3、继承：**使子类可以自动地拥有（复制）父类的全部属性和操作。

**4、多态性：**使得在多个类中能够定义同一个操作或属性名，并在每一个类中有不同的实现的一种方法。

5、软件分类：

**功能：**系统软件、支撑软件、应用软件。

**规模：**微型软件、小型软件、中型软件、大型甚至超大型软件。

**工作方式：**实时软件、分时软件、交互式软件、批处理软件。

**服务对象：**项目软件、产品软件。

**使用频率：**一次性使用软件、使用频率较高的软件。

**影响程度：**一般性软件、关键性软件。

**6、软件的生命周期**：软件定义、软件开发、软件的使用与维护。

7、软件**开发模型**：瀑布模型、渐增模型、演化模型、螺旋模型、智能模型。

8、系统建模的**三要素**：方法、工具和过程

**9、软件危机产生的原因**主要有两个：一是与软件本身的特点相关；二是软件开发和维护的方法不正确。

**10、UML的特点、优点**：唯一性、连续性、维护性、复用性和逐步完善

**11、面向对象的三大重要特征**：封装性、继承性和多态性

12、软件开发方法**从结构化开发方法、模块化开发方法到面向对象开发方法**是一个渐进的演变过程

13、软件生命周期描述了一个软件**从定义、开发、使用、维护到服用**的全过程

14、面向对象的**基本概念**有：对象、类急气封装性、多态性、继承性和消息传递

15、软件开发过程由**客户端需求分析、系统分析、系统设计和系统实现以测试与维护**四个阶段组成

16、面向对象系统的开发过程**以体系结构为中心，以用例为驱动，是一个反复、渐增的过程**

17、面向对象技术采用**以类为中心的封装、继承、多态等不仅支持软件复用，而且使软件维护工作可靠有效，可实现软件系统的柔性制造。**

**第二章**

1、UML的五种视图：用例视图、逻辑视图、构件视图、进程视图和配置视图

2、UML的三大类模型图是：用例模型图、静态模型图和动态模型图

3、用例模型描述的是外部执行者主要用于需求分析阶段

4、UML的静态建模机制包括：类图、对象图、包图、构件图、配置图

5、UML的动态模型包括4种图：状态图、活动图、顺序图、合作图

6、软件的开发过程即生命周期划分为开始、详细规划、系统构造、移交四个阶段。

7、UML开发过程中的核心活动成分是：分析、设计、实现、测试、配置和一些核心支持活动。

8、UML开发过程的产物包括两大类：模型和文档

9、UML软件开发过程的基本特点：用例驱动系统、以体系结构为中心、螺旋上升式的开发过程、以质量控制和风险管理为目标

10、UML中的扩展机制包括三种：构造型、标记值和约束。构造型用于对模型元素进行分类，在已有的基本模型元素上定义新的模型元素。标记值也称特性规格说明，他和约束一起直接对摸个模型元素附加一些特性和语义。

11、软件项目开发过程包括的具体工作内容是：业务建模、需求分析、设计、实现和测试。

12、UML软件开发过程的基本特征是：以用力驱动软件开发全过程，以系统体系结构为中心，以质量控制和风险管理为目标，采用反复迭代、循环、渐增是的螺旋上升式开发过程。

13、UML以用例对象为中心，以系统体系结构为主线，采用循环、迭代、渐增的方式进行开发。

14、UML的最终产物就是最后提交的可执行文件的软件系统和相应的软件文档资料

15、UML的需求分析建模中，用例模型图必须与用户反复交流并加以确认。

16、UML分析和设计模型由三类模型图表示，三类模型图是：用例模型图、静态模型图和动态模型图。

17、UML的软件统一开发过程，即生命周期按时间顺序可以划分为，开始，详细设计，系统构造和移交四个阶段及阶段中一系列的循环重复。

18、UML开发过程是一种二维结构软件开发过程，软件项目开发过程流程包括的核心工作内容是，分析，设计，实现，测试和配置。

19、UML中的五个不同的视图可以完整地描述出所建造的系统，这五种视图是用例视图、逻辑视图、构件视图、进程视图和配置视图。

20、UML中有10种基本图可以完整地描述出所有建造的系统，这10种视图是用例图、类图、对象图、包图、构件图、配置图、序列图、活动图、状态图和合作图。

**第三章**

1、可行性研究：经济可行性、技术可行性和法律可行性。

2、需求分析的目的是深入描述软件功能和性能，确定软件设计的约束和软件同其他系统元素的接口细节，定义软件的其他有效性需求。

3、用例图包含的模型元素有系统、执行者、用例以及他们间的不同关系，如继承、关联、依赖等。

4、执行者是指在系统外部与系统交互的人或其他系统

5、执行者分为“人”执行者和“外部系统”执行者两类。

6、UML中用例间的关联主要有4种：继承关联、扩展关联、包含关联和使用关联

7、UML的活动图示系统动态行为建模的图形工具之一，用来表示完成一个操作所需要的活动，或者是一个用例实例的活动。活动图实际上也是一种流程图，它描述活动的序列，即系统从一个活动到另一活动的控制流。活动图特别适合秒速动作流和并发处理行为。

8、UML的客户需求分析模型包括用例模型、类图、对象图和活动图组成。

9、UML客户需求分析使用的CRC卡上责任一栏的内容主要描述类的属性和操作

10、UML客户需求分析产生的用例模型描述了系统的功能要求

11、在UML的需求分析建模中，用例模型必须与用户反复交流并加以确认。

12、在UML的需求分析建模中，对用例模型中的用例进行细化说明应使用活动图

13、活动图中的分劈和同步接合图符是用来描述多进程的并发处理行为

14、UML软件开发过程需求分析阶段产生的模型由三类模型图表示。它们是：用例模型图、静态模型图和动态模型图。

15、CRC卡中的描述由类名、类特征、类类型、责任和协作者共五部分组成。

16、软件项目的目的的可行性研究分析中，技术可行性研究包括风险分析、资源分析、技术分析三部分组成。

17、在UML软件开发过程的需求分析阶段，建立用例模型的步骤分为，确定系统的范围和边界，确定系统的执行者和用例，对用例进行描述，定义用例之间的关系和审核用例模型。

18、用例图中以实线方框表示系统的范围和边界，在系统边界内描述的是用例，

在边界之外描述的是执行者。

19、用例模型中的执行者可以是“人”执行者也可以是“外部”系统执行者。

用例模型中的用例之间的关联有使用关联、扩展关联。包含关联和继承关联。

**第四章**

1、根据建立的用户需求模型，在系统分析阶段要进一步确立三个模型系统模型：对象静态模型，对象动态模型和系统功能模型。

2、类之间的关系有关联、聚集、继承、依赖、细化等。

3、包是UML的模型元素之一，包可以包含其他包和类。包之间可以有关系，入依赖等。

4、包一种分组机制，它把一些模型元素组织成语义上相关的组，包中拥有或涉及的所有模型元素叫做包的内容。

5、UML的客户需求分析、系统分析和系统设计阶段产生的模型，其描述图符完全相同。

6、类和对象都有属性，他们的差别是：类描述了属性的类型，而对象的属性必须有具体值。

7、UML系统分析阶段产生的包图描述了系统的系统体系层次结构。

8、设计模式在面向对象系统设计中式设计方法的一种形式

9、“对象容器”设计模式对有限的对象进行管理，它不能创建对象

10、在UML软件开发过程系统分析阶段产生的对象模型有三种模型。他们是：对象的静态模型，对象的动态模型和对象的系统功能处理模型。

11、共享聚集的部分对象可以是任何整体的一部分，表示事物的整体/部分关系较的情况，整体段的重数应该是n。

12、在UML软件开发过程的需求分析和系统分析阶段，建立对象类模型的步骤分为寻找确定对象类、定义接口、定义类之间的关系、建立对象类图和建立系统包图。

13、组合聚集是指整体拥有它的部分，他具有抢的物主身份，表示事物的整体/部分关系较强的情况。部分生存在整体中，不可分离他们与整体一起存在或消亡。整体的充数必须是1或0。

14、系统分析是在客户需求分析规格说明的基础之上对其进行的分析

15、类有实例，它的实例是一个对象。

16、在UML中，包用来表示一个模型组织的分组机制，包没有实例。

**第五章**

1、消息分为四种控制流，分别是简单消息、同步消息、一步消息、和返回消息。

2、顺序图用来描述对象间的交换行为。他注重消息的时间顺序，即对象间消息的发送和接收的顺序。顺序图还揭示了一个特定场景的交互，即系统执行期间发生在某个时间点的对象之间的特定交互，他适合描述实时系统中的时间特性和时间约束。

3、合作图和顺序图都可用来描述系统对象间的交互。顺序图强调的是一组对象间的操作调用的时间顺序，合作图则强调这组对象之间的关系。

4、UML系统设计的一般步骤包括系统对象设计、系统体系结构设计和系统设计的优化

5、顺序图和合作图主要用与对用例图中控制流的建模，用他们来描述用例图的行为。

6、顺序图的模型元素有对象、消息、链接等，这些模型元素表示某个用例中的若干个对象和对象之间所有传递的消息，来对系统的行为建模。

7、顺序图描述一组对象之间消息的传递顺序

8、顺序图和合作图建立了UML面向对象开发过程中的对象动态交互模型

9、在UML软件开发过程产生的对象动态模型中消息有四种类型，他们是简单消息，同步消息、异步消息和返回消息。

10、顺序图和合作图用来表达对象之间的交互，是描述一组对象如何合作完成某个行为的模型化工具进程是一个动作流，能够与其它进程并发执行

11、线程是内部的一个动作流，能够与其他线程并发执行

12、主动对象是一个拥有进程或线程的对象，能初始化控制活动，可以独立并发运行

13、被动对象是一个必须由其他对象发来的消息进行触发才执行动作的对象。

14、交互图描述系统中对象间的交互行为。每一个交互都有发送者和接受者，他们可以是整个系统、一个子系统、一个用例、一个对象类或一个操作。

**第六章**

1、UML软件开发过程中产生的状态图描述了系统中一个对象所具有的各种状态和这个对象从一中状态到另一状态的转换，以及影响对象这些状态的事件等，它主要用来描述某个对象从一个状态到另一个状态编号迁移的控制流

2、活动中的事件有entry事件、exit事件、do事件

3、状态的改变---迁移

4、状态图可以表现一个对象在生存期的行为、所经历的状态序列、引起状态转移的事件以及因状态转移引起的动作。

5、状态图描述一个对象在不同事件的驱动下发生的状态迁移。

6、一个同步并发迁移图符可以有多个源状态或目标状态，他们可以把一个控制分解为并行运行的并发线程，或将多个并发线程结合成单个线程

7、活动图中动作状态之间的迁移不是靠事件触发的，

8、状态图和活动图建立了UML面向对象开发过程中的对象动态状态模型

9、顺序状态表明状态之间的迁移时串行的，即一个接一个顺序迁移

10、状态图还可以用复合迁移的同步并发迁移图符来表示并发子状态

11、状态图中的进程是个动作流，能够与其它进程并发执行

12、在活动图中，虚箭头线也成为对象流，对象流表示动作状态或活动状态与对象之间的关联

13、活动图即可以描述对象的动态行为，还可以用来描述用例

14、活动图中活动状态的迁移不是由事件按进行触发，一个活动执行完毕自动进入下一个多栋状态

15、状态图和活动图描述系统中某个系统对象的一系列状态变化。

**第七章**

1、在UML中，软件构件分为源代码构件、二进制构件和可执行代码构件图由这些构件、接口以及构件之间的关系组成。

2、系统体系结构式用来描述系统各部分的结构、结构以及他们用于通信的一种机制。

3、UML可以描述硬件之间的互联关系，也能描述硬件单元上的软件系统的分布

4、软件体系结构是对系统的用例、类、对象、结构以及相互间的交互和协作进行描述

5、硬件系统体系结构要对系统的构件、结点的配置进行描述。

6、构件时软件系统体系结构中定义的概念和功能在物理体系结构中的实现。

7、配置图由结点和结点之间的联系组成，描述了处理器、设备和软件构件欲行时的体系结构。

8、配置图的基本元素有结点、构件、对象、连接、依赖等。

9、系统体系结构建模可以分为软件系统体系结构建模和硬件系统体系结构建模

10、构件时软件系统体系结构类、对象、他们间的关系和协作种定义的概念和功能在物理体系结构中的实现

11、软件构件分为源代码构件、二进制构件和可执行代码构件

12、构件图主要用于建立系统的静态模型

13、构件图中的构件没有实例，只要在配置图中才能标识构件的实例

14、配置图由结点和结点之间的联系组成，描述了处理器、设备和软件构件运行时的体系结构。

15、结点之间、结点与构件之间的联系包括通信关联、依赖联系等。

**第八章**

1、设计模式的分类：按照模式的目的将其分为创建型、结构性和行为型

设计模式独立于具体的编程语言

2、设计模式是面向对象软件工程中的一个重要概念，是由软件模式分支中衍生出来的一个解决具体问题的重要方案之一。

3、4.4.5节介绍的“对象集合管理器”模式就是本章介绍的简单工厂

单例模式属于对象创建型，它保证一个类仅有一个实例

4、在面向对象设计中，设计模式是系统可复用的基础，正确地使用设计模式，有助于快速开发出可复用的系统。

5、设计模式就是对特定问题的描述或解决方案，往往直接对应一段程序代码。

6、简单一点讲，模式就是解决特定问题的经验，实质上就是软件的复用。

7、软件模式基本形成了概念模式、设计模式和编程模式三个重要的类别。

8、工厂模式有三种形态：简单工厂模式，抽成工厂模式和工厂方法模式

9、一个设计模式有四个基本要素：名称、问题、解决方案、和效果

10、设计模式按照模式的目的将其分为创建型、设计和编程者三种类型的设计模式分别描述了对象在创建、组合以及相互作用的过程中如何降低他们之间的耦合性、提高复用性的种种成功方案。

11、设计模式的作用和研究意义表现在：优化是设计经验、极高的复用性、丰富的表达能力和极低的耦合度

12、创建型模式就是描述怎么样创建一个对象，他隐藏了对象创建的具体的细节，是程序代码不依赖具体的对象

13、结构型模式处理类或对象的组合，即描述累和对象之间怎样组织起来形成更大的结构从而实现新的功能。

14、行为型设计模式描述算法以及对象之间的任务（职责）分配，他所描述的不仅仅是类或对象的设计模式，还有他们之间的设计模式。