



UML概述

1

内容

- 什么是UML
- UML的历史
- UML的主要特点
- UML的构成
- UML中的视图
- UML的公共机制
- UML的应用领域
- 支持UML的工具
- UML各种模型图实例

2

什么是UML?

- UML是Unified Modeling (建模) Language
 - 模型从全局上把握系统的全貌及其相关部件之间的联系
- UML是用于软件开发过程的面向对象建模语言
 - 面向对象技术是对真实世界的抽象思维方式
- UML是Unified (统一) Modeling Language
 - 统一各种建模方法的表示差别，并集它们所长于一身

3

什么是UML?

- 建立一种统一方法和语言的意义：
 - 面对众多的方法，用户难以辨别是否适合于自己当前工作的需要。
 - 不同的表示方法妨碍了用户之间的交流。
 - 不同的方法本身也在不断地融合。
- UML统一了各种不同的软件分析和设计的建模语言，成为系统建模语言的事实上的标准。

4

什么是UML?

Unified Modeling Language (统一建模语言) 是对象管理组织 (OMG) 制定的一个通用的、可视化的建模语言标准，可以用来可视化 (visualize)、描述 (specify)、构造 (construct) 和文档化 (document) 软件密集型系统的各种工件 (artifacts, 又译制品)

5

课堂习题

- UML的全称是 ()
 - A. Unify Modeling Language
 - B. Unified Modeling Language
 - C. Unified Modem Language
 - D. Unified Making Language

6

课堂习题

- 选B。

7

课堂习题

- 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 (1分)。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 (1分)，还支持从需求分析开始的软件开发的全过程 (1分)。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 (2分)。

8

课堂习题

- 一个UML模型只描述了一个系统_____，它并没告诉我们系统是怎么做。
- 答案：要做什么

9

UML的历史

UML之前的OO方法：

- Shlaer/Mellor的OOSA方法, 1988
- R. WirfsBrock的职责驱动CRC卡片方法, 1990
- P. Coad & E. Yourdon的OOA/OOD方法, 1991
- Grady Booch的Booch方法, 1991
- James Rumbaugh的OMT, 1991
- Ivar Jacobson的OOSE方法, 1992
-

10

UML的历史

- 三种方法受到推崇
 - Booch: Booch '93 《Object-Oriented Analysis and Design with Applications》
 - OMT: Rumbaugh '91 《Object-Oriented Modeling and Design》
 - OOSE: Ivar Jacobson '92 《Object-Oriented Software Engineering》

成功地使用用例的概念是面向对象技术走向第二代的标志

11

课堂习题

- OMT方法是由下面哪位科学家提出的 ()
 - A. Booch
 - B. Rumbaugh
 - C. Coad
 - D. Jacobson

12

课堂习题

- 选B。Booch提出的是Booch方法；Coad&Yourdon提出的是OOA/OD方法；Jacobson提出的是OOSE方法。

13

课堂习题

- OOSE方法是由下面哪位科学家提出的（ ）
 - A. Booch
 - B. Rumbaugh
 - C. Coad
 - D. Jacobson

14

课堂习题

- 选D。

15

课堂习题

- 第二代面向对象技术的标志是（ ）
 - A. 用例
 - B. UML语言
 - C. 活动图
 - D. 组件图

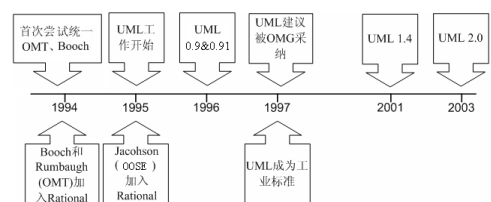
16

课堂习题

- 选A。

17

UML的历史



18

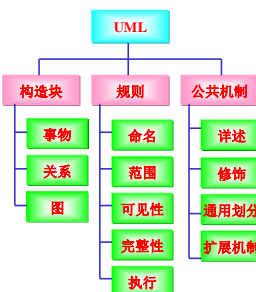
UML的特点

- 统一的标准, 已经被OMG接受为标准建模语言
- 面向对象, 支持面向对象开发
- 可视化, 表示能力很强大
- 独立于开发过程, 可以适用于不同软件过程
- 概念明确, 表示简洁, 结构清晰, 容易学习和掌握

19

UML的构成

- 基本构造块: 也就是建模元素, 是模型的主体
- UML规则: 也就是支配基本构造块如何放在一起的规则
- 公共机制: 运用于整个UML模型中的公共机制、扩展机制



20

课堂习题

- 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

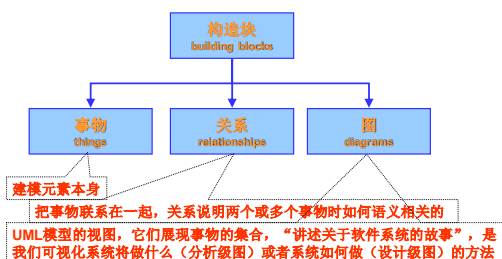
21

课堂习题

- 选A。

22

构造块



23

课堂习题

- 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

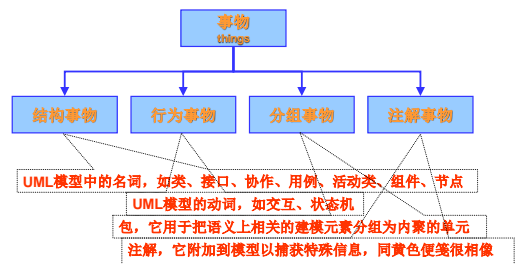
24

课堂习题

- 选A。

25

事物



26

课堂习题

- UML中的事物包括：结构事物，分组事物，注解事物和（ ）
 - A. 实体事物
 - B. 边界事物
 - C. 控制事物
 - D. 动作事物

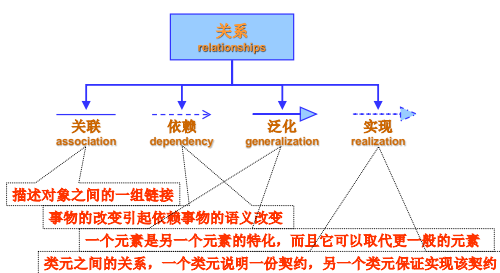
27

课堂习题

- 选D。

28

关系



29

课堂习题

- 实现的符号用_____表示。

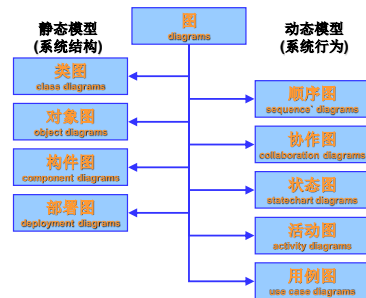
30

课堂习题

- 虚线加空心三角箭头。

31

图



32

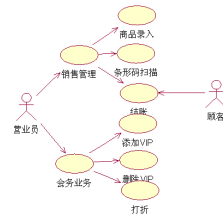
UML 9种图

类图：类以及类之间的相互关系	静态图	结构
对象图：对象以及对象之间相互关系		
构件图：构件及其相互依赖关系	实现图	构
部署图：构件在各节点上的部署		
顺序图：强调时间顺序的交互图	交互图	行为
协作图：强调对象协作的交互图		
状态图：类所经历的各种状态	行为图	为
活动图：活动以及活动间的约束关系		
用例图：需求捕获，测试依据	用例图	

33

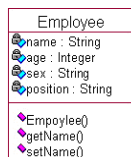
用例图

- 用例图描述了一个系统提供的功能单元。用例图的主要目的是帮助开发团队以一种可视化的方式理解系统的功能需求，包括基于基本流程的“角色”关系，以及系统内用例之间的关系。
- 使用用例图可以表示出用例的组织关系，这种组织关系包括整个系统的全部用例或者是完成相关功能的一组用例。
- 在用例图中画出某个用例方式是在用例图中绘制一个椭圆，然后将用例的名称放在椭圆的中心或椭圆下面的中间位置。



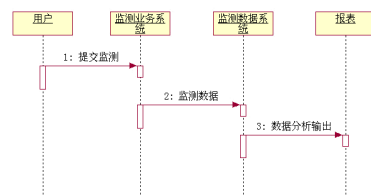
类图

- 类图显示了系统的静态结构，表示了不同的实体（人、事物和数据）是如何彼此关联起来。
- 类图可用于表示逻辑类，逻辑类通常就是用户的业务所谈及的事物，比如说学生、学校等。类图还可用于表示实现类，实现类就是程序员处理的实体。



序列图

- 序列图显示了一个具体用例或者用例的一部分的一个详细流程。它几乎是自描述的，序列图不仅可以显示了流程中不同对象之间的调用关系，还可以很详细地显示对不同对象的调用。
- 序列图有两个维度：垂直维度，也称时间维度，以发生的时间顺序显示消息或调用的序列；水平维度显示消息被发送到的对象实例。



课堂习题

- 下面哪个视图属于UML语言的交互图 ()
 - A. 行为图
 - B. 状态图
 - C. 实现图
 - D. 顺序图

43

课堂习题

- 选D。

44

课堂习题

- 下面不属于UML语言的交互图的是 ()
 - A. 沟通图
 - B. 状态图
 - C. 交互浏览图
 - D. 顺序图

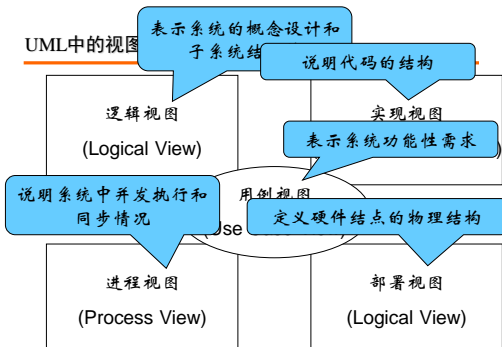
45

课堂习题

- 选B。

46

UML中的视图



47

UML中的视图

- 用例视图
- 逻辑视图
 - 静态结构在类图和对象图中进行描述；动态模型在状态图、顺序图、协作图以及活动图中进行描述
- 实现视图
 - 构件图
- 进程视图
 - 状态图、协作图和活动图
- 部署视图
 - 部署图

48

课堂习题

- UML视图不包含()。
 - A. 用例视图
 - B. 用户视图
 - C. 构件视图
 - D. 逻辑视图

49

课堂习题

- 选B。

50

课堂习题

- UML语言包含几种视图 ()
 - A. 3
 - B. 5
 - C. 7
 - D. 9

51

课堂习题

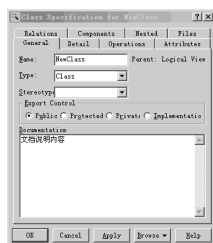
- 选B。

52

UML的通用机制

1. 规则说明

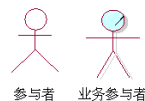
- 如果把模型元素当成一个对象来看待，那么模型元素本身也应该具有很多的属性，这些属性用于维护属于该模型元素的数据值。
- 属性是使用名称和标记值（Tagged Value）的值来定义的。标记值指的是一种特定的类型，可以是布尔型、整型或字符型，也可以是某个类或接口的类型。



UML的通用机制

2. 修饰

- 在UML的图形表示中，每一个模型元素都有一个基本符号，这个基本符号可视化地表达了模型元素最重要的信息。
- 用户也可以把各种修饰细节加到这个符号上以扩展其含义。这种添加修饰细节的做法可以为图中的模型元素在一些视觉上的效果上发生一些变化。



UML的通用机制

3. 通用划分

- 通用划分是一种保证不同抽象概念层次的机制。通常我们可以采用两种方式进行通用划分，一种是对类和对象的划分，另外一种是对接口和实现的分离。
- 类和对象的划分是指类是一个抽象而对象是这种抽象的一个实例化。接口和实现的分离是指接口声明了一个操作接口，但是却不实现其内容，而实现则表示了对该操作接口的具体实现，它负责如实地实现接口的完整语义。
- 类和对象的划分保证了实例及其抽象的划分，从而使得对一组实例对象的公共静态和动态特征无需——管理和实现，只需要抽象成一个类，通过类的实例化实现对对象实体的管理。

UML的扩展机制

1. 构造型

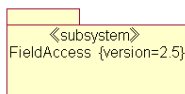
- 简而言之，构造型扩展机制的目的就是基于一个已存在的模型元素再重新定义一个新的模型元素。
- 构造型的一般表现形式为使用“<<”和“>>”包含构造型的名称在里面，例如<<use>>、<<extends>>等。



UML的扩展机制

2. 标记值

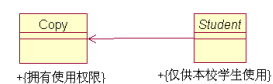
- 标记值是由一对字符串构成，这对字符串包含一个标记字符串和一个值字符串，用来存储着有关模型元素或表达元素的一些相关信息。
- 标记值，可以被我们用来扩展UML构造块的特性，可以根据我们的需要来创建详述元素的新元素。
- 标记值可以与任何独立元素相关，包括模型元素和表达元素。标记值是当我们需要对一些特性进行记录的时候而给定元素的值。



UML的扩展机制

3. 约束

- 约束机制用于扩展UML构造块的语义，允许建模者和设计人员可以增加新的规则和修改现有的规则。
- 约束可以表示在UML的规范表示中不能表示的语义关系，特别是当陈述全局条件或影响许多元素的条件时，约束特别有用。
- 约束使用大括号和大括号内的字符串表达式表示，即约束的表现形式为(约束的内容)。约束可以附加在表元素、依赖关系，或注释上。



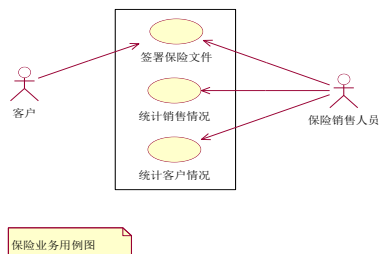
UML的应用领域

- UML最常用的是建立软件系统的模型
- UML还可以用于描述非软件领域的系统，如机械系统、企业机构等。
- UML适用于系统开发的不同阶段，包括分析、设计、实现和测试等。

支持UML的工具

- Rational Rose
- Borland Together
- ArgoUML
- StarUML
- Microsoft visio
-

UML各种模型图实例



61

课堂习题

● 下面哪个符号代表注释 ()

- A. B.
- C. D.

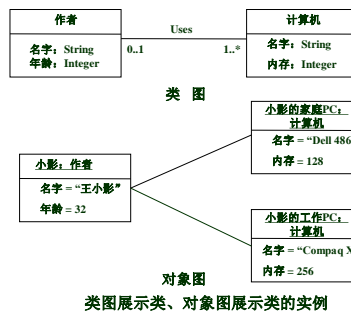
62

课堂习题

● 选D。

63

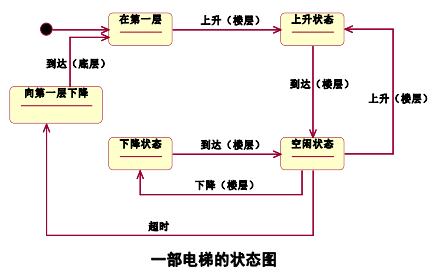
UML各种模型图实例



类图展示类、对象图展示类的实例

64

UML各种模型图实例



一部电梯的状态图

65

课堂习题

● 下面哪个UML视图是描述一个对象的生命周期的 ()

- A. 类图
B. 状态图
C. 协作图
D. 组件图

66

课堂习题

- 选B。

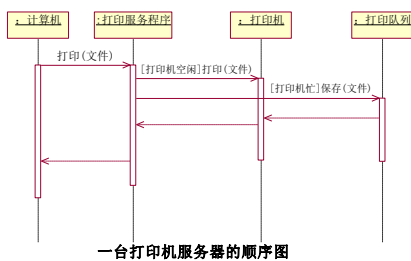
67

课堂习题

- 状态图只是对_____对象建立模型。
- 答案：一个

68

UML各种模型图实例



一台打印机服务器的顺序图

69

课堂习题

- 生命线是UML视图中哪个图形的组成部分 ()
 - A. 类图
 - B. 状态图
 - C. 活动图
 - D. 顺序图

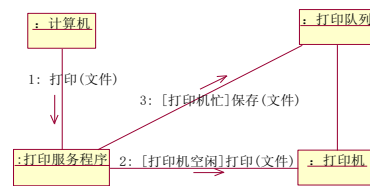
70

课堂习题

- 选D。

71

UML各种模型图实例



一台打印机服务器的协作图

72

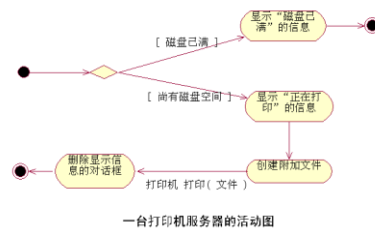
课堂习题

- 在协作图中通过_____表示出消息的时间顺序。

答案：消息编号

73



UML各种模型图实例





74

课堂习题

- 下面哪个图形代表活动 ()

A.  B. 

C.  D. 

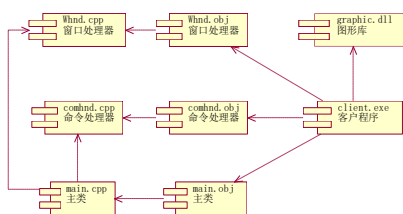
75

课堂习题

- 选D。

76

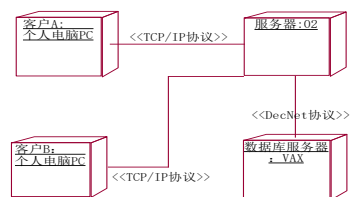
UML各种模型图实例



反映程序代码构件之间的依赖关系的构件图

77

UML各种模型图实例



描述一个系统的物理体系结构的部署图

78