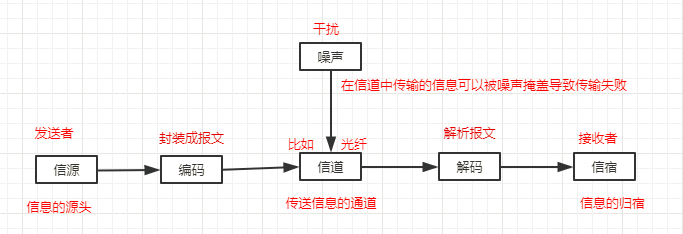
# 信息化与信息系统\*概述

本节课程需要掌握的考点

1. 信息的特征P2（了解）、信息的质量属性p3（掌握）、信息的传输模型P4（掌握）
2. 信息系统的特征 P5（了解）
3. 信息化5个层次 P8、信息化的主体手段途径目标 P8(掌握)
4. 两网、一站、四库、十二金P9 (了解)
5. 信息化六要素及其地位P9-10 (掌握)
6. 信息系统生命周期五个阶段及其内容p10-12(掌握)
7. 信息系统开发方法：结构化、面向对象、原型法、面向服务的方法特征及其优缺点p12-17 （掌握）
8. 计算机网络：OSI协议7层及其作用协议P18、IEEE802协议P18-19、TCP/IP协议4层及其协议作用P19-21、网络交换P21、网络设置及其作用P21-22、无线网络P22、网络存储P33-34、可用性和可靠性P34-35（掌握）

* 信息的质量属性：
  + 1. 精确性 对事物状态描述的精准程度
    2. 完整性 对事物状态描述的全面程度，完整信息应包括所有重要事实
    3. 可靠性 指信息的来源、采集方法、传输过程是可以信任的，符合预期
    4. 及时性 指获得信息的时刻与事件发生时刻的间隔长短
    5. 经济性 指信息获取、传输带来的成本在可以接受的范围之内
    6. 可验证性 指信息的主要质量属性可以被证实或者证伪的程度
    7. 安全性 信息可以被非授权访问的可能性，可能性越低，安全性越高

* 信息技术 核心是传输技术



* 1. 信源：产生信息的实体，信息产生后，由这个实体向外传播、如QQ使用者
  2. 信宿：信息的归宿或接收者，如使用QQ的另一方（当然这一方也是信源），它通过计算机屏幕接收QQ使用者发生的文字
  3. 信道：传播信息的通道，如TCP/IP网络
  4. 编码器：对编码信息进行加密在编码
  5. 译码器：把信道上送来的信号（原始信息与噪声的叠加）转换成信宿能接受的信号
  6. 噪声：干扰，可以来自于信息系统分层结构的任何一次，到噪声携带的信息达到一定程度的时候，在信道中传输的信息可以被噪声掩盖导致传输失败。

## 1.1信息系统与信息化

系统的特性：

1. 目的性 2) 整体性 3)层次性 4)稳定性 5)突变型 6)自组织性 7)相似性 8)相关性 9) 环境适应性

信息系统的特性：开放性、脆弱性、健壮性

* 信息化五个层次：
  + 产品信息化 智能电视、智能灯具
  + 企业信息化 生产制造系统、ERP、CRM、SCM
  + 产业信息化 农业、工业、交通运输、生产制造业、服务业等传统产业
  + 国民经济信息化 金融、贸易、投资、计划、通关、营销等组成一个信息大系统
  + 社会生活信息化 智慧城市、互联网金融
* 信息化
* 主体 全体社会成员、包括政府、企业、事业、互联网金融
* 手段 基于现代信息技术的先进社会生产工具
* 途径 创建信息时代的社会生产力，推动社会生产关系及社会上层建筑的改革
* 目标 使国家的综合实力。社会的文明素质和人民的生活质量全面提升
* 两网、一站、四库、十二金(了解)
  + - 两网 指政务内网和政务外网
    - 一站 政府门户网站
    - 四库 人口、法人单位、空间地位和自然资源。宏观经济等四个基础数据库
    - 十二金

办公业务资源系统、宏观经济管理系统建设

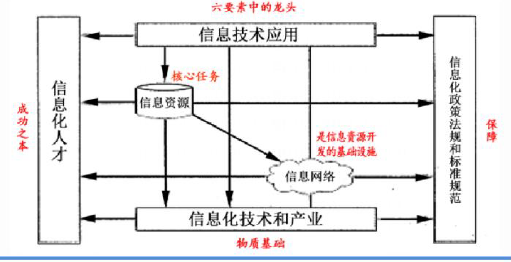
金税、金关、金财、金融监管（含金卡）、金审

金盾、金保、金农、金水、金质

信息化体系六要素

* + 信息技术应用 龙头（主阵地）
  + 信息资源 核心任务
  + 信息网络 基础设施
  + 信息技术和产业 物资基础
  + 信息化人才 成功之本
  + 信息化法规 保障

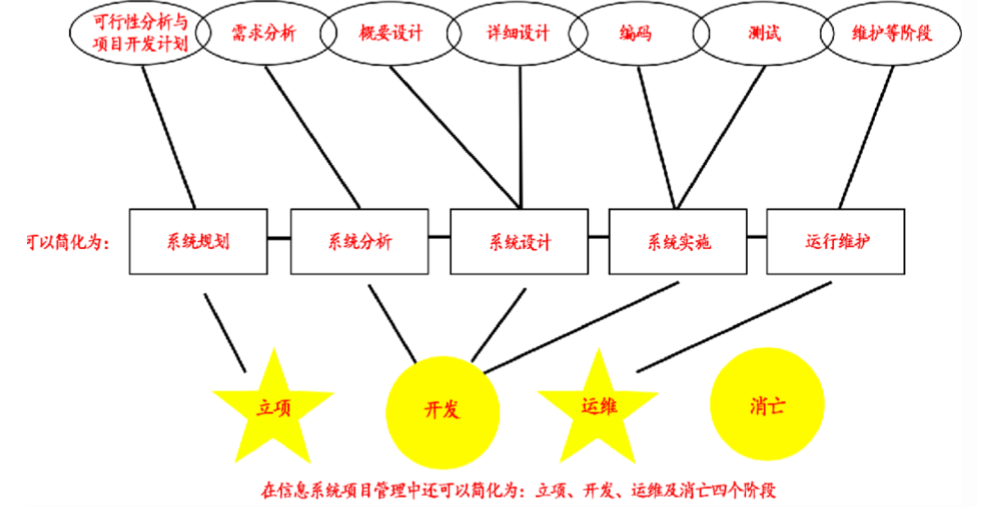
记忆口诀：鹰头、鸡基、人本、法保、织一织热的心、网基



图形记忆口诀：鹰上（应用）、下鸡（技术）、左人（人才）、右龟（规范）

阶段划分：

* 系统规划（可行性分析与项目开发计划）
* 系统分析（需求分析）
* 系统设计（概要设计、详细设计）
* 系统设施（编码、测试）
* 运行维护





记忆技巧：划分即实行，掌握每个阶段的产物

1、系统规划阶段：对组织的环境、目标及现行系统的状况初步调查，根据组织目标和发展战略，确定信息系统的发展战略，对建设新系统的需求做出分析和预测，研究建设新系统的必要性和可能性。给出拟建设的备选方案，写出可行性研究报告

2、系统分析阶段：根据系统设计计划书所确定的范围，对现行系统进行详细调查，描述现行系统的业务流程，指出现行系统的局限性和不足之处，确定新系统的基本目标和逻辑功能需求，即提出新系统的逻辑模型。（做什么）

3、系统设计阶段：根据系统说明书中规定的功能要求，考虑实际条件，具体设计实现逻辑模型的技术方案，也就是设计新系统的物理模型。又称为物理设计阶段，可分为总体设计（概要设计）和详细设计两个阶段。技术文档是系统设计说明书、

4、系统实施阶段：将设计的系统付诸实施的阶段。任务包括计算机等设备的购置、安装和调试、程序的编写和调试、人员培训、数据文件转换、系统调试与转换等、这个阶段的特点是几个相互联系、相互制约的任务同时展开，必须精心安排、合理组织。系统测试之后写出系统测试分析报告。

5、系统运行和维护阶段：系统投入运行后，需要经常进行维护和评价，记录系统运行的情况，根据一定的规则对系统进行必要的修改，评价系统的工作质量和经济效益。

## 1.2信息系统开发方法

**结构化的方法**：

包括结构化方法、面向对象方法、原型化方法、面向服务的方法

精髓是自顶向下、逐步求精和模块化设计

主要特点：

1、开发目标清晰化

2、开发工作阶段化

3、开发文档规范化

4、设计方法结构化

主要缺点：

1、开发周期长

2、难以适应需求变化

3、很少考虑数据结构

**面向对象（OO）方法**:

系统分析、系统设计和系统实现三个阶段之间已经没有“缝隙”

使用结构化方法进行自顶向下的整体划分；然后，自底向上地采用OO方法进行开发

结构化方法和OO方法仍是两种在系统开发领域中相互依存的、不可替代的方法

面向对象的方法

1、面向对象（OO）方法认为，客观世界是由各种对象组成的，任何事物都是对象

2、与结构化方法类似，OO方法也划分阶段，但其中的系统分析、系统设计和系统实现三个阶段之间已经没有“缝隙”。也就是说，这三个阶段的界限变得不明确

3、OO方法使系统的描述及信息模型的表示与客观实体相对应，符合人们的思维习惯，有利于系统开发过程中用户与开发人员的交流和沟通，缩短开发周期。面向对象设计主要有三大特征：封装性、继承性和多态性，使用面向对象方法构建的系统具有更好的复用性

4、一些大型信息系统的开发，通常是将结构化方法和OO方法结合起来。首先，使用结构化方法进行自顶向下的整体划分；然后，自底向上的采用OO方法进行开发。因此，结构方法和OO方法仍是两种在系统开发领域中相互依存的、不可替换的方法。

**面向对象方法OO：**对象、类、集成、封装、消息、多态。关键点：在于是否建立一个全面、合理、统一的模型既能反应问题域又被计算机系统求解域所接受

1. 面向对象的分析OOA：了解问题涉及的对象、对象的关系与操作，构造问题的对象模型
2. 面向对象的设计OOD：对OOA结果整理
3. 面向对象的程序设计OOP：系统功能的编码。

OO方法的分支：Coad/Yourdon、Booch、OMT(系统分析、系统设计、对象设计和实现)、OOSE

**面向对象概念解析**

--1—什么是对象：只要是客观存在的事物都是对象

--2—什么是类：类是确定对象讲会拥有的特征（属性）和行动（方法），它不是具体客观存在的东西

--3—对象的属性：对象的属性就是对象具备的各种特征，每个对象的每个属性都具有特点的值。

--4—对象的方法就是对象执行的操作，也就是说对象能干什么？这就成为方法。

面向对象和面向过程的区别：

1. 面向对象是将实物高度抽象化
2. 面向过程是一种自顶向上的编程
3. 而面向对象是自下先建立抽象模型然后再使用模型

补充知识点：

**对象**：对象是由数据及其操作所构成的封装体，对象包含三个基本要素，分别是对象标识、对象状态和对象行为；是系统中用来描述客观事物的一个模块，是构成系统的基本单元。用计算机语言来描述，对象是由一组属性和对这组属性进行的操作构成的。

**类**：现实世界中实体的形式化描述，类将该实体的属性（数据）和操作（函数）封装在一起。

类和对象的关系可以总结为：

1. 每一个对象都是某一个类的实例；
2. 每一个类在某一时刻都有零或更多的实例；
3. 类是静态的，对象是动态的
4. 类是生成对象的模板

类和对象的关系可理解为，对象是类的实例，类是对象的模板。如果将对象比作房子，那么类就是房子的设计图纸。

**抽象**：通过特定的实例抽取共同特征后形成概念的过程，对象是现实世界中某个实体的抽象，类是一组对象的抽象。

**封装**：封装是将相关的概念组成一个单元，然后通过一个名称来引用他。面向对象封装是将数据和基于数据的操作封装成一个整体对象，对数据的访问或修改只能通过对象对外提供的接口进行。

补充面向对象的知识点

**继承**：继承表示类直接的层次关系，继承又可分为单继承和多继承，继承自父类的属性特征，不需要在子类中进行重复说明

**多态**：使得在多个类中可以定义同一个操作或属性名，并在每个类中可以有不同的实现。多态使得某个属性或操作在不同的事情可以表示不同类的对象特性。多态，是面向对象的程序设计语言最核心的特性。多态，意味着一个对象有着多重特征，可以在特定的情况下，表现不同的状态，从而对应着不同的属性和方法

**接口**：接口就是对操作规范的说明，其只说明操作应该做什么

**消息**：体现对象间的交互，通过它向目标对象发送操作请求

**组建**：小时软件系统可替换的、物理的组成部分，封装了模块功能的实现。组建应当是内聚的，并具有相对稳定的公开接口。

**模式**：描述了一个不断重复发生的问题，以及该问题的解决方案。其包括特定环境、问题和解决方案三个组成部分。应用设计模式可以更加简单和方便的去复用成功的软件设计和架构，从而帮助设计者更快更好的完成系统设计

**复用**：软件复用是指将已有的软件及其有效成分用于构造新的软件或系统。

**原理**：

系统开发初期，必须明确系统的功能要求，确定系统边界

根据用户初步需求，利用系统开发工具，快速地建立一个系统模型展示给用户，在此基础上与用户交流，最终实现用户需求的信息系统快速开发的方法

**分类**：

**是否实现功能**：

**水平原型**：

也称为行为原型，用来探索预期系统的一些特定行为，并达到细化需求的目的

通常只是功能的导航，但并未真实实现功能

主要用在界面上

**垂直原型**：

也称为结构化原型，实现了一部分功能

主要用在复杂的算法实现上

**最终结果**：

**抛弃式原型**：用在解决需求不确定性、二义性、不完整性、含糊性

**演化式原型**：

为开发增量式产品提供基础，逐步将原型演化成最终系统

必须易于升级和优化的场合，特别适用于Web项目

**特点**：

1. 可以使系统开发的周期缩短，成本和风险降低、速度加快，获得较高的综合开发效益。
2. 以用户为中心来开发系统的，用户参与的程度大大提高，开发的系统符合用户的需求，因而增加了用户的满意度，提高了系统开发的成功率。
3. 由于用户参与了系统开发的全过程，对系统的功能和结构容易理解和接受，有利于系统的移交，有利于系统的允许和维护。

**优点**:

1. 主要在于能有效地确认用户需求
2. 用于那些需求不明确的系统开发
3. 对分析层面难度大、技术层面难度不大的系统，适合于原型开发
4. 而对于技术层面的困难远大于其分析层面的系统，则不宜用原型法

**面向服务的方法**：

1. 进一步将接口的定义与实现进行解耦
2. 提高系统可复用性、信息资源共享和系统之间的互操作性

**OIS模型：**

1. 物理层

该层包括物理连网媒介，如电缆连线连接器。该层的协议产生并检测电压以便发送和接收携带数据的信息。有RS232、V.35、RJ-45、FDDI。

1. 数据链路层

它控制网络层与物理层之间的通信。它的主要功能是将从网络层接收到的数据分隔成特定的可被物理层传输的帧。协议有IEEE802.3/.2、HDLC、PPP、ATM

1. 网络层

其主要功能是将网络地址（例如，IP地址）翻译成对应的物理地址（例如，网卡地址），并决定如何将数据从发送方路由到接收方。有IP、UDP、SPX。

1. 传输层

主要负责确保数据可靠、顺序、无错地从A点传输到B点、如提供建立、维护和拆除传送连接的功能；选择网络层提供最合适的服务；在系统之间提供可靠的透明的数据传送，提供端到端的错误恢复和流量控制、有TCP、UDP、SPX。

1. 会话层

负责在网络中的两节点之间建立和维持通信，以及提供交互会话的管理功能，如三种数据流方向的控制，即一路交互、两路交替和两路同时会话模式。有RPC、SQL、NFS

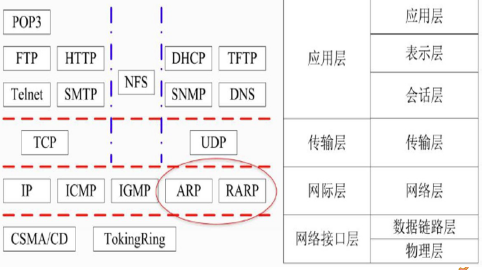
1. 表示层

如同应用程序和网络之间的翻译官，数据将按照网络能理解的方案进行格式化；管理数据的解密和加密、数据转换、格式化和文本压缩。有JPEG、ASCII、GIF、MPEG。

1. 应用层

负责对软件提供接口以使程序能使用网络服务，如事务处理程序、文件传送协议和网络管理等。有HTTP、Telnet、FTP、SMTP。

|  |  |  |  |
| --- | --- | --- | --- |
| 层次 | 名称 | 主要功能 | 主要设备及协议 |
| 7 | 应用层 | 实现具体的应用功能 | POP3、FTP、HTTP、Telnet、SMTP、DHCP、TFTP、SNMP、DNS |
| 6 | 表示层 | 数据的格式与表达、加密、压缩 |
| 5 | 会话层 | 建立、管理和终止会话 |
| 4 | 传输层 | 端到端的连接 | TCP、UDP |
| 3 | 网络层 | 分组传输和路由选择 | 三层交互机，路由器 ARP、RARP、IP、ICMP、IGMP |
| 2 | 数据链路层 | 传送已帧为单位的信息 | 网桥、交换机、网卡 PPTP、L2TP、SLIP、PPP |
| 1 | 物理层 | 二进制传输 | 中继器，集线器 |



TCP/IP协议

1. 应用层
   1. FTP（文件传输协议）

网络上两天计算机传送文件的协议

FTP在客户机和服务器之间需建立两天TCP连接，一条用于传送控制信息（使用21号端口），另一条用于传送文件内容（使用20号端口）

* 1. TFTP（简单文件传输协议，UDP）

用来在客户机与服务器之间进行监督文件传输的协议，提供不复杂、开销不大的文件传输服务。提供不可靠的数据流传输服务

* 1. HTTP(超文本传输协议)

从WWW服务器传输超文本到本地浏览器的传送协议

* 1. SMTP(简单邮件传输协议，TCP)提供可靠且有效的电子邮件传输的协议
  2. DHCP(动态主机配置协议，UDP) 负责IP地址的分配，有固定分配、动态分配和自动分配
  3. Telnet(远程登录协议TCP)允许用户登录进入远程计算机系统
  4. DNS(域名系统) 转换工作称为域名解析转换
  5. SNMP(简单网络管理协议) 网络管理

1. 传输层
   1. TCP 是可靠的，面向连接的、全双工的数据传输服务，输数据量比较少，且对可靠性要求高的场合。
   2. UDP 一种不可靠的、无连接的协议，传输数量大，对可靠性要求不是很高，但要求速度快的场合。
2. 网络层
   1. IP 无连接的和不可靠的，他将差错监测和流量控制之类的服务授权给了其他的各层协议
   2. ICMP（网络控制报文协议） 专门用于发送差错报文协议
   3. IGMP（国际足管理协议） 允许Internet中的计算机参加多播，是计算机用做向相邻多路由器报告多目组成成员的协议
   4. ARP（地址解析协议） IP地址向物理地址的转换。每块网卡都有唯一的地址
   5. RARP（反向地址分析协议）物理地址向IP地址的转换

## 1.3常规信息系统集成技术

**网络协议：**

局域网协议：IEEE802规范

组成：

802.1（802协议概论）、802.2（逻辑链路控制层LLC协议）

802.3（以太网的CSMA/CD载波监听多路访问/冲突监测协议）

802.4（令牌总线Token Bus协议）、802.5（令牌环Token Ring协议）

802.6（城域网MAN协议）、802.7（FDDI宽带技术协议）

802.8（光纤技术协议）、802.9(局域网上的语音/数据集成规范)

802.10（无线局域网WLAN标准协议）

分类：

|  |  |  |  |
| --- | --- | --- | --- |
| 分类 | 类型 | 传输速度 | 传输介质 |
| IEEE 802.3 | 标准以太网 | 10Mb/s | 细同轴电缆 |
| IEEE802.3u | 快速以太网 | 100Mb/s | 双绞线 |
| IEEE802.3z | 千兆以太网 | 1000Mb/s | 光纤或双绞线 |

广域网协议：

PPP点对点协议、ISDN综合业务数字网、xDSL、DDN数字专线、X.25、FR帧中继、ATM异步传输模式

**网络交换**：

* 物理层交换 如电话网
* 链路层交换 二层交换，对MAC地址进行变更
* 网络层交换 三层交换，对IP地址进行变更
* 传输层交换 四层交换，对端口进行变更，比较少见
* 应用层交换

**网络设备**：

|  |  |  |
| --- | --- | --- |
| 互联设备 | 工作层次 | 主要功能 |
| 中继器 | 物理层 | 对接收信号进行再生和发送，只起到扩展传输距离用，对高层协议是透明的，但是要个数有限（例如，再以太网中只能使用4个） |
| 网桥 | 数据链路层 | 根据帧物理地址进行网络直接的信息转发，可缓解网络通信繁忙度，提高效率。只能够连接相同MAC层的网络 |
| 路由器 | 网络层 | 通过逻辑地址进行网络之间的信息转发，可完成异构网络之间的互联互通，只能连接使用相同网络层协议的子网 |
| 网关 | 高层（第4-7层） | 最复杂的网络互联设备，用于连接网络层以上执行不同协议的子网 |
| 集线器 | 物理层 | 多端口中继器 |
| 二层交换机 | 数据链路层 | 是指传统意义上的交换机，多端口网桥 |
| 三层交换机 | 网络层 | 带路由功能的二层交换机 |
| 多层交换机 | 高层（第4-7层） | 带协议转换的交换机 |

**无线网络的产品**：无线网卡、无线AP、无线网桥和无线路由器

**网络存储技术**：

1、直接附加存储（DAS）

存储设备通过SCSI电缆直接连到服务器

其本身是硬件的堆叠，存储操作依赖于服务器，不带有任何存储操作系统

2、网络附加存储（NAS）

1、通过网络接口与网络直接相连，由用户通过网络访问，支持即插即用

2、NAS存储设备类似于一个专用的文件服务器，去掉了通用服务器的大多数计算功能，而仅仅提供文件系统功能，从而降低了设备的成本

3、支持多种TCP/IP协议，主要是NFS（网络文件系统）和CIFS（通用Internet文件系统）来进行文件访问，性能特点是进行小文件级的共享存取

3、存储区域网络（SAN）

特点： 通过高速光纤连接，极度可扩展型；通过专用交换机将磁盘阵列与服务器连接起来的高速专用子网，它没有采用文件共享存取方式，而是采用块（block）级别存储

分类：

* + - * FCSAN 热插拔性、高速带宽、远程连接、连接设备数量大等
      * IPSAN 基于IP网络实现数据块级别存储方式的存储网络，由于设备成本低，配置技术简单，可共享和使用大容量的存储空间
      * IBSAN 机构设计的非常紧密，大大提高了系统的性能、可靠性和有效性，能缓解各硬件设备之间的数据流量拥塞

**网络接入:**

有线接入：PSTN、ISDN、ADSL、FTTx+LAN和HFC

无线接入：无线电波作为传输媒介

GPRS、3G、4G接入

WiFi和移动互联接入（4G）



高速以太网的宽带技术FTTx+LAN（光纤+局域网）

FTTC（光纤到路边）FTTZ（光纤到小区）FTTB（光纤到楼）FTTF（光纤到流层）FTTH（光纤到户）

**网络工程：**

网络规划：网络需求分析、可行性分析、对现有网络的分析和描述

网络设计：

核心层：通过高速转发通信，提供优化、可靠的骨干传输结构

汇聚层：核心层和接入层的分界面，完成网络访问策略控制、数据包处理、过滤、寻址，以及其他数据处理的任务

接入层：允许终端用户连接到网络

网络实施：

**网络设计工作**：

1. 网络拓扑结构设计 考虑因素有：地理环境、传输介质与距离以及可靠性
2. 主干网络（核心层）设计 根据用户方网络规模大小、网上传输信息的种类和用户方可投入的资金等因素来考虑
3. 汇聚层和接入层设计 汇聚层的存在与否，取决于网络规模的大小
4. 广域网连接与远程访问设计 根据网络规模的大小、网络用户的数据量来选择
5. 无线网络设计 适用于很难布线的地方（比如受保护的建筑物、机场等）或者经常需要变动布线结构的地方（如展览馆等） 对于城市范围的网络接入也能适用，可以设想一个采用无线网络的IS可以为一个城市的任何角落提供高速互联网接入
6. 网络安全设计 网络系统的硬件、软件及其系统中的数据受到保护
7. 设备选型

**数据仓库：**面向主题的、集成的、非易失的、且随时间变化的数据集合，用于支持管理决策

组成：

1、数据源 数据仓库系统的基础，是整个系统的数据源泉

2、数据的存储与管理 是整个数据仓库系统的核心

3、OLAP服务器

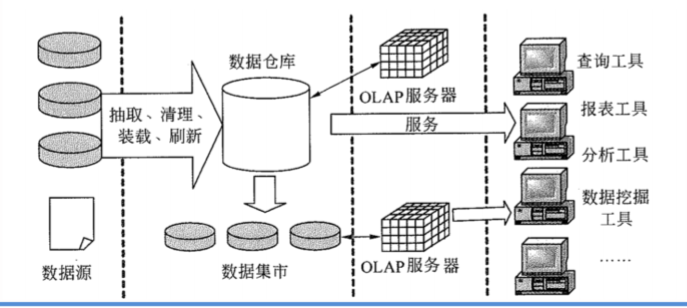
ROLAP：基本数据和聚合数据均存放在RDBMS之中

MOLAP：基本数据和聚合数据均存放于多维数据库中

HOLAP：基本数据存放于RDBMS之中，集合数据存放于多维数据库中

4、前端工具

各种查询工具、报表工具、分析工具、数据挖掘工具以及各种基于数据仓库或数据集市的应用开发工具



**中间件**:

定义： 1、在一个分布式系统环境中处于操作系统和应用程序之间的软件

2、一种独立的系统软件或服务程序，分布式应用软件借助这种软件在不同的技术之间共享资源，中间件位于客户机服务器的操作系统之上，管理计算机资源和网络通信

分类：

底层型中间件：

主流技术有JVM（Java虚拟机）、CLR（公共语言运行库）、ACE（自适配通信环境）、JDBC(Java数据库连接)和ODBC（开放数据库互连）代表产品主要有SUNJVM和MicrosoftCLR

通用型中间件：主流技术有CORBA（公共对象请求代理体系结构）、J2EE、MOM（面向消息的中间件）和COM代表产品主要有IONAOrbix、BEWebLogic和IBMMQSeries

集成型中间件：主流技术有WorkFlow和EAI(企业应用集成) 代表产品有BEAWebLogic 和 IBMWebSphere

应用：

1、为了完成系统底层传输层的集成，可以采用CORBA技术

2、为了完成不同系统的信息传递，可以采用消息中间件产品

3、为了完成不同硬件和操作系统的集成，可以采用J2EE中间件产品

## 1.4软件工程

**分类**：

1、业务需求

反映企业或客户对系统高层次的目标要求，项目投资人、购买产品的客户、客户单位的管理人员、市场营销部门或产品规划部门

2、用户需求 描述用户的具体目标，或用户要求系统必须能完成的任务，描述用户能使用系统来做什么

3、系统需求 从系统的角度来说明软件的需求，包括功能需求、非功能需求和设计约束等，也称为行为需求，规定了开放人员必须在系统中实现的软件功能，用户利用这些功能来完成任务，满足业务需要

**分类2**：

1、功能需求 通过系统特性的描述表现出来的，所谓特性指一组逻辑上相关的功能需求，表示系统为用户提供某项功能（服务），使用户的业务目标得以满足

2、非功能需求 系统必须具备的属性或品质，例如，可维护性、效率等

3、设计约束 限制条件或补充规约，通常是对系统的一些约束说明，例如，必须采用国有自主知识产权的数据库系统，必须运行在UNIX操作系统之下等

**分类3**：

1、常规需求 用户认为系统应该做到的功能或性能，实现越多用户会越满意

2、期望需求 用户想当然认为系统应具备的功能或性能，但并不能正确描述自己想要得到的这些功能或性能需求。如果期望需求没有得到实现，会让用户感到不满意。

3、意外需求 兴奋需求，是用户要求范围外的功能或性能（但通常是软件开发人员很乐意赋予系统的技术特性），实现这些需求用户会更搞笑，但不实现也不影响其购买的决策。

**特点**：无二义性、完整性、一致性、可测试性、确定性、可跟踪性、正确性、必要性

**获取的方法**： 用户访谈、问卷调查、采样、情节串联板、联合需求计划

**SA方法：**

1、核心是数据字典

2、模型  
 1、数据模型 实体联系图（E-R图） 描述实体、属性、以及实体之间的关系

2、功能模型 数据流图（DFD） 从数据传递和加工的角度，利用图形符合通过逐层细分描述系统内各个部件的功能和数据在它们之间传递的情况，来说明系统所完成功能

3、行为模型（状态模型STD） 通过描述系统的状态和引起系统状态转换的事件，表示系统的行为，指出作为特定事件的结果将执行那些动作。

**软件需求规格说明书（SRS）：**

需求开发活动的产物，使项目干系人与开发团队对系统的初始规定有一个共同的理解

内容：1.范围2.引用文件3.需求4.合格性规定5.需求可追踪性6、尚未解决的问题7、注解8、附录

**需求验证：**

1、内容

1、SRS正确的描述了预期的、满足项目干系人需求的系统行为和特征

2、SRS中的软件需求是从系统需求、业务规格和其他来源中正确推导而来的

3、需求是完整的和高质量的

4、需求的表示在所有地方都是一致的

5、需求为继续进行系统设计、实现和测试提供了足够的基础

2、方法 需求评审、技术评审、需求测试

**UML**

**结构：**构造块、规则、公共机制

UML具有的语言特征：

1、不是一种可视化的程序设计语言，而是一种可视化的建模语言

2、不是过程，也不是方法，但允许任何一种过程和方法使用它；

UML适用于迭代式开发过程

**四种关系：**

1、依赖 两个事物之间的语义关系，其中一个事物发生变化会影响另一个事物的语义

2、关联 描述一组对象之间连接的结构关系

3、泛化 一般化和特殊化的关系，描述特殊元素的对象可替换一般元素的对象

4、实现 类之间的语义关系，其中的一个类指定了由另一个类保证执行的契约

**5个系统视图：**

1、逻辑视图 设计视图，表示了设计模型中在架构方面具有重要意义的部分，即类、子系统、包和用例实现的子集

2、进程视图 是可执行线程和进程为活动类的建模，它是逻辑视图的一次执行实例，描述了并发和同步结构

3、实现视图 对组成基于系统的物理代码的文件和构建进行建模

4、部署视图 对构件部署到一组物理节点上，表示软件到硬件的映射和分布结构

5、用例视图 是最基本的需求分析模型

1. 关联 不同类的对象之间的结构关系，它在一段时间内将多个类的实例连接在一起，对象实例之间的关系，而不表示两个类之间的关系
2. 依赖 两个类A和B，如果B的变化可能会引起A的变化，则称类A依赖于类B
3. 泛化 一般事物与该事物中的特殊种类之间的关系，也就是父类与子类之间的关系，继承关系是泛化关系的反关系，也就是说，子类继承了父类，而父类则是子类的泛化。
4. 共享聚集 聚合关系，它表示类之间的整体与部分的关系，部分可能同时属于多个整体，部分与整体的生命周期可以不相同。例如：汽车和车轮是聚合关系，车子坏了，车轮还可以用，车轮坏了，可以换一个新的
5. 组合聚集 组合关系，它也是表示类之间的整体与部分的关系，与聚合关系的区别在于，组合关系中的部分职能属于一个整体，部分与整体的生命周期相同，部分随着整体的创建而创建，也随着整体的消亡而消亡。
6. 实现关系 将说明和实现连续起来，接口是对行为而非实现的说明，而类中则包含了实现的结构。一个或多个类可以实现一个接口，而每个类分别实现接口中的操作

关联：连接模型元素及链接实例，用一条实线来表示。

依赖：表示一个元素以某种方式依赖于另一个元素，用一条虚线加箭头来表示



聚合：表示整体与部分的关系，用一条实现加空心菱形来表示



组合：表示整体与部分的有一关系，用一条实线加实心菱形来表示

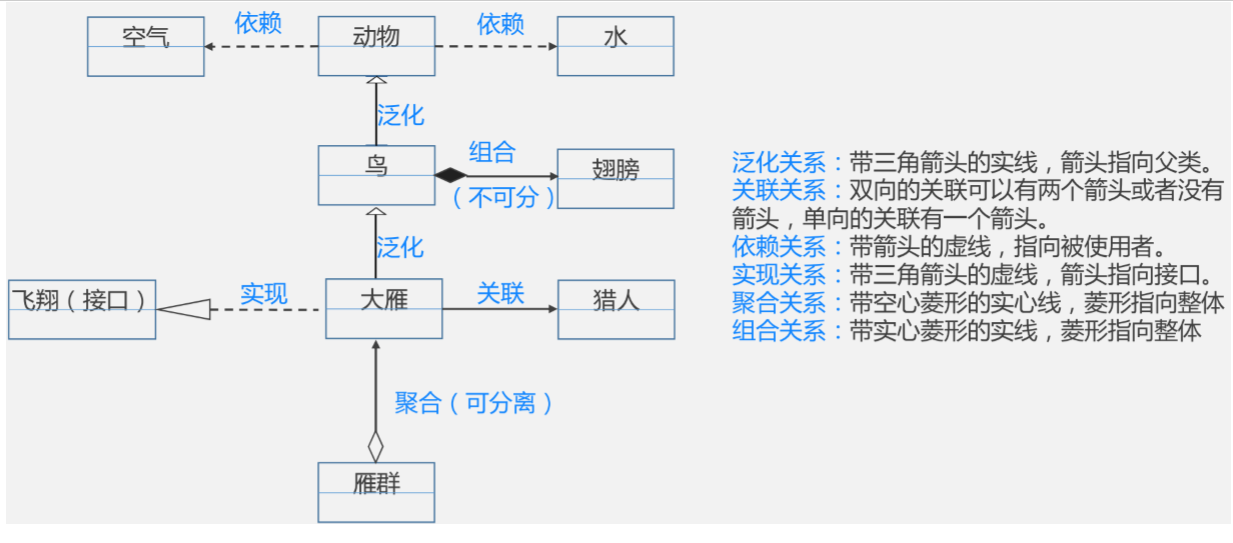


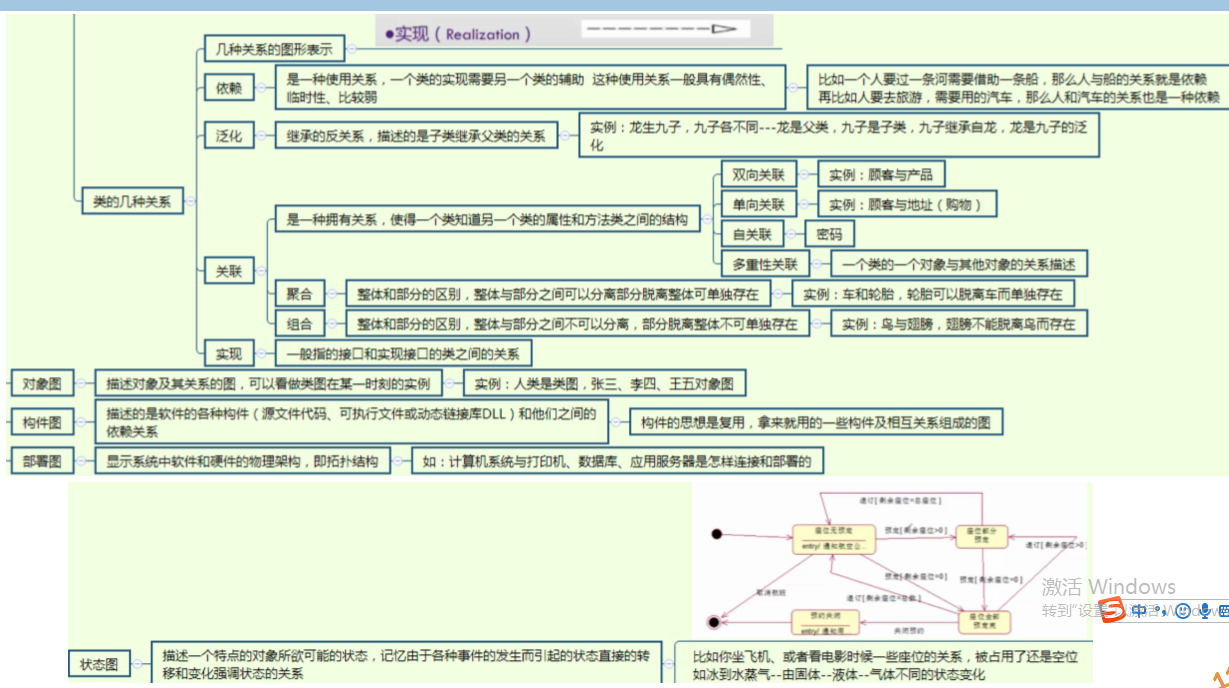
泛化：表示一般与特殊的关系，用一条实线加空心箭头来表示



实现：表示类与接口的关系，用一条虚线加空心箭头来表示







**14种图：**

1. 类图 描述一组类、接口、协作和它们之间的关系，是最常见的图
2. 对象图 描述一组对象以及它们之间的关系，描述了在类图中所建立的事物实例的静态快照
3. 构件图 描述一个封装的类和它的接口、端口、以及由内嵌的构建和连接件构成的内部结构
4. 组合机构图 描述结构化类（例如，构件或类）的内部结构，包括结构化类与系统其余部分的交互点。组合结构图用于画出结构化类的内部内容
5. 用例图 描述一组用例、参与者及它们之间的关系
6. 顺序图（序列图） 一种交互图，展现了一种交互，由一组对象或者参与者以及它们之间可能发送的消息构件
7. 通信图 一种交互图，它强调收发消息的对象或参与者的结构组织，强调的是时序，强调的是对象之间的组织结构（关系）
8. 定时图（计时图） 一种交互图，它强调消息跨越不同对象或参与者的实际时间，而仅仅只是关心消息的相对顺序
9. 状态图 描述一个状态机，它由状态、转移、事件和活动组成
10. 活动图 将进程或其他计算结构展示位计算内部一步步的控制流和数据流
11. 部署图 描述对运行时的处理节点及在其中生存的构建的配置
12. 制品图 描述计算机中一个系统的物理结构。制品包括文件、数据库和类似的物理比特集合
13. 包图 描述由模型本身分析而成的组织单元，以及它们之间的依赖关系
14. 交互概览图 是活动图和顺序图的混合物

**软件架构**

**核心**：能否达到架构级的软件复用，也就是说，能否在不同的系统中，使用同一个软件架构

**软件架构风格**：

1. 数据流风格 批处理序列架构风格和管道/过滤器
2. 调用/返回风格 主程序/子程序、数据抽象和面向对象，以及层次结构
3. 独立构件风格 进程通信和事件驱动的系统
4. 虚拟机风格 解释器和基于规则的系统
5. 仓库风格 数据库系统、黑板系统和超文本系统

**评估人员关注**：质量属性

**相关名词**：

敏感点：一个或多个构件（和/或构件之间的关系）的特性

权衡点：影响多个质量属性的特性

**评估方式**：

调查问卷（或检查表）的方式

基于场景的方式

最为常用

分类：

架构权衡分析法（ATAM）

软件架构分析法（SAAM）

成本效益分析法（CBAM）

描述方面:

刺激 场景中解释或描述项目干系人怎样引发与系统的交换部分

环境 描述的是刺激发生时的情况

响应 系统是如何通过架构对刺激作出反应的

作用：分析软件架构对场景的支持程度，从而判断该架构对这一场景所代表的质量需求的满足程度。

基于度量的方式

1.5软件设计

设计模式：  
分类：

结构化设计：

一种面向数据流的方法

以SRS和SA阶段所产生的DFD和数据字典等文档为基础

自顶向下、逐步求精和模块化的过程

面向对象设计

原则：

高内聚：模块内部个成分之间的联系程度

低耦合：模块之间联系的程度

区别：

OOD是OOA方法的延续，其基本思想包括抽象、封装和可扩展性，其中可扩展性主要通过继承和多态来实现

目的：可以方便地复用成功的软件设计

要素：模式名词、问题、目的、解决方案、效果、实例代码和相关设计模式

处理范围分：

类模式：处理类和子类之间的关系和静态关系

对象模式：处理对象之间的关系和动态性

目的和用途分：

创建型模式：用于创建对象，包括工程方法模式、抽象工程模式、原型模式、单例模式和建造者模式

结构型模型：用于处理类或对象的组合，包括适配器模式、桥接模式、组合模式、装饰模式、外观模式、享元模式和代理模式

行为型模式：用于描述类或对象的交互以及职责的分配，包括职责模式、命令模式、解释器模式、迭代器模式、中介者模式、备忘录模式、观察者模式、状态模式、策略模式、模板方法模式、访问者模式

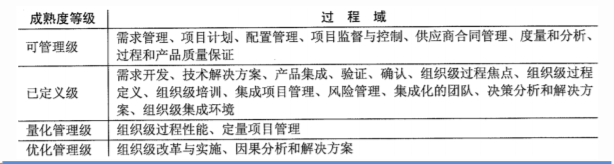
能力成熟度模型集成：

能力成熟度模型集成（CMMI）

定义：继承了CMM的阶段表示法和EIA/IS731的连续式表示法，这两 方法各有优缺点，均采用统一的24个过程域，它们在逻辑上是等价的，对同一个组织采用两种模型分别进行CMMI评估，得到的结论应该是相同的

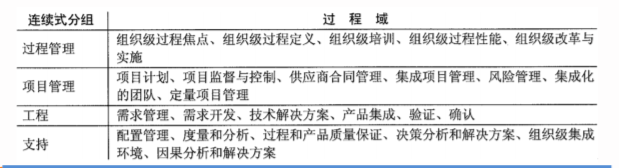
分类：

1、阶段式模式



保持4个成熟等级，但关键过程域做了一些调整和扩充

2、连续式模型



将24个过程域按照功能分为过程管理、项目管理、工程和支持四个过程组

**软件测试**

**测试方法分：**

静态测试：被测试程序不在机器上运行，而采用人工检测和计算机辅助静态分析的手段对程序进行检测，包括对文档的静态测试和对代码的静态测试，对文档的静态测试主要以检查单的形式进行，对代码的静态测试一般采用桌前检查、代码走查和代码审查

**动态测试**：

定义：在计算机上实际运行程序进行软件测试

方法：

白盒测试：结构测试，用软件单元测试中，将程序看作是一个透明的白盒，测试人员完全清楚程序的结构和处理算法，按照程序内部逻辑结构设计测试用例，检测程序中主要执行通路是否能按预定要求正确工作，方法有控制流测试、数据流测试和程序变异测试，使用人工检查代码的方法来检查代码的逻辑问题，逻辑覆盖，语句覆盖、判定覆盖、条件覆盖、条件/判定覆盖、条件组合覆盖、修正的条件/判定覆盖和路径覆盖功能测试

黑盒测试：功能测试，用于集成测试、确认测试和系统测试中，将程序看作是一个不透明的黑盒，完全不考虑程序的内部结构和处理算法，只检查程序功能是否能按照SRS的要求正常使用，程序是否能适当地输入数据并产生正确的输出信息，程序运行过程中能否保持外部信息（如文件和数据库等）的完整性等。方法：等价类划分、边界值分析、判定表、因果图、状态图、随机测试、猜错法和正交试验法

**单元测试**：也称为模块测试，测试的对象是可独立编译或汇编的程序模块、软件构件或OO软件中的类（统称为模块），目的是检查每个模块能否正确的实现设计说明中的功能、性能、接口和其他设计约束等条件，发现模块内可能存在的各种差错。

**集成测试：**检查模块之间，以及模块和已集成的软件之间的接口关系，并验证已集成的软件是否符合设计要求

**确认测试：**

**目的：**主要用于验证软件的功能、性能和其他特性是否用户需求一致

**分类：**

**内部确认测试：**由软件开发组织内部按照SRS进行测试

**Alpha测试：**由用户在开发环境下进行测试

**Beta测试：**由用户在实际使用环境下进行测试，通过后，才能把产品发布或交付给用户

**系统测试**：对象是完整、集成的计算机系统，目的是在真实系统工作环境下，验证完整的软件配置能否和系统正确连接，并满足系统/子系统设计文档和软件开发合同规定的要求

**配置项测试**：对象是软件配置项，目的是检验软件配置项与SRS的一致性

**回归测试**：

目的：测试软件变更之后，变更部分的正确性和对变更需求的符合性，以及软件原有的、正确的功能、性能和其他规定的要求的不损害性

对象：未通过软件单元测试的软件，在变更之后，应对其进行单元测试，未通过配置项测试的软件，在变更之后，首先应对变更的软件单元进行测试，然后再进行相关的集成测试和配置项测试。未通过系统测试的软，在变更之后，首先应对变更的软件单元进行测试，然后再进行相关的集成测试、配置项测试和系统测试。因其他原因进行变更之后的软件单元，也首先应对变更的软件单元进行测试，然后再进行相关的软件测试

**面向对象的测试系统**：

1、封装性 必须考虑到信息隐藏原则对测试的影响，以及对象状态与类的测试序列

2、继承性 必须考虑到继承对测试充分性的影响，以及误用引起的错误

3、多态性 必须考虑到动态绑定对测试充分性的影响、抽象类的测试、以及吴用对测试的影响

**软件调试与测试的区别**：

1、测试的目的是找出存在的错误，而调试的目的是定位错误并修改程序以修改错误

2、调试时测试之后的活动，测试和调试在目标、方法和思路上都有所不同

3、测试从一个已知的条件开始，使用预先定义的过程，有预知的结果；调试从一个未知的条件开始，结束的过程不可预计

4、测试过程可以事先设计，进度可以事先确定；调试不能描述过程或持续时间

**软件测试的管理：**

过程管理**：**

组成：测试活动管理，测试资源管理

要点：软件测试应由相对独立的人员进行。根据软件项目的规模、完整性级别和测试类别，软件测试可由不同机构组织实施

配置管理：将测试过程中产生的各种工作产品纳入配置管理，由开发组织实施的软件测试，应将测试工作产品纳入软件项目的配置管理，由独立测试组织实施的软件测试，应建立配置管理库，将被测试对象和测试工作产品纳入配置管理

**评审工作：**

测试就绪评审：测试执行前对测试计划和测试说明等进行评审

测试评审：测试完成后，评审测试过程和测试结果的有效性，主要对测试记录和测试报告进行评审

**补充**：

1、软件测试是为了发现错误而执行程序的过程，是根据程序开发阶段的规格说明及程序内部结构而精心设计的一批测试用例（输入数据及其预期结果的集合），并利用这些测试用例去运行程序，以发现程序错误的过程。故软件测试应尽可能在实际运行试验环境下进行。软件测试不再只是一种仅在编码阶段完成后才开始的活动，而是应该包括在整个开发和维护过程中的活动，他本身也是实际产品构造的一个组成部分。

2、软件测试有如下几个原则：

1、软件开发人员即程序员应当避免测试自己的程序（单元测试除外）

2、应尽早的和不断地进行软件测试

3、用测试用例要正确的态度：第一，测试用例应当由测试输入数据和预期输出结果这两部分组成；第二，在设计测试用例时，不仅要考虑合理的输入条件，更要注意不合理的输入条件。

4、要充分注意软件测试中的群集现象，也可以认为是“80-20原则”。不要以为几个错误并且解决这些问题之后，就不需要测试了。反而这里是错误群集的地方，对这段程序要重点测试，以提高测试投资的效益。

5、严格执行测试计划，排除测试的随意性，以避免发生疏漏或者重复无效的工作、

6、应当对每一个测试结果进行全面检查

7、妥善保存测试用例、测试计划、测试报告和最终分析报告，以备回归测试及维护之用。

**EAI**：

