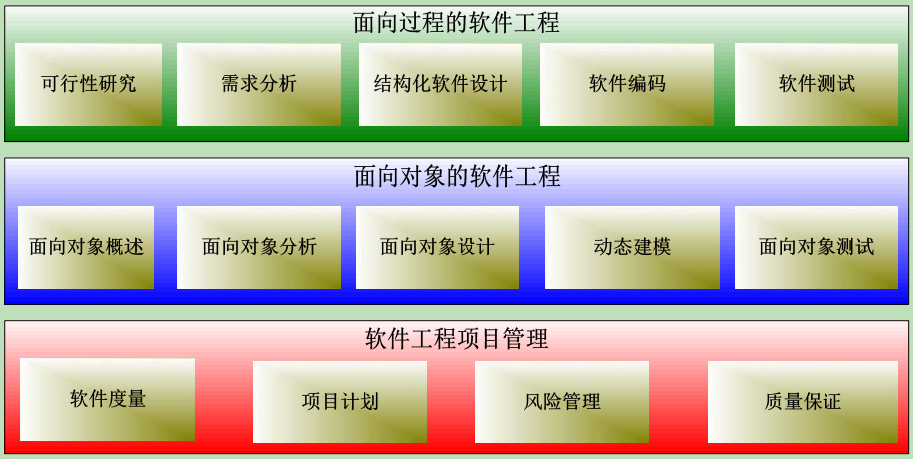
软件工程
第8讲 面向对象的方法学

贾西平
Email: jiaxp@126.com

课程主要内容



内容提纲

- 面向对象的基本概念
- 面向对象建模
- 对象模型
- 动态模型
- 功能模型

3

内容提纲

- **面向对象的基本概念**
- 面向对象建模
- 对象模型
- 动态模型
- 功能模型

4

Peter Coad和Edward Yourdon提出用下列等式认识面向对象方法：

面向对象 = 对象（object）

+ 分类（classification）

+ 继承（inheritance）

+ 通过消息的通信（communication with messages）

可以说，采用这四个概念开发的软件系统是面向对象的

5

面向对象的基本思想

将一个实际问题看成是一个或几个对象的集合。

6

面向对象分析

在系统所要求解的问题中找出对象以及它所属的类，并定义对象与类。

7

面向对象设计

把系统所要求解的问题分解为一些对象及对象间传递消息的过程。

8

面向对象实现

- 把数据和数据处理过程结合为一个对象。

9

面向对象的基本概念

•对象（object）:客观世界的实体，是一组**属性**以及这组属性上的**专用操作**的**封装体**。

– 一个对象通常由对象名、属性和操作三部分组成

•属性（attribute）：每个对象都有它自己的**属性值**，表示其状态。

– 通常是一些数据，也可以是另一个对象。

– 对象中的属性**只能通过**该对象所提供的操作来存取或修改。

•操作（operation）：也称**方法**或**服务**，规定了对象的行为，表示对象所能提供的服务。

10

封装

- **封装**（encapsulation）：一种信息隐蔽技术，用户只能看见对象封装界面上的信息，对象的内部实现对用户是隐蔽的。
- **封装的目的**：使对象的使用者和生产者分离，使对象的定义和实现分开。

11

举例

- 一个学生，一条信息，一本书都可以是一个对象；
- 学号、姓名、性别、年龄和专业等数据描述了一个学生对象的状态，是学生对象的属性。
- 学生信息登记、学生专业查询等方法定义了学生对象的操作。

12

类

- **类（Class）**
  一组具有相同属性和相同操作的对象的集合。
- 每个对象都是其所属类的一个实例(instance)
- 类是创建对象的模板，同一个类的每个对象都具有相同的结构和行为。

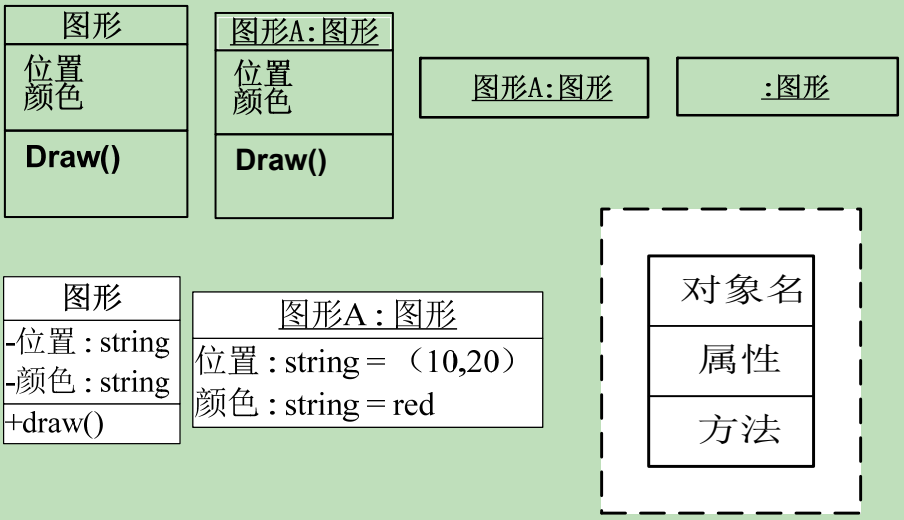
13

类的说明

- 类是一个抽象数据类型的实现，描述了属于该类型的所有对象的性质。
  - 例如，Integer是一个类，它描述了所有整数的性质(包括整数的算术运算和大小比较)，“2”、“3”和“5”等这些具体整数都是Integer这个类的对象，都具备算术运算和大小比较的处理能力。
- 类**属性**是对象状态的抽象，用数据结构来描述；
- 类**操作**是对象行为的抽象，用操作名和实现该操作的方法来描述

14

对象的表示



15

类的层次

- 一个类的上层可以有超类(Superclass)，超类也称基类；
- 一个类的下层可以有子类(Subclass)，子类也称派生类。
- 类之间的结构关系主要有两种：
  - 一般与特殊结构关系
  - 整体与部分结构关系

16

类的结构关系：一般与特殊

- **一般与特殊结构关系**又称为分类结构关系，是“is a”关系。
- 例如，飞机和交通工具都是类，飞机是一种特殊的交通工具，它们之间是“is a”关系。
- 上层类:一般性、共性;下层类:特殊、具体。
- 现实世界中的**一般化**的抽象关系用类的这种结构关系来描述

17

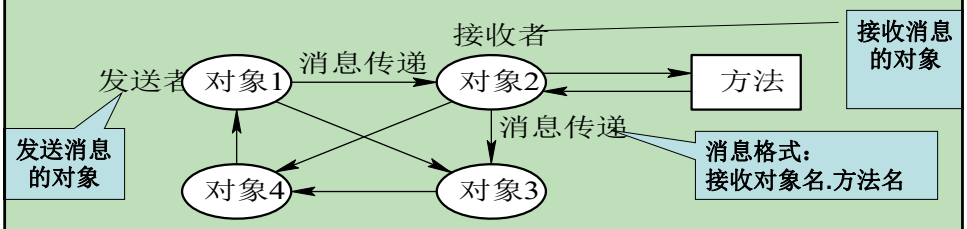
类的结构关系：整体与部分

- **整体与部分结构关系**称为组装结构关系，是“has a”关系。
- 例如，飞机和发动机都是类，发动机是飞机的一部分，它们之间是“has a”关系。
- 上层类：整体；下层类：部分、成员。
- 现实世界中的**组成**的抽象关系用类的这种结构关系来描述

18

消息

- **消息**用来请求对象执行某一处理或回答某些信息要求。
- 对象间通信通过消息传递来实现。
- 在面向对象程序设计中，程序的执行是靠在对象间传递消息来完成的。



19

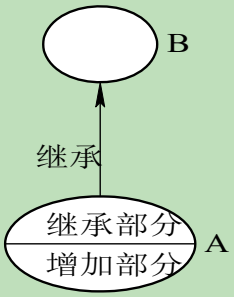
方法

- **方法**是类中操作的实现过程。
- 方法包含方法名、参数和方法体。
- 一个对象接收到一条消息后，它所包含的方法决定对象该做怎样的处理。
- 对象的内部信息隐蔽(私有)，对象间只能通过消息来连接，对象私有数据用其方法访问。

20

继承性

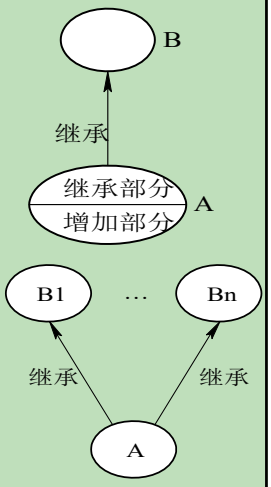
- 当类A不但具有类B的属性，而且还具有自己的独特属性时，称类A**继承**了类B，继承关系常称“**即是**”(**is a**)关系
- 当类A继承类B时，类A是类B的子类，而类B是类A的超类。
- 子类A由两部分组成：继承部分和增加部分。
  - 继承部分是从B继承来的
  - 增加部分是专为A编写的新代码



21

继承的特点及分类

- 继承具有**传递性**。
- 继承分类：单重继承和多重继承
- **单重继承**（或简单继承）：一个类至多只能直接继承一个类。
  - 简单继承是树型结构。
- **多重继承**：一个类可以直接继承多个类。
  - 多重继承是网状结构。



22

封装性

- **封装是一种信息隐蔽技术，用户只能见到对象封装界面上的信息，对象内部对用户是隐蔽的。**
- 用户只知道某对象是“做什么”的，不知道“怎么做”。
- 封装将外部接口与内部实现分离，用户不必知道行为实现的细节，只须用消息来访问该对象。

23

多态性

- **多态性**是指同一个操作作用于不同的对象上可以有不同的解释，产生不同的执行结果。
  - 例如“画”操作，作用在“矩形”对象上，则在屏幕上画一个矩形，作用在“圆”对象上，则在屏幕上画一个圆。
- 不同等级层次中的类可共享一个行为名称，各自按自己的需要来实现这个行为，得到不同的结果.
- 多态性机制允许每个对象以自己的解释方式来响应共同的消息。
- **优点：**增加了软件系统的灵活性、可理解性和可维护性，减少了信息冗余，提高了软件的可重用性和可扩充性

24

内容提纲

- 面向对象的基本概念
- **面向对象建模**
- 对象模型
- 动态模型
- 功能模型

25

模型与建模

- **模型**是为了理解问题而对问题作出的一种抽象，是对问题的一种**无歧义**的描述。
- 模型由一组**图示符号**和组织这些符号的**规则**组成，利用它们来定义和描述问题域中的术语和概念。
- 利用模型可以规范地表示问题的解
- 建模的目的：
  - 减少复杂性(把复杂问题分解、化简)
  - 促使软件开发人员透彻地理解问题

26

面向对象模型

- **面向对象建模**是用面向对象技术和方法来定义、描述问题域中的软件解。
- 用面向对象方法开发软件，通常需要建立**对象模型**、**动态模型**和**功能模型**三种模型。
- **对象模型**：描述系统的数据结构，是目标系统中最关键、最基本、最核心的模型；
- **动态模型**：描述系统的控制结构；
- **功能模型**：描述系统的功能。

27

内容提纲

- 面向对象的基本概念
- 面向对象建模
- **对象模型**
- 动态模型
- 功能模型

28

对象模型

- **作用：**描述系统的静态结构，包括构成系统的类和对象、它们的属性和操作，它们之间的联系。
  - 面向对象方法是以对象为基础来构造系统
  - 对象模型为建立动态模型和功能模型提供了实质性的框架。
  - 对象模型是基础，动态模型和功能模型在此基础上创建。
- **建立对象模型的目标：**从问题域中提炼出对目标系统有价值的概念

29

类的命名

- 使用**名词**或**名词短语**命名
  - 例如，图书、课程、公司职员等；
- 用具体应用领域人们习惯的专业术语作类名
- 不要太随便，或刻意创造；
- 类名应该无歧义、简洁、具有描述性

30

结构的表示符号

- 在面向对象分析与设计中，**结构**是问题域复杂关系的表示，与系统的任务直接相关，目标系统的任务决定了系统的结构。
- 结构一般分为两类：
  - **一般-特殊结构**
  - **整体-部分结构**

31

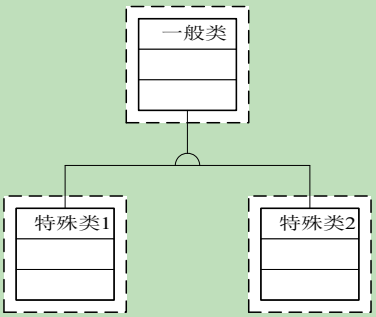
一般-特殊结构

- **一般 - 特殊结构**也称**分类结构**，或**归纳结构**，它是一个类与它的子类之间的分类关系
- 高层类说明一般性的属性，低层类说明特殊属性。
- 低层类对象 **“即是(is a)”**高层类对象的某种特殊情况，继承了在高层类中定义的属性和服务。
- 一般-特殊关系具有继承性，一般类和对象的属性和方法被定义后，即可在特殊类和对象中使用

32

一般-特殊结构的表示

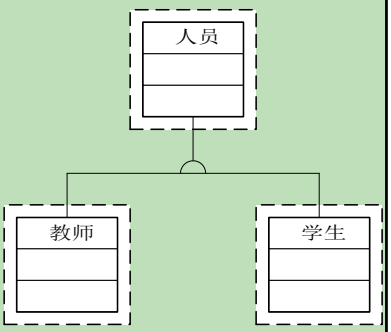
- 顶部是一个一般类，下部是若干个特殊类，之间用**线**和**半圆型标记**连接；
- 半圆型的标记表明图形是一般 - 特殊关系，这种表示法是**有向**的，从半圆中心画一条线所指到的是特殊类。
- 一般类与特殊类之间的**连线端点应连接到类**(而不是对象)，表明是类之间的关系



33

一般-特殊结构的表示实例

- 在关于人员的管理系统中，可以将人员定义成一般类，将教师和学生定义成人员的特殊类。
- 它们之间构成一般 - 特殊关系
表示如右图



34

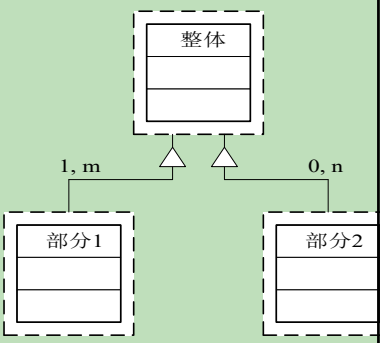
整体-部分结构

- **整体 - 部分结构**也称**组装结构**(**聚集结构**)，它反映了对象之间的**构成关系**。
- 能将具有特殊的整体 - 部分关系的类组织到一起。
- 对于在**问题域**和**系统任务的边界区域**中识别类非常有用
- **传递性**：如果A是B的一部分，B是C的一部分，则A也是C的一部分

35

整体-部分关系表示

- 在顶部是一个整体对象(用类-&-对象符号表示的对象)，下部是组成该整体的若干个部分对象(用类-&-对象符号表示的对象)，它们之间用线与三角标记连接。
- **三角标记**表明整体 - 部分关系，是**有向**的。
- 整体画在上部，部分画在下部
- 关系线的**终点位置**是**外框上**，反映**对象之间**的映射关系
- 结构线的每一端标有一个**数量**或**数量的区域**，表示整体可以拥有的部分数（或部分可以拥有的整体数），当**数量为1时可省略**。



36

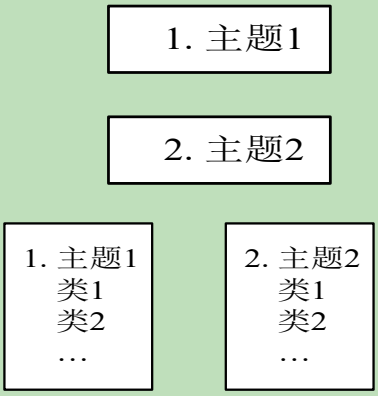
主题

- **主题**：一种指导开发者或用户研究大型复杂模型的一种机制，有助于分解大型项目以便分组承担任务。
- 可以给出面向对象分析和设计的模型总体概貌
- 主题所依据的原理是整体 - 部分关系的扩充

37

主题的表示

- 主题有两种表示形式：简单表示形式和扩展表示形式。
- **简单表示形式**只标出主题名和编号（如上图）；
- **扩展表示形式**除了标出主题名和编号外，还要标出主题所包含的类（如下图）



38

属性

- **属性**用于描述类的特性
- 一个属性是一个数据项（状态信息）
- 类中对象都有相应的属性值
- **属性表示：**放在类表示符号的中间部位

39

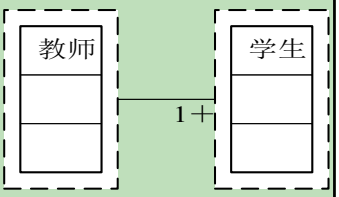
关联关系

- 反映**对象之间**相互依赖、相互作用的关系。
- 一个关联关系就是一个问题域映射模型，反映了某个对象对其他对象的需求。
- 反映了对象之间的静态关系，是二元关系

40

关联的表示

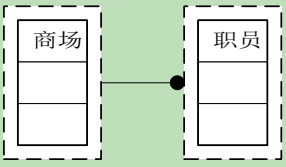
- 关联用两个对象之间的**实线**来表示。
- 关联关系分为**一对一**(1:1)、**一对多**(1:m)和**多对多**(m:n)等三种基本类型
- 例如
  - 一个学校有一个校长，学校与校长是一对一的关系；
  - 一个教师教许多学生，教师与学生是一对多的关系；
  - 一个学生可选修多门课程，一门课程可被多个学生选修，学生与课程是多对多的关系



41

阶数

- 对象的每条关联关系上均标有数字(m)或者范围(m，n)，说明该对象对其他对象的约束。
- 该数字或范围表明可能发生的映射数目或范围。如：
  - “○”表示零个或一个
  - “●”表示零个或多个
  - “1+”表示一个或多个
  - “1，3”(1～3)表示该范围的上下限。
  - 固定数目的连接可使用单个数字来表示



42

服务

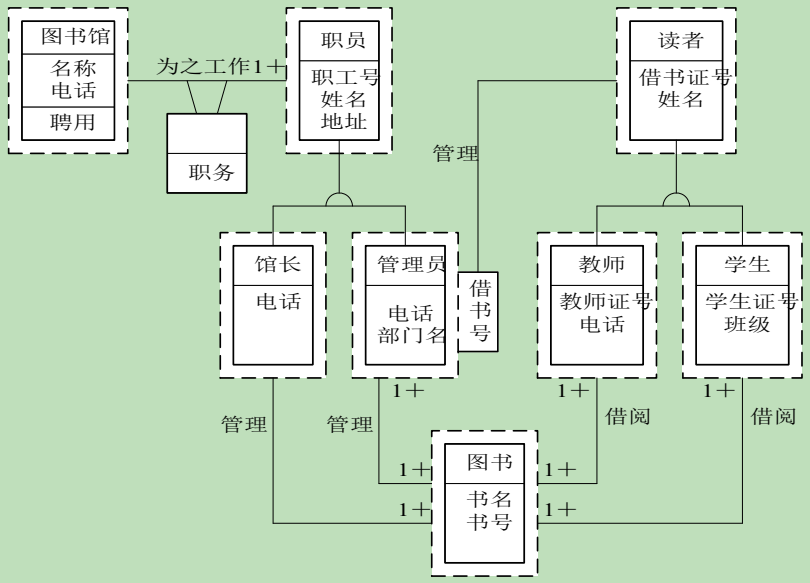
- **服务**：某对象所具有的特定行为；
- 一个服务就是收到一条消息之后所执行的处理。
- **服务表示**：放在类表示符号的下部

43

对象模型举例

- 一个图书借阅管理的对象模型，它表明该公司有许多名职员为之工作，职员又进一步划分为馆长和管理员两类；
- 馆长可对图书借阅工作进行管理，每名管理员可管理多本图书，每本图书可由多名管理员管理；
- 图书拥有许多读者，借书证号惟一地确定一个读者；
- 读者又可分为教师和学生两类，每名教师或学生可以借阅数本图书，每一种图书可供多名教师或学生借阅。

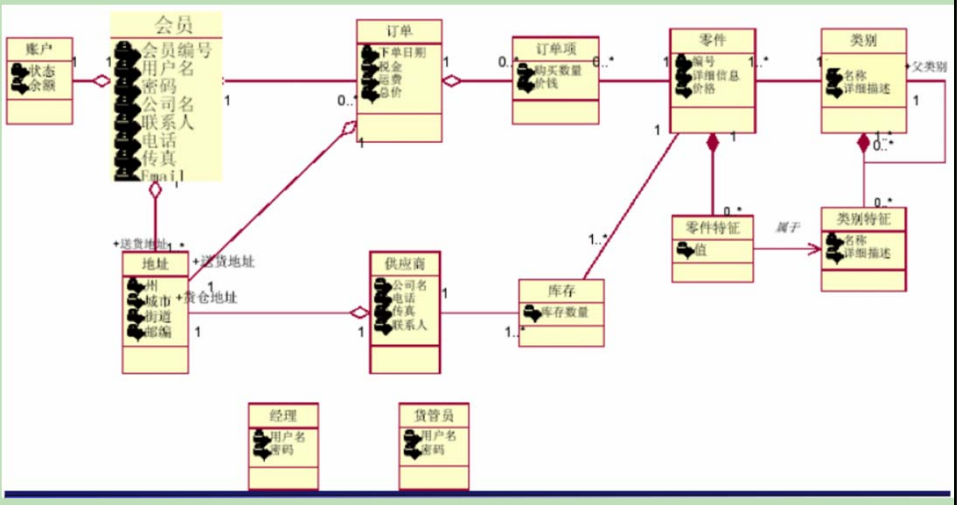
44



45

图书借阅管理的对象模型

类图（修订4）



46

内容提纲

- 面向对象的基本概念
- 面向对象建模
- 对象模型
- **动态模型**
- 功能模型

47

动态模型

- **动态模型**表示**瞬时的、行为化的系统的控制性质**，描述了系统的**控制结构**。
- 动态模型侧重于系统的控制逻辑，包括**状态图**和**事件追踪图**。
- **状态图**用来描绘对象的**状态**、触发状态转换的**事件**以及对象的**行为**(对事件的响应)；
- **事件追踪图**侧重于说明发生于系统执行过程中的一个**事件序列**。
- 每个类的动态行为用一张状态图来描绘，各个类的状态图通过共享事件组合起来，从而构成系统的动态模型。

48

事件

- **事件**是引起对象状态转换的控制信息。
- 模型中，各对象之间**相互触发**，一个触发行为称作一个**事件**。
- 对象对事件的响应方式包括：
  - **改变自己的状态**
  - **形成一个新的触发行为**

49

事件类

- **事件类**由各个独立事件的共同结构和行为抽象组成。
- 有些事件类传送的是简单的“要发生某事件”的信息，有些传送的是数据值。
- 由事件传送的数据值叫**属性**。
- 属性可以在事件类名之后用括号列出，如下表：

事 件 类	属　性	描　述
列车出发 按下鼠标按钮 数字拨号	线路、班次、城市 按钮、位置 数字	**列车出发**(线路、班次、城市) **按下鼠标按钮**(按钮、位置) **数字拨号**(数字)

50

状态

- 状态是对对象属性值的一种抽象。
- 对象所具有的属性值称为**对象的状态**。
- 各对象之间相互触发(即作用)，就形成了一系列的状态变化

51

状态的特性

- **状态具有时间性**
  - 事件表示时刻，状态代表时间间隔。
  - 一个对象在接收事件前后是两个不同的状态
- **状态具有持续性**
  - 需要一段时间间隔表示一个状态。
  - 状态与事件相互依赖，一个事件可将两个状态分开，一个状态可将两个事件隔开

52

行为

- **行为**：被事件触发的对象达到某种状态时，所做的一系列处理操作。

53

脚本

- **脚本**，也叫**场景**(Scenarios)，是指系统在某一执行期间内出现的一系列事件。
- 脚本通常**起始于**一个系统外部的输入事件，**结束于**一个系统外部的输出事件。

54

使用电话的脚本

编号	事　件	编号	事　件
1	呼叫者拿起电话	10	呼叫者拨号(4)
2	响拨号声	11	电话鸣响声
3	呼叫者拨电话号码(2)	12	接收者拿起电话
4	拨号声停	13	停鸣响声
5	呼叫者拨号(3)	14	电话接通
6	呼叫者拨号(3)	15	通电话
7	呼叫者拨号(6)	16	接收者挂断电话
8	呼叫者拨号(6)	17	电话中断
9	呼叫者拨号(5)	18	呼叫者挂断电话

55

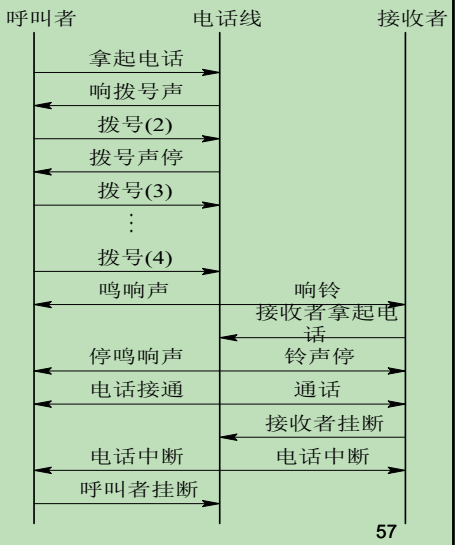
事件追踪图

- 描述发生于系统执行过程中的一个特定“场景”(脚本)，是完成系统某个功能的一个事件序列。
- 写好脚本后，需要确定事件追踪：
  - 标识每个事件的发送者对象和接收者对象
  - 用事件追踪图按事件序列顺序，来表示事件、事件的发送者对象和事件的接收者对象

56

事件追踪图的画法

- 竖线表示对象
- 带箭头的横线表示事件
- 箭头从发送者对象指向接收者对象
- 时间自上向下延续，与间隔的空间无关，没有精确的时序
  例：打电话时间追踪图（见右图）



57

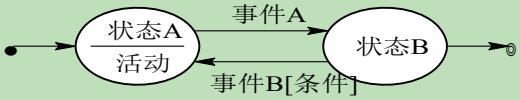
状态图

- 状态图是一个**状态与事件的网络**，侧重于描述**每一类对象的动态行为**。
- 状态是某一时刻中属性特征的概括，而状态转换则表示这一类对象在何时，对系统内外发生的哪些事件作出何种响应

58

状态图的画法

- 用圆形框或椭圆框表示状态，框内可标上状态名也可不给状态命名，行为在框内用关键字do(后接冒号)标明。
- 用箭头线表示从一个状态到另一个状态的转换，附加在箭头线上的短语标明触发此状态转换的事件名
- 如果事件是条件事件，须在事件名后加一个方括号并写上状态转换的条件。
  - 条件事件当条件满足时，该事件的发生才能引起状态的转换。
- 用实心圆表示初始状态，用一对同心圆表示最终状态



59

状态图与事件追踪图

- 状态图描述**一个对象**的个体行为
- 事件追踪图描述**多个对象**所表现出来的集体行为
- 两者从不同角度说明同一系统行为

60

内容提纲

- 面向对象的基本概念
- 面向对象建模
- 对象模型
- 动态模型
- **功能模型**

61

功能模型

- 功能模型
  - 由多个**数据流图**组成
  - 表明了从外部输入，通过操作和内部存储，直到外部输出的整个的数据流情况
  - 包括了对象模型内部数据间的限制。
- 数据流图**不指出控制或对象的结构信息**，它们在动态模型和对象模型中已描述。
- 建立功能模型有助于软件开发人员更深入地理解问题域，改进和完善自己的设计

62

三种模型之间的关系

- 三者相互补充、相互配合，从不同侧面描述了待开发系统
  - **对象模型侧重于描述系统数据结构**，定义了“做什么”的实体
  - **动态模型侧重于描述系统控制结构**，规定在何种状态下，接受什么事件的触发而“做什么”
  - **功能模型侧重于描述系统功能，**指明了系统应该“做什么”
- 动态模型和功能模型以对象模型为基础
- 三种模型都包含了数据、控制和操作等共同概念，各自描述的侧重程度不同

63

内容回顾

- 面向对象的基本概念
- 面向对象建模
- 对象模型
- 动态模型
- 功能模型

64

思考题

- 对象、属性、操作的概念？
- 什么是类？
- 什么是封装？
- 什么是继承性？
- 什么是多态性？
- 什么是消息、方法？
- 用面向对象方法开发软件要建立哪些模型？

65

推荐视频

- https://www.bilibili.com/video/BV1Q741157ve?p=51
- https://www.bilibili.com/video/BV1Q741157ve?p=52
- https://www.bilibili.com/video/BV1Q741157ve?p=53
- https://www.bilibili.com/video/BV1Q741157ve?p=54
- https://www.bilibili.com/video/BV1Q741157ve?p=55
- https://www.bilibili.com/video/BV1Q741157ve?p=56
- https://www.bilibili.com/video/BV1Q741157ve?p=57

66

谢谢大家！
感谢清华大学刘强老师的视频资源！

11:36