

关于UML

老师上课说的:	1
第一章: UML概述	1
第二章: 面向对象技术和建模基础	3
第三章: UML建模工具简介	4
第四章: 用例和用例图	4
第五章: 类图和对象图	6
第六章: 顺序图和通信图	7
第七章: 状态机图和活动图	10
第八章: 组件图和部署图	12
第九章: 包图、组合结构图、定时图、交互概览图和概要图	12
复习策略与答题技巧 (ChatGPT-5.1旗舰模型)	13
对比记忆	13

老师上课说的:

选择, 简答, 分析, 设计

分析: 各种图 (6种)

上课讲的基本上都涉及, 参考教材课件

p2前面配套资源下载, 各种文件

课件看课堂的

第一张: 基本概念: 发展, 组成, 特点, 关系

第二张: 面向对象 (概念, 特点, 封装继承表达, 基本阶段, 方法, p28建模, 概念, 优点), 建模基础

第三张: 基本...

第四章: 重点内容 (不能错类型!) 使用...

第五章: 重点内容, 类图 (元素, 关系...稍微复杂一点), 概念类图, 区分什么时候使用...

第六章: 顺序图, 通信图

第七章: 状态图; 活动图 (基本, 带泳道)

第八章: 组建图 (软件), 部署图 (物理节点), 了解基本概念, 组成, 作用和用途即可

第九章: 包图, 后面不用设计要了解概念, 概要图不看

(以下黑色内容由DeepSeek生成)

(以下蓝色内容由我自己添加)

(以下红色内容为书后的答案, 删掉了我觉得没意思的东西)

第一章: UML概述

知识点:

1. 软件工程发展: 从“软件危机”到结构化方法, 再到面向对象方法的演变。
2. 面向对象方法的特点:

- **封装**: 隐藏对象内部细节, 仅通过公共接口访问。
 - **继承**: 子类自动拥有父类的属性和方法, 实现代码复用和层次分类。
 - **多态**: 同一操作作用于不同对象产生不同行为 (覆盖/重载) 。
 - **抽象**: 忽略非本质细节, 专注于关键特征。

3. UML的定义与组成：统一建模语言，是一种可视化的标准建模语言。

4. UML的视图与模型图: 2+4+1模型 (2种静态结构、4种动态行为、1种物理架构)。

5. 面向对象分析与设计 (OOA/OOD) 的关系:

- **OOA**: 理解问题域, 识别对象和类, 建立概念模型 (做什么)。
 - **OOD**: 将分析模型转化为系统设计模型, 考虑实现技术 (怎么做)。

6. UML 的事物

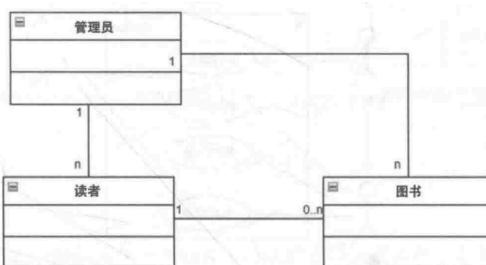
- 1) 构建事物: 静态部分, 描述概念、物理元素 (类、接口、协作、用例、构建、节点)
 - 2) 行为事物: 动态部分, 描述跨越空间事件的行为 (交互、状态机)
 - 3) 分组事物: 组织部分, 描述事物组织结构 (包: 把元素编制成组的机制)
 - 4) 注释事物: 解释部分 (注释: 对元素进行约束、简单解释)

7. UML的关系

1) 依赖 (Dependency) : 两个模型元素之间的语义关系

(独立元素) - - - - - → (依赖元素)

2) 关联 (Association) : 一个对象和另一个对象之间的关系



3) 泛化 (Generalization) : 一般化-特殊化的关系

(子类) —————→ (父类)

4) 实现 (Realization) : 类之间的语义关系

(实现) -----> 接口

8. 图的分类

- 1) 构建结构图: 包图、类图、对象图、组合结构图、组建图、部署图、概要图
 - 2) 行为建模图: 用例图、活动图、状态机图、通信图、顺序图、定时图、交互概览图

系统开发阶段: 单元测试->集成测试->系统测试->验收测试

教材后面的练习及答案：

1. UML事物有哪些？

答：UML的事物包括：构建事物、行为事物、分组事物和注释事物。

2. UML 关系有哪些？

答：UML的关系包括：依赖、关联、泛化和实现。

3. UML图有哪些？其中哪些是静态图？哪些是动态图？

答：UML的图包括：用例图、类图、对象图、状态机图、活动图、顺序图、通信图、构件图、部署图、包图、组合结构图、定时图、交互概览图和概要图。

4. 简述什么是UML？

答：UML是一种可视化的统一过程建模语言，用于面向对象的分析与设计建模，它遵循特定的规则，允许创建各种模型，与开发过程无关。

第二章：面向对象技术和建模基础

知识点：

1. 核心概念：

- 对象：具有状态、行为和标识的实体。
- 类：一组具有相同属性和方法的对象的抽象。
- 接口：声明了一组操作，但不包含实现。

2. UML的三大要素：事物、关系、图。

3. 建模的基本原则：抽象、分解、模块化、复用。

4. 面向对象方法的优点：更贴近人类思维，提高可维护性、可复用性和可扩展性。

5. 面向对象方法的常见概念

1) 类：图匪可u才嘎吧的方法和数据描述一组对象共同行为和属性

2) 类（静态）；对象（动态）

3) 封装：把对象的全部属性、方法结合在一起；信息隐蔽，隐藏内部实现细节，提供外部接口。`public`（公有）、`protected`（受保护的，同一个包、子类可以访问）、`private`（私有，对象内部访问）、`default`（默认，当前目录下可访问）

4) 封装的优点：方便类和对象操作，降低错误修改属性的可能性；提高松散耦合关系、提高系统独立性；提高程序复用性；针对大型开发系统降低开发风险。

5) 继承：新类继承状态、行为（一般类、特殊类的层次模型）（冯伟单重继承、多重继承）

6) 多态性：方法重载（静态多态）、方法重写（动态多态）

教材后面的练习及答案：

1. 什么是对象，具有什么特征？

答：对象是面向对象的基本构造单元，是系统中用来描述客观事物的一个实体，一个对象由一组属性和对属性进行操作的一组方法组成。

2. 简述面向对象的开发过程。

答：面向对象的开发过程主要包括四个阶段：面向对象的需求分析OOA、面向对象的系统设计OOD，面向对象的程序实现OOP和面向对象的系统测试OOT。

3. UML在面向对象开发过程中起到的作用是什么？

答：UML在面向对象开发过程中起到的作用是多方面的，它不仅提供了一种标准化的建模语言，还支持面向对象的主要概念，如类、对象、继承、多态性和关系等。UML的使用有助于软件开发人员更好地理解、设计和实现应用程序，从而提高软件开发的效率和质量。

4. 什么是建模，有什么优点？

答：建模是创建或模仿系统、概念或项目的模型的过程，采用可视化的精确的符号，抽象系统的不同视图。软件建模实质上是捕捉系统本质的过程，把问题领域转移到解决领域的过程。其优点在

于：1) 使用模型，便于从整体上、宏观上把握问题，以便更好地解决问题；2) 软件建模可以加强软件工作人员之间的沟通，便于提早发现问题；3) 模型可为代码生成提供依据，帮助系统可视化；4) 模型可进行详细说明，便于在构造系统时决策的文档化管理。

5. 什么是面向对象技术？

答：面向对象技术是一种以对象为核心，采用人类在认识客观世界的过程中普遍运用的思维方法，直观、自然地描述客观世界中的有关事物来构造软件系统的方法。

第三章：UML建模工具简介

没什么好考的

第四章：用例和用例图

知识点：

1. 作用：从用户角度描述系统功能需求，界定系统边界。

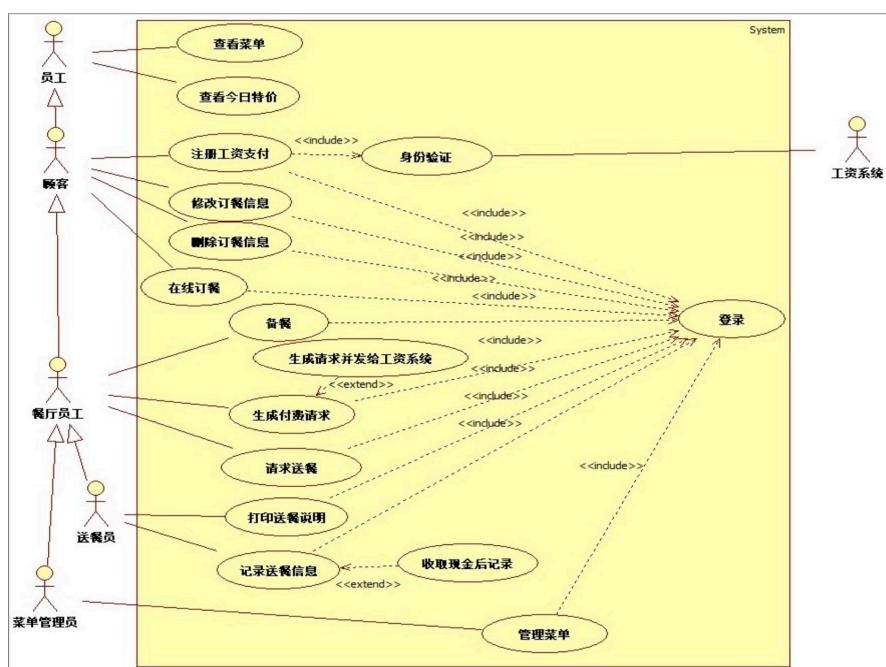
2. 核心元素：

- 参与者：系统外部与系统交互的角色（人、其他系统）。
- 用例：系统为参与者提供的一个有价值的功能单元。
- 系统边界：矩形框，内是用例，外是参与者。

3. 关系：

- 关联：参与者和用例之间的通信。
- 包含：<include>，基础用例必须执行被包含用例（如：登录包含验证密码）。
- 扩展：<extend>，基础用例可能在特定条件下执行扩展用例（如：下单可能扩展使用优惠券）。
- 泛化：参与者之间或用例之间的“父子”继承关系。

4. 分析设计题关键：准确识别参与者和用例；正确使用包含与扩展（区分“必须”和“可能”）。



5. 用例的定义

1) 用例是对一个参与者 (actor) 使用系统的一项功能时所进行的交互过程的一个文字描述序列。

2) 用例是系统、子系统或类和外部的参与者 (actor) 交互的动作序列说明, 包括可选的动作序列和会出现异常的动作序列。

6. 用例的本质: 用例是对一组动作的描述, 描述参与者可以感受到的系统功能和服务。

7. 用例图的基本元素: 参与者、用例、系统边界、关系

8. 用例描述可以包含的内容: 用例编号、用例名称、用例概述、范围、主参与者、次要参与者、项目相关人利益说明、前置条件 (用例启动时, 参与者应该处于什么状态?) 、后置条件 (用例结束时系统应置于什么状态?) 、成功保证、基本事件流 (常规、预期路径的描述) 、扩展事件流 (异常情况、选择分支) 。

9. 建立用例图的步骤

1) 识别参与者

2) 识别用例

教材后面的练习及答案:

1. 什么是参与者? 如何确定系统的参与者?

答: 参与者是系统外部的一个人或者物, 它以某种方式参与了系统的执行过程。参与者不是特指人, 是指系统以外的, 在使用系统或系统交互中所扮演的角色, 可以是人, 也可以是事物, 或是其他系统。

2. 什么是用例? 如何确定系统的用例?

答: 用例指系统外部参与者可以感受到的系统的服务或者功能。

3. 用例之间有哪些关系? 对每一种关系, 请举出一个实际的例子, 并画出用例图。

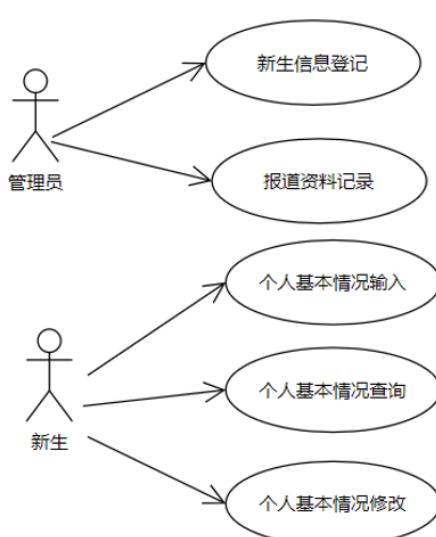
答: 用例之间的关系主要包括: 包含关系、扩展关系、泛化关系和精化关系。包含关系例子: 删除图书和修改图书包含查询图书; 扩展关系例子: 图书管理系统中, 交罚款是还书的扩展用例; 泛化关系例子: 精确查询图书和精确查找图书; 精化关系: 交话费和手机充话费。图略。

4. 试画出学生选课系统的用例图。

答: 图略, 建议先对学生选课系统进行需求分析, 参与者主要包括管理员、学生和教师, 学生主要使用选课用例, 管理员使用学生信息维护、课程信息维护等用例, 教师可查看课程信息和学生信息等。

5. 学生管理系统中有一个模块是报到登记, 具体流程是: 在新生入校报到时, 进行新生信息登记, 记录学生的报到资料、个人基本情况的输入、查询、修改等。

问题: 写出在上述需求描述中出现的Actor, 并根据上述描述绘制其用例图。

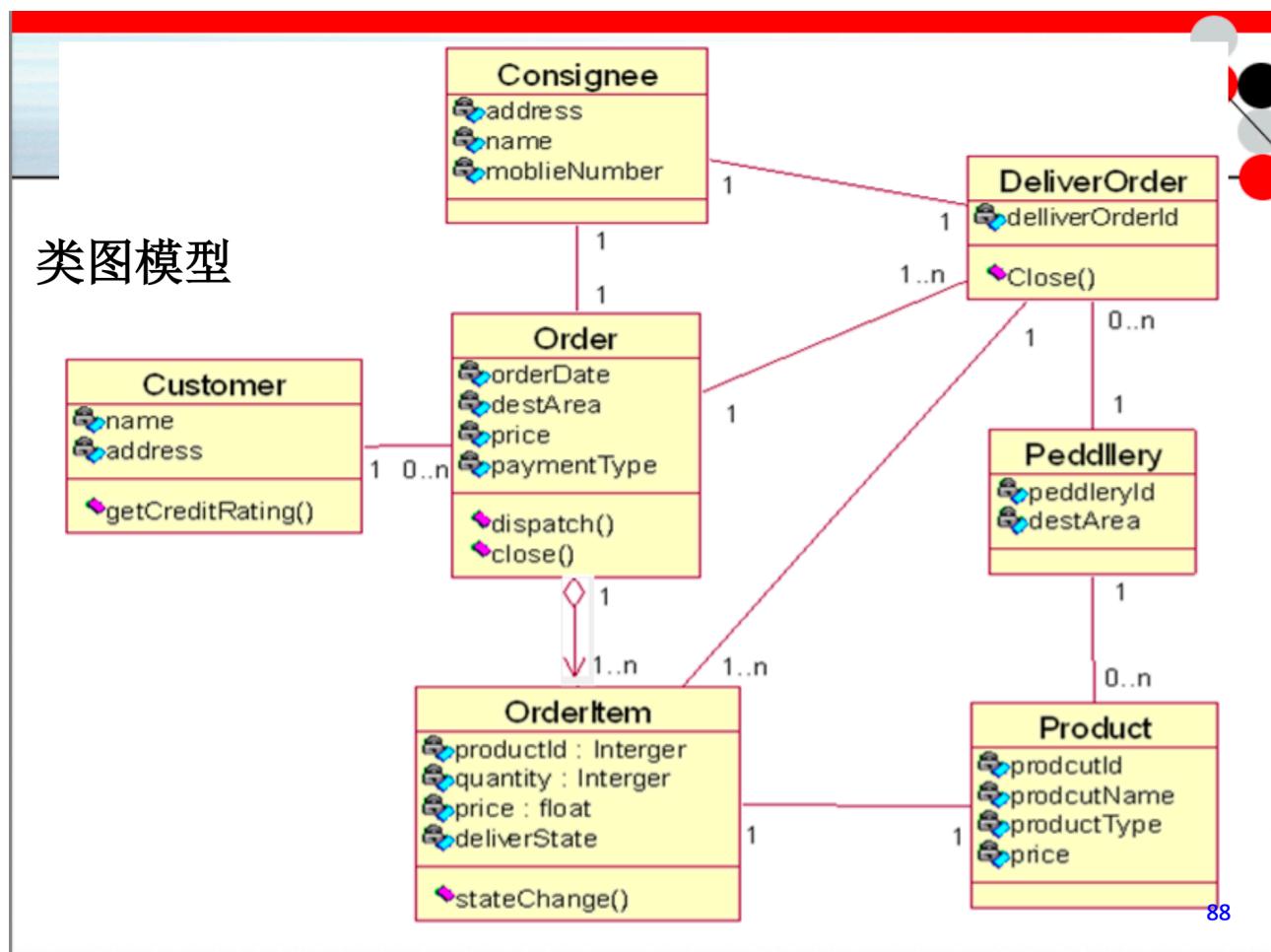


答：出现的执行者有：管理员和新生，用例图如下所示，需要说明的是：新生信息登记仅实现学号分配和院系、班级等的录入，其他信息都可由学生在个人基本情况输入中填写。

第五章：类图和对象图

知识点：

- 作用：描述系统的静态结构，展示类、属性、方法及类之间的关系。
- 类的表示：三层矩形（类名、属性、方法）。可见性：+(public), -(private), #(protected)



3. 关系的类型与辨析（绝对重点！）：

关系	语义	UML表示	代码体现	辨析关键
依赖	一个类的变化可能影响另一个类	虚线箭头	局部变量、参数、静态方法调用	使用关系，临时性、弱
关联	类之间有长期、结构性的连接	实线（可带箭头）	成员变量（引用）	结构性、长期
聚合	整体-部分，部分可独立于整体存在	空心菱形实线	成员变量	has-a, 生命周期独立

关系	语义	UML表示	代码体现	辨析关键
组合	更强整体-部分，部分与整体共存亡	实心菱形实线	成员变量（且负责创建/销毁）	contains-a, 生命周期一致
泛化	一般-特殊，继承关系	实线空心三角箭头	extends (Java)	is-a
实现	类实现接口	虚线空心三角箭头	implements	实现契约

4. **分析设计题关键**: 根据语义描述判断关系类型; 绘制符合逻辑的类图; 区分概念类图 (分析阶段) 和设计类图 (设计阶段) 。
5. **类图概要**: 表示系统中类和类之间的关系, 系统静态结构的描述
6. **属性的含义**: 描述类所表示的事物的静态性质
7. **属性的格式**: [可见性#-+]属性名[: 类型][多重性][初始值][{特性}]
8. **操作的含义**: 对类的对象所能做的事情的一个抽象, 描述类所表示的事物的动态性质。
9. **操作的格式**: [可见性]操作名[(参数列表): 返回类型]{特性}]

教材后面的练习及答案:

1. 试论述类与用例的区别。

答: 类是对一组具有相同属性、操作、关系和语义的对象的抽象, 其为有形或者无形的实体, 一般以名词形式出现。而用例是系统对外看到的功能, 是具体做什么事情, 一般以动宾形式出现。

2. 类主要由哪几部分组成?

答: 类是由属性和操作组成。

3. 试运用本节所学的静态建模技术找出用户管理模块中的所有的类。

答: 对于学生信息管理系统中的用户来说, 主要有: 用户、用户类型、所属机构等。

4. 请说一说类、抽象类和接口的区别。

答: 接口中只有方法, 没有属性, 且方法为抽象方法, 不做实现; 抽象类中可以有属性, 也可以有方法, 方法中必须有未实现的方法, 但是也可以有有实现的方法; 类中有属性和实现的方法。

5. 什么是依赖? 它与关联有什么区别?

答: 依赖表示两个或多个模型元素之间语义上的关系, 一个元素的改变可能会影响其他元素, 则表示一个元素依赖于另一个元素。

6. 什么是泛化? 泛化是否就是类的继承, 如果不是, 请说明理由。

答: 泛化又称继承, 是一种存在于一般元素和特殊元素之间的分类关系, 表示特殊元素在一般元素的基础上可以增加新的属性或操作。

7. 试论述聚合和组合的异同。

答: 聚合是表示整体与个体之间的关系, 组合是表示整体与部分之间的关系。相比来说, 组合的耦合性较强, 聚合的耦合性则较弱。组合中, 整体和部分之间具有相同的生命周期, 组合中的整体消失后, 个体也不复存在。聚合中整体与个体则有不同的生命周期, 当整体消失后, 个体依然可以存在。

第六章：顺序图和通信图

知识点:

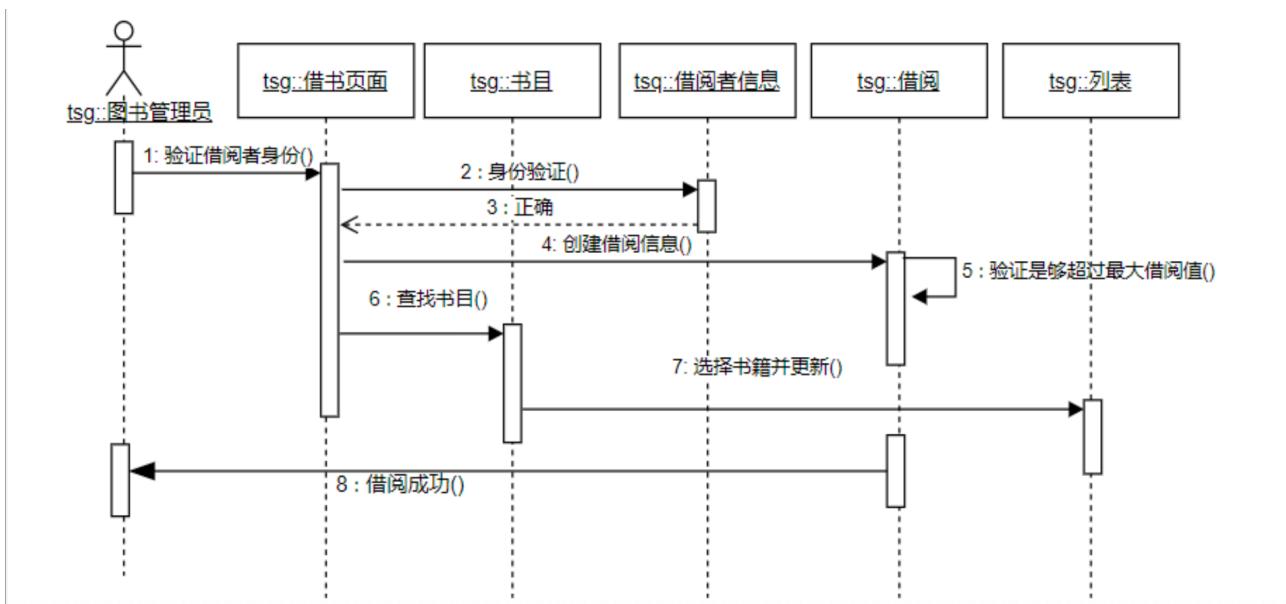
1. **作用**: 强调时间顺序的对象交互, 描述用例或操作的具体执行流程。
2. **核心元素**:

- **对象生命线**: 垂直虚线, 表示对象存在的时间。
- **激活条**: 生命线上的矩形, 表示对象执行操作的时间段。
- **消息**:
 - 同步消息 (实心箭头) : 等待返回。
 - 异步消息 (开放箭头) : 不等待返回。
 - 返回消息 (虚线箭头) : 可省略。

3. 组合片段 (常用于分析题) :

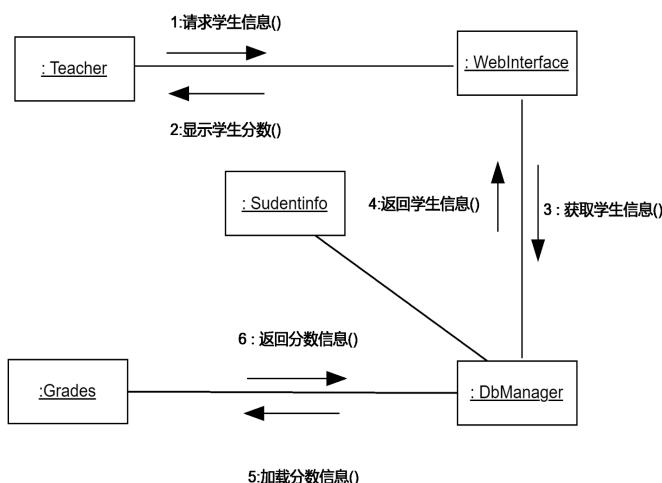
- **alt**: 条件分支 (if...else) 。
- **opt**: 可选片段 (if) 。
- **loop**: 循环片段 (for, while) 。
- **par**: 并行片段。

4. 创建与销毁对象: 消息指向对象本身表示创建, X标记表示销毁。

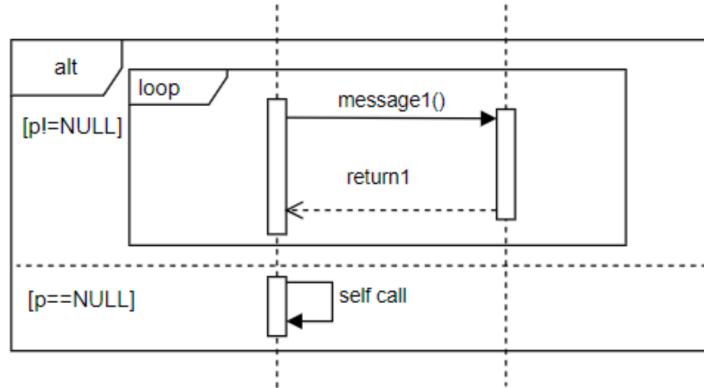


通信图知识点:

- **作用**: 强调对象间结构链接的交互, 与顺序图信息等价, 可互相转换。
- **核心元素**: 对象、链接 (对象间的关联) 、消息 (标有序号, 如1, 1.1, 2) 。
- **与顺序图的主要区别**: 通信图不强调时间顺序, 但清晰展示对象间的连接关系。



1. 交互图的两种形式：顺序图+通信图
2. 浏览顺序图的方法：从上到下查看对象间交换的消息
3. 顺序图（Sequence图）的组成元素：角色actor+对象object+生命线lifeline+激活期（activation）/控制焦点（focus of control）+消息message
4. 顺序图格式：编号[条件]：事件
5. 组合片段：seq（顺序执行）、alt（特定条件选择交互）、opt（相当于switch）等



6. 顺序图、通信图区别：顺序图从实践角度，通信图从空间角度
7. 通信图的作用：描绘对象之间消息传递情况、显示对象机器交互关系、表示一个类操作实现
8. 顺序图和通信图的区别：
 - 1) 侧重点不同，顺序图强调时间顺序交互，通信图强调发送和接收消息的组织结构
 - 2) 顺序图可以反映对象创建、激活、销毁生命周期，通信图没有
 - 3) 通信图反映动作路径，消息必须有顺序号，顺序图没有

教材后面的练习及答案：

1. 顺序图由角色、对象、生命线、激活和（ ）组成。

A. 关系 B. 消息 C. 用例 D. 实体

答：B。

2. 顺序图中消息有几种类型？

答：简单消息、同步消息、异步消息、返回消息。

3. 返回消息是必须的吗？

答：返回消息是可选择的，它依赖建模的具体/抽象程度，一般为了顺序图阅读方便，每个消息都有返回消息。

4. 顺序图中的消息顺序号是必须的吗？

答：不是必须的，顺序图中，生命线代表对象从产生到结束的存活时间，消息调用时按照从上往下的顺序调用，故序号可以有也可以没有。如果序号有，也必须按照从上往下的顺序进行标号。

5. 在顺序图和通信图中，分别应该如何表示“循环”结构？

答：顺序图中，对于循环调用的消息，在对应的消息名称前面添加循环次数标识“*[condition]”即可，如“*[1..10]:Message()”，表示该消息需要循环调用10次。通信图也类似，也是在消息名称的前面加上循环次数标识，表示满足具体条件下，该消息循环调用的次数。

6. 通信图有哪些基本元素组成？各有什么含义？

答：通信图强调参与一个交互对象的组织，它由以下基本元素组成：活动者、对象、连接和消息。

(1) 活动者

活动者（Actor）发出主动操作的对象，负责发送初始消息，启动一个操作。

(2) 对象

对象 (Object) 是类的实例，负责发送和接收消息。一个协作代表了为了完成某个目标而共同工作的一组对象。对象的角色表示一个或一组对象在完成目标的过程中所应该起的作用。通信图中的对象与顺序图中的对象元素概念基本相同，表示方式也相同，只不过没有生命线，而且在通信图中，无法表示对象的创建和撤销，所以对象在通信图中的位置没有限制。

(3) 链接

链接 (Link) 用线条来表示，也称为链，链接表示两个对象共享一个消息，位于对象之间或参与者与对象之间。

表示两个或多个对象间的独立连接，是关联的实例。通信图中，关联角色是与具体语境有关的暂时的类元之间的关系，关系角色的实例也是链。链表示为一个或多个相连的线或弧。

(4) 消息

消息 (Message) 的含义与顺序图中的消息基本类似。在通信图中，不带有消息的通信图标明了交互作用发生的上下文，而不表示交互。它可以用来表示单一操作的上下文，甚至可以表示一个或一组类中所有操作的上下文。如果关联线上标有消息，图形就可以表示一个交互。

7. 顺序图和通信图有什么关联？

答：顺序图和通信图都是动态模型中的交互图，表示对象之间的交互关系，两者语义上是等价的，可以相互转换。区别在于顺序图是一个二维图，强调的是时间序列，从上往下显示消息交互的时间顺序；而通信图强调的是空间，重点显示对象之间的链接及其交互关系。

8. 关于通信图的描述，下列哪个不正确（ ）。

- A. 通信图作为一种交互图，强调的是参加交互的对象的组织；
- B. 通信图是顺序图的一种特例
- C. 通信图中有消息流的顺序号；
- D. 在StarUML工具中，通信图可在顺序图的基础转换生成；

答：B。

9. 一个对象和另一个对象之间，通过消息来进行交互。消息交互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

答：C。

10. 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。

消息序列可用两种图来表示，分别是（ ）。

- A. 状态机图和顺序图
- B. 活动图和通信图
- C. 状态机图和活动图
- D. 顺序图和通信图

答：D。

第七章：状态机图和活动图

【状态图】知识点：

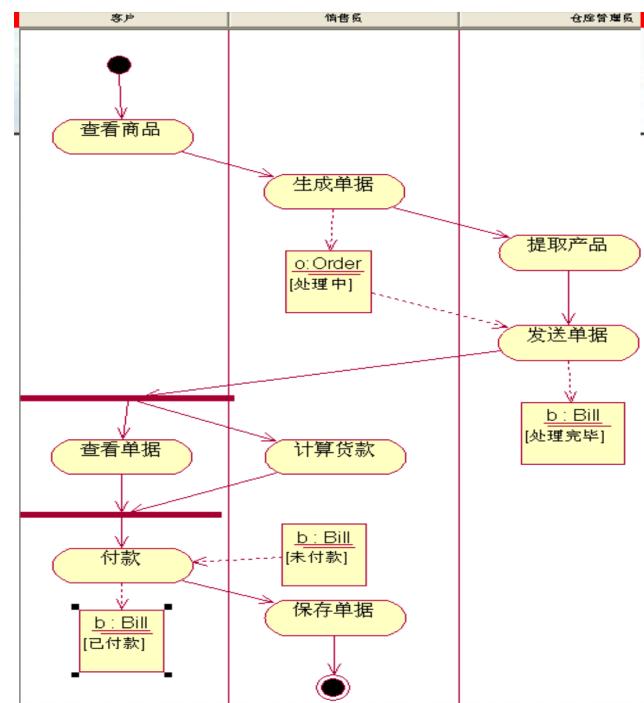
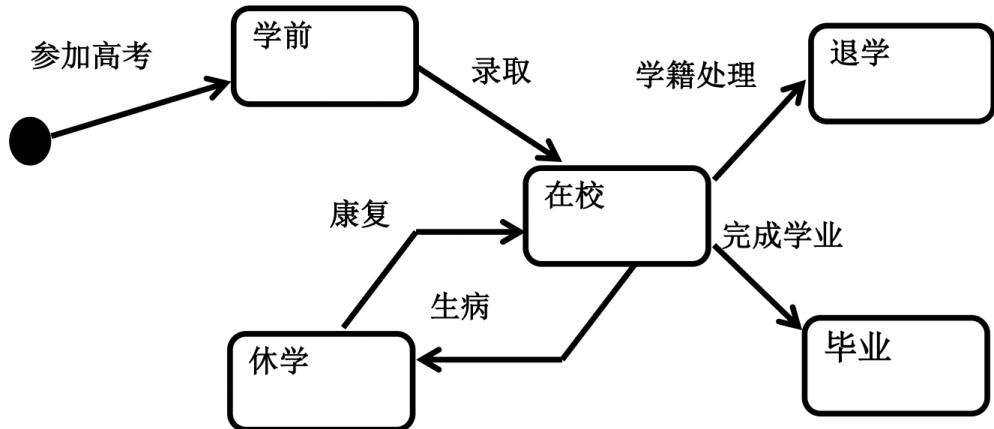
1. 作用：描述一个对象在其生命周期内，响应事件的状态序列及转移。

2. 核心元素：

- 状态：初态（实心圆）、终态（同心圆）、中间状态（圆角矩形）。
- 转换：事件[监护条件]/动作。
- 复合状态与子状态：状态内嵌子状态。
- 历史状态：H，记住退出时的子状态。

【活动图】知识点：

• 状态机图:



1. **作用:** 描述系统或业务的工作流程（活动的顺序和并行），类似流程图但支持并发。
 2. **核心元素:**
 - 动作节点：圆角矩形。
 - 控制节点：初态、终态、决策（菱形）、合并（菱形）、分叉（粗线，开始并行）、汇合（粗线，结束并行）。
 - 对象节点：输入/输出的数据。
 3. **带泳道的活动图:** 将活动按职责（如用户、系统）分配到不同纵向区域，清晰展示任务分工。
1. 状态机的组成：状态、转换、事件、活动、动作
2. 状态机是用于描述状态和状态转换的
3. 状态机用于描述一个对象在生存期间的状态行为，表示对象响应时间所经历的状态序列以及伴随的动作
4. 状态机用于说明对象在其生命周期中响应时间所经历的状态序列以及其对这些事件的响应

5.

教材后面的练习及答案：

1. 下面（ ）不是UML中的静态视图。

- A. 状态机图 B. 用例图 C. 对象图 D. 类图

答：A。

2. （ ）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

答：A。

3. 下列关于状态机图的说法中，正确的是（ ）。

- A. 状态机图是UML中对系统的静态方面进行建模的5种图之一。
B. 状态机图是活动图的一个特例，状态机图中的多数状态是活动状态。
C. 活动图和状态机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. 状态机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。

答：C。

4. 对反应型对象建模一般使用（ ）。

- A. 状态机图 B. 顺序图 C. 活动图 D. 类图

答：A。

5. 请选择UML中合适的图来描述图书管理系统中图书馆业务功能模块。该模块包括借书、还书、预约借书等功能。

答：根据图书管理系统中的借书、还书和预约借书绘制活动图即可。注意电子图书和线下实体图书的管理系统有所不同。以下示例图为线下实体图书馆的借书活动图，对于还书和预约借书，根据实际业务过程仿照绘制即可。

第八章：组件图和部署图

【组件图】知识点：

1. 作用：描述系统软件组件的物理结构及依赖（如可执行文件、库、源代码文件）。

2. 核心元素：

- 组件：带<<component>>的矩形，或带组件图标的矩形。
- 接口：提供接口（“球”），需求接口（“窝”）。
- 依赖关系：组件之间通过接口连接。

【部署图】知识点：

1. 作用：描述系统运行时的硬件节点结构及软件在其上的部署情况。

2. 核心元素：

- 节点：立方体，表示硬件设备或软件运行环境（如服务器、PC、数据库）。
- 工件：带<<artifact>>的矩形，表示具体的物理文件（如.exe, .jar）。
- 部署：将工件部署到节点上。
- 通信路径：节点间的连接线。

第九章：包图、组合结构图、定时图、交互概览图和概要图

知识点：

1. 作用：对模型元素进行分组管理，体现命名空间和层次结构，用于逻辑架构设计。

2. 核心元素：

- **包**：带标签的文件夹图标。
 - **关系**：
 - **依赖**：一个包的元素使用另一个包的元素（虚线箭头）。
 - **泛化**：包之间的继承。
 - **嵌套**：包中包含子包。
3. **包与组件的区别**：包是逻辑分组（设计时），组件是物理模块（实现时）。

复习策略与答题技巧（ChatGPT-5.1旗舰模型）

1. **选择/简答题**：熟记各图定义、作用、核心元素、关系的精确语义。
2. **分析题**（常考6种图：用例、类、顺序、通信、状态、活动）：
 - **步骤**：① 审题判断用哪种图；② 提取关键对象/状态/活动；③ 分析逻辑流程；④ 对照标准检查关系/消息/条件是否正确。
 - **常见错误点**：聚合组合混淆、包含扩展误用、消息顺序错乱、监护条件遗漏。
3. **设计题**：
 - **类图**：从需求中找名词（类）、动词（方法），分析类之间的关系（重点！）。
 - **顺序图**：明确参与对象，按时间顺序画出消息，注意返回和组合片段。
 - **状态图**：明确一个对象的所有状态，触发状态变化的事件和条件。
 - **活动图**：明确业务流程的开始和结束，识别并行活动和判断点，使用泳道分配职责。
4. **资源利用**：务必以课堂课件和教材配套资源（P2下载）为主，例题和习题是关键。

对比记忆

1. 用例图的关系

关系	箭头长啥样	由哪里指向哪里
关联	开放箭头	由参与者指向用例
包含<<include>>	虚线开放箭头	由包含指向被包含
扩展<<extend>>	虚线开放箭头	由扩展用例指向用例
泛化	实线三角形箭头	由范围更大的指向范围更小的用例
分组		

2. 类图的关系：

关系	箭头长啥样	由哪里指向哪里
关联	开放箭头实线(单向关联)、实线无箭头(双向关联)。一般不用箭头	-
聚合	空心菱形，另一边可以有箭头	由总体指向部分
组合	实心菱形，另一边可以有箭头	由总体指向部分
泛化	空心三角形	由小类指向大类

3. 顺序图/通信图

消息的类型	箭头长啥样	由哪里指向哪里
同步消息	实心三角形实线	调用者 → 接收者
异步消息	开放箭头实线	发送者 → 接收者
同步且立即返回消息	开放箭头虚线	接收者 → 调用者

** 通信图简化为开放实线箭头，不严格区分同步/异步